A Secure Erasure Method for Fragmented JPEG Images

Xu Chang
School of Cyber Security, Shandong University of Political Science and Law, Jinan, 250014, Shandong, China
changxumail@163.com

ABSTRACT

Data has become a fundamental factor of production and a core strategic resource for the country. Data security has become a major issue related to national security and economic and social development, as well as the protection of personal privacy data. This article proposes a secure erasure method for JPEG images stored in fragmented storage by analyzing the storage, deletion, and recovery of data under different file systems, the causes of file fragmentation, and the principles and algorithms of data erasure. Using file system structure to locate data, using multi-level erasure algorithm to overwrite data, and performing data deletion and structural adjustment on the system structure information of the corresponding file system to achieve secure erasure of JPEG image files. The experimental results show that the proposed method is capable of securely erasing fragmented JPEG image files, has the ability to prevent recovery, and has high practical value.

CCS CONCEPTS

• **Security and privacy** → Systems security; File system security.

KEYWORDS

JPEG, File Fragments, Erasure Technology, Data Recovery

ACM Reference Format:
Xu Chang. 2024. A Secure Erasure Method for Fragmented JPEG Images. In *2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (DEAI 2024), January 19–21, 2024, Hongkong, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3675417.3675519

1 INTRODUCTION

With the development of the new generation of information technology and the integration of human production and life, various types of data are massively gathered and growing rapidly, which has had a significant and profound impact on people's lives and economic development. Data has become a fundamental factor of production and a core strategic resource for a country. Data security is not only related to personal factors, but also affects important areas such as politics, economy, and national defense. Once leaked and maliciously exploited, it will cause significant harm to national security and social stability. Data security has become a major issue related to national security and economic and social development. Therefore, data security is a major issue related to national security, economic and social development, and personal interests.

Some important government and defense departments have extremely high confidentiality requirements for data, and there is also an urgent need for confidentiality of core data of enterprises or personal privacy data. A large number of electronic devices not only enrich and facilitate our daily lives, but also provide a breeding ground for crime [8]. Traditional methods such as file deletion and disk formatting cannot truly completely erase data. Especially when discarded and damaged storage devices flow into the second-hand market, data information is often recovered by criminals using recovery software and extracting private information for sale or engaging in illegal criminal activities. Therefore, how to safely and quickly erase disk data without damaging the disk has also received increasing attention.

JPEG balances quality and performance, and is the most widely used still image compression standard. It is the default or supported format for most smartphones, tablets, digital cameras, and other products. In 2021, social networks spread over 100 million images per hour and over 3 billion images per day [2]. Digital images are spreading continuously and at an extremely high rate, indicating that image information plays an important role in daily life. In addition, frequent creation, modification, and deletion of files on traditional storage devices can cause files to become fragmented and out of order on the disk. Garfinkel conducted fragmentation statistics on the disk of the file system, showing that the JPEG file fragmentation rate is 16%. Typical NAND flash devices [9] (such as solid-state drives, USB drives, etc.) introduce a flash transfer layer (FTL) due to compatibility issues, resulting in natural fragmentation [3]. The fragmentation and disorder of data can increase the difficulty of data erasure, easily ignoring file system details, resulting in incomplete data erasure and providing clues for data recovery.

Based on this, this article proposes a secure erasure method for JPEG fragmented storage images. By analyzing the storage, deletion, and recovery principles of JPEG images in different file systems, the relevant data structures of JPEG fragmented files are thoroughly cleared, and multi-level erasure algorithms are called to complete the secure erasure of fragmented image data. The experiment shows that the proposed method can securely erase JPEG fragmented data, and has high resilience and application value.

2 RELATED WORK

2.1 Data recovery technology

Data recovery technology refers to the technique of using certain strategies, methods, or means to recover data when the electronic data storage medium is damaged, causing some or all of the data to be inaccessible and unreadable, so that the lost information

DEAI 2024, January 19–21, 2024, Hongkong, China

ACM ISBN 979-8-4007-1714-7/24/01
https://doi.org/10.1145/3675417.3675519

Table 1: The beginning and end of different image files

File Type	File Header	End of File
JPEG (jpg)	FFD8FF	FF D9
PNG (png)	89504E47	AE 42 60 82
GIF (gif)	47494638	00 3B

can be reproduced. Data recovery technology is neither backup nor prevention, but a remedial measure for electronic data storage media caused by failures or improper operations. Therefore, there are also some special situations that are difficult to recover, such as data being overwritten. The classification of data recovery can be roughly divided into four types: software recovery, hardware recovery, database system or closed system recovery, and coverage recovery.

2.2 Data Carving technology

Data carving technology relies on the file system structure for data recovery [4]. When the file system is damaged or incomplete, it is difficult to fully utilize its functionality. Especially for fragmented storage data, it is difficult to determine the type, ownership, and order of data partitioning. Data carving [1] does not rely on file system information, but is based on the theoretical foundation of information technology and computer science. It extracts data from the seemingly unstructured original binary data stream to form a file, and reasonably explains the actual meaning of the file data. By analyzing and repairing the binary data of the file, it achieves file carving and repair. Based on the internal features of the image file, JPEG fragmented data can be searched outside the file system and the data can be directly extracted in the DATA area to recover the image. The main principles include the following aspects.

2.2.1 File structure analysis. File carving technology first analyzes the structural information of files such as data regions, file headers, and file endings. By searching, identifying, and analyzing structural information, accurately locate the important data location of the file to be recovered, providing a foundation for subsequent carving and repair. The common file header and footer flags are shown in the Table 1.

2.2.2 Data recovery algorithms. File carving technology involves various data recovery algorithms such as data fragment classification, adjacency judgment and reorganization, completion of file header data, and repair of file tail data. Through data carving algorithms, the classification, judgment, and reorganization of fragmented data can be effectively restored to complete data carving. In addition, partially lost data can also be repaired and restored.

2.2.3 File replication strategy. File replication technology needs to select corresponding file replication strategies based on different scenarios and specific application requirements. Data consistency and integrity can be maximized through techniques such as block by block replication, or differential replication.

Table 2: Different erasure algorithms

Data erasure algorithm	Overwrite count
Dod5220.22-M (ECE)	7
AR380-19	3
Schneier	7
Gutmann	35
TSSIT OPS-II	7
gost p50739-95	2
HMG IS5 (enhancement)	3

2.3 Data erasure technology

Data erasure technology, also known as data destruction technology or file shredding technology, aims to destroy electronic data in order to achieve the goal of information confidentiality that cannot be leaked, recovered, or repaired [5]. Disk storage uses magnetic recording technology to store binary data, and data erasure is a reverse operation that arises from the behavior of data recovery and data carving technology. The most commonly used way to delete data in daily life is to directly delete files in the operating system, then clear the recycle bin, or use Shift+Delete to completely delete them. However, this type of deletion only removes part of the file's structure or marks modifications, and the true data of the file is not deleted from the disk. Data can also be restored through specific software programs. The second method is to overwrite all data on the disk with 0, 1, random numbers, etc. to achieve the true deletion of data. By overwriting different data in the corresponding data area of the file, the irrecoverability of the file is achieved. Among them, the second method, also known as second level erasure or data erasure, erasing confidential data on the disk in this way is more reassuring.

The reason why data can be "recovered" is that the operating system does not directly destroy the data area when deleting files. The behavior of these tools is to repeatedly reset the corresponding file entity data in the data area, such as repeatedly writing to the magnetic track with various garbage data. According to the method of writing data and the intensity of damage, data erasure can be divided into several algorithms. In terms of data erasure algorithms, each country has proposed its own data erasure standards, such as the national standard (three pass, seven pass), the US Department of Defense's Dod5220.22-M, Schneier, and Gutmann algorithms, Russia's gost p50739-95, Canada's TSSIT OPS-II, as shown in Table 2. In the middle, the Gutmann algorithm will perform 35 repeated erasures on the data area, which is enough to damage the data to the point where it cannot be restored using ordinary file recovery tools; The strength of Dod5220.22-M is more flexible, and data can be damaged to the point where it cannot be restored using the corresponding hardware devices.

2.4 JPEG compression format

JPEG [7, 10, 11] (Joint Photographic Experts Group) is a product of the JPEG standard, developed by the International Organization for Standardization (ISO), and is a compression standard for continuous tone still images. The JPEG compression principle uses steps such as discrete cosine transform, quantization, and entropy encoding to

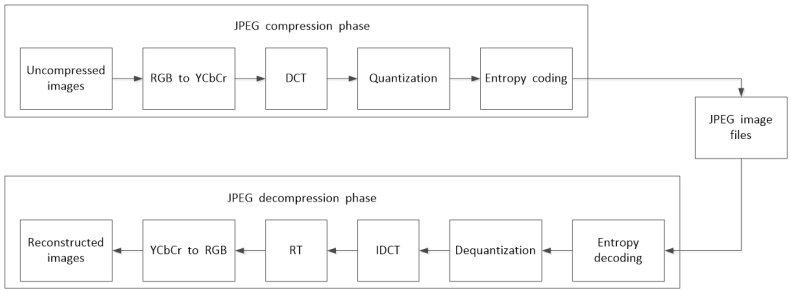


Figure 1: JPEG image compression and decompression process

convert images to the frequency domain. Then, quantization and entropy encoding are used to reduce the size of image files, achieving effective image compression. This glossy compression method can significantly reduce the size of image files while maintaining image quality, thus achieving more efficient image transmission and storage, as shown in Figure 1.

3 ANALYSIS OF DATA STORAGE PRINCIPLES

3.1 Analysis of File Storage Principles under FAT32

The FAT32 architecture consists of four parts: DBR, FAT, directory, and data, which manage disk space and file storage on a cluster basis.

- DBR mainly uses BPB to record some basic information, such as the number of bytes in each sector, the number of sectors in each cluster, and the number of reserved sectors. The positions of FAT, FDT, and DATA can be located through the number of reserved sectors, FAT occupied sector size, and FAT number in BPB.
- The FAT table is used to record the situation of file data occupying clusters, usually with two FAT tables. The second FAT table is used to back up the first FAT table. The FAT table uses a cluster chain to record the clusters occupied by the file, with each 32-bit binary being a FAT table entry used to record information about the occupied clusters. Each FAT table entry records the cluster number of the next cluster, and the last cluster records the “FF FF FF 0F” end tag.
- The directory stores all the files and subdirectory information in the root directory, with each item being 32 bytes in size, including the file name, length, deletion price, attributes, creation time, and the most important starting cluster number.
- The data is the file data saved in the cluster during file storage, which is the actual data of the file.

3.2 Analysis of File Storage Principles under NTFS

NTFS, like FAT32, is allocated on a cluster basis. In the NTFS file system, all data on the disk appears in the form of files, and even the management information of the file system is stored in the form of a set of files, namely meta files$ MFT is a very important meta file composed of file records (FR).

3.2.1 MFT.. MFT (Master File Table) is the core of NTFS, a type of NTFS data structure specifically used to store file records as a meta file. It is also a file itself, which stores the properties of all files in the file system. To read the content of a file, you need to first locate the file in the main file table, and then locate the location where the file data is stored.

3.2.2 Metafile. After formatting the partition to NTFS, a lot of important system information is written, which is the meta file of NTFS, including MFT. The first 16 file records in the main file table are usually fixed, and all of these files start with ”$” to indicate that they are hidden files in the system and cannot be modified by users.

3.2.3 File Records. MFT manages files through file records, with each file record corresponding to a different file. Each FR has a fixed size and occupies 2 sectors, consisting of a file record header and an attribute list. File records are stored continuously in MFT, numbered sequentially starting from 0. The file created by the user will also generate corresponding FR records for the corresponding file information, and the FR number will be recorded in the directory entry.

3.3 Analysis of the causes of image fragmentation

The reasons for fragmented image file storage include:

(1) Frequent creation, deletion, and modification of files by different organizational strategies in the operating system (such as FAT32) can easily lead to fragmented storage. As shown in the figure, files B and C are stored continuously, while file A is divided into three parts, presenting a fragmented and discontinuous storage state; As shown in the figure 2.

(2) Different from traditional disk storage media, SSD [12] and other NAND storage media are compatible with traditional modes and introduce a Flash Translation Layer (FTL) consisting of three modules: address mapping, garbage collection, and wear balancing. Among them, the wear leveling algorithm introduced to ensure uniform wear of storage devices results in completely disordered and fragmented storage of image files.

(3) Image hiding technology first divides files into different numbers of smaller blocks, shuffles the order, and stores them in different file data on storage devices to achieve image file hiding and confidentiality. If the order and location of the data blocks are known, it is convenient to reorganize them into the original file, but if not known, it is very difficult.

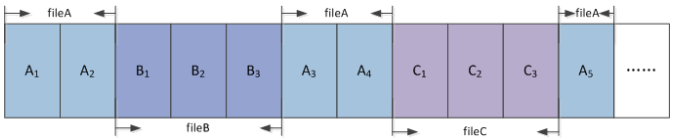


Figure 2: Image Fragmentation and Continuous Storage

3.4 ANALYSIS OF PRINCIPLES OF DATA DELETION AND RECOVERY

3.4.1 The principle of deleting and restoring FAT32 image files.

- File deletion principle of FAT32. After the file is deleted, the file data will not be cleared. The system will set the cluster chain in the FAT table of the file to 0, indicating that the corresponding cluster can be allocated to other files for use; And set the first byte of the directory entry corresponding to the files in the root directory to E5, and at the same time set the two higher bytes of the first cluster to "00 00 00". The data in the data area remains unchanged until a new file is written and overwritten.
- File recovery principle of FAT32. Based on the first byte in FDT, determine whether the file has been deleted (E5 deletion flag), or determine the target file through methods such as file name, file type, file deletion date, file creation date, and file last access date. If the first two bytes of the file cluster are missing and the address cannot be found, they can be replaced by the two higher bytes of the file created at the closest time to other files, verify if it is correct, usually with high accuracy. If the files are stored continuously, the file data can be found based on the first cluster number and file size in the directory entry of the file to be recovered in the root directory to complete the data recovery; If the file is fragmented for storage, the starting position is determined based on the first cluster number, and the file features are classified and the adjacent fragments are determined. After reorganization, the data is replicated.

3.4.2 The principle of deleting and restoring NTFS image files.

- The principle of NTFS files system deletion. After deleting a file in an NTFS volume, the system will make changes in the following areas: firstly, the byte at the offset 16H of the FR header corresponding to the file is set to "00H"; The second is to clear the directory entries corresponding to the root directory files; The third is to modify the occupation record of the deleted FR in the DataRun attribute of $MFT B0H; The fourth is to modify the identification of the cluster occupied by the data of the erased image in the DataRun [6] attribute of $BITMAP 80H; The data has not changed.
- The principle of NTFS files system recovery. The NTFS file system recovers data, enters the MFT of the file system, and jumps to file record 5 (which is one of the metadata files of the NTFS system and the root directory of the file system. It mainly stores some information of files in the root directory of the virtual USB drive. By analyzing the A0H attribute of the MFT, the index entries of the files can be found). According to the A0H attribute of the root directory MFT (index allocation attribute, which defines the start and end VCN of the root directory index, as well as the start LCN and number of clusters occupied by the data run), data can be run to jump to the index allocation data of the root directory. View the index table of the root directory, find the index entry of the target file, and analyze the index entry to obtain the MFT reference number of the file. Jump to the MFT file record of the target file, check its 80H attribute, and obtain the starting cluster number and number of clusters occupied by the data in the attribute body during operation to locate the data area of the target file and complete data recovery.

4 JPEG FRAGMENTED STORAGE IMAGE ERASURE METHOD

4.1 Choose a multi-level erasure algorithm

Multiple erasure methods are used, including the DOD5220.22-M three erasure method, the DOD5220.22-M standard seven erasure method, and the Gutmann method. The higher the level required, the more times it is overwritten, the more thorough the erasure, and the longer the execution time.

4.2 Design of image erasure method for fragmented storage

Based on the principles of fragmented file storage, deletion, and recovery in the file system mentioned above, a fragmented storage JPEG image recovery method based on the Windows file system is designed as shown in the figure 3.

- Step 1: First determine the file system type. If it is a FAT32 file system, jump to Step 2; Otherwise, transfer to Step3;
- Step 2: Locate the DBR, read the BIOS parameter table (BPB) in the DBR, obtain the number of reserved sectors, FAT occupied sectors, and FAT numbers. According to formulas 1 and 2, jump to FAT and locate the data area of the deleted file based on the first cluster number in the directory entry of the deleted image file and the cluster chain of the deleted file in the FAT table. Jump to Step4 to overwrite data;
- Step 3: Locate DBR, read the BIOS parameter table (BPB) in DBR, or locate the root directory through the 5th FR of MFT, search for the directory entry of the deleted image file in the root directory, obtain the file record number of the deleted file, and use the file record number to jump to the FR of the deleted file in MFT, Find the 80H in FR to obtain the data area of the deleted image file (two possibilities: when the file is small, directly in the 80H attribute, when the file is large, DataRun needs to be calculated), and jump to Step4 to overwrite the data;

Figure 3: Flowchart of the proposed method

- Step 4: calls the erase algorithm, overwrites the data area, and jumps to Step 5;
- Step 5: Conduct research on erasure algorithms. After multiple secure erasures of data, it is necessary to erase the structural information of the file system. If it is a FAT32 file system, jump to Step 6; If it is an NTFS system, jump to Step7;
- Step 6: Overwrite the directory entry of the deleted image over the directory entry of the erased image; Jump to Step8;
- Step 7: Clear the FR of the erased image; Jump to Step9;
- Step 8: Clear the cluster chain of the erased image and jump to Step 12;
- Step 9: Clear the directory entries of the erased images in the root directory, and adjust the directory entries in the root directory according to the B+tree principle; Jump to Step10;
- Step 10: Modify the DataRun in the $MFT B0H attribute to clear the FR identifier occupied by deleted images; Jump to Step11;
- Step 11: Modify the identification of the cluster occupied by the data of the erased image in the $BITMAP 80H attribute DataRun, and jump to Step 12;
- Step 12: Complete the overwrite of deleted images and clean up the file structure.

5 EXPERIMENTAL RESULTS AND ANALYSIS

The testing environment is to use a 64GB USB flash drive as the storage disk, with a cluster set size of 4096 bytes (8KB). Format it as FAT32 file system and NTFS file system in sequence. Simulate practical applications, such as TXT, RMVB, GIF, PDF, BMP, ZIP, DOCX, DOC. Wait for them to be randomly collected and written until they cannot be written, and then simulate normal application scenarios. After 20 batches of file creation and deletion, let them naturally generate fragments with random fragment sizes and positions. Choose JPEG images with sizes ranging from 12.8K to 5.18M and name them test1-test20.

By copying storage devices and performing routine deletion on one of them, using Meiya Forensic Master for evidence collection

Table 3: Different erasure algorithms

Recovery & Forensic Tools	Anti forensics &recovery ratio (Regular method)	Anti forensics &recovery ratio (proposed method)
WinHex	40%	100%
Easyrecoery	50%	100%
Forehost	40%	100%
Scalpel	35%	100%

can obtain image evidence from test1-test20. After using data carving tools such as WinHex, Easyrecovery, Forehost, and Scalpel for data recovery, it was found that the failure rate of WinHex was 40%, Easyrecovery was 50%, Forehost was 40%, and Scalpel was 35%. After analysis, WinHex, Easyrecovery, and Forehost only rely on file wrap recovery and fail to properly recover fragmented data, while Scalpel can achieve fragmented carving of partially fragmented data. Using the proposed method for erasing data on another storage device, the Meiya Forensic Master was unable to obtain evidence for image evidence. Similarly, WinHex, Easyreecoery, Forehost, Scalpel, and other tools were also unable to recover images. The anti evidence and recovery rate was 100%. The experiment showed that this method can achieve secure erasure of fragmented stored JPEG images, as shown in Table 3.

6 CONCLUSIONS

This article proposes a secure erasure method for JPEG images stored in fragmented format. By locating data through the file system structure, using a multi-level erasure algorithm to overwrite the data, and deleting and cleaning the corresponding record structure of the file system, the goal of secure deletion is achieved; Finally, the use of fragmented storage images for secure erasure is checked and analyzed. The experimental results show that daily forensics tools, data recovery and data carving tools cannot obtain or recover safely erased data, and this method has strong security.

ACKNOWLEDGMENTS

This work is supported by Undergraduate Teaching Reform Research Project of Shandong Province under Grant M2022233; it is also supported by Projects of Shandong University of Political Science and Law under Grant 2021Z04B; Teaching Research and Teaching Reform Project of Shandong University of Political Science and Law under Grant 2021JGA001.

REFERENCES

[1] Yang Xiaodong. Data self destruction technology in data security systems [J]. Electronic Technology and Software Engineering, 2021, (02): 249-250
[2] Sina Finance, spreading 3 billion images worldwide every day, drives the economy and hides hidden crises, https://baijiahao.baidu.com/s?id$=$1701991358318741606&wfr$=$spider&for$=$pc 2021.
[3] Xu, Chang, Jian, *et al.* JPEG fragment Caring based on Pixel Similarity of MED_ED [C]//The 38th China Control Conference. 2019.
[4] Wang Lin. Application and Exploration of Data Recovery Technology in Computer Forensics [J]. Computer Knowledge and Technology, 2023,19 (12): 80-82. DOI: 10.14004/j.cnki.ckt.2023.0598.
[5] Gao Zhipeng, Xu Zhiqiang, Wu Shixiong, *et al.* Research on Hard Disk Self Erasure Technology [J]. Information Network Security, 2012, (12): 60-64.
[6] Wu Shuhui. Research on NTFS file recovery method based on file name search [J]. Computer Age, 2023, (03): 119-123. DOI: 10.16644/j.cnki.cn33-1094/tp.2023.030.28.
[7] Xu Linglong, Zhang Yujin, Wu Yun. Detection and localization of tampered areas in dual JPEG compressed images [J]. Optoelectronics •Laser, 2023,34 (12): 1271-1278. DOI: 10.16136/j.roel2023.12.0523.
[8] Wu Xianyan. Research on Key Technologies for Image File Sculpture[D]. Harbin Institute of Technology, 2017
[9] Matsui C , Sun C , Takeuchi K .Design of Hybrid SSDs With Storage Class Memory and NAND Flash Memory[J].Proceedings of the IEEE, 2017, PP(9):1-10.DOI:10.1109/JPROC.2017.2716958.
[10] Yuan L , Ebrahimi T .Image privacy protection with secure JPEG transmorphing[J].IET Signal Processing, 2017, 11(9):1031-1038.DOI:10.1049/iet-spr.2016.0756.
[11] Ruchaud N , Dugelay J L .JPEG-based scalable privacy protection and image data utility preservation[J].IET signal processing, 2018(7).DOI:10.1049/iet-spr.2017.0413.
[12] Yang P , Xue N , Zhang Y ,et al.Reducing Garbage Collection Overhead in {SSD Based on Workload Prediction[C]//11th {USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 19).2019.

Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

DeepFake Face Image Detection based on Improved VGG Convolutional Neural Network

Xu Chang[1,2], Jian Wu[1,2], Tongfeng Yang[1] , Guorui Feng[1,2]

1. School of Cyber Security, Shandong University of Political Science and Law, Jinan 250014

2. Key Laboratory of Evidence-Identifying in Universities of Shandong(Shandong University of Political Science and Law), Jinan 250014, China

E-mail: changxumail@163.com

Abstract: DeepFake can forge high-quality tampered images and videos that are consistent with the distribution of real data. Its rapid development causes people's panic and reflection. In this paper we presents an improved VGG network named NA-VGG to detect DeepFake face image, which was based on image noise and image augmentation. Firstly, In order to learn the tampering artifacts that may not be seen in RGB channels, SRM filter layer is used to highlight the image noise features; Secondly, the image noise map is augmented to weaken the face features. Finally, the augmented noise images are input into the network to train and judge whether the image is forged. The experimental results using the Celeb-DF dataset have shown that NA-VGG made great improvements than other state-of-the-art fake image detectors.

Key Words: DeepFake, Image Detection, VGG

1 Introduction

With the continuous development of artificial intelligence technology, especially the emergence of deep learning, image and video editing becomes more and more easy. Different from common tampering techniques such as spreading, copy move, and remove. DeepFake relies on the technology of deep learning. Through the algorithm of deep learning, it can identify the photos of different angles, postures and expressions of the target characters (such as celebrities, politicians, etc.), and then continuously train to automatically generate the fake pictures, and cover them to the faces of the original video characters to form the " DeepFake videos " [1]. Compared with PS and other image modification and tampering technologies, the reason why "DeepFake is worrisome is that it is a combination of high authenticity, pervasiveness and rapid evolution [2]. In November 2017, DeepFake has been widely used on Reddit for its production of many pornographic videos in the United States, which has attracted attention from all walks of life and gained a great reputation on the Internet. In January 2018, the application using DeepFake was officially launched, which further intensified the spread of DeepFake videos. The object of face swapping has also rapidly expanded from celebrities and politicians to friends, classmates and colleagues. The development of DeepFake naturally triggered people's panic and reflection. Therefore, the major technology enterprises also began to make joint action with the academic community to avoid further negative impact on the fierce discussion of whether and how to regulate the DeepFake technology.

*This work is supported by Open Fund of the Key Lab of Forensic Science,Ministry of Justice,China (Academy of Forensic Science);Program for Young Innovative Research Team and Big Data and Artificial Intelligence Legal Research Collaborative Innovation Center in Shandong University of Political Science and Law; Projects of Shandong Province Higher Educational Science and Technology Program under Grant No.J16LN19, J18KA357, J18KA383; Shandong Province Soft Science Research Project under Grant No.2019RKB01369.

Fig. 1. Deepfake images of Tommy Lee Jones and Ian McKellen

With the continuous improvement of computer computing power and the continuous reduction of hardware price, as well as the high integration of deep learning tools such as tensorflow[3] and keras [4], technicians with certain professional foundation can forge high-quality forged images and videos consistent with the real data distribution through convolution automatic encoder [5] and Generative adversarial networks (GAN) [6].Deepfake image as shown in Fig. 1 In addition, smart phones and desktop applications such as deepnude [7], faceapp and fakeapp [9] make it easy for the general public without a deep technical background to produce videos or pictures that are hard to identify with the naked eye. In particular, counterfeiting some shocking content on social media can be spread quickly without verification.

Deepfake videos on the Internet are mainly produced in the following three ways, including computer apps, service portals, and market services. Among them, computer apps refer to the tools used to create DeepFake videos, most of which have the most of these apps provide 'facewatching' capabilities; online service portals provide users with service

portals functions as online businesses for generating and selling custom deepfakes. Users can upload photos by uploading Market services refer to individual deep fake creators who advertise their services on forums and online markets.

So far, the published DeepFake tools have been widely used to produce pornographic videos of fake celebrities or revenge porn, which seriously infringes the personal rights and property rights of citizens. Such porn has been banned by reddit, twitter, ponhub and other websites [8]. On the other hand, they have also been used to produce fake news, such as forging politicians' specific statements, creating political tensions and destroying society stability, national security and international order. At present, governments are also considering these issues. In January 2020, Facebook has deleted DeepFake videos that according its standards [9]; DeepFake will destroy the trust mechanism of the social community. Human beings will enter a post truth era, and the truth will no longer exist. People only believe what they are willing to believe. As a result, social consensus is difficult to aggregate and the trust model will be broken. AI is like a double-edged sword. It can also bring harm. It can help [10]. DeepFake makes it more and more difficult to distinguish the real image from the processed image. More and more researches are devoted to the fight against counterfeiting in the digital world. We need to use all available tools to identify the true and the false, and prevent the criminals from using the tampered images for immoral business or political activities. But AI can also be used to detect fake images that human eyes can't see. Making and recognizing face swapping is like a cat and mouse game. The Deepfake technology is changing with each passing day, and the counterfeiting technology should also be iterative. In the future, in the face of increasingly severe trust crisis, it is urgent to need effective DeepFake image detection methods and technologies. It is becoming more and more important and challenging to study how to distinguish whether a picture or video is true.

It is for this reason that we proposed a DeepFake image detection VGG network (NA-VGG) based on image noise and image augmentation, as shown in Fig.2. Our contribution of this work is summarized as follows: first. The RGB image to be detected is used to highlight the image noise information through SRM filter layer to obtain the image noise characteristic image as the input of the network. Second, the noise image is flipped horizontally / vertically, Third, in view of the poor visual quality of a large number of depth forgery data sets, which cannot truly reflect the depth forgery image spread on the network, this experiment uses the Celeb-DF dataset [11], the experimental results show that that NA-VGG has great improvements in detecting DeepFake face images.

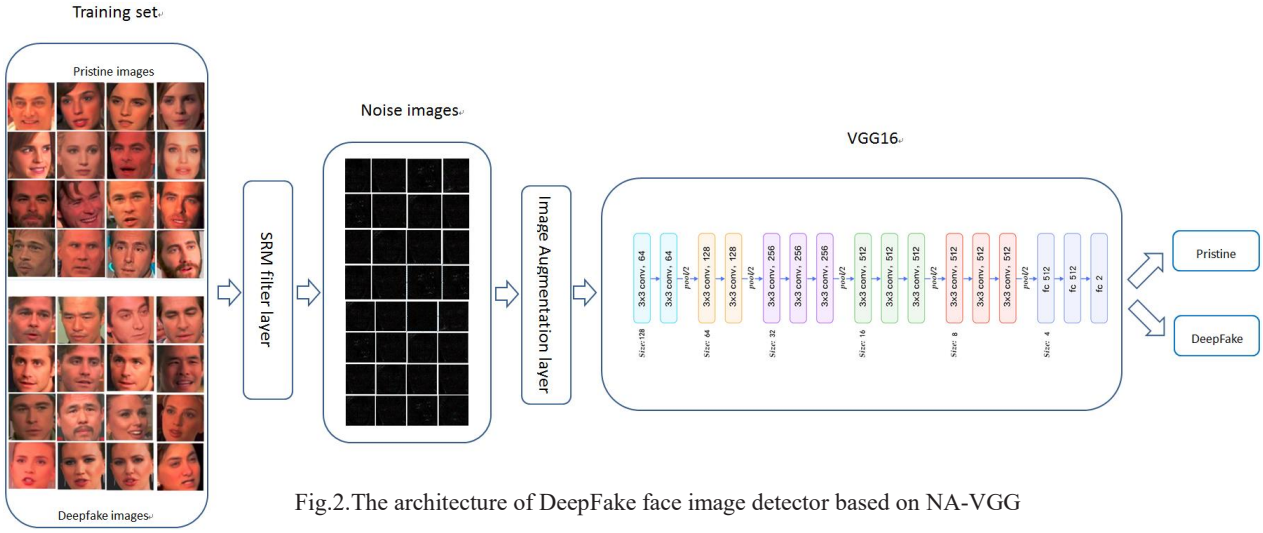


Fig.2.The architecture of DeepFake face image detector based on NA-VGG

2 Related Work

Digital Media Forensics. The purpose of digital media forensics technology [9] is to automatically evaluate the integrity of images or videos, and judge whether they have been tampered, synthesized, spliced or not only by the digital media itself. It is a passive forensics technology, which does not need to deal with media information in advance. Its universality and practicability make it an important research field. Traditional passive forensics technology of digital media mainly includes passive forensics based on traces left during forgery, passive forensics based on the consistency of digital media imaging equipment, and passive forensics based on the statistical characteristics of digital media itself [12]. Different from the traditional image editing tools such as PS, the emergence and continuous development of DeepFake technology, zero technology users can use desktop applications or apps to tamper with the image with one key, which is not only convenient to use, but also difficult to identify the authenticity of the faked image.In 2018, David Güera et al. [12] proposes a temporal-aware pipeline to automatically detect deepfake videos, convolutional neural network (CNN) is used to extract frame level features, which are then used to train a recurrent neural network (RNN);Peng Zhou et al. [20] proposes a two-stream Faster R-CNN network and train it end-to-end to detect the tampered regions given a manipulated image. In 2019,Chih-Chung Hsu et al. [23]proposes a deep learning-based methd to detect the fake image by combining the contrastive loss;Yuezun Li et al. [24] Based on the existing DeepFake algorithm, only limited resolution images can be generated, and distortion is needed to match the original face in the source video. It is proved that convolutional neural network (CNN) can effectively capture the artifacts left in the distortion matching process. The

above DeepFake algorithm improves the detection accuracy of DeepFake image from different perspectives, but most of the datasets used are of rough quality, and most of the tampering marks can be seen by the naked eye, resulting in its weak detection and weak generalization ability. With the continuous development of DeepFake technology, the forgery image quality is getting higher and higher, which leads to the high quality circulated in the re detection network of the above methods The performance of image decreases rapidly.

Generative Adversarial Networks(GAN). Deep learning has made breakthroughs in computer vision, voice and other application fields [14]. Compared with traditional machine learning, deep learning has a good representation ability, and it can automatically obtain abstract features. The model of deep learning can be roughly divided into discriminant model and generative model. Among them, the generative model has a direct impact on the real world modeling performance, and requires a lot of prior knowledge; in addition, because the real world data is too complex, it needs too much computation to fit the model, which makes the generative model become a very challenging and difficult to solve machine learning problem. Until 2014, Goodfellow et al. [15] proposed a new generation model inspired by the two person zero sum game in the game theory (that is, the sum of the two people's interests is zero, and the gains of one person are exactly the losses of the other person). The network system is composed of a generator G and a discriminator D. Both the generator and the discriminator can use the depth neural network which is currently hot in research [16]. The optimization process of GAN is a two-player minimax game with value function V (D,G) as formula (1), and the ultimate goal of optimization is to achieve the Nash equilibrium [17], which avoids the calculation of the distribution function brought by repeated application of Markov chain learning mechanism, and does not need the lower limit of variation or approximate inference, thus greatly improving Application efficiency [18]. At present, GAN has been continuously improved, such as DCGAN[13], CoGAN[22], ProGAN[25], StyleGAN[26] and other GAN networks, which have been successfully applied in the field of image generation and video generation. The quality of generation has been continuously improved, and it is difficult for the naked eye to recognize.

$$\min_G \max_D V(D,G) = \mathrm{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathrm{E}_{z\sim p_z(z)}\left[\log\left(1-D(G(z))\right)\right] \quad (1)$$

where x represents the real picture, z represents the noise of input G, G (z) represents the picture generated by G network, D (x) represents the probability that D network judges whether the real picture is real,D (g (z)) is the probability that D judges whether the picture generated by G is real.

Convolutional neural network(CNN). CNN is a kind of feedforward neural network. It is one of the most representative neural networks in the field of deep learning. It has made many breakthroughs in the field of image analysis and processing. Compared with other neural network structures, CNN needs relatively few parameters, which makes it can be widely used and has gradually become a commercial application, almost wherever there are images, there will be CNN. The current popular CNN include LeNet5, AlexNet, AlexNet, ResNet, GoogLeNet and VGG[19], among which VGG is developed by researchers from the Visual Geometry Group of Oxford University and Google DeepMind Company. VGG explores the relationship between the depth of CNN and its performance, and successfully constructs a CNN of 16-19 layers. It is proved that increasing the depth of the network can affect the final performance of the network to a certain extent. Colleagues who reduce the error rate enhance the mobility, and the generalization of migrating to other image data is also very good. Therefore, this paper uses VGG16 to extract the Noise features of the DeepFake images, and improves and optimizes the network structure to detect the DeepFake face images.

3 NA-VGG for Deepfake Image Detection

3.1 SRM Filter Layer

Simple image splicing, moving and other tampering operations have obvious differences in contrast. The authenticity can be judged by extracting relevant features. While the high-quality DeepFake face image is different, and its fidelity is very high. RGB channel is not enough to deal with different forgeries [20], especially in the case of deep forgeries without obvious splicing boundary and contrast, the effect of feature extraction using RGB channel is not good. Therefore, this paper focuses on image noise rather than semantic image content to determine the authenticity of the image, uses SRM filter in image forensics [21] to extract local noise feature map from RGB image, and takes the local noise distribution data of the image as the network input, and then uses the noise feature to provide the basis for image processing for authenticity classification.

The SRM filter kernel used in this paper, its weights are shown in Fig.3, sets the kernel size of SRM filter layer in noise flow as 5 × 5 × 3, and the output channel size of SRM layer as 3. Figure 4 shows the noise characteristic of the image to be examined after passing through the SRM layer. It can be seen that the acquired image is not the content of the image, but highlights the image noise, which contains the tampering artifacts that may not be seen in the RGB channel.

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}$$

Fig. 3. The SRM filter kernel used to extract noise features

3.2 Image Augmentation

Image Augmentation can translate, flip, rotate, zoom and enhance the existing image data to generate new images for training or testing. This operation can increase the number of pictures by several times, thus greatly reducing the possibility of over fitting. In this paper, we use the ImageDataGenerator class of Keras framework to augment the image to be examined, mainly through the random horizontal flip function of horizontal flip and vertical flip

function to weaken the features of human face, highlighting the detection of DeepFake trace features.

Fig. 4. Noise features of test images

3.3 VGG for Noise Feature Extraction

In our work, VGG16 is selected as the basic architecture. The convolution layer and pooling layer of VGG16 can be divided into different block, numbered Block 1 ~ Block 5 from the front to the back. Each block contains several volume layers and a pool layer, including 13 volume layers, 3 fully connected layers, and 5 pool layers. We add SRM filter layer and image Augment layer in front of VGG16 network, and propose a kind of VGG network based on noise and image augmentation (NA-VGG).

4 Experiments

4.1 Dataset

The development and evaluation of a DeepFake image detection algorithm need to meet two conditions: one is the need for large-scale datasets for training; the other is that the data in the datasets should truly reflect the quality of DeepFake images or videos on the Internet. Celeb-DF [22] is a new deepfake video dataset proposed by Yuezun Li et al. The dataset contains 408 original videos obtained from YouTube, and 795 DeepFake videos are synthesized from these real videos. Table 1 shows the average AUC performance of Celeb-DF dataset and UADFV, DeepFake-TIMIT(LQ,HQ), FF++ / DF, Celeb-DF and other datasets in different DeepFake detection methods. It can be seen that, compared with the previous DeepFake datasets, because of the high quality of DeepFake video in the Celeb-DF, the characteristics of artifacts are not obvious, and some detection models will reduce the performance on the Celeb-DF, resulting in the low detection accuracy.

Table 1: Average AUC performance of different methods on each dataset

Database	Average AUC performance
UADFV	78.7%
DeepFake-TIMIT(LQ)	73. 8%
DeepFake-TIMIT(HQ)	66.6 %
FF++ / DF	76.1 %
Celeb-DF	48.7 %

4.2 Image Dataset Preprocessing

In this paper, Celeb-DF is used for training evaluation. Firstly, the image is extracted from the DeepFake video. In this paper, Python + Opencv is used to extract the image by frame in the video. In order to ensure that the proportion of real image and DeepFake image is basically balanced, so in folder Celeb-real, 40 discontinuous images are captured for each video, and 10 discontinuous videos are captured for each video by Celeb-synthesis. 12416 training set images (including 5334 original images and 7082 DeepFake images) are extracted, 1376 verification set images (including 573 original images and 803 DeepFake images) and 1376 test set images (including 552 original images and 552 DeepFake images) are extracted 824 DeepFake images). The images of training set and test set have no repetition and are taken from different videos. They are disjoints sets. Secondly, we use classifier "haarcascade_frontalface_alt.xml" of OpenCV for face location and capture, and save the training set, verification set and test set of DeepFake detection.

4.3 Parameter Settings

Resizing of every image to 128*128,The optimizer is set to SGD for end-to-end training of the complete model with a learning rate of 0.01,decay of 1e-6, momentum of 0.9, and nesterov of True . The loss function is set to categorical_crossentropy as formula (2).

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}\left(y_{ij} * \log\left(\hat{y}_{ij}\right)\right) \qquad (2)$$

where M represents the total number of categories, n represents the number of samples, Yij is the actual result, y^ is the predicted result.

4.4 Results and Analysis

The experimental results are shown in Statistical table of the comparison of average AUC score of each detection method on Celeb-DF (see Table 3). The results show that the accuracy of our method in detecting DeepFake images is much higher than that of several DeepFake detection models using Celeb-DF data set in reference [22]. Other DeepFake detection models are trained in other datasets with obvious artifact features such as low resolution, color mismatch and visible boundary. The learned features may not be applicable in high-quality DeepFake dataset Celeb-DF, resulting in performance degradation. In addition, from the experimental results, it can be seen that the SRM filter can enhance the image noise by 16.8% compared with the VGG16 network, and the image augmentation by 12.5%, which shows that the SRM filter can highlight the image noise feature, and the image augmentation is effective to improve the detection accuracy.

Table 2: AUC performance of different methods on Celeb-DF

Methods	Average AUC performance
Average of Several Methods[22]	48.7%
Two-Stream[10]	55.7%
VGG16	56.4 %
SRM filter +VGG16	73.2 %
NA-VGG	85.7 %

5 Conclusion

In this paper we have presented a VGG network based on noise and image augmentation (NA-VGG) to detect the DeepFake face image. Firstly, the RGB image to be detected is used to highlight the image noise information through SRM filter layer, and then the image noise characteristic image is obtained as the input of the network. Secondly, the image noise is flipped horizontally / vertically, and weakened by data augmentation .Finally, the experimental results using the Celeb-DF have shown that by using SRM filtering to highlight image noise and image augmentation to weaken face features, we can learn tampering artifacts that may not be seen in RGB channels. The results show that NA-VGG made great improvements in detecting DeepFake face images. In future work, we plan to introduce Siamese network and RGB feature during training to further improve the accuracy.

References

[1] J.Y. Zhu, T. Park, P. Isola and A.A. Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In: Proceedings of the IEEE International Conference on Computer Vision. 2017:2223-2232.

[2] L.S. Wang, On the integrated regulation of "deep forgery" intelligent technology, Oriental Law, 2019:1–14.

[3] M. Abadi et al, Tensorflow: A system for large-scale machine learning. Proceedings of the USENIX Conference on Operating Systems Design and Implementation, 16:265–283, 2016.

[4] F. Chollet, et al, https://github.com/fchollet/keras. Keras, 2015.

[5] A. Tewari et al, Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017: 1274–1283.

[6] G. Antipov, M. Baccouche, and J.-L. Dugelay, Face aging with conditional generative adversarial networks. arXiv:1702.01983, 2017.

[7] The state of deepfake,deeptrace, http://www.deeptracelabs.com ,2019:4-7.

[8] D. Güera , Edward J. Delp, Deepfake Video Detection Using Recurrent Neural Networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance(AVSS), 2018:1–6.

[9] SoHu, Facebook banned deepfake video before the presidential election,http://www.sohu.com/a/365259905_99-956743,2020.

[10] P. Zhou, X.T. Han, V. I. Morariu and L.S. Davis , Two-stream neural networks for tampered face detection, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[11] Y. Li, X. Yang, P. Sun, H.G. Qi and S.W. Lyu, Celeb-DF: A New Dataset for DeepFake Forensics, arXiv preprint arXiv:1909.12962, 2019.

[12] R.C. Chen, Research on passive forensics of digital media based on object edge analysis.Hunan University,2012.

[13] R. Alec, M. Luke, C. Soumith, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv preprint arXiv:1511.06434,2016.

[14] W.L. Wang, Z.R. Li. Research progress of generative countermeasure network, Journal on Communications, 2018.

[15] I. Goodfellow, J. Pouget-abadie, M. Mirza, et al, Generative adversarial nets[C]//International Conference on Neural Information Processing Systems. 2014: 2672–2680.

[16] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. Cam-bridge, UK: MIT Press, 2016.

[17] K.F. Wang,C. Gou, Y.J. Duan, Y.L. Lin, X.H. Zheng ,F.Y.Wang, Generative Adversarial Networks: The State of the Art and Beyond.Acta Automatica Sinica, 43(3),2017.

[18] I. Goodfellow, Generative adversarial networks[J], arXiv: arXiv1701.00160, 2017.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.

[20] Peng Zhou, Xintong Han, Vlad I. Morariu, Larry S. Davis, Learning Rich Features for Image Manipulation Detection, arXiv: arXiv 1805.04953,2018.

[21] J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3): 868-882,2012.

[22] M.Y. Liu, O. Tuzel, Coupled Generative Adversarial Networks, arXiv:1606.07536,2016.

[23] Chih-Chung Hsu , Yi-Xiu Zhuang, and Chia-Yen Lee, Deep fake image detection based on pairwise learning, Preprints, 2019.

[24] Yuezun Li, Siwei Lyu.Exposing, DeepFake videos by detecting face warping artifacts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019:46–52.

[25] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation, arXiv:1710.10196,2018.

[26] T. Karras, S. Laine, T. Aila, A Style-Based Generator Architecture for Generative Adversarial Networks, arXiv:1812.04948,2019.

[27] Y.J. Zhang, A Course of Image Processing and Analysis(2nd Edition).Beijing:Posts & Telecom Press,2016.

File recovery of high-order clearing first cluster based on FAT32

Xu Chang[1,2], Jian Wu[1,2], Fanchang Hao[2,3],Pengtao Liu[1,2],Guorui Feng[1,2]

[1] School of Cyber Security, Shandong University of Political Science and Law, 250014, Jinan,China
{changxumail, jinanwujian, ptwave,fengguorui}@163.com
[2] Key Laboratory of Evidence-Identifying in Universities of Shandong(Shandong University of Political Science and Law), 250014, Jinan,China
[3] School of Computer Science and Technology,Shandong Jianzhu University,250101, Jinan, China
haofine@hotmail.com

Abstract. The core technical problems of file recovery are the first cluster number clearing of high-order two bytes and the fragmentation in the file. The algorithm proposed in this paper makes full use of file system information such as FAT, creation time, file identification and fragmentation characteristics, recover easy ones and the files difficult to recover which are recoverable at first, and then analyzes the status of remaining fragments after eliminating many interference of fragments. Experimental results show that this algorithm can significantly reduce the complexity of fragments and improve the accuracy of file recovery.

Keywords: FAT32, File recovery, Fragmentation, file identification.

1 Introduction

With the rapid development of information technology and intelligent equipment, electronic information storage equipment is rapidly popularized and has been continuously applied to various fields of human life and work. People got a great convenience in obtaining, storing and sharing data, which brought many challenges at the same time. Hackers, viruses, disoperation, storage medium quality defects and many other factors are threatening the security of data [1].Once the data is lost, the loss cannot be measured by money. On the other hand, in the process of cybercrime, with the enhancement of the criminal suspect's Anti-detection consciousness and the use of anti-forensics technology, especially the deletion of some sensitive and important data information, forensics becomes more difficult [2]. Data recovery is a technology to recover all or part of the data from the storage medium that has been destroyed or deleted. It has important application value and quickly becomes one of the key research areas to protect user data, fight computer crime and forensics.

For the convenience of use and management, most of the data is stored as files in specific file systems. Currently, the Windows system has a market share of over 92.6% [3], and its main application file systems are FAT32 and NTFS. In particular, the FAT32 file system is supported by numerous digital devices (NTFS has a lot of advantages, but is not suitable for mobile storage). Since mainstream storage devices generally have large capacity, which make the probability of high-order zero clearing after file deletion is very high. Frequent file operations lead to more fragmentation on disk. After analyzing more than 350 hard disks including FAT, NTFS, UFS and other file systems, Garfikel[10] et al. found that probability of fragmentation is as high as 42%, the probability of MS word file was 17%, and JPEG file was 16% on the disk.

The main difficulty in file recovery is the high order clearing of the first cluster number after file deletion [4] and file fragmentation [10]. For file recovery, most existing tools (e.g. FTK, Encase ,The Sleuth kit) are based on the assumption that the files stored sequentially in hard disks, Recover the file by extracting the block of data between the header and the footer, which caused them cannot handle the fragmented files[12]. The research on the zeroing of high order two bytes mainly focuses on creation time proximity, identification code comparison, time and feature code combination, etc., and the accuracy needs to be improved. However, recover the file correctly is impossible even if the first cluster is precisely located. This paper makes full use of the file system information such as FAT, creation time, file feature and shard feature to propose a data recovery method that can improve the precise location of the high-order clearing first cluster and deal with file fragments better.

The rest of this paper is organized as follows.Section2 talks about related works. The theory of file deletion and recovery of FAT32 will be introduced in Section3, followed by the proposed file recovery algorithm (Section4).Section 5 shows the experimental results and Section 6 concludes the paper.

2 Related work

Currently, file recovery can be divided into two major categories [6]: one is file recovery, and the other is filing Carving. File recovery is based on file system meta-information to recover Data. By scanning the file allocation table in the file system to locate the file start sector and the end of the sector to recover files, extract the data between header and footer mainly use the file system information or extract the size of data from header which can be acquired in FDT, The recovery efficiency is high for the files stored in the device continuously, but the accuracy of the data that cannot accurately locate the first cluster number or file header or fragment should be improved.

File carving works without using File system information. It attempts to recover and reconstruct files from the ostensibly unstructured original disk image (the binary data stream), without relying on the file system of the source disk image. Paper [1,5] suggests that in view of the existence of the creation time of files that are completely deleted in FDT, the high order two bytes of files that are similar to the creation time of deleted files can be found to replace the high order two bytes of deleted files. In Paper [4], a rapid data recovery scheme is proposed, which determines the storage

space of the file according to the cluster number in the sector where the file header feature code is located and the file size in the remaining directory items. In addition, the method of extracting effective information from the compound document structure is also discussed. In paper [11], the author proposed a method to recover lost file data by comprehensive utilization of various feature information, that is, the information of the file itself was deleted and other information of the creation time or storage location adjacent to it was also used. The creation time was combined with the feature code of the file for correlation and comparison to determine the two bytes of high order. Paper [12] provides an algorithm based on CED which used to evaluate if two data blocks are consecutive in the same file. Paper [13] use CNNs proposed CNN-based detectors for aligned and nonaligned double JPEG compression detection and explored the capability of CNNs to capture DJPEG artifacts directly from images.

In summary, most of the current researches focus on the recovery of a certain type of files or separate the recovery and carving of files. Most researches on file recovery ignore the fragments and only consider the case of continuous storage, which has a very good effect on the continuous storage of files. However, the fragment of files leads to the failure to recover files correctly even if the first cluster is precisely located. On the other hand, most of the researches on file carving ignore the file system structure and only study the fragments of data. Many metadata can play an important reference role in duplication carving, like FAT table. However, in reality, most storage devices are fragmented and also retain all or part of the metadata information of the file system. Making full use of this useful information can effectively improve the accuracy of file recovery.

Theory of file deletion and recovery of FAT32

3.1 FAT32 file system

FAT32 file system consists of DBR（DOS Boot Record and its reserved sectors, FAT（File Allocation Table）, FDT(File Directory Table) and DATA [7], as shown in Fig. 1.

DBR. BPB in DBR is particularly important for data recovery, which records the file storage format of this partition, starting sector, hard disk media descriptor, end sector, FAT number, size of allocation unit and root directory size and other important information.

FAT. A registry of where files are stored on the hard disk, recording the different file clusters. The clusters occupied by files are stored in the FAT in the form of cluster chain. The previous cluster stores the serial number of the next cluster.

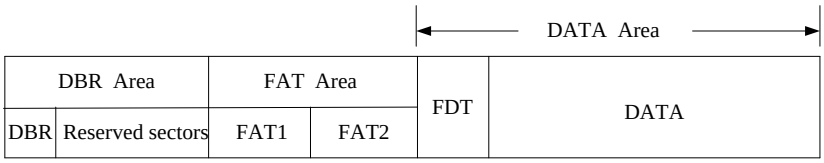


Fig. 1. FAT32 file system structure.

FDT. Also known as the DIR root area, all files and their subdirectories in the root directory directory table (FDT) have a "directory entry." Record the file name, extension, file creation date, creation time, access date, access time, last write date, last write time, file length, starting cluster of high and low bits and other information.

DATA. It is the FAT32 file system's true data storage area, occupying most of the hard disk space. When the file is deleted, the data in the data area corresponding to the file is still saved in the corresponding cluster, but this part of data is garbage data corresponding to the operating system. If new data is written, the cluster will be used as the idle cluster.

3.2 Fragmentation

Most files are stored sequentially on storage devices, but frequent file operations such as adding, deleting and modifying will lead to partial file fragmentation storage. File fragment is when a file is divided into several parts, scattered on a disk, but all parts are present and undamaged. These parts are called file shards. These fragments may be stored in order (as shown in file 1 in Fig. 2) or out of order(as shown in file 2 in Fig.2). A complete continuous file is one where the data in the file is contiguous. The research in this paper is based on a continuous or ordered debris model with complete structure and no loss.

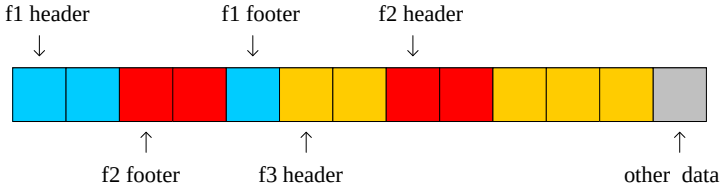


Fig. 2. Structure of FAT32 files system.

3.3 File identification

Common Office documents such as Word, Excel, PowerPoint, Visio and thumbnail file Thumbs. Db and other composite documents are a kind of file storage structure. Composite documents have obvious file header and file tail characteristics, which can

be used to identify file types. For example, in Word document format with the ".docx "extension, storage starts with hexadecimal keywords like" 0x504B030414." JPEG files start with "0xFFD8FFE0" or "0xFFD8FFE1" as the file header and end with "0xFFD9". The header and tail identifiers for some common files are shown in the following Table 1.

Table 1. Identifications of common files.

file type	header	footer
Word2010	”0x504B030414”	
Word97~2007	“0xD0CF11E0A1”	
JPEG	“0xFFD8FFE0”or”0xFFD8FFE1”	“0xFFD9”
PDF	“0x25504446”	“0x2525454F46”
GIF	“0x47494687A” or”0x474689A”	“0x003B”
WMV	“0x3026B2758E66CF11”	
ZIP	“0x504B0304”	

3.4 Analysis of file deletion operations

When deletes the file on disk, it does not really delete or overwrite the file data, but only modify some attributes of the file system. Just because the data still exists, it is not completely deleted, which provides us with the possibility of data recovery. This is also the main reason to be able to restore deleted data. The deletion of files in FAT32 file system involves the following four situations [1,4]. Changes of various parts of the file system before and after deletion by different deletion methods are analyzed as follows:

Shift+Delete. The FAT of the deleted file should be cleared, as shown in Fig. 3. The first byte of FDT is changed to "E5", and the top two bytes of the first cluster number are zeroed, as shown in Fig. 4. Therefore, the accurate location of the first cluster is not possible, which makes file recovery more difficult.

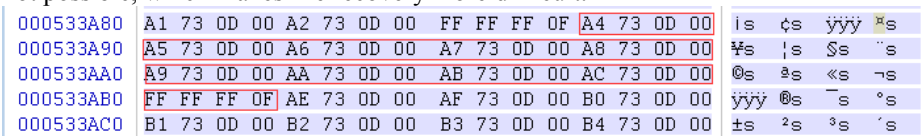


Fig.3. Status of FAT before deletion.

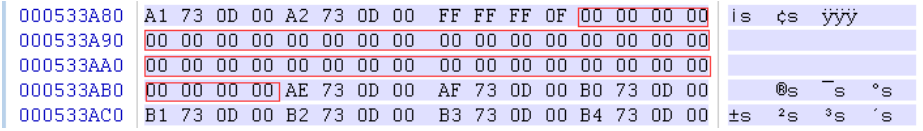


Fig.4. Status of FAT after deletion.

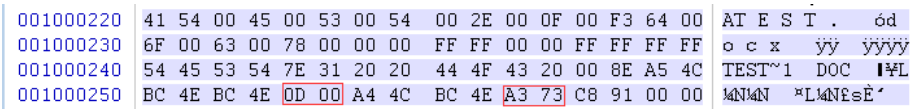


Fig.5. Status of FDT before deletion.

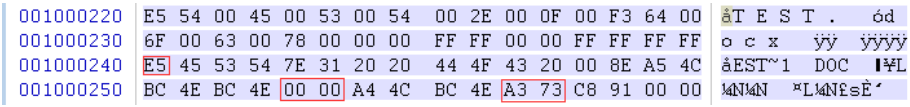


Fig.6. Status of FDT after deletion.

Delete. Enter the recycle bin first, and empty the recycle bin. The FAT item of deleted file is cleared, the first byte of FDT is changed to "E5", the first cluster number has no change before and after deletion, the first cluster location of the file is accurate, and recovery is relatively easy to achieve.

Completely delete the subdirectory. The directory FAT empty, FDT the first byte to "E5", the first cluster number of two bytes clear; The FAT table item of the file in the folder is cleared, the first byte of the FDT directory item is changed to "E5", the first cluster high position two bytes are not zero, you can locate the first cluster.

Format Disk. The FAT and FDT corresponding to the file are cleared, and the data still exists. The recovery of formatted files can only be achieved by reading binary data in storage devices. One method can be used to recover files by locating the start and end marks of files. However, for files that are stored discontinuously due to fragmentation, the recovery success rate is low. Another method can use data reconstruction technology [8,9,10] for recovery to identify, group, sort and reorganize file fragments.

The deletion of removable devices such as U disk is slightly different from the above deletion analysis. For example, after the deletion and thorough deletion of flash drive and the deletion of files in folders, the high-order clearing first cluster, this makes it difficult to locate the first cluster accurately. This paper mainly focuses on the zero-clearing of the first cluster height by two bytes after file deletion, which makes it impossible to locate the first cluster accurately, and causes the recovered files to be unable to be opened or to be restored as garbled codes.

4 The proposed algorithm of high-order clearing first cluster based on FAT32

4.1 Recover methods of high level clearing

According to the above analysis in section3, after the file is completely deleted, the first cluster number in FDT of the file directory entry is cleared to zero by two bytes, while the lower two bytes remain unchanged. To correctly locate the starting location of the file data, you need to recover the data exactly by getting two bytes high again by various means. The main methods include:

According to the creation time. The probability of starting cluster height being the same for files created at the same time is very high. Therefore, by looking for the files with the closest creation time in the same directory, we can refer to their height of two bytes.

Combined with the file identification. The initial cluster address obtained by method above is used to locate the cluster, and the file identification is used to determine whether it is consistent with the file type of the file to be recovered in FDT.

When using the creation time to locate the first cluster number, if the storage device is fragmented, the address of the first cluster of different files with the same creation time varies greatly, so it is difficult to ensure the accuracy. Using the method of combining creation time with file identification, if more files of the same type are in the same directory or all files of the same type, it may cause miscalculation.

4.2 The proposed algorithm

In this paper, a more accurate method is proposed to determine whether the cluster's two bytes high are accurate or not, and to avoid the problem of low data recovery accuracy caused by fragmentation. From easy to difficult, the algorithm proposed by continuous data is divided into four modules, which are FAT idle state extraction, no high-order reset file recovery, high-order reset file recovery and fragments processing.

The algorithm first by the second, three modules will determine the correct documents to get rid of, again to the rest of the document analysis, a small amount of divided in this way can obviously reduce the complexity of the debris.

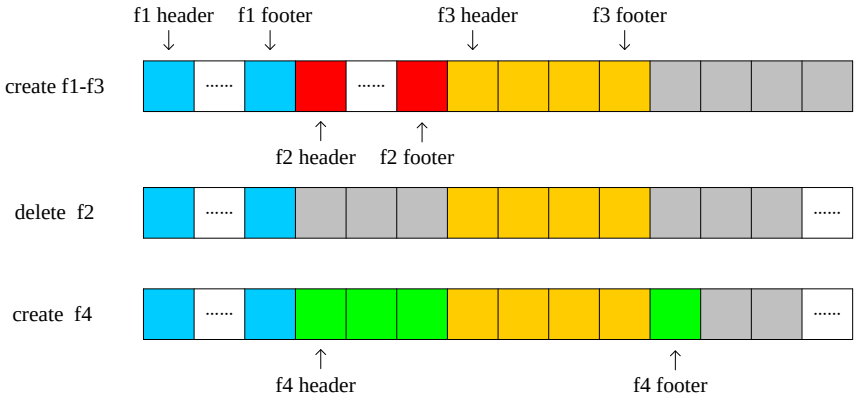


Fig.7. f3 and f4 deleted at the same time, because of the head and tail with f3 to store data in a row, easy to restore, can restore its first, in the fourth module lamination processing algorithms to restore the f4, can eliminate interference of debris, improve accuracy.

Idle state extraction of FAT. Traverse the FAT table, extract the state of the free cluster, mark the free cluster as "N", means none. In the data recovery process of the second and third modules, cluster marks for successful recovery of files occupied by successful recovery marks, cluster Numbers in doubt mark file name, and file length and other information. In case the fourth module is called when analyzing shard data.

File recovery with no high-order reset. Traverse the file directory, find deleted files of FDT, extract the deleted file's first cluster number, locating the files occupy the first bunch of location and match the file type, according to the length of the file to compare the end-of-file, if the head and tail, length is correct can be set for the first cluster to be successful, and the FAT table to restore files occupy clusters are marked correctly. (for files without tail feature identification, tail data can be located and judged according to the file length. If there is data before the tail and no data after the tail, the deleted file can be determined. Otherwise, set the first cluster mark corresponding to the FAT table, such file name, file type and length. The FAT status table has been prepared for reanalysis and use.

File recovery with high-order reset. Traverse the file directory, delete files search, according to preliminary access creation time first cluster number and verification, by matching the same file type, compare the end-of-file, according to the length of the file on the file tag and without the file tag files for recovery, and FAT free cluster status table, use the same methods above. The specific algorithm of this step is as follows:

Step1: traverse the root directory, extract the file directory items with deleted mark files, analyze FDT, and extract the creation time of deleted files.

Step2: according to the creation time of the extracted deleted file, find the existing disk file closest to the creation time of the deleted file, extract the two bytes high of the first cluster number, and merge the two bytes low in the deleted file FDT into the position of the first cluster number of the deleted file.

Step3: jump to the corresponding cluster according to the extracted first cluster number, analyze whether the file identification information in the cluster matches the file type in the deleted file FDT, and judge whether the corresponding cluster in the FAT state table is idle. If all match jump to Step4, otherwise jump to Step5.

Step4: extract and delete the file type in the file FDT, judge whether it has file tail information, if there is a jump to Step6, otherwise jump to Step7.

Step5: if the corresponding cluster is not an idle cluster, the first cluster location error, record the deleted file recovery state is "F", means "Fail"; Jump to Step7 if file type does not match.

Step6: extract and delete the last four bytes in the file FDT, namely the file length. Skip the corresponding file length from the starting position of the first cluster to see whether the end of the file matches the end of the file of this type. If match, skip to Step8; otherwise, skip Step9.

Step7: locate FAT table. Locate to the end of the file according to the location of the first cluster extracted and the length of the file. If the data before the end exist (including the end), jump to Step8 after no data. Otherwise jump to Step9.

Step8: extract the data of the deleted file according to the obtained first cluster number and file length, save it as the recovery success file, and mark the corresponding cluster as "V" and means "Victory" in the FAT table idle state table.

Step9: identify "FT" and means further treatment in the first cluster in the FAT status table, and delete the name, file length and file type information of the file, which has been prepared for the FAT status analysis of module four teams.

Fragmented data processing. After the processing of the first three steps, the remaining idle clusters are greatly reduced, which can improve the speed and accuracy of the analysis. Clusters with successful recovery, questionable clusters, and idle clusters in the FAT idle state table are identified. This module mainly analyzes the status of doubt cluster and idle cluster, and analyzes the data with fragments phenomenon when deleting files. If the file f4 in figure 7 is deleted, the nearest footer should be searched according to f4's header, and then the free cluster data directly from header and footer should be restored. The specific recovery algorithm is as follows:

Step1: scan FAT idle state table and read the clusters marked "FT" in sequence.

Step2: search for the nearest end tag matching the file type in the free cluster after the first cluster.

Step:3: get the free cluster data between the first cluster and the end tag of the deleted file in the FAT free state table, and calculate the extracted data length.

Step4: compare the extracted data length with the file length in the FAT idle state table, if consistent, restore the file. And mark "V" in FAT idle state table. If the FAT state table is not modified in case of inconsistency, skip to Step1 and continue to scan the next file.

Step5: end of this round of scanning.

5 Experimental results and analysis

The test environment is: Intel(R) Pentium(R) CPU g3250 3.2GHz, 4G memory, Windows7, ST1000DM 003-1er162. VHD (Microsoft Virtual Hard Disk format) technology was used to Virtual a Hard Disk with a size of 8G. After loading, DiskGenius was used to format it into FAT32 file system with the cluster size of 4096B(8KB) cluster size of 4096 bytes (8KB). In order to simulate real usage scenarios, common file types of different sizes, such as RMVB, GIF, PDF, BMP, ZIP, DOCX, TXT, etc. are randomly collected and written to them until they cannot be written, and then fragments are generated naturally by simulating the principle of fragmentation generation, with random fragment size and location. Fragmentation produces a simulation of the use of real storage devices, randomly deleting some files, and then writing files, 10 times. Finally, 100 files of different types were randomly deleted to complete the production of test data. The algorithm proposed in this paper

is tested and compared with common recovery software. Experimental results are shown in Table 2:

Table 2. Analysis of experimental results.

Name	Number of success	Success rate
Foremost	168	84%
Scalpel	183	91.5%
Proposed algorithm	189	94.5%

The results show that the proposed algorithm is more accurate than the tools include foremost and scalpel, which improved the accuracy of file recovery.

5 Conclusion

In this paper, we analyzed two problems existing in data recovery and different file deletion methods. Through FAT32 file system analysis and the fragmentation model ， we proposed a new method to recovery deleted files, using creation time combined with file id FAT table cluster state judgment, characteristic of the high two bytes store by continuous and matching accuracy, through the analysis of the high two bytes and verify accurate correct deleting files for recovery, analysis of the residual FAT table free cluster state to avoid the interference of file fragments to restore to algorithm consists of four modules, respectively for the FAT idle state extraction, no high-order reset file recovery, high-order reset file recovery and fragments processing. Experimental results show that the proposed method can effectively improve the accuracy.

Acknowledgements

This work is supported by Projects of Shandong Province Higher Educational Science and Technology Program under Grant J18KA357, J16LN19, J18KA383;Doctoral Research Fund of Shandong Jianzhu University under GrantXNBS1810;Teaching Innovation project of Shandong University of Political Science and Law under Grant 2017JYB009; projects of Shandong University of Political Science and Law, Grant No. 2016Z03B, 2015Z03B.

References

1. Wei Liu. In-depth disclosure of data recovery technology (2nd edition). Beijing:Electronic industry press.248--257 (2016).
2. Bo Guo,Youquan M.,. A computer forensics method based on FAT32 file system. Computer application and software. 1,260--262(2010)

3. Windows 10 Operating system version of the latest market share data released,http://notebook.it168.com/a2019/0131/5155/000005155901.shtml,(2019)
4. Deming Yang.Fast recovery method of effective data based on FAT32.Journal of Computer Applications, 9,2500--2503(2012)
5. Haiyang Fan，Lipeng W.,. Research on data recovery technology of FAT32 file system, Science and technology information,36,55--57(2013)
6. Hong G.,. Research on JPEG file duplication based on thumbnail. Hangzhou dianzi university,(2012)
7. Shijian Dai,Yanhui Tu. Data recovery technology (classic reproduction version).Beijing: Electronic industry press.(2014)
8. Y.B. Tang, J.B. Fang, K.P. Chow, S.M. Yiu, Jun Xu, Bo Feng,Qiong Li, Qi Han.Recovery of heavily fragmented JPEG files. pp:108-117,Digital Investigation (2016)
9. Digital Assembly.Smart Carver.http://digital-assembly.com/ products/smartcarver-dc3/,(2015)
10. S.L. Garfinkel. Carving contiguous and fragmented files with fast object validation . Digital Investigation, 2007, 4: 2-12.
11. Guangyu G.,Shujuan Z.,. A file recovery method in a FAT32 file system. Network new media technology.2,36--41(2016)
12. Yanbin T., Junbin F., K.P. Chow, S.M. Yiu, Jun Xu, Bo Feng,Qiong Li, Qi Han.Recovery of heavily fragmented JPEG files. pp.108-117,Digital Investigation 18 (2016)

Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

JPEG fragment Carving based on Pixel Similarity of MED_ED

Xu Chang[1,2], Jian Wu[1,2], Fanchang Hao[2,3]

1. School of Cyber Security, Shandong University of Political Science and Law, Jinan 250014
E-mail: changxumail@163.com

2. Key Laboratory of Evidence-Identifying in Universities of Shandong(Shandong University of Political Science and Law), Jinan 250014, China
E-mail: jinanwujian@163.com

3. School of Computer Science and Technology,Shandong Jianzhu University,Jinan.250101
E-mail: haofine@hotmail.com

Abstract: JPEG fragment carving technology is a hot and difficult research field in data recovery and computer forensics. Most of the adjacent Pixel Similarity algorithms for JPEG images have high accuracy when the pixels are continuous and smooth, Low accuracy will occur when image contains sharp areas such as there are strips, vertical strips and angles. SoD and ED only focus on the similarity between adjacent edge pixels, rather than the integrity and smoothness of the whole image. MED uses the surrounding pixel values to predict the target pixel value, which is better than SoD and ED in the prediction of pixel integrity. In order to solve this problem, we provide a new Euclidean distance adjacent pixel detection algorithm MED_ED based on median edge detector, according to the local stability and the characteristics of MED. Secondly, the design algorithm to judge the original cluster size, according to the Features of the file system, in order to solve the problem of inefficiency under the situation of the current storage device capacity soaring and high image quality. The experimental results show that the proposed method can effectively improve the accuracy and efficiency.

Key Words: JPEG fragments, File carving, MED_ED

1 Introduction

Image files are more convincing than other documents, because of the characteristics of objectivity, truthfulness, clarity and intuition. Therefore, they play an important role in computer crime forensics. JPEG standard has been widely supported by vendors because of its good performance, and has achieved great success in the market [1]. As the most widely used Static Image Compression Standard, JPEG file is the default or supported format of most electronic products, such as most smart photo phones, Tablets, digital cameras, etc. In 2014[2], the number of photos uploaded to the Internet will reach 900 billion, which shows that picture information plays an important role in our daily life. Therefore, the research on JPEG Carving has attracted great attention of forensic researchers, and has rapidly become one of the key technologies to combat computer crime, and a hot research area of data recovery and computer forensics.

As electronic data, image files are easy to be destroyed. Criminals often try their best to destroy the traces of crime, delete and destroy the data to avoid suspicion, or divide, disrupt and hide confidential image files in other data areas, resulting in the failure to reproduce confidential images. In addition, frequent creation, modification and deletion of traditional storage devices (such as mechanical hard disks) can lead to fragmentation and disorder of files. Garfinkel [3] makes fragmentation statistics on more than 350 disks of FAT, NTFS and UNIX file systems. It shows that the probability of fragmentation of important files such as e-mail, word documents and images in computer forensics is relatively high, and shows the fragmentation rate of JPEG files is 16%. Typical NAND flash devices (such as solid-state hard disk, U disk, etc.) are naturally fragmented by introducing flash transfer layer (FTL) due to compatibility problems. The fragmentation and disorder of data will make the accuracy of traditional data recovery technology decrease sharply or even completely invalid, which will cause great difficulties for computer forensics. File carving does not depend on the metadata of the file system. It mainly aims at the process of recovering and reconstructing files from the seemingly unstructured binary data stream in the case of file fragmentation and data disorder. Therefore, the key problems to be solved include fragment recognition, grouping, sorting and reorganization, and the core technical problem is to design a pixel similarity algorithm to judge whether the next data block is adjacent to the previous data block and belongs to the same file or not.

2 Related work

Because of widely used, JPEG naturally becomes the focus of investigation in digital forensics [5]. File carving is beneficial to data recovery and computer forensics, which can recover data without file system or fragmentation on disk, and also extract deleted or hidden fragmented fragments to become evidence. Early file Carving was mainly based on the header/footer of the file. It searched for the header character sequence "FF D8 FF E0" or "FF D8 FFE1", and the end of the file sequence was "FFD9". The data between the

*This work is supported by Projects of Shandong Province Higher Educational Science and Technology Program under Grant J18KA357, J16LN19, J17KB182, J18KA383;Shandong Province Soft Science Research Project under Grant 2018RKB01174; Doctoral Research Fund of Shandong Jianzhu University under GrantXNBS1810;Teaching Innovation project of Shandong University of Political Science and Law under Grant 2017JYB009; projects of Shandong University of Political Science and Law, Grant No. 2016Z03B, 2015Z03B.

header and the end Regard as a file. Foremost, an open source tool was a command line tool based on the recovery of the header and footer of the file. In recent years, many scholars and researchers have studied it from different perspectives and introduced other algorithms or ideas into it, forming many new theories.

JPEG file carving technology is generally divided into simple carving, header/maximum length carving, Smart carving, double-fragment gap carving [3], mapping carving [6], and graph theory carving [7]. In 2015, A.Pal [8] research team also studied in-depth the methods of monitoring the occurrence of image file fragments, and proposed a general document carving model, Smart Carver, Which divide file carving into three steps: pretreatment, reorganization and reorganization. In the pretreatment stage, data fragments are decompressed, classified and predicted in the collation stage, and in the reorganization stage, data fragments are reorganized into target files. The fragmentation granularity considered in the model is sector. In 2015, Y.B. Tang[9] provide a new pixel similarity method (CED) to judge if two data blocks are adjacent in the same JPEG image and a fragmentation point detection algorithm based on the algorithm; In 2017, M. Barni[10] provide to use CNNs for aligned and nonaligned double JPEG compression detection. In particular, they explored the capability of CNNs to capture DJPEG artifacts directly from images. Results show that the proposed CNN-based detectors achieve good performance even with small size images. In 2014, G.H. Na [11] proposed a fragment type judgment method based on byte distribution statistics, which achieved good results in the test of JPEG images. After analyzing the internal structure and content features of JPEG image file compression data. In 2014, B. Zhang [12] proposes a JPEG image carving algorithm based on forward matching distance, which makes full use of the content features of JPEG image file compression data.

In summary, most of the adjacent Pixel Similarity algorithms for JPEG images have high accuracy when the pixels are continuous and smooth, Low accuracy will occur when image contains sharp areas such as stripe, vertical bar and angle. The research on that problem needs to be improved urgently. In addition, most of the methods are based on sector to match adjacent pixels, which is inefficient in the current situation of soaring storage capacity and improving image quality. The accuracy and efficiency of JPEG file carving still need to be improved.

3 The proposed pixel similarity of MED_ED

3.1 Causes of fragmentation

Because of file system organization strategy, storage compatibility strategy or data hiding, files on storage devices are mostly stored in three forms: continuous storage, fragmented storage and incomplete storage.

- Frequent operation
  Frequent operations on traditional storage devices will lead to fragmentation of some files, like add, delete, modify, etc. For example (see Fig. 1), file A and file B are stored sequentially. When file A is deleted, the clusters of A (cluster is composed of sectors) is newly written to a smaller file C. At this time, there will be some space between file C and file B. If a large file D is written, the space between file C and file B cannot accommodate the file. At this time, file D needs to be divided into two parts, and fragmentation will occur.

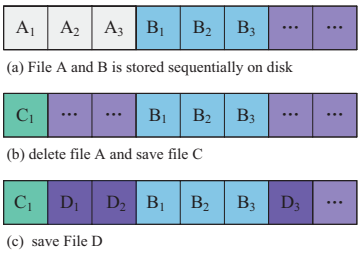


Fig 1:The cause of file fragmentation

- Natural fragmentation
  NAND storage media such as SSD are different from traditional disk storage media. In order to adapt to the common file system, Flash Translation Layer (FTL) was introduced. Which generally composed of three modules: address mapping, garbage collection and wear balance. Wear-leveling algorithm guarantees that storage devices are uniformly worn. Because of wear-leveling algorithm is proprietary and unknown to digital researchers, it is difficult to determine the correct sequence of file blocks. When file system metadata is lost, files on NAND disks will be completely disordered and fragmented.
- Image hiding
  Image hiding can divide a secret file into several smaller data blocks, and then distribute them in disorder among other data on disk in order to achieve the purpose of hiding and confidentiality. Hidden data can easily and nondestructively reorganize files according to the location and order of fragments, but for other users or forensics, it is very difficult to reorganize such fragmented data.

3.2 Related terminologies

Several terms will be used in the description of this paper, include cluster fragments/piece, fragmented file, segment, fragment point. The explanation is shown as follows (see Table 1).

Table 1: Related terms

Term	Definition
Cluster	It consists of one or more sectors, Different file system clusters have different sizes.
Fragments /Piece	Consisting of one or more clusters, part of a file.
Fragmented file	A file consisting of more than two fragments stored discontinuously.
Segment	Composed of adjacent fragments. it may or may not belong to the same file.
Fragment point	Breakpoint between two fragments.

3.3 Pixel Similarity of fragments

The adjacent pixels of images have strong correlation which taken by image devices. In most cases, if a picture contains other file data blocks, it will lead to decoding errors.

This is because adjacent pixels jump at the edge of data blocks. Because the local pixels of JPEG image are continuous and smooth in most cases, it is a common solution to solve the problem of fragmentation and disorder by calculating similarity of compressed pixels between JPEG image files. The pixel values at the junction of two data blocks can be compared after decoding. It shows two consecutive data blocks X, and The number of adjacent pixels are 7(see Fig. 2). If the pixels of two data blocks are smooth enough or meet a certain threshold through the pixel similarity checking algorithm, and there is no sharp area, then the two data blocks are adjacent.

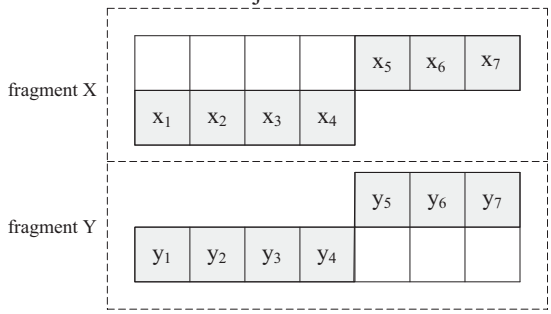


Fig.2: Pixel correlation of two adjacent fragments

3.4 Pixel Similarity algorithms

At present, most image carving tools or algorithms are based on PM, SoD, MED and other adjacent pixel prediction algorithms:

Pixel Matching (PM): judging whether adjacent pixels are equal, if equal, then the weight is added 1, the higher the weight is, the higher the matching degree is.

Sum Of Differences (SoD) [9]: judging the sum of the differences of RGB values between two adjacent data blocks as formula (1), the lower the SoD value, the higher the matching degree.

$$SoD = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \quad (1)$$

The value of Xi in block X is the RGB value of the pixel point, and Yi is the RGB value of the corresponding Xi in block Y, where i={1,2... n}, n is the width of the pixel.

Median Edge Detection (MED)[14,16]: a median edge detector, can use the pixel values above, left and upper left corner of a pixel to predict. Suppose x is the pixel to be predicted, A is the left pixel of x, B is the upper pixel of x, and C is the upper left corner of X as formula (2).

$$\hat{x}_{MED} \triangleq \begin{cases} min(a,b) & if\ c \geq \max(a,b) \\ max(a,b) & if\ c \leq \min(a,b) \\ a+b-c & otherwise \end{cases} \quad (2)$$

Euclidean distance(ED) [15]: is another similarity method commonly used in device independent color space to measure color difference as formula (3). ED has been successfully applied in document carving experiment. The experiment shows that it is more accurate than SoD. The lower the ED value, the higher the matching degree.

$$ED = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3)$$

3.5 Pixel Similarity of MED_ED

SoD and ED only focus on the similarity between adjacent pixels of the boundary, rather than the integrity and smoothness of the whole image. If two data blocks are similar (for example, both images have sky) or there are strips, vertical strips, angles (such as windows, branches, etc.), it is difficult to use SoD or ED to match correctly. MED uses the surrounding pixel values to predict the target pixels, which is better than SoD and ED in terms of local pixel integrity. In order to improve the accuracy and reliability of adjacent pixel computation, a new Euclidean distance adjacent pixel detection algorithm MED_ED based on median edge detector is proposed in this paper.

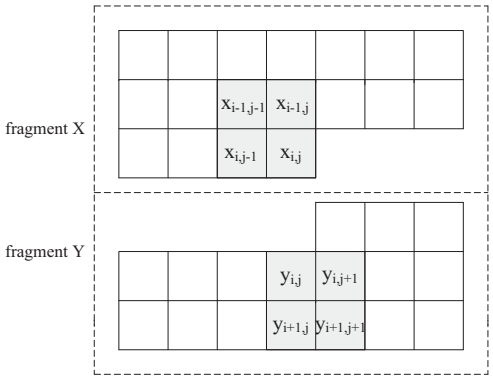


Fig.3: Illustration of computing MED_ED

Smooth regions in images generally have the same RGB values, and data fluctuations generally occur in areas with sharp areas. Despite this fluctuation, the variation pattern of RGB values in a small range is similar and stable, regardless of whether the region is in a smooth or a sharp area [9]. First, MED algorithm is used to predict the pixel values of fragment A and fragment B edges (see Fig. 3), , and then ED is used to calculate the Euclidean distance of the predicted adjacent pixel values. The definition of MED_ED (Euclidean distance based on Median Edge Detection)as formula (4), the pixel value of $x_{(i,j)}$ is calculation using (5),and $y_{(i,j)}$ is calculation using formula (6).

$$MED\_ED = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(\hat{x}_{i,j} - \hat{y}_{i,j})^2} \quad (4)$$

$$\hat{x}_{i,j} \triangleq \begin{cases} min(x_{i,j-1}, x_{i-1,j}) & x_{i-1,j-1} \geqq max(x_{i,j-1}, x_{i-1,j}) \\ max(x_{i,j-1}, x_{i-1,j}) & x_{i-1,j-1} \leqq min(x_{i,j-1}, x_{i-1,j}) \\ x_{i,j-1} + x_{i-1,j} - x_{i-1,j-1} & otherwise \end{cases} \quad (5)$$

$$\hat{y}_{i,j} \triangleq \begin{cases} min(y_{i,j+1}, y_{i+1,j}) & y_{i+1,j+1} \geqq max(y_{i,j+1}, y_{i+1,j}) \\ max(y_{i,j+1}, y_{i+1,j}) & y_{i+1,j+1} \leqq min(y_{i,j+1}, x_{i+1,j}) \\ y_{i,j+1} + y_{i+1,j} - y_{i+1,j+1} & otherwise \end{cases} \quad (6)$$

4 JPEG image fragment Carving model based on Pixel Similarity of MED_ED

Modern disk has large capacity and JPEG file always have high pixels, more complex data on the actual storage medium, unpredictable degree of disorder and incompleteness. In most studies, it is inefficient to compute the similarity of adjacent pixels using sector as binary data stream. For example, FAT32 file system cluster defaults to 4096 bytes. If

the cluster size is determined, the detection time of adjacent pixels can be greatly reduced, and the detection efficiency and accuracy can be improved. If the FAT table information contained in a 4-bit mirror file is opened for sectors, if the FAT table entry is 4 bytes, the FAT32 file system can be determined. The method of judging cluster size is as follows:

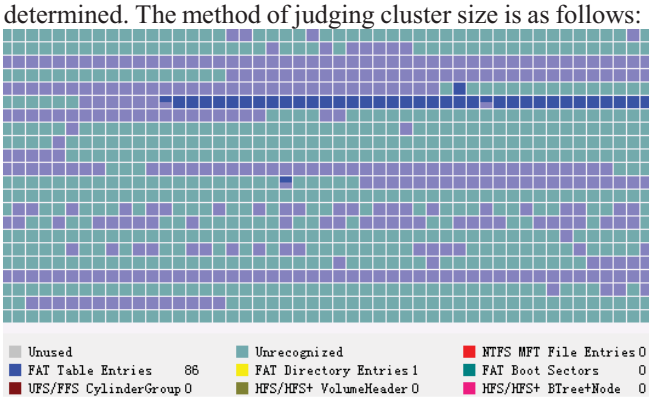


Fig.4: FAT information contained in a mirror file

4.1 Judging Cluster Size

Step1: Search the disk for DBR residual data, locate the 0x0D location, and read the number of bytes per cluster.

Step2: If DBR does not exist, flag sectors can be searched to predict the original cluster size. As shown in the figure, there are FAT table data blocks, and when opened, the FAT directory entries are found to be 4 bytes, which in most cases can be inferred as FAT32 file system, the default cluster size is 4K.

Step3: Verify the predicted cluster size, randomly extract 50 sectors within the cluster for entropy detection, and determine whether each sector data is the same type of data. The cluster size can be determined by validating correctly, while the cluster size can be set to 512 bytes in case of failure.

4.2 Fragment recognition

Generally, a large number of different types of files are stored on disk, such as JPEG, DOCX, XLSX, PDF, ZIP, MP3, etc. There are many algorithms for file type classification, including BFA, BFC, FHT [17], k-means, RoC, k-nearest-neighbors, NLP and so on. Some algorithms have achieved very high recognition rate, recognition rate of common algorithms (see Table 2). In this paper, we use tool Oscar to identify JPEG file fragments, which improved by Karresand [18]. Oscar which uses RoC features to recognize JPEG file clusters. Experiments show that the recognition rate of JPEG files reaches 99%.

Table 2: Recognition rate of common algorithms

Algorithm	Recognition rate
BFA	27.5%
BFC	45.83%
FHT	95.83%
RoC	97%

4.3 Group fragments into Segment

The identified fragments are stored in the original storage order of the disk, and the number of files to be carved is determined by searching for JPEG header markers. The header-missing file is not considered in this paper. The process of grouping algorithm is as follows:

Step1: Search for header and footer flags, search for header flags, jump to step 2, otherwise jump to Step3; stop if has no header or footer flag.

Step2: Determine the header Hi of the JPEG file, continue to search downward for the header and footer flag of the JPEG file, jump to Step3 if the header Fi is searched, jump to Step4 if the header H_{i+1} is searched, and stop without the header and footer flag.

Step3: Search for Fi at the end of the file, and use the data (including F_i) between the previous file header H_i or the starting position H_0 and F_i as a group of S_i (segments). Continue to search, if you find the header flag, jump to Step5, and jump to Step6 at the end of the search.

Step4: Search for other file header H_{i+1}, then the data block between file header Hi and file header H_{i+1} (excluding H_{i+1}) is a group of S_i. i++, jump to Step1;

Step5: Search the JPEG header H_{i+1}, then the data blocks between Fi and H_{i+1} (excluding H_{i+1}) are a set of S_{i+1}, i++, jump to Step2; Search the file footer F_{i+1}. The data blocks between F_i and F_{i+1} (including F_{i+1}) are a set of S_{i+1} and jump to Step1.

4.4 JPEG decoding

After determining the header in Segments, the data behind the SOS segment is decoded by clusters. The order of decoding in JPEG standard is from left to right and from top to bottom. Breakpoints in data blocks may occur anywhere in the JPEG image, including within the MCU. For smaller JPEG files, the MCU may appear after several lines, and for larger JPEG images, it may appear on the first MCU line. Until the offset is the size of the cluster, it is initialized as fragments $\{f_0\}$, and then decodes the next cluster, and uses the Pixel Similarity of MED_ED to determine whether it is adjacent.

4.5 Computation of Pixel Similarity

We use the Euclidean distance between fragments is calculated by the ED algorithm when the Pixels of image is smooth (Formula 3). If the pixel ratio of two data blocks is smooth enough and there is no jump, the neighborhood is represented. If the decoding is correct but the jump occurs, the improved MED_ED algorithm (Formula 4) is used to judge. The adjacency of Segments judgment also uses this method.

4.6 Recombination

When all JPEG Segments are arranged in the correct order, the final image reorganization is completed. The integrity of each part is verified by JPEG image format.

5 Experimental results and analysis

The SD card with 8G size is used to make test storage device to simulate the real usage scenario. After formatting the SD card, the real usage mode is simulated. Common file

types of different sizes, such as DOCX, TXT, XLSX, AVI, RMVB, are randomly collected and written until they are unable to write, and then some files are randomly deleted. Repeated 20 times, the simulation fragment generation principle naturally produces fragments, and the size and location of debris are random. Delete some data in SD card and write 100 pictures (10 JPEG pictures taken by mobile phone, 1920×1080 pixels, 90 downloaded by network, uniformly converted to 1024×768). Under the file system, 5 pictures are randomly selected and manually divided into 2 pieces and 5 pieces into 3 pieces. Making SD card image, ignoring the file system, using the sculpture complex model proposed in this paper, different pixel prediction algorithms are used for neighborhood prediction. The experimental results are shown in Statistical table of different adjacent pixel detection algorithms (see Table 3).

Table 3: Statistical table of different adjacent pixel detection algorithms

Algorithm	Number of Carved	Number of success	Success rate
SoD	85	81	81%
ED	88	86	86%
MED_ED	93	92	92%

The results show that the proposed MED_ED pixel similarity algorithm is more accurate than the SoD and ED algorithms, which can correctly carving fragmented JPEG files.

6 Conclusion

In this paper, the cause of fragmentation and the adjacent pixel similarity algorithm are studied, aiming at the poor accuracy of the JPEG image which contains sharp areas such as strips, vertical strips and angles. Most of the adjacent Pixel Similarity algorithms for JPEG images have Low accuracy will occur when image contains sharp areas. In order to solve this problem, we provide a new algorithm MED_ED based on median edge detector, according to the local stability and the characteristics of MED. Secondly, the design algorithm to judge the original cluster size, according to the Features of the file system, in order to solve the problem of inefficiency under the situation of the current storage device capacity soaring and high image quality. The experimental results show that the proposed method can effectively improve the accuracy and efficiency. In the future, we will study JPEG files with missing headers and explore ways to construct virtual headers according to their structures.

References

[1] Y.J. Zhang.A Course of Image Processing and Analysis(2nd Edition).Beijing：Posts & Telecom Press,2016.

[2] ifanr, On the Internet, a picture wins a thousand words,https://www.ifanr.com/476411, 2014.

[3] S.L. Garfinkel. Carving contiguous and fragmented files with fast object validation . Digital Investigation, 2007, 4: 2-12.

[4] A. Pal, N. Memon. The evolution of file carving[J].Signal Processing Magazine, IEEE,2009, 26 (2):59-71.

[5] M. Xu,L Huang,H P Zhang,J Xu,N Zheng.Recovery method for JPEG file fragments with missing headers.Journal of Image and Graphics，2013.18 (1) :24-35.

[6] M. Cohen. Advanced carving techniques. Digital Investigation . 2007, 4(3-4).119-128.

[7] N. Memon, A. Pal. Automated reassembly of file fragmented images using greedy algorithms.IEEE Transactions on Image Processing, 2006,15(2): 385-393.

[8] Digital Assembly.Smart Carver.http://digital-assembly.com/products/smartcarver-dc3/,2015.

[9] Y.B. Tang, J.B. Fang, K.P. Chow, S.M. Yiu, Jun Xu, Bo Feng,Qiong Li, Qi Han.Recovery of heavily fragmented JPEG files. Digital Investigation 18 2016, 108-117.

[10] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, S. Tubaro .Aligned and non-aligned double JPEG detection using convolutional neural networks. Vis. Commun. Image R. 49 ,2017,153–163.

[11] G. H. Na, Shim K S, Moon K W, et al. Frame-based recovery of corrupted video files using video codec specifications. IEEE Transactions on Image Processing， 2014, 23(2): 517-526.

[12] B.Zhang.JPEG Image Carving Algorithm Based on Forward Distance. Dalian:Dalian University of Technology,2014.

[13] Y.B.Yang,F.K.Wang. Wear Leveling Aware FTL for Hybrid Solid State Disks.Beijing: Chinese Journal of Computers.2018,41(10):2379-2393.

[14] M. J. Weinberger, G. Seroussi, G. Sapiro. The LOCO-I lossless image compression algorithm; principles and standardization into JPEG-IS. HPL-98-193, Nov. 1998.

[15] B. Li, L. Wang, Y. Sun, Q. Wang. Image fragment carving algorithms based onpixel similarity. In: Multimedia information networking and security(MINES), 2012 fourth international conference on. IEEE; 2012. 979-82.

[16] L.D. Hou, W. Zhang, K. Chen, et al., “Reversible Data Hiding in Color Image with Grayscale Invariance,” IEEE Trans. on Circuits and Systems for Video Technology, 2018.

[17] McDaniel M, Heydari M H. Content based file type detection algorithms.Proceedings of the 36th Annual Hawaii International Conference on System Sciences, IEEE, 2003:10 pp.

[18] Karresand M, Shahmehri N. Oscar—file type identification of binary data in disk clusters and ram pages.Security and Privacy in Dynamic Environments.Springer US,2006:413-424.

ATLANTIS PRESS

Advances in Engineering Research, volume 166

3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)

Research on Recognition and Carving Techniques of Monitoring Video Fragments based on Structural Features

Xu Chang[a], Jian Wu[b], Shanshan Pei[c]

Shandong University of Political Science and Law, Jinan, Shandong Province, China

[a]changxumail@163.com, [b]jinanwujian@163.com, [c]peishan616@163.com

Keywords: Fragments Recognition; Video Carving; AVI

Abstract. The purpose of this paper is to research how to identify and restructure the binary data fragments which were stored on device efficiently and accurately. Take widely used AVI video file format for example, we proposed a method to identify fragments and carve them into files based on structural features, after the studying of carving technical principle of video and AVI file structure. At last, make a simulation to test the method. The results show that it can recover AVI video files from storage devices without Meta system information and achieved very high percentage of accuracy.

Introduction

With the development of information technology, equipment for road monitoring or vehicle video forensics is widely used. As an important source of evidence, monitor video plays an important role in order management and traffic accident handling. Usually, the data of monitor video is huge, and it is often stored in sequential or non-sequential form in storage devices. Because of limited storage space, repeated erasures are needed during use, which will make the data severely fragmented and caused a great obstacle to forensic.

The traditional recovery method relies on system meta-information. It is difficult to recover when the meta-information is incomplete, even if the data is not covered. Data carving technology overcomes the shortcomings of traditional data recovery techniques. It attempts to recover and reconstruct files from an unstructured raw disk image binary data stream, without relying on the file system of the source disk image, carving the "shape" of certain documents from a digital "plane" under the conditions of automatic or minimal manual intervention [1]. This article focuses on AVI format video files that are widely used in surveillance video and driving recorders. It focuses on how to efficiently and accurately identify and extract target fragments in seemingly indistinguishable binary fragmented data, and reorder and reconstruct the fragment files.

As an important part of computer forensic analysis, its tools or methods has continuously improved, and theoretical research is constantly deepening and improving[2]. Video files are complex and have a large amount of data. In the early days, There is less research on video files, Most of the research focus on case analysis which taken by specific video tools. The data carving methods are not universal, such as WS. Van Dongen [3] recovers the video which recorded by Samsung Video Recorder; Huang Bugen [4] recovers the MP4 video in the SANYO camera.

In recent years, in order to cope with the various types of document processing and complex fragments carving requirements, many researchers have introduced related field theories from different perspectives. For example, foreign researchers have introduced graph theory and greedy algorithms into document carving, which brought many new carving theory and methods, enriched carving theory. Andrew B. Lewis [5] researched the compressed data and proposed a method of carving MPEG video. In 2013, Huang Wei [1] used a combination of keyword and binary gap carving to complete the carving; in 2014, Gi-Hyun Na [6] proposed a frame-based MPEG video carving method, carving MPEG video successfully. In 2015, Liu Liying [7] proposed an AVI data extraction method based on the internal features of AVI files and an AVI fragment reorganization method based on the frame length information in each frame and file index. Li Zichuan [8] proposed a debris-level search technique for video file data by recording time in WFS non-universal file system. Xia Rong et

al. [9] proposed a method for efficient video recovery forensics using WinHex scripting tool. In general, the current research on the theory and tools has made great progress. However, there are still many problems that need to be resolved in the face of a wide range of equipment and document types and different levels of damage.

Analysis of AVI Format

AVI is a digital video and audio file format that conforms to the RIFF file specification. It is the most complex RIFF file currently in use, and can simultaneously store audio and video data. RIFF files use four-character code (FOURCC) to characterize data types. The structure of the entire AVI file is: a RIFF header, a list for describing media stream formats, a list for storing media stream data, and an optional index block. The expanded structure of the AVI file is roughly as shown in Fig.1:

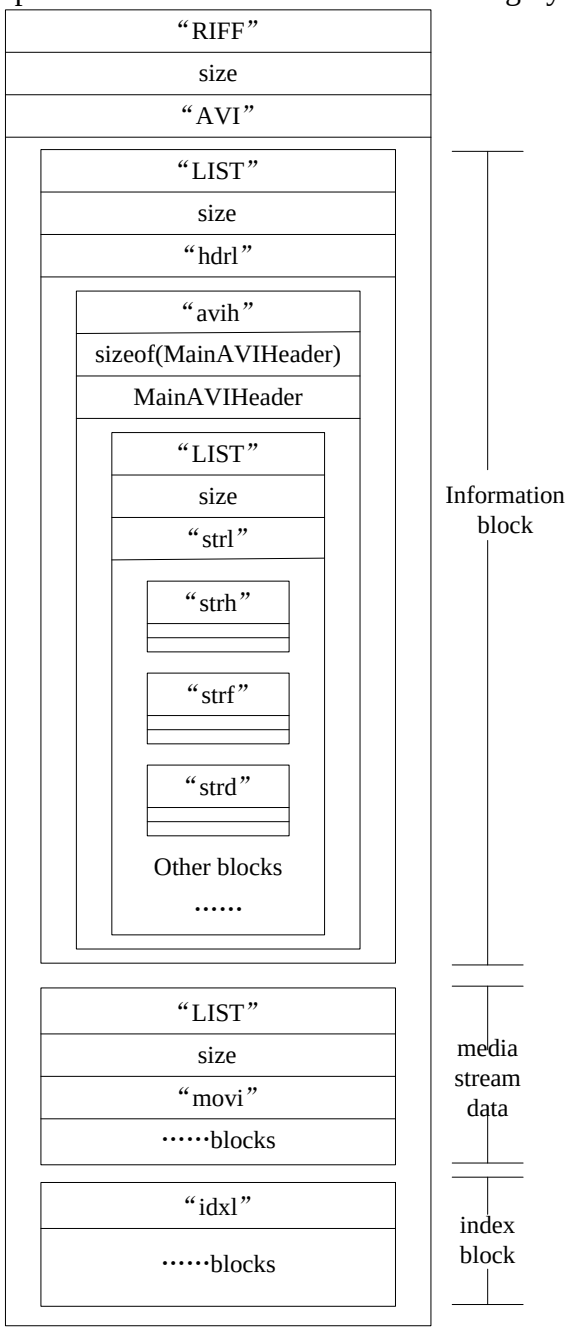
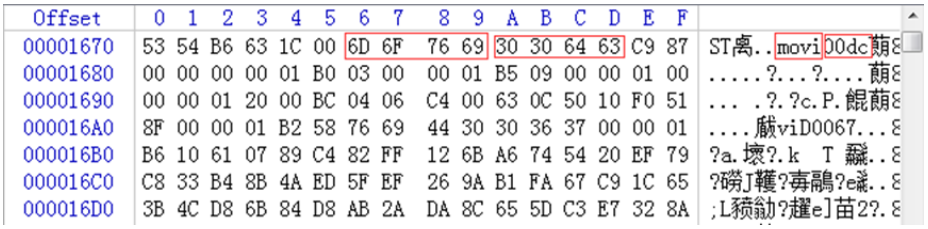


Fig.1 Expanded structure of AVI

The List block with ID “hdrl” is used to describe the format information of each stream in the AVI file (each media data in the AVI file is called a stream). The list nests a series of blocks and sub-lists, including an "avih" block and one or more "strl" sub-lists for recording global information about AVI files (eg, number of streams, width and height of video images). Each "strl" sublist contains at least

one "strh" block containing the header information of the stream and a "strf" block that specifies the specific format of the stream, "strd" block (requires configuration information for the codec to be saved) and "strn" block (The name of the save stream) is optional.

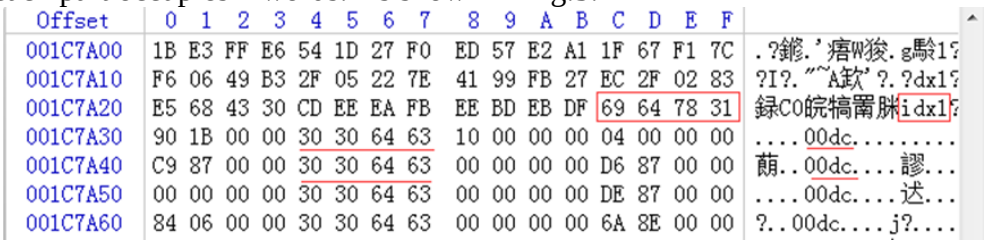
The list with ID "movi" is used to save the real media stream data. When the AVI file contains multiple streams, the data block uses FOURCC codes to represent the types of data blocks of different streams. The four-character code consists of a 2-byte type code and a 2-byte stream number. The standard type codes are defined as follows: "db" (uncompressed video frame), "dc" (compressed video frame), "pc" palette change, "wb" (audio data), as shown in Fig.2 "00dc" Compress video frames.



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00001670	53	54	B6	63	1C	00	6D	6F	76	69	30	30	64	63	C9	87
00001680	00	00	00	00	01	B0	03	00	00	01	B5	09	00	00	01	00
00001690	00	00	01	20	00	BC	04	06	C4	00	63	0C	50	10	F0	51
000016A0	8F	00	00	01	B2	58	76	69	44	30	30	36	37	00	00	01
000016B0	B6	10	61	07	89	C4	82	FF	12	6B	A6	74	54	20	EF	79
000016C0	C8	33	B4	8B	4A	ED	5F	EF	26	9A	B1	FA	67	C9	1C	65
000016D0	3B	4C	D8	6B	84	D8	AB	2A	DA	8C	65	5D	C3	E7	32	8A

Fig.2 Data of media stream

The index block is followed by media stream data, which represented by the AVI file “idx1”. The index block uses consecutive 16 bytes to index each media data block in the AVI file to record the label, attribute, location of the sub-block relative to the "movi" list and sub-block length in the data block, and each part occupies 4 words. As shown in Fig.3.



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
001C7A00	1B	E3	FF	E6	54	1D	27	F0	ED	57	E2	A1	1F	67	F1	7C
001C7A10	F6	06	49	B3	2F	05	22	7E	41	99	FB	27	EC	2F	02	83
001C7A20	E5	68	43	30	CD	EE	EA	FB	EE	BD	EB	DF	69	64	78	31
001C7A30	90	1B	00	00	30	30	64	63	10	00	00	00	04	00	00	00
001C7A40	C9	87	00	00	30	30	64	63	00	00	00	00	D6	87	00	00
001C7A50	00	00	00	00	30	30	64	63	00	00	00	00	DE	87	00	00
001C7A60	84	06	00	00	30	30	64	63	00	00	00	00	6A	8E	00	00

Fig. 3 Index block

The List block with the ID "JUNK" is used to represent special data. It is used to fill internal data. The application program generally ignores the actual meaning of these data blocks.

Fragments Recognition and Carving based on Structural Features

The carving process for video files is divided into two stages: data extraction stage and data fragment reconstruction.

Identification and Extraction of Fragments. Scan the original image in terms of bytes, and extract header information, media stream data information, and index information of the suspected AVI file according to the characteristics of the data structure. During the search, the data is located by flag bits, and each frame data is combined with the data length according to the searched flag bit and data length to determine the integrity of the data. Save the fragmented data if the data is complete.

Identification and Extraction of Header Information. AVI video files are complex in structure and can be extracted from data blocks in the data area or extracted from header information. The break zone of the file also appears mainly in the data area "movi" list. The list of file headers is usually considered as complete and generally does not appear fragmented. Therefore, extracting the header information can get the size and attribute information of the file, which helps to carry out the work of carving.

After the “RIFF” flag is located, scanning continues. If the subsequent data can satisfy the AVI file header structure feature, the data is considered to be a complete AVI header. Search for "52 49 46 46" in the image to search for file header information. If it exists, skipping 12 bytes can find the next flag bit "LIST", and turn it to "INFOISFT" and other flags until the "movi" flag. Starting from the "RIFF" flag and before the "movi" flag is a typical AVI file header. After the header information is extracted, analyze and save the detailed structure.

Identification and Extraction of Media Stream Data. The AVI file is a complex audio and video file. The main data area can be divided into video streams, audio and video alternating mixed streams and other forms. In general, it consists of a video stream and an audio stream. The two stream numbers are "00" and "01". The video data FOURCC is encoded as "db" or "dc" and the audio FOURCC is encoded as "wb". Therefore, the suspect data frame can be located by searching for the FOURCC flag. This article uses the following methods to identify and extract data fragmentation.

Step1: Search FOURCC code, record the offset O, jump to Step2;

Step2: After extracting the FOURCC code, four bytes of sub-data length Li, jump to Step3;

Step3: Jump to the (O+Li) offset position and judge the FOURCC code for the next sub data exist or not. If not exist jump to Step1. If exist, the data block before saving the initial offset O to the next FOURCC code position is a fragment Fi, which is stored in the fragment library, fragment length and the relative position of “movi” are recorded at the same time.

Extraction of Index Block. The FOURCC code of the AVI index tag is "idx1". This data has a small proportion in the AVI file and is not easily divided into fragments in general. Therefore, this article directly searches for the location "idx1" and extracts the index data based on the index length information.

Reconstruction of Fragments. Carving the video fragments needs to reconstruct three parts of the file, contains header, media stream data and index block. First, read the extracted file header information and analyze the length; second, read the index block information, and sequentially extract 16 words of tag, attribute, relative position and length information. Compare the relative positions and data lengths of the data fragments with the data from fragment library. If match, add the flag "hit". If not, generate empty data and add the flag "miss" to construct media stream data area information; third, read additional index block data again; finally reconstruct each part of data required by the video file and compare it with the data length in header. If match, means carve completely. The specific reconstruction process is shown in Fig.4.

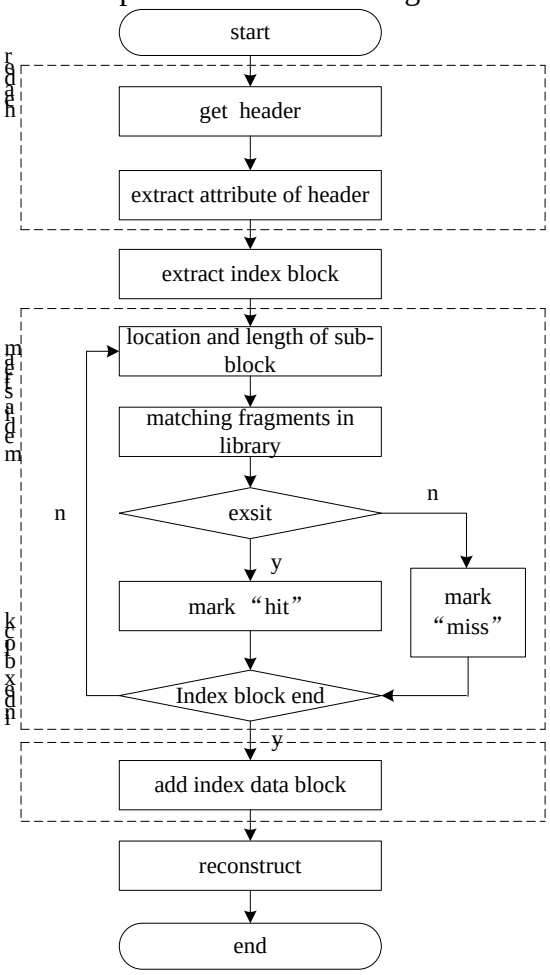


Fig.4 Flowchart of fragments reconstruction

Experiment and Result Analysis

This article uses a 4G U disk to make a test disk to simulate a real-life usage scenario. This U disk has been used for more than one year every day and has not been formatted during this period. Randomly collects 200 commonly used file types such as docx, txt, jpg, avi, etc. Including four avi files, followed by Cap1.avi, Cap2.avi, Cap3.avi, Cap4.avi. Then read, write and delete, In order to generate file fragments naturally. Ensure the fragment size and locations are random. During the experiment, the U disk file system is ignored, the target fragment is extracted from the "undifferentiated" binary fragmented data, and the video file based on structural features is reconstructed to recover the required video files. After the experiment, the four files were successfully carved and played normally, as shown in Fig.5.

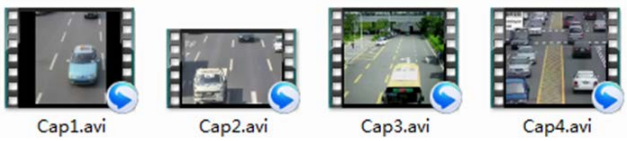


Fig.5 successfully carved AVI video files

The recovered file is not significantly different from the source file by manual observation. Using Winhex to view the data size and comparing with the source file, the file carving effect is shown in Table 1.

Table 1. Carving result statistics

File Name	**Comparison before and after recovery**			
	File size after carving (bytes)	***Original file size***	***Miss number***	***Carving rate***
Cap1.avi	3248144	3,248,562	0	99.98%
Cap2.avi	9205533	9,226,192	1	99.77%
Cap3.avi	1838587	1,873,348	1	98.14%
Cap4.avi	10187084	10,187,920	2	99.99%

In the simulated mirroring environment, the target video file is relatively small, and the degree of fragmentation is lighter, so the recovery effect is better. According to the experimental data analysis, the proposed method based on the structural characteristics of the video file carving complex data carving is feasible and effective. Especially suitable for non-file system storage devices, the data is relatively complete fragmentation data. For fragmented data resulting from incomplete coverage, only the identified fragmented files can be used to maximize the carving based on the existing structural features. Like the file index header and index block are completely covered at the same time, the extracted condition cannot be recognized.

Conclusion

AVI format is widely used in video surveillance applications. The research on carving of AVI video files is of great significance and it is helpful for the development of forensics and electronic recovery technologies. Due to the wide variety of existing file types, it is difficult to implement universal file reconstruction tools that are currently applicable to all files, and more cases are studied for specific types of data. The next step will be to further study the case where the structural features are incomplete and the fragments are out of order. Identify fragments depend on the content features and design algorithm which are used to implement the judgment and reorganization of adjacent fragment data.

Acknowledgment

This work was funded by the Science and Technology Project of Universities in Shandong Province, with the project approval number: J16LN19; the university-level project of Shandong University of Political Science and Law, and the project approval numbers are: 2016Z03B, 2016Z04B, 2015Z03B,

2016JYA001; Key Laboratory of Evidence-Identifying in Universities of Shandong (Shandong University of Political Science and Law).

References

[1] Huang Wandi, Wang Zhongxia, Wu Zhendong. carving of AVI files in electronic forensics. Chinese Journal of Forensic Sciences, 2013 (3), pp. 57-61.

[2] Nicholas Mikus. An analysis of disc carving techniques . Master thesis. Monterey: Naval Postgraduate School, 2005, pp.23–25.

[3] Van Dongen W S. Case study: Forensic analysis of a Samsung digital video recorder. Digital Investigation, 2008, 5(1): pp.19-28.

[4] Huang Bugen, Huang Zheng, Liu Jianjun. Restoration of Deleted Video in SANYO Digital Camera. Journal of Internet Information and Security, 2011, pp 143-155.

[5] Lewis A B. Reconstructing compressed photo and video data [D]. Cambridge: University of Cambridge, 2012.

[6] Na G H, Shim K S, Moon K W, et al. Frame-based recovery of corrupted video files using video codec specifications. IEEE Transactions on Image Processing, 2014, 23(2), pp. 517-526.

[7] Liu liying. A study of AVI Video Carving [D], Nanjing University of Posts and Telecommunications, 2015.

[8] Li Zichuan. Research on Data Search and Recovery Technology of WFS Video Surveillance System, Journal of China Interpol College, 2015, pp.42-45.

[9] Xia Rong, Wu Bin, Yuan Wenqin. Research on surveillance video recovery technology, industry and application security, 2017, pp. 126-129.

Neural Networks 184 (2025) 107070

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

CMFX: Cross-modal fusion network for RGB-X crowd counting

Xiao-Meng Duan, Hong-Mei Sun *, Zeng-Min Zhang, Ling-Xiao Qin, Rui-Sheng Jia *

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

ARTICLE INFO

Keywords:
RGB-X crowd counting
Cross-modal fusion
Feature rectification
Feature decoding

ABSTRACT

Currently, for obtaining more accurate counts, existing methods primarily utilize RGB images combined with features of complementary modality (X-modality) for counting. However, designing a model that can adapt to various sensors is still an unsolved issue due to the differences in features between different modalities. Therefore, this paper proposes a unified fusion framework called CMFX for RGB-X crowd counting. CMFX contains three core components: fast feature aggregation module (FFAM), cross-modal feature interaction module (CFIM), and cross-modal feature decoding module (CFDM). Specifically, FFAM aims to enhance the fusion representation capability of low-level multimodal features through lightweight mixed attention. CFIM can fully realize the interaction and fusion of high-level features by rectifying the feature information of two modalities and exploring their potential correlations. In addition, CFDM employs a novel graph convolution block for refining and preserving cross-modal features at high-level and low-level. To validate CMFX, this paper unifies, for the first time, two modalities complementary to RGB images, namely depth and thermal. After extensive experiments on three public datasets RGBT-CC, DroneRGBT, and ShanghaiTechRGBD, we found that CFMX performs excellently in these two multimodal combinations. Code: https://github.com/Duanxm9/CMFX.

1. Introduction

Crowd counting is a pivotal field in computer vision, attracting more and more researchers' attention. It seeks to estimate the number of individuals within a specific video or image to meet the urgent needs in urban administration, public safety, transportation planning, etc. Currently, significant progress has been made in generating crowd density maps using optical information from RGB images. However, solely focusing on RGB images fails to provide detailed information about the crowd under specific conditions, resulting in a decline in counting performance. For example, crowds in RGB images are almost difficult to recognize under conditions of occlusion or poor illumination, and direct detection of crowds from RGB images becomes difficult. Therefore, with the advancement of sensor technology, different sensors have attracted attention for their ability to provide richer information for RGB images. For instance, depth images can provide more precise information about the shape, distance, and relative position of the human body, while thermal images can capture the thermal characteristics of a target through specific infrared imaging. However, existing multimodal methods are only applicable to a single modality and cannot be well applied to multiple modal combinations. Therefore, constructing a unified framework that can fuse RGB images and different modal images still poses a challenge.

Recently, with the continuous advancement and extensive application of depth and infrared cameras, cross-modal crowd counting methods have become increasingly popular. Compared to RGB images, depth images can provide distance information between the camera and the objects for each pixel. Therefore, researchers have proposed many RGB-D cross-modal methods. For example, Lian et al. (2021) designed a dual-path guided detection network (DPDNet) for crowd counting by using RGB-D data. Zhang et al. (2021b) designed a novel RGB-D fusion method that is a blend of RGB and depth images for training. Li et al. (2022a) introduced a method called CmCaF, which efficiently combines the complementary features from RGB images and depth images through cycle-attention. Liu et al. (2023) designed a collaborative cross-modal attention network called CCANet to fully leverage feature information of both modalities. With the advancement of Binocular Intelligent Integrated Thermal Imaging cameras, the combination of RGB images and thermal images has a significant impact on crowd counting. For example, Liu et al. (2021) designed a cross-modal collaborative representation learning framework that effectively learns complementary information between different modalities. Li et al. (2023) designed a network called CSA-Net, which focuses on bi-modal information fusion while aggregating multi-scale contextual information. Zhou et al. (2022)

* Corresponding authors.
E-mail addresses: shm0221@163.com (H.-M. Sun), jrs716@163.com (R.-S. Jia).

https://doi.org/10.1016/j.neunet.2024.107070
Received 28 February 2024; Received in revised form 20 November 2024; Accepted 17 December 2024
Available online 18 December 2024

designed a dual-branch network, where the data augmentation module fuses complementary features in the bi-modal modality to combine various rich receptive fields. Zhou et al. (2023) proposed a network called (MC^{3}Net), which uses ConvNext as its infrastructure to handle three input streams from RGB, T, and RGB-T. Overall, the aforementioned RGB-D and RGB-T multimodal fusion methods, despite employing a simple multimodal interactive fusion strategy, are typically fused for a specific single modality, making it difficult to extend to encompass combinations involving other modalities. Therefore, to flexibly integrate information between different modal pairs and leverage the potential of complementary features across modalities, designing a unified RGB-X crowd counting network would be of significant advantage.

Thus, we propose a generalized network called CMFX for RGB-X crowd counting. Specifically, CMFX contains two substreams: RGB modality stream and X-modality stream. Firstly, for simple and efficient fusion of low-level features, this paper introduces a fast feature aggregation module (FFAM) that accomplishes the interaction and fusion of the first two layers of features through channel and spatial correlations. Secondly, to facilitate effective interaction and fusion of high-level cross-modal semantic information, this paper proposes a cross-modal feature interaction module (CFIM) to mutually rectify the bi-modal feature information and explore the potential correlations and interactions between modalities. Furthermore, to sufficiently aggregate high-level and low-level features, this paper proposes a cross-modal feature decoding module (CFDM). This module utilizes a novel graph convolution block to address the complementarity between semantics and details of features. In summary, the network not only performs multimodal information fusion at the channel and spatial levels but also cross-modal fusion at the sequence-to-sequence level, thus realizing the generalization capability of the model. In general, the main contributions are as follows:

1) We design a universal RGB-X crowd counting framework called CMFX, which can be uniformly employed for RGB-D and RGB-T multimodal fusion.
2) We propose a fast feature aggregation module (FFAM) to accomplish the fusion of low-level features simply and efficiently, along with a cross-modal feature interaction module (CFIM) that rectifies high-level cross-modal information and explores their potential correlations for better integration and adaptive fusion of high-level semantic features.
3) We develop a cross-modal feature decoding module (CFDM) constructed using an efficient graph convolution block designed to further integrate high-level and low-level features. The module both preserves long-range attention at each stage and highlights salient features by suppressing irrelevant regions to improve counting accuracy.
4) CMFX is considered and evaluated in RGB-D and RGB-T tasks, and extensive experiments demonstrate that it can be applied to various modal combinations with optimal performance.

Important symbols and their descriptions in this paper are listed in Table 1.

Table 1
Important symbols and their descriptions.

Symbols	Description	Symbols	Description
FFAM	Fast feature aggregation module	AAP	Adaptive average pooling
CFIM	Cross-modal feature interaction module	CA	Channel attention
CFDM	Cross-modal feature decoding module	SA	Spatial attention
GCB	Graph convolution block	F_R^i	Intermediate features from RGB modality
MAE	Mean absolute error	F_X^i	Intermediate features from X-modality
RMSE	Root mean squared error	D^i	Feature obtained through FFAM
GAME	Grid average mean absolute error	D^j	Feature obtained through CFIM
PSNR	Peak signal-to-noise ratio	F_D	The predicted density map
SSIM	Structural similarity	L	Loss function
AMP	Adaptive max pooling	w/o	Without

2. Related works

2.1. Attention mechanism

The attention mechanism is vital in a range of computer vision tasks, such as crowd counting, and image classification. It adjusts the model's focus on the input by assigning weights to different features or positions, enabling it to focus more on important information. This enhances the robustness and generalization of the model. Chen et al. (2018) introduced a reverse attention-guided network to improve the model by exploring the missing detail information. Hu et al. (2018) designed a block called ``Squeeze-and-Excitation (SE)'', which flexibly concerns the relationship between channels. The introduction of SE has been a key catalyst in propelling the advancement of attention mechanisms. For instance, Wang et al. (2020) formulated an Efficient Channel Attention Module (ECA), which can realize cross-channel interactions by utilizing the channel attention in SENet, thus effectively reducing the complexity of the model. Despite the ability of channel attention to focus on important feature maps, it overlooks crucial positional information. To complement the channel attention information, Chen et al. (2017) proposed a novel network (SCA-CNN) that integrates spatial and channel attention to better focus on feature maps. Woo et al. (2018) designed a CBAM that captures important information in the feature map by combining spatial and channel attention. Li et al. (2022b) designed a network called IA-MFFCN, which employs reverse attention to minimize the effect of noise on counting accuracy. Hu et al. (2022) proposed a network called Le-SKT, in which modules use attention mechanisms to enhance salient information. For RGB-D and RGB-T tasks, attention mechanism applies as well. For instance, Li et al. (2022a) proposed a model that includes cycle-attention to efficiently fuse bi-modal feature information. Tang et al. (2022) designed a three-branch network using an Information Improvement Module to adaptively fuse multimodal information. The experimental results of the above methods demonstrate that the simultaneous application of channel attention and spatial attention yields the best performance. However, efficiently utilizing two attention mechanisms for feature fusion at low levels is still worth exploring. Therefore, this paper designs FFAM, a channel and spatial hybrid attention constructed at minimal cost to enhance the representation of low-level cross-modal information.

2.2. Graph convolutional networks

Recently, Graph Convolutional Networks (GCNs) have gained significant attention in deep learning and have been extensively utilized for diverse computer vision applications. For instance, Han et al. (2022) first designed a network called Vision GNN (ViG), which uses graph convolutional networks to process images. This architecture utilizes the K-nearest neighbors (KNN) algorithm to connect all patches in the segmented image. This operation not only captures long-range dependencies similar to those in Transformer but also irregular and complex objects. Landrieu and Simonovsky (2018) proposed the concept of a superpoint graph, which utilizes edge features as nodes to encode contextual relationships in a 3D point cloud. Michieli et al. (2020) introduced a module based on the adjacency graph, which considers the detected objects and the relationships between them as nodes and edges of the graph. Rahman and Marculescu (2023) proposed a network called G-CASCADE, which refines multi-level feature maps generated by hierarchical transformer encoders through efficient graph convolutional

blocks. This design allows the model to enhance local and global features while maintaining long-range dependency information. Pan et al. (2024) introduced a model called GETANet, which effectively integrates contextual information from RGB and thermal images using a Dual Graph Convolutional Module (DGCM), thereby improving the accuracy of crowd counting tasks. The application of Graph Convolutional Networks (GCNs) in this framework enhances the model's understanding of spatial relationships between individuals. Experimental results from existing methods demonstrate that the use of graph convolutional blocks is an effective approach for refining spatial features. Therefore, to fully integrate the fusion information obtained from different stages of the backbone network, this paper designs CFDM that employs graph convolution to consider the spatial relationships between image features. This module not only preserves the long-range dependency information but also suppresses irrelevant background information to efficiently predict crowd density maps.

2.3. Multi-Modal crowd counting

Although previous methods for crowd counting mainly relied on RGB images and achieved remarkable success, they are insufficient to effectively solve problems in complex situations. Therefore, to obtain accurate density maps, more and more researchers are focusing on multimodal fusion methods, such as RGB-D and RGB-T approaches.

2.3.1. RGB-D crowd counting

Different from RGB images, depth images offer structural and spatial position information of crowds to overcome the constraints faced by RGB images in unconstrained scenes. Shi et al. (2019) designed a network named PACNN that utilizes perspective information of the crowd as additional knowledge to detect changes in the crowd. Yang et al. (2019) designed a network called DECCNet. This network utilizes a bidirectional cross-modal attention mechanism to enhance counting and is considered as an approximate RGB-D counting solution. Lian et al. (2019) proposed a method to improve detection networks using density maps. At the same time, they released a dataset called ShanghaiTechRGBD, which greatly advances the progress of cross-modal crowd counting tasks. Li et al. (2022a) introduced a model that fuses RGB and depth features using cycle-attention and designed a new FC supervision to optimize the model. Liu et al. (2023) introduced a two-stream network called CCANet, which adaptively fuses multimodal features.

2.3.2. RGB-T crowd counting

Different from RGB images, thermal images excel in adapting to low illumination as they capture thermal radiation emitted from the surfaces of objects, effectively compensating for the limitations of RGB images. Peng et al. (2020) proposed a dataset called DroneRGBT and designed a multimodal network (MMCCN). Zhang et al. (2021a) proposed a network called I-MMCCN containing Block Mean Absolute Error loss to improve MMCCN. Liu et al. (2021) proposed a dataset called RGBT-CC while developing a framework for cross-modal collaborative representation learning to comprehensively capture multimodal information. Tang et al. (2022) proposed a TAFNet, which is an adaptive fusion network containing three branches of RGB, T, and RGB-T. Li et al. (2023) designed a network called CSA-Net, which is not only capable of fusing information from RGB modality and thermal modality but also of aggregating multi-scale information. Pan et al. (2023) proposed a network called CGINet, which performs cross-modal fusion by focusing on features at various levels.

Although the aforementioned methods have achieved remarkable results, most of them are only applicable to single sensors and cannot work effectively across different sensors. For instance, CmCaF (Li et al., 2022a) and CCANet (Liu et al., 2023), designed for RGB-D crowd counting, perform less well for RGB-T crowd counting. Similarly, CSA-Net (Li et al., 2023) and MC^{3}Net (Zhou et al., 2023), designed for RGB-T crowd counting, perform poorly in RGB-D crowd counting. Therefore, we propose a unified framework that can be effectively applied to different multimodal combinations. Compared to existing methods, CMFX can fully utilize the potential of diverse modal features, which is crucial for handling RGB-X crowd counting with various complements and uncertainties.

3. Proposed method

We first present the proposed model followed by detailed descriptions of the FFAM, CFIM, and CFDM. Finally, we also describe the loss functions.

3.1. Overview

This paper proposes a unified dual-branch structure called CMFX. Its general framework is shown in Fig. 1. CMFX consists of two CNN-based branches: one for extracting features from RGB images and the other for extracting features from X-modality images. Considering that features extracted at different levels contain different information, this paper designs FFAM for the fusion of low-level multimodal features and CFIM for the interaction and fusion of high-level multimodal features. And CFDM performs the final fusion of multi-level features to achieve the final density map estimation and counting. The specific framework is shown in Fig. 2.

From Fig. 2, firstly, given two inputs to CMFX, namely RGB image ($F_R \in \Re^{3\times H\times W}$) and X image ($F_X \in \Re^{1\times H\times W}$), we input them into two similar VGG-19 networks for feature extraction. Due to the differences between different modalities, direct fusion may lead to a decrease in counting accuracy. Furthermore, there are also differences in information across different levels. The low-level features primarily capture the details and textures of the image, while the high-level features emphasize the semantic information. Therefore, to solve these differences, this paper designs two different fusion modules. Specifically, in the first two layers of the backbone network, this paper proposes FFAM to interact and fuse shallow features simply and swiftly, which not only reduces computational complexity but also improves counting performance. When dealing with the last three layers of deep features, this paper designs CFIM in the middle of the two branches, which firstly uses the bimodal information to rectify each other, and then explores the potential correlations between them to fully facilitate the interaction and fusion of cross-modal information. Furthermore, at the end of the network, this paper employs a novel graph convolution block to integrate the fused feature maps obtained from the above two modules to generate the final density map. In the remaining part, this paper provides detailed descriptions of each module and explains the loss functions. Also, we will use X to denote the complementary modality, which can be depth images or thermal images.

3.2. Fast feature aggregation module (FFAM)

When extracting the low-level feature maps of the network, they may include basic information such as edges, colors, and textures. However, employing intricate fusion operations can significantly increase computational complexity while simultaneously reducing precision. Therefore, to enhance the efficiency of the model, this paper proposes FFAM, which can effectively highlight crowd information and weaken complex background information, so that the low-level features can be better interacted and fused. The structure of FFAM is depicted in Fig. 3.

Specifically, the module comprises channel attention $CA(\cdot)$ and spatial attention $SA(\cdot)$. Given intermediate features $F_R^i \in \Re^{C\times H\times W}$ and $F_X^i \in \Re^{C\times H\times W}$ from two modalities, where $i = 1, 2$ represents the first two layers of the backbone network, we concatenate the inputs F_R^i and F_X^i along the channel dimension and then apply a 3×3 convolutional block to generate the F_f^i, which is used for the input of channel attention. The

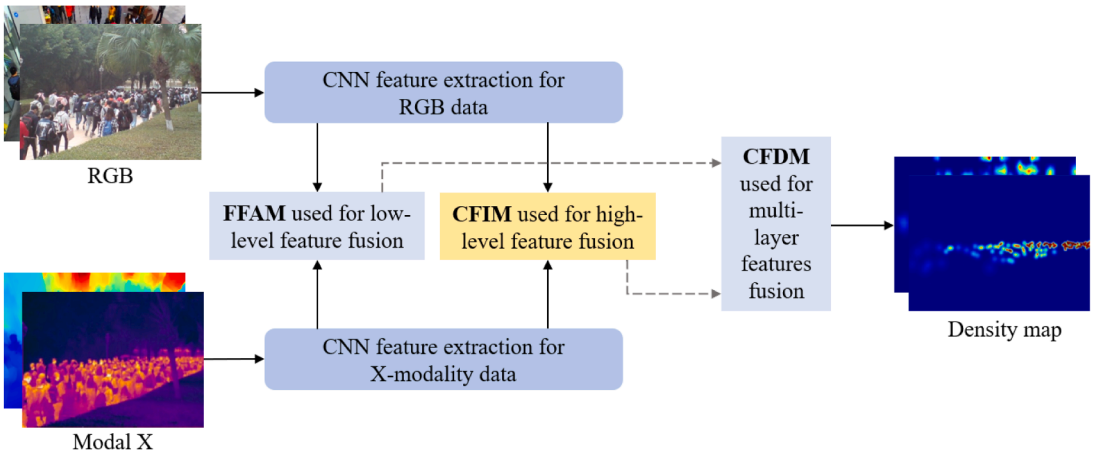


Fig. 1. The general framework of CMFX.

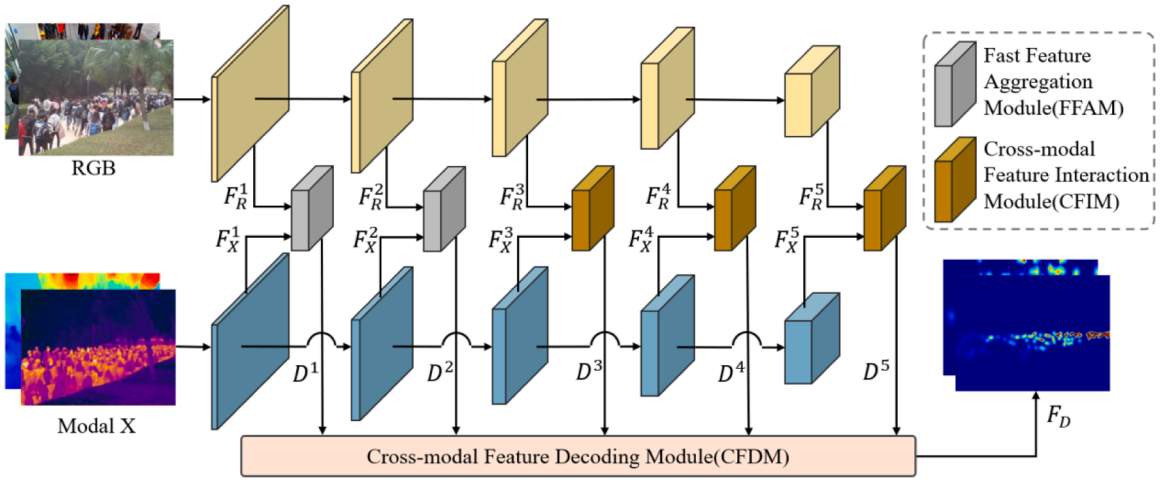


Fig. 2. Architecture of CMFX. This network comprises three modules: FFAM, CFIM, and CFDM. FFAM is utilized for processing the features of the first two layers of the network. CFIM is employed for the interaction and fusion of high-level features. CFDM is used to integrate multiple cross-modal feature maps obtained from FFAM and CFIM to obtain accurate density maps.

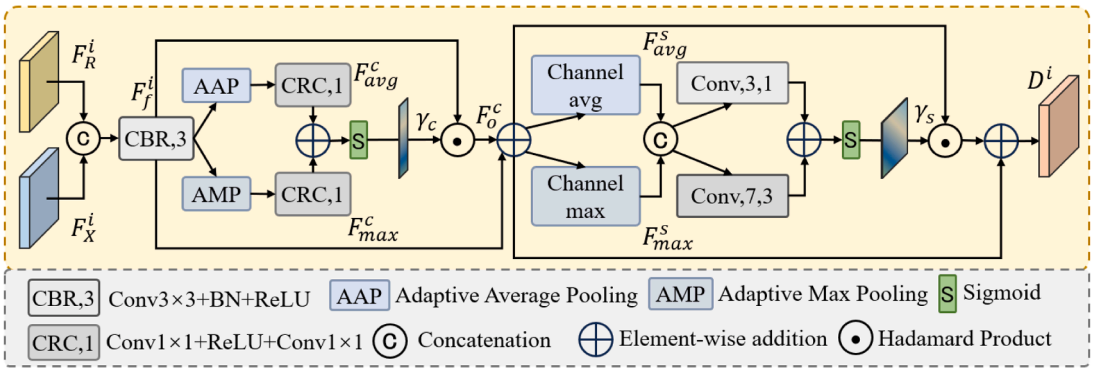


Fig. 3. Architecture of FFAM. This module is used to process cross-modal features in the first two layers of the backbone network.

overall procedure can be summarized as follows:

$$F_f^i = CBR\left(Concat\left(F_R^i, F_X^i\right)\right) \tag{1}$$

$$FFAM\left(F_R^i, F_X^i\right) = SA\left(CA\left(F_f^i\right)\right), i = 1, 2 \tag{2}$$

where *Concat* denotes concatenation, and *CBR* is a 3 × 3 convolution followed by BN and ReLU.

3.2.1. Channel attention (CA)

Channel attention aims to focus on important cross-modal feature maps and highlight meaningful information. Specifically, first, to aggregate spatial information between cross-modal feature maps and reduce parameter complexity, this paper adopts global adaptive average

pooling and global adaptive maximum pooling. Subsequently, each pooling result undergoes a 1 × 1 convolutional layer for channel compression to aggregate features. Then, a ReLU activation is applied followed by a 1 × 1 convolutional layer to restore the original channels and emphasize important channel information, thereby generating two feature descriptors $F^c_{avg} \in \Re^{C\times1\times1}$ and $F^c_{max} \in \Re^{C\times1\times1}$. Subsequently, the results obtained from the two branches are combined through element-wise addition and then adopted a sigmoid function to get a sequence of channel weight, denoted as $\gamma_c \in \Re^{C\times1\times1}$. Finally, the original fusion feature F^i_f is multiplied with γ_c and added to the initial input F^i_f to obtain the final feature vector F^c_o. In short, the process is computed as follows:

$$F^c_{avg} = CRC\left(AAP\left(F^i_f\right)\right) \tag{3}$$

$$F^c_{max} = CRC\left(AMP\left(F^i_f\right)\right) \tag{4}$$

$$CA\left(F^i_f\right) = F^i_f \odot \partial_1\left(F^c_{avg} \oplus F^c_{max}\right) \tag{5}$$

$$F^c_o = CA\left(F^i_f\right) + F^i_f \tag{6}$$

where *AAP* is global adaptive average pooling, *AMP* represents global adaptive maximum pooling, *CRC* denotes two 1 × 1 convolutions and a ReLU sandwiched in between, ⊙ stands for element-by-element multiplication, and ∂_1 represents a sigmoid function.

3.2.2. Spatial attention (SA)

Spatial attention is designed to focus on key locations in the cross-modal feature maps. Similarly, this paper utilizes average pooling and maximum pooling to aggregate channel information in the feature maps generated by CA, resulting in two 2D feature maps, denoted as $F^s_{avg} \in \Re^{1\times H\times W}$ and $F^s_{max} \in \Re^{1\times H\times W}$, which represent the average pooling features and max pooling features of the channels, respectively. Then, they are connected and passed through parallel 3 × 3 and 7 × 7 convolutions to obtain richer multi-scale information. Subsequently, the results obtained from the two branches are combined through element-wise addition and adopted a sigmoid function to gain a spatial weight map $\gamma_s \in \Re^{1\times H\times W}$. Finally, the output feature of CA is multiplied with γ_s and added to the input F^c_o to get the final fusion vector D^i. Briefly, this process is as follows:

$$F^s_{avg} = C_{avg}\left(F^c_o\right) \tag{7}$$

$$F^s_{max} = C_{max}\left(F^c_o\right) \tag{8}$$

$$D^i = F^c_o \odot \partial_1\left(Conv_3\left(Concat\left(F^s_{avg}, F^s_{max}\right)\right) \oplus Conv_7\left(Concat\left(F^s_{avg}, F^s_{max}\right)\right)\right) + F^c_o \tag{9}$$

where C_{avg}, C_{max}represent the average pooling and maximum pooling on the channel dimension, and $Conv_3$, $Conv_7$ represent 3 × 3 and 7 × 7 convolutions, respectively.

3.3. Cross-modal feature interaction module (CFIM)

Although the information between different modalities is complementary, there is still noise within them, which can affect the balance of feature fusion between modalities. However, we can filter noise through the mutual rectification of the information in the modalities. To this end, this paper designs CFIM for the post-three-stage of the backbone network, which aims to rectify features and explore potential correlations between modalities to achieve better interaction and fusion of multimodal information. This module includes two stages: feature rectification and feature fusion. The framework of CFIM is depicted in Fig. 4.

3.3.1. Feature rectification stage

In this stage, the paper rectifies the information of the modalities

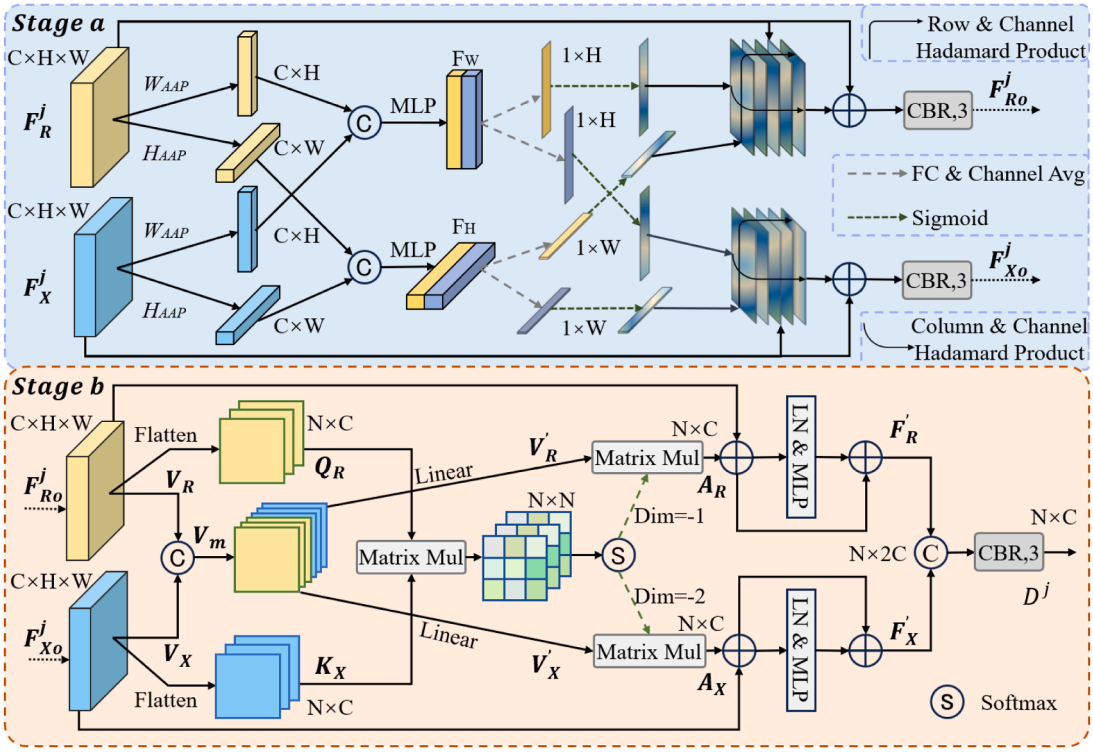


Fig. 4. Architecture of CFIM. This module is divided into two stages: feature rectification and feature fusion, which are used to process the last three layers of features of the backbone network. The output of feature rectification stage is operated as input for feature fusion stage.

through channel and spatial attention in the axial direction, aiming to mitigate the impact of noise between different modalities on the performance of the model. The construction of feature rectification stage is depicted in *Stage a* in Fig. 4. For the RGB feature map $F_R^j \in \Re^{C\times H\times W}$ and the X-modality feature map $F_X^j \in \Re^{C\times H\times W}$ at intermediate layers, with $j = 3, 4, 5$ representing the last three layers of the backbone network, this paper encodes each channel of the above two feature maps along the row-column direction using feature aggregation in both spatial dimensions. In this way, this paper achieves the capture of long-range dependencies in one spatial dimension while preserving relative position information in another dimension. Specifically, this paper applies adaptive average pooling operations to the feature maps of two modalities separately in the row and column directions, resulting in four aggregated features. They can be expressed as:

$$Z_R^h = H_{AAP}\left(F_R^j\right), Z_R^w = W_{AAP}\left(F_R^j\right) \tag{10}$$

$$Z_X^h = H_{AAP}\left(F_X^j\right), Z_X^w = W_{AAP}\left(F_X^j\right) \tag{11}$$

where H_{AAP}, W_{AAP}denote the average pooling operation along the row-column direction, and $Z_R^h \in \Re^{C\times W}$, $Z_R^w \in \Re^{C\times H}$, $Z_X^h \in \Re^{C\times W}$, $Z_X^w \in \Re^{C\times H}$represent the aggregated features obtained from the average pooling operation along the row-column direction for RGB feature maps and X-modality feature maps, respectively. To rectify the encoding features, this paper concatenates the aggregated features of the two modalities along the row and column directions, respectively, and feeds them into the MLP to improve the interaction of the features. They are represented as follows:

$$F_H = f_{mlp}\left(Concat\left(Z_R^h, Z_X^h\right)\right) \tag{12}$$

$$F_W = f_{mlp}\left(Concat\left(Z_R^w, Z_X^w\right)\right) \tag{13}$$

where *Concat* denotes concatenation along the channel direction, f_{mlp} refers to a shared MLP function, and $F_H \in \Re^{2C\times W}$, $F_W \in \Re^{2C\times H}$ represent intermediate features in the row-column direction, respectively. In addition, to better capture the complementary information between them, this paper splits the two intermediate feature maps in the row-column direction by using a fully connected layer (FC) and pooling operation in the channel dimension, and then spatially encodes these feature maps after a sigmoid function, which is calculated as follows:

$$W_R^h = \partial_1(fcc(F_H)), W_R^w = \partial_1(fcc(F_W)) \tag{14}$$

$$W_X^h = \partial_1(fcc(F_H)), W_X^w = \partial_1(fcc(F_W)) \tag{15}$$

where *fcc* denotes FC and channel pooling operation, ∂_1 represents a sigmoid function, and $W_R^h \in \Re^W$, $W_R^w \in \Re^H$, $W_X^h \in \Re^W$, and $W_X^w \in \Re^H$ denote the sequence of attentional weights in the row-column direction for RGB modality and X-modality, respectively, which can help the network to localize the objects of interest in the row-column direction. Subsequently, we apply a dot product for each input feature map using two weight sequences along the row and column directions separately for feature rectification and region of interest localization. To ensure the stability of the features, we employ residual connections. Additionally, we use a 3×3 convolutional block to generate the outputs of this stage, namely F_{Ro}^j and F_{Xo}^j. These outputs are then adapted to serve as inputs for *Stage b*. This process can be represented as:

$$F_{Ro}^j = CBR\left(F_R^j \otimes W_R^h \otimes W_R^w + F_R^j\right) \tag{16}$$

$$F_{Xo}^j = CBR\left(F_X^j \otimes W_X^h \otimes W_X^w + F_X^j\right) \tag{17}$$

where $\otimes$ denotes the dot product, *CBR* stands for a 3×3 convolution followed by BN and ReLU.

3.3.2. Feature fusion stage

The traditional cross-attention mechanism generates a set of *Query*, *Key*, and *Value* for each input, which increases the computational complexity. To solve this issue, we can employ matrix multiplication between the patch sequence matrices of the two modalities to obtain a similarity matrix. Then, probability calculations are performed for the two dimensions of the similarity matrix to obtain complementary weight information for each modality. This not only reduces complexity but also explores potential correlations between different modalities for better feature fusion. The structure is shown in *Stage b* in Fig. 4. Specifically, we take the two output feature maps, $F_{Ro}^j \in \Re^{C\times H\times W}$ and $F_{Xo}^j \in \Re^{C\times H\times W}$, from *Stage a* as inputs. Firstly, F_{Ro}^j is flattened to obtain the 2D patch sequences $Q_R \in \Re^{N\times C}$, $V_R \in \Re^{N\times C}$, and similarly, F_{Xo}^j is flattened to obtain the 2D patch sequences $K_X \in \Re^{N\times C}$, $V_X \in \Re^{N\times C}$. To better explore the connections and convey more complementary information between two modalities, this paper cascades V_R and V_X to generate $V_m \in \Re^{N\times 2C}$, and then maps the V_m with a channel number of $2C$ to V_R' corresponding to RGB modality and V_X' corresponding to X-modality using a Linear layer with a channel number of C. This paper uses the above representation vectors to compute the attention graph:

$$V_m = Concat(V_R, V_X) \tag{18}$$

$$V_R', V_X' = Linear(V_m) \tag{19}$$

$$M = \frac{Q_R \times (K_X)^T}{\sqrt{d}} \tag{20}$$

where V_R', V_X' denote the fusion feature matrices obtained by linear layer operations, and $M \in \Re^{N\times N}$ represents the similarity attention map between modalities. To better explore the complementary information between different modalities, when softmax function is computed with the dimensions of columns:

$$A_R = softmax_{d=-1}(M)\cdot V_R' \tag{21}$$

when the softmax function is computed with the dimensions of rows,

$$A_X = softmax_{d=-2}(M)\cdot V_X' \tag{22}$$

where d denotes the computational dimension of softmax, $A_R \in \Re^{N\times C}$ represents the association state from tokens in the X-modality to tokens in the RGB modality to obtain relevant information about X-modality from RGB image, and $A_X \in \Re^{N\times C}$ represents the association state from tokens in the RGB modality to tokens in the X-modality to obtain relevant information about RGB image from X-modality. Subsequently, the interaction feature maps are obtained by a linear layer, a multilayer perceptron and residual connections:

$$F_R' = MLP\left(LN\left(F_{Ro}^j + A_R\right)\right) + F_{Ro}^j + A_R \tag{23}$$

$$F_X' = MLP\left(LN\left(F_{Xo}^j + A_X\right)\right) + F_{Xo}^j + A_X \tag{24}$$

Finally, to achieve a comprehensive fusion of multimodal information, the feature maps of two modalities are concatenated and processed through *CBR* to obtain the output of this stage:

$$D^j = CBR\left(Concat\left(F_R', F_X'\right)\right) \tag{25}$$

where D^j denotes the output feature map obtained by fusing cross-modal information from the last three layers of the backbone network.

3.4. Cross-modal feature decoding module (CFDM)

Current common methods for crowd counting usually employ

decoding heads at the end of the network for feature aggregation and density estimation tasks. However, this approach ignores the richness of multi-scale semantic information. Although subsequent improvements have addressed the scale issue by employing aggregated feature information and dilated convolution in the multi-scale regression heads, they still fail to effectively capture long-range dependency information. Therefore, to better aggregate multiple feature maps generated by the backbone network and fully capture local and global contextual information, this paper proposes CFDM. This module utilizes a novel graph convolutional block to integrate multi-layer features, thereby generating effective crowd density maps. The structure of CFDM is depicted in Fig. 5.

Specifically, to integrate multi-scale features and eliminate their differences, this paper first performs feature filtering on the D^1 generated by FFAM. Firstly, D^1 is downsampled and then subjected to a 1×1 Convolution, ReLU activation, and 3×3 convolution to obtain $D^{1,out}$, aligning its features with those of the second layer. Afterwards, it is concatenated with D^2 through a channel, followed by two CR (3×3 convolution, ReLU) and an intermediate downsampling operation, resulting in multi-scale mixed feature $D^{2,out}$ that matches D^3. We apply similar processing on the D^4 and D^5 generated by the CFIM to obtain a mixed feature $D^{4,out}$ with the same size as D^3. Afterward, we concatenate these three feature maps to obtain D^{out}. This process can be represented as follows:

$$D^{1,out} = CRC\left(Down_2\left(D^1\right)\right) \tag{26}$$

$$D^{2,out} = CR\left(Down_2\left(CR\left(Concat\left(D^{1,out}, D^2\right)\right)\right)\right) \tag{27}$$

$$D^{5,out} = CRC\left(Up_2\left(D^5\right)\right) \tag{28}$$

$$D^{4,out} = CR\left(Up_2\left(CR\left(Concat\left(D^{5,out}, D^4\right)\right)\right)\right) \tag{29}$$

$$D^{out} = CR\left(Concat\left(D^{2,out}, D^3, D^{4,out}\right)\right) \tag{30}$$

where $Down_2$ represents down-sampling with a stride of 2, Up_2 represents up-sampling with a stride of 2, CRC stands for 1×1 convolution, ReLU and 3×3 convolution, $Concat$ represents concatenation along the channel dimension, and CR means 3×3 Convolution and ReLU.

After aggregating multi-scale features, this paper adopts two methods to further refine the feature map to capture long-range attention features and local perceptual information. Among them, the graph convolution block (GCB) is used to capture long-range attention features, while the spatial attention (SA) is used to perceive local features. This can not only extract the global relationships in the crowd, but also eliminate noise in local details. Finally, the obtained feature maps are transformed into the final density map output using a decoding head. Specifically, this paper adopts the Grapher design in Vision GNN (Han et al., 2022) as the benchmark. The GCB consists of a graph convolution layer (GC) and two 1×1 convolutional layers followed by BN and ReLU. The specific process is as follows:

$$D^{g,in} = CBR\left(D^{out}\right) \tag{31}$$

$$GC\left(D^{g,in}\right) = GELU\left(BN\left(DC\left(D^{g,in}\right)\right)\right) \tag{32}$$

$$GCB\left(D^{g,in}\right) = CBR\left(GC\left(D^{g,in}\right)\right) \tag{33}$$

$$D^{g,out} = SA\left(GCB\left(D^{g,in}\right) + D^{g,in}\right) \tag{34}$$

where CBR is a 1×1 convolution followed by BN and ReLU. DC represents a graph convolution in a dense dilated K-nearest neighbor (KNN) graph. SA represents spatial attention and is consistent with FFAM. At last, the predicted density map is acquired after a decoding head:

$$F_D = Conv_3\left(D^{g,out}\right) \tag{35}$$

where $Conv_3$ represents a 3×3 convolution and ReLU.

3.5. Loss function

To train the model, we used the weighted sum of the count loss and Bayesian loss as the loss function. The goal of count loss is to calculate the loss value by comparing the predicted count of individuals with the actual count. Whereas, Bayesian loss utilizes point labeling to build a density probability function, after which the density of each pixel is estimated by taking the sum of the product of the contribution probability to obtain the expected counts for each marker. The loss function is calculated as follows:

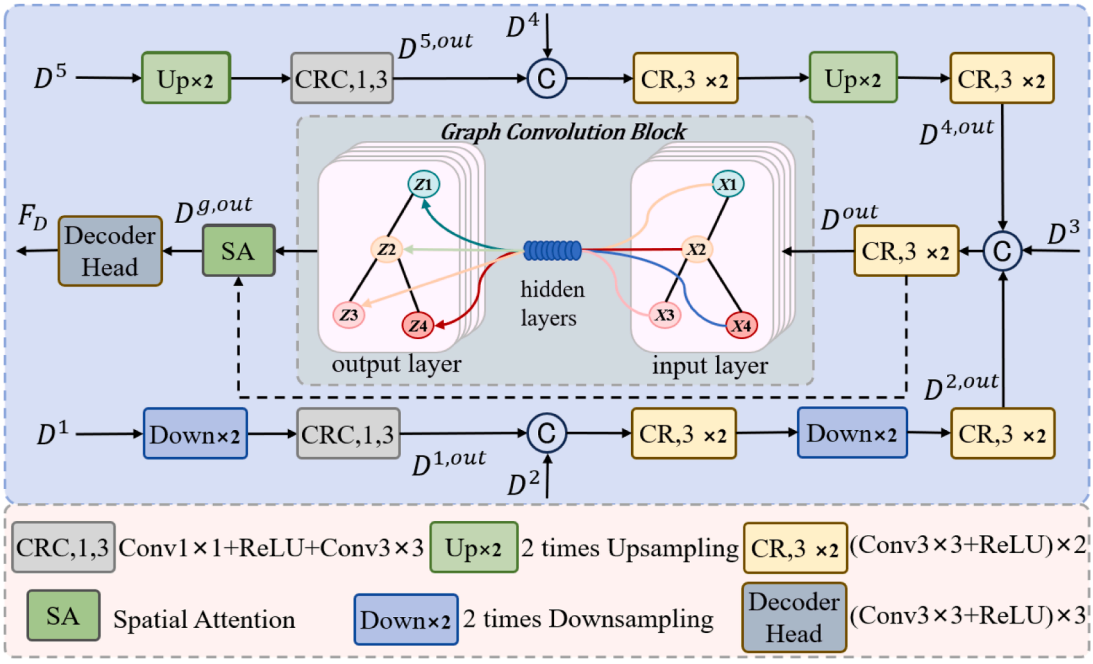


Fig. 5. Architecture of CFDM. This module focuses on better aggregation of multiple levels of feature maps generated by both FFAM and CFIM.

$$L_C = L_1(P, G) \tag{36}$$

$$L_{Bayes} = \sum_{1}^{n} F(1 - E[c_n]) \tag{37}$$

$$L = \lambda_1 L_C + \lambda_2 L_{Bayes} \tag{38}$$

where P and G represent the predicted and actual number of individuals respectively, Lc is the counting loss, F denotes the distance function (L1 distance),$E[c_n]$ indicates the Bayesian loss calculated by using the posterior label probability and estimated density map, λ_1 and λ_2 indicate the loss parameter, which is set to 1, respectively.

4. Experiment details and datasets

4.1. Implementation details and evaluation metrics

In this study, we adopted PyTorch (Paszke et al., 2019) to implement the CMFX. The backbone network utilized for the model is the BL (VGG-19) (Ma et al., 2019). During training, each input is a pair of RGB-D or RGB-T images. The experiments were performed on a single NVIDIA GEFORCE RTX 2080Ti, boasting 11 GB of memory. To train the model, we performed 200 epochs using Adam's algorithm (Kingma & Ba, 2014) with a learning rate of 10^{-5}.

We used mean absolute error (MAE), root-mean-square error (RMSE (Zhou et al., 2018)), and grid average mean absolute error (GAME) as metrics to evaluate the performance of our model. MAE represents the average of the absolute prediction error for images, which intuitively indicates the difference between the predicted and actual counts. RMSE amplifies the impact of errors through squaring, making it highly sensitive to outliers and capable of reflecting the distribution of prediction errors. GAME divides the sample into 4^l grids and calculates the error within each grid, allowing for a more accurate assessment of the model's spatial prediction capability:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - G_i| \tag{39}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - G_i)^2} \tag{40}$$

$$GAME(l) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{4^l}\left|P_i^j - G_i^j\right| \tag{41}$$

where n denotes the number of images, P_i and G_i represent the predicted counts and the actual counts in the ith image, and P_i^j and G_i^j denote the predicted counts and the actual counts in the jth region of the ith image. Notably, GAME (0) is equivalent to MAE.

4.2. Datasets

This paper validated the effectiveness of CMFX on two RGB-T datasets (RGBT-CC and DroneRGBT) and one RGB-D dataset (ShanghaiTechRGBD). Specifically, RGBT-CC dataset (Liu et al., 2021) contains 2030 pairs of representative RGB-T images, showcasing scenes like shopping centers, streets, and stations. Among these pairs, 1013 pairs are captured in bright environments, while the rest are captured under poor lighting conditions. In addition, the same split as the previous work (Liu et al., 2021), 50 % of the images (1030 pairs of images) are for training, 40 % of the images (800 pairs of images) are for testing, and 10 % of the images (the remaining images) are used for validation. DroneRGBT dataset (Peng et al., 2020) comprises 3600 pairs of labeled RGB-T images and includes a total of 175,698 people. Moreover, as in the same split of previous work (Peng et al., 2020), 50 % of the images are used for training and 50 % for testing. What sets this dataset apart from previous ones is that it is captured using drones, featuring scenes such as parks, parking lots, and campuses. This dataset is divided into three classes based on illumination: dark, dusk, and light. Each class is further divided into two parts, one for training and the other for testing. While ShanghaiTechRGBD dataset (Lian et al., 2019) includes 2193 pairs of RGB-D images with an average of 65.9 individuals per image. In this dataset, we also use the same split for the training and testing as the previous work (Lian et al., 2019), 55 % of the images (1193 pairs of images) are allocated for training while 45 % of the images (the remaining images) are designated for testing.

5. Experimental results and analysis

In this section, a series of experimental results will be shown to validate the effectiveness of CMFX. In Section 5.1, this paper compares our method with existing methods on the ShanghaiTechRGBD dataset. In Section 5.2, this paper compares our method with existing methods on the RGBT-CC dataset and DroneRGBT dataset. In Section 5.3, this paper conducts extensive ablation experiments to assess the influence of each module on the overall network's performance, while also validating the effectiveness of CMFX under different illumination conditions.

5.1. Results on RGB-D dataset

In our study, we evaluate our approach with existing methods on the ShanghaiTechRGBD dataset. The comparative methods include MCNN (Zhang et al., 2016), CSRNet (Li et al., 2018), BBSNet (Fan et al., 2020), HDFNet (Pang et al., 2020), MMCCN (Peng et al., 2020), CSRNet+IADM (Liu et al., 2021), DEFNet (Zhou et al., 2022). The results are displayed in Table 2.

According to the results in Table 2, CMFX significantly outperforms other methods in five evaluation metrics when the input image is RGB-D. Compared with existing methods, CMFX focuses on rectifying and transferring the information of different modalities and uses CFIM to filter out the noise caused by differences between different modalities, effectively achieving the interaction and fusion of multimodal information. Hence, CMFX leads to a substantial enhancement in crowd counting on the ShanghaiTechRGBD dataset. The visualization results are shown in Fig. 6(a). The difference between the predicted count and the actual count of the second image is 0.9. Regardless of whether the number of crowds in the image is large or small, our model's prediction results are very close to the true value. As shown in Fig. 7(a), most of the points are distributed in the neighborhood of the diagonal line, which indicates that the predicted counts and the actual counts on this dataset are very close to or almost equal to each other. This indicates that for RGB-D crowd counting, CMFX can accurately predict the count of individuals using the geometric information of the crowds.

5.2. Results on RGB-T datasets

We compare our method with existing methods on the RGBT-CC dataset and the DroneRGBT dataset. The comparative methods on the RGBT-CC dataset include MCNN, CSRNet, BL (Ma et al., 2019), BBSNet,

Table 2
Comparison results of different methods on ShanghaiTechRGBD dataset.

Methods	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE
MCNN	9.66	12.40	16.89	23.88	13.23
CSRNet	9.52	13.17	17.36	24.07	13.75
BBSNet	7.29	9.67	13.38	18.56	10.67
HDFNet	7.13	10.14	13.90	19.11	10.06
MMCCN	7.29	9.34	12.56	17.45	10.53
CSRNet+IADM	6.73	9.10	11.66	15.42	9.51
DEFNet	9.30	10.41	12.26	15.33	13.35
Ours	**6.28**	**8.10**	**10.92**	**15.25**	**8.80**

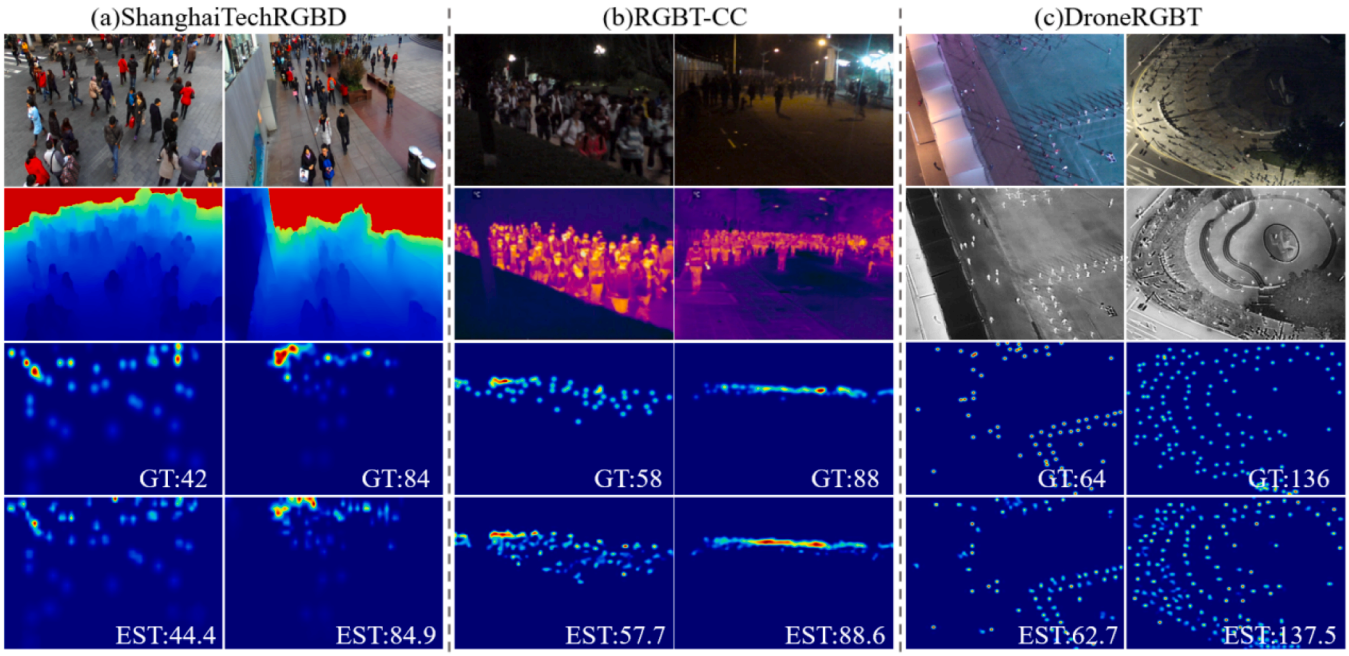


Fig. 6. Visualization results on different datasets. From top to bottom: RGB image, depth/thermal image, ground truth and predicted density map.

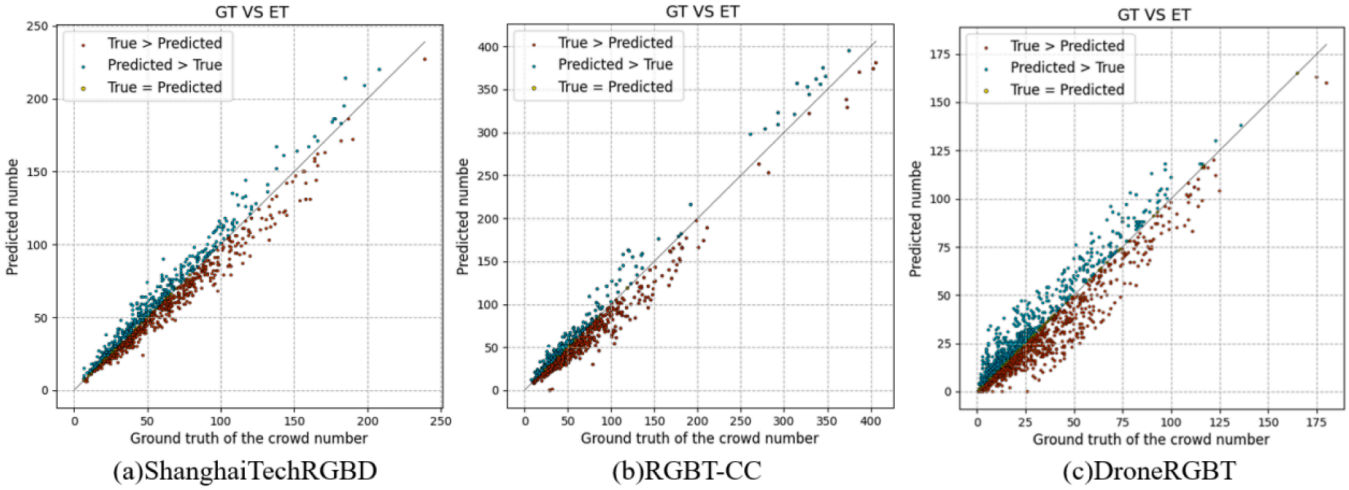


Fig. 7. Comparison of predicted and actual counts on three datasets. In these figures, the points closer to the diagonal line indicate that the predicted counts are closer to the actual counts.

MMCCN, CSRNet+IADM, DEFNet, TAFNet (Tang et al., 2022), LIECCNet (Hu & Li, 2023), CSANet (Li et al., 2023), CGINet (Pan et al., 2023). In addition, the comparative methods on the DroneRGBT dataset include MCNN, CSRNet, BBSNet, MMCCN, BL+IADM (Liu et al., 2021), FIDTM (Liang et al., 2022), DEFNet, CGINet. And the results are presented in Tables 3 and 4.

From Tables 3, 4, our method exhibits excellent performance on both RGBT-CC dataset and DroneRGBT dataset, surpassing the existing methods. As shown in Table 3, compared with the existing optimal method CGINet, CMFX improves 6.8 %, 4.1 %, 2.2 %, 5.7 %, 5.6 % on the five metrics of GAME (0), GAME (1), GAME (2), GAME (3), and RMSE, respectively. The visualization results are shown in Fig. 6(b). The difference between the predicted value and the actual value of the first image is 0.3. This demonstrates the ability of thermal images to supplement RGB images by providing crowd locations and distributions, enabling the model to distinguish between crowds and backgrounds. In addition, as shown in Table 4, on the DroneRGBT dataset, compared

Table 3
Comparison results of different methods on RGBT-CC dataset.

Methods	Backbone	GAME (0)	GAME (1)	GAME (2)	GAME (3)	RMSE
MCNN	Multi-column CNN	21.89	25.70	30.22	37.19	37.44
CSRNet	CSRNet	20.40	23.58	28.03	35.51	35.26
BL	VGG-19	18.70	22.55	26.83	34.62	32.67
BBSNet	ResNet-50	17.88	22.05	26.47	34.48	30.73
MMCCN	ResNet-50	13.82	17.83	22.20	29.64	24.36
CSRNet+IADM	CSRNet	17.94	21.44	26.17	33.33	30.91
DEFNet	VGG-16	11.90	16.08	20.19	27.27	21.09
TAFNet	VGG-16	12.38	16.98	21.86	30.19	22.45
LIECCNet	VGG-19	13.28	17.38	22.20	30.04	26.53
CSANet	CSRNet	12.45	16.46	21.48	30.62	21.64
CGINet	ConvNext-Tiny	12.07	15.98	20.06	27.73	20.54
Ours	VGG-19	**11.25**	**15.33**	**19.62**	**26.14**	**19.38**

Table 4
Comparison results of different methods on DroneRGBT dataset.

Methods	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE
MCNN	20.45	26.57	35.57	46.45	27.30
CSRNe	9.99	12.73	17.63	28.16	16.29
BBSNet	15.55	17.42	20.45	23.97	27.03
MMCCN	12.76	14.00	15.73	18.22	20.63
BL+IADM	9.77	12.91	17.08	22.61	15.76
FIDTM	15.84	20.45	27.04	34.42	24.43
DEFNet	10.87	12.13	13.95	16.58	17.93
CGINet	8.37	9.97	12.34	15.51	13.45
Ours	**6.75**	**8.88**	**11.87**	**14.69**	**11.05**

with the CGINet, CMFX improves on these five metrics by 19.4 %, 10.9 %, 3.8 %, 5.3 %, and 17.8 %, respectively. The visualization results are shown in Fig. 6(c). The difference between the predicted value and the actual value of the first image is 1.3. Meanwhile, as shown in Fig. 7(b) and (c), the predicted counts and the actual counts on these two datasets are very close to or almost equal to each other. In summary, CMFX focuses on the rectification and interaction of information between different modalities rather than simply fusing multimodal information. It can explore potential correlations between modalities and focus on region-of-interest information and global contextual information, thereby achieving better performance. This further validates the effectiveness of CMFX in obtaining multimodal information.

5.3. Ablation studies

In this part, we carried out a lot of ablation experiments with the RGBT-CC dataset as an example, proving the progressiveness of CMFX.

5.3.1. Effectiveness of FFAM, CFIM, CFDM

This paper designs FFAM, CFIM, and CFDM to rectify and fuse multimodal features. Thus, to verify the effect of each module, this paper extracts them from the network for experimentation respectively. The results are displayed in Table 5.

5.3.1.1. CMFX without FFAM (w/o FFAM). This paper uses a simple fusion method instead of FFAM in the first two layers of the backbone network. As depicted in Table 5, FFAM improves the evaluation metrics. In addition, we present the visualization results without the use of the FFAM in the fifth column of Fig. 8. Through comparison, we find that adding FFAM is beneficial for improving the prediction quality of density maps. This is mainly because FFAM can capture the texture and details of images at low levels, and can effectively highlight high-density crowd areas while suppressing background noise.

5.3.1.2. CMFX without CFIM (w/o CFIM). This paper uses a simple fusion method instead of CFIM in the last three layers of the backbone network. As presented in Table 5, CFIM leads to a notable enhancement in network. In addition, we present the visualization results without the use of the CFIM in the sixth column of Fig. 8. Through comparison, it is evident that CFIM has improved the accuracy of crowd prediction and the quality of density maps. This improvement is due to the CFIM's capability to uncover latent correlations between different modalities at deeper network layers, which suppresses noise information between modalities through row-column correction. Additionally, it facilitates the interaction and fusion of high-level cross-modal information via cross-attention mechanisms.

Table 5
Ablation studies of the core components of CMFX.

Methods	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE
Ours	**11.25**	**15.33**	**19.62**	**26.14**	**19.38**
w/o FFAM	12.22	16.35	20.88	27.99	21.95
w/o CFIM	12.89	16.65	20.93	28.16	23.61
w/o CFDM	11.99	16.37	21.56	29.89	22.51

5.3.1.3. CMFX without CFDM (w/o CFDM). This paper uses a simple decoding head at the end of the model to replace CFDM. As shown in Table 5, without CFDM, the final predicted density maps will be inaccurate. The final column of Fig. 8 illustrates that without the use of the CFDM, the density map can only roughly represent the areas of the crowd, lacking the ability to refine the density estimation. In contrast, when the CFDM is employed, both the accuracy and quality of the density map are significantly improved. This enhancement is primarily due to the CFDM's effectiveness in aggregating multiple feature maps generated by the backbone network and utilizing graph convolution to refine both local information and global contextual cues.

In summary, these three core components are crucial for the successful implementation of the unified CMFX. If any of the core components in the proposed model are missing, the counting performance will be affected and degraded.

5.3.2. Effectiveness of CMFX under different illumination conditions

To validate the precision of CMFX across various illumination conditions, this paper compares the proposed method with existing methods in bright and dark environments on the RGBT-CC dataset. The comparative methods include CSRNet+IADM (Liu et al., 2021), TAFNet (Tang et al., 2022). The results are presented in Table 6.

According to Table 6, it can be observed that CMFX has improved by 23.5 %, 19.8 %, 17.8 %, 17.7 %, and 12.7 % on five evaluation metrics under bright conditions compared to TAFNet. Meanwhile, under dark conditions, CMFX performs better than existing optimal methods. This result shows that CMFX has strong robustness to lighting changes and can deliver more precise predictions in both bright and dark conditions.

5.3.3. Effectiveness of the overall architecture

To validate the effectiveness of the overall architecture depicted in Fig. 2, this paper performs comprehensive ablation experiments to investigate the rationality of applying FFAM to the first two layers and CFIM to the last three layers of the network. The results are presented in Table 7.

From Table 7, applying FFAM to the first two layers and CFIM to the last three layers achieves optimal counting accuracy. This is because FFAM can effectively integrate low-level features such as contours and textures between different modalities. Simultaneously, high-level features are rich in semantic and location information, while CFIM can significantly enhance the model's global modeling capability and promote the complementary enhancement of information between different modalities. Although this design may lead to increased model parameters and a decline in inference speed, the substantial improvement in counting accuracy reflects a necessary trade-off for a more comprehensive integration and utilization of cross-modal complementary advantages. This further validates the rationality of the model.

5.3.4. Effectiveness of the fusion strategy

To validate the effectiveness of the proposed strategy of employing two independent backbone networks to process RGB and X-modality images separately, this paper performs ablation experiments on the RGBT-CC dataset. The comparative methods include JL_DCF (Fu et al., 2021), RISNet (Wang et al., 2024). The results are shown in Table 8.

From Table 8, it is evident that CMFX employing a dual-branch structure performs the best in counting accuracy. Specifically, the JL_DCF achieves its best accuracy in a single-branch structure with DenseNet161 as the backbone compared to other backbone networks, but it still falls short of the best performance of current other methods. Notably, JL_DCF demonstrates the least number of parameters when using VGG-16 as the backbone. Additionally, we compared another state-of-the-art single-branch network called RISNet and found that its

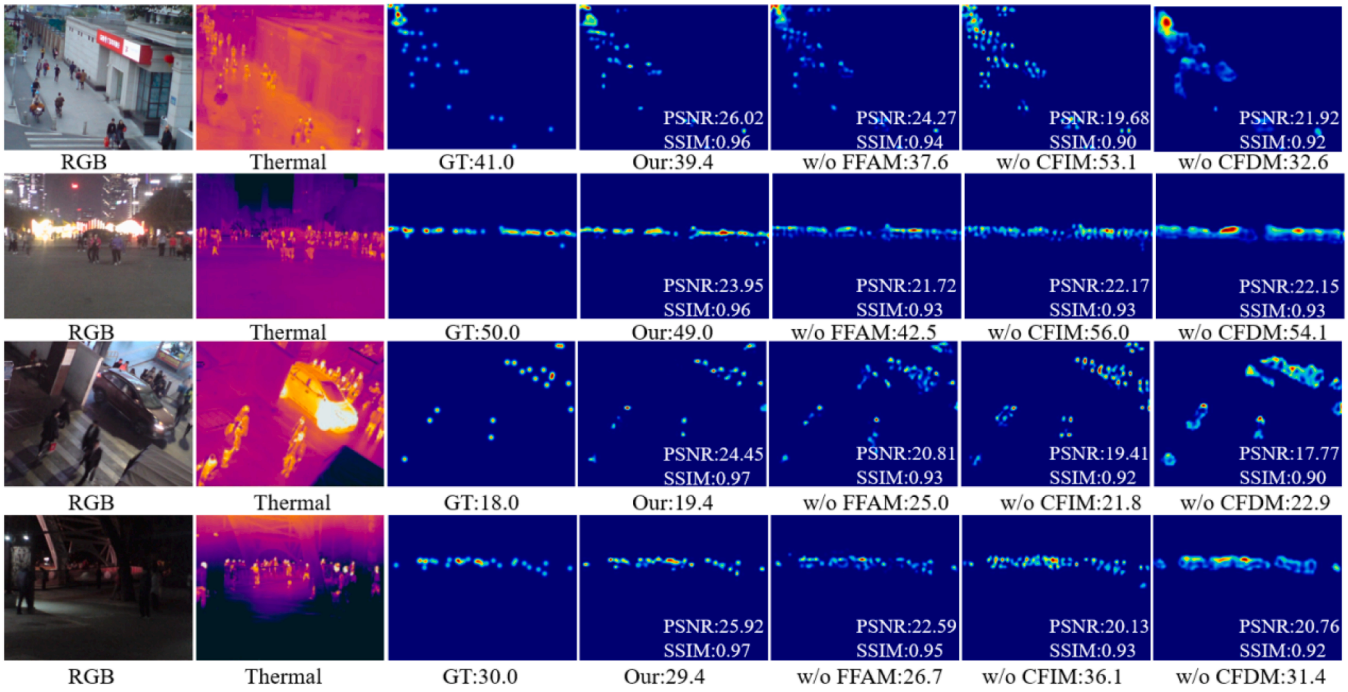


Fig. 8. Comparison of ablation and visualization results of different modules.

Table 6
Performance under different illumination conditions.

Illumination	Methods	GAME (0)	GAME (1)	GAME (2)	GAME (3)	RMSE
Brightness	CSRNet+IADM	20.36	23.57	28.49	36.29	32.57
	TAFNet	15.57	20.65	26.67	36.17	24.25
	Ours	**11.91**	**16.56**	**21.93**	**29.77**	**21.19**
Darkness	CSRNet+IADM	15.44	19.23	29.79	30.28	29.11
	TAFNet	14.20	19.20	24.00	31.63	27.50
	Ours	**12.17**	**15.60**	**19.66**	**26.33**	**21.66**

counting performance improved when using VGG-19 as the backbone. We also conducted experiments on the single-branch and dual-branch fusion strategies of CMFX. The results indicate that CMFX using the single-branch outperforms other advanced single-branch networks, with a parameter count of only 18.41 M. This is attributed to the ability of FFAM and CFIM to accurately capture and utilize key features of the modalities, while CFDM refines the multi-scale multimodal features. When employing the dual-branch strategy, CMFX exhibits optimal performance. This advantage arises from the dual-branch approach's ability to fully leverage the rich color and texture details in RGB images while extracting temperature distributions or depth information from X-modality images. These features are crucial for accurate counting in complex scenes. Traditional single-branch methods that process different modalities may lead to information redundancy, where specific modal features can be diluted, thereby limiting the effective fusion of complementary information in subsequent stages. In contrast, the dual-branch strategy adopted in this study effectively captures and utilizes the key features of each modality. Moreover, although the counting performance of the single-branch approach is slightly inferior, it has a lower parameter number, making it more suitable for devices with limited computing resources.

Table 7
Ablation studies of the overall architecture on the RGBT-CC dataset.

Methods	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE	Params	FLOPS	FPS
CMFX w/o(FFAM&CFIM)	14.52	18.46	22.89	30.12	25.87	26.60M	28.49G	15.14
FFAM(1–5)	12.79	16.59	20.75	28.02	23.14	28.23M	29.87G	13.40
FFAM(1–4)+CFIM(5)	12.50	16.61	20.66	27.87	22.90	29.72M	30.15G	12.06
FFAM(1–3)+CFIM(4–5)	12.12	16.26	20.02	27.52	21.34	30.96M	31.94G	10.94
FFAM(1–2)+CFIM(3–5)(Ours)	**11.25**	**15.33**	**19.62**	**26.14**	**19.38**	32.45M	33.27G	9.82
FFAM(1)+CFIM(2–5)	11.68	15.80	20.33	28.04	21.10	33.87M	34.93G	7.41

Table 8
Ablation studies of the fusion strategy.

Methods	Venue	Architecture	Backbone	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE	Params
JL_DCF	TPAMI 2021	Single Branch	DenseNet161	14.38	18.55	22.24	29.41	25.13	29.14M
			ResNet101	15.11	19.91	25.02	32.99	25.25	23.42M
			VGG-16	14.76	19.29	23.21	32.09	28.68	21.87M
RISNet	CVPR 2024	Single Branch	PVT_v2_b2	19.90	25.96	31.48	40.29	32.58	26.80M
			VGG-19	13.72	17.33	21.85	29.55	27.26	22.11M
Ours	–	Single Branch	VGG-19	13.42	17.20	21.61	28.73	28.19	**18.41M**
		Dual Branch		**11.25**	**15.33**	**19.62**	**26.14**	**19.38**	32.45M

5.3.5. Ablation of the internal structure of FFAM

To demonstrate the lightweight and efficiency of the convolution channel attention and spatial attention used in FFAM, we conducted an ablation study on the RGBT-CC dataset. The results are presented in Table 9.

Specifically, FFAM focuses on preserving the texture and edge details of images at the lower levels of the network, as convolution operations are more efficient and precise when handling these low-level features. From the comparison results of cross attention, the combination of transformer spatial and channel attention in Table 9, and the results of the layered combination ablation study of FFAM and CFIM in Table 7, it can be seen that the strategy of using convolution channel and spatial attention in the first two levels has higher accuracy, lightweight. Furthermore, the visualization results in Fig. 8 clearly show that the use of FFAM effectively suppresses background interference, facilitating the generation of more accurate density maps and exhibiting superior performance in crowd counting. This effect may stem from FFAM's ability to effectively enhance important information while mitigating background noise in the early stages, ensuring better integration of texture and detail.

5.3.6. Ablation of the internal structure of CFDM

To validate the effectiveness of utilizing graph convolution block for multi-scale feature fusion in CFDM, we conducted an ablation study on the RGBT-CC dataset. The results are presented in Table 10.

In Table 10, we present the details of CFDM settings. After completing the multimodal information fusion, effectively integrating multi-scale fusion information is crucial. Traditional methods typically use simple decoders to integrate feature maps and ultimately generate density maps. However, as indicated in the first row of Table 10, this approach can't yield optimal results. We then experimented with downsampling the first two layers of features and upsampling the last two layers, applying the regression head after integrating multiscale features at the third layer; yet, this method also failed to achieve superior performance. Notably, when we incorporated convolution block operations during the integration process, as shown in the third row of Table 10, the results improved. This indicates that convolutional filtering plays a significant role in integrating multiscale features. However, relying solely on convolutional aggregation can't resolve the generated density maps appearing blurred. As illustrated in Fig. 8, the graph convolution block in CFDM effectively refines global features and significantly enhances the accuracy of the density maps. Experimental results further substantiate the importance of using convolutional aggregation combined with graph convolution for refining crowd regions when integrating multiscale features in CFDM.

5.3.7. Complexity analysis of CMFX

To validate the complexity of CMFX, we compared it with the recently advanced multimodal crowd counting methods on the RGBT-CC dataset. The results are shown in Table 11.

From Table 11, CMFX demonstrates superior accuracy and image quality compared to four mainstream methods: BL+IADM, DEFNet, MC^{3}Net, and RISNet. The visualization results in Fig. 9 indicate that the density maps generated by CMFX exhibit significantly better detail handling than those generated by other methods. Although the integration of multimodal information has led to an increase in parameters and complexity, resulting in a lower frames per second (FPS) performance, CMFX still retains a certain level of real-time processing capability. Therefore, we conclude that CMFX has a significant advantage in enhancing crowd counting accuracy and image quality, further demonstrating its applicability in practical applications.

Table 10
Ablation studies of CFDM, where `C' represents convolution block, `D' represents downsampling, `U' represents upsampling, `RL' represents the regression layer, `S' represents spatial attention, `G' represents the graph convolution block.

Methods	GAME (0)	GAME (1)	GAME (2)	GAME (3)	RMSE
CU(54321)+RL(3)	12.22	16.32	21.82	30.54	23.52
U(54)+D(12)+RL(3)	12.45	16.50	22.07	30.71	24.60
CU(54)+CD(12)+RL (3)	11.99	16.37	21.56	29.89	22.51
CU(54)+CD(12)+SRL (3)	11.84	15.89	20.25	27.67	22.24
CU(54)+CD(12)+ GSRL(3)	**11.25**	**15.33**	**19.62**	**26.14**	**19.38**

6. Discussion

6.1. Data distribution

To verify the random distribution of the training, validation, and testing data in our dataset, we conducted a six-fold cross-validation on the RGBT-CC dataset. The results are shown in Table 12.

Specifically, we randomly divided the 1030 pairs of samples from the original train set into five subsets, each containing 206 samples, while treating the original 200 validation samples also as a subset. In each experiment, we selected five subsets (about 1000 samples) for training, with the remaining subset (about 200 samples) used as the validation set to assess model performance. Finally, we evaluated the test set using the weight configuration that performed best during the validation stage. As shown in Table 12, by comparing the results of the six-fold cross-validation, we found that the initial results (Fold 5) were situated between the best and worst outcomes. This finding further validates the randomness of our dataset partitioning, ensuring that the training and evaluation of the model are not influenced by data bias, thereby proving the reliability of the results.

6.2. Backbones study

To validate the compatibility of the backbone network with the three modules and assess the performance of the overall architecture, we conducted many experiments, and the results are presented in Table 11. We incorporated two popular Transformer-based backbone networks: SegFormer and TransNext, along with two widely used convolutional-based backbone networks: ResNet and ConvNext, to analyze the compatibility of CMFX's three modules with these backbones. The results indicate that when using SegFormer and TransNeXt as backbones, both counting accuracy and image quality are suboptimal. This is primarily due to the tendency of Transformers as feature extractors to cause attention dispersion, which hinders the effective refinement of crowd density maps. In contrast, although the convolutional-based networks (ResNet and ConvNext) demonstrate competitive performance, VGG-19 stands out in terms of accuracy and image quality, while also exhibiting competitive inference speed (FPS=9.82).

Table 9
Ablation studies of FFAM. TCA stands for Transformer Channel Attention, TSA stands for Transformer Spatial Attention.

Module	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE	Params	FLOPS	FPS
CrossAttention	12.33	16.44	20.73	28.38	23.33	37.51M	36.85G	6.67
TCA+TSA	12.72	16.69	20.98	28.35	25.23	36.14M	38.31G	5.46
FFAM	**11.25**	**15.33**	**19.62**	**26.14**	**19.38**	**32.45M**	**33.27G**	9.82

Table 11
Comparison of the complexity of different methods and backbones.

Methods	Venue	Backbone	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE	PSNR	SSIM	FPS
BL+IADM	CVPR2021	VGG-19	15.61	19.95	24.69	32.89	28.18	17.64	0.8743	16.59
DEFNet	TITS2022	VGG-16	13.15	16.74	20.83	27.69	23.09	20.88	0.9059	9.97
MC^{3}Net	TITS2023	ConvNext	11.77	15.29	19.59	28.11	21.00	19.73	0.8929	10.29
RISNet	CVPR2024	VGG-19	13.72	17.33	21.85	29.55	27.26	20.19	0.8871	13.54
Ours	–	SegFormer	16.39	20.96	25.66	33.23	27.56	19.67	0.8859	10.12
		TransNext	13.51	17.25	21.31	28.32	24.95	20.91	0.9056	9.14
		ResNet	12.14	16.07	20.15	27.66	20.05	20.74	0.9194	9.54
		ConvNext	11.91	15.85	19.93	26.81	19.70	21.64	0.9153	10.80
		VGG-19	**11.25**	**15.33**	**19.62**	**26.14**	**19.38**	**22.12**	**0.9277**	9.82

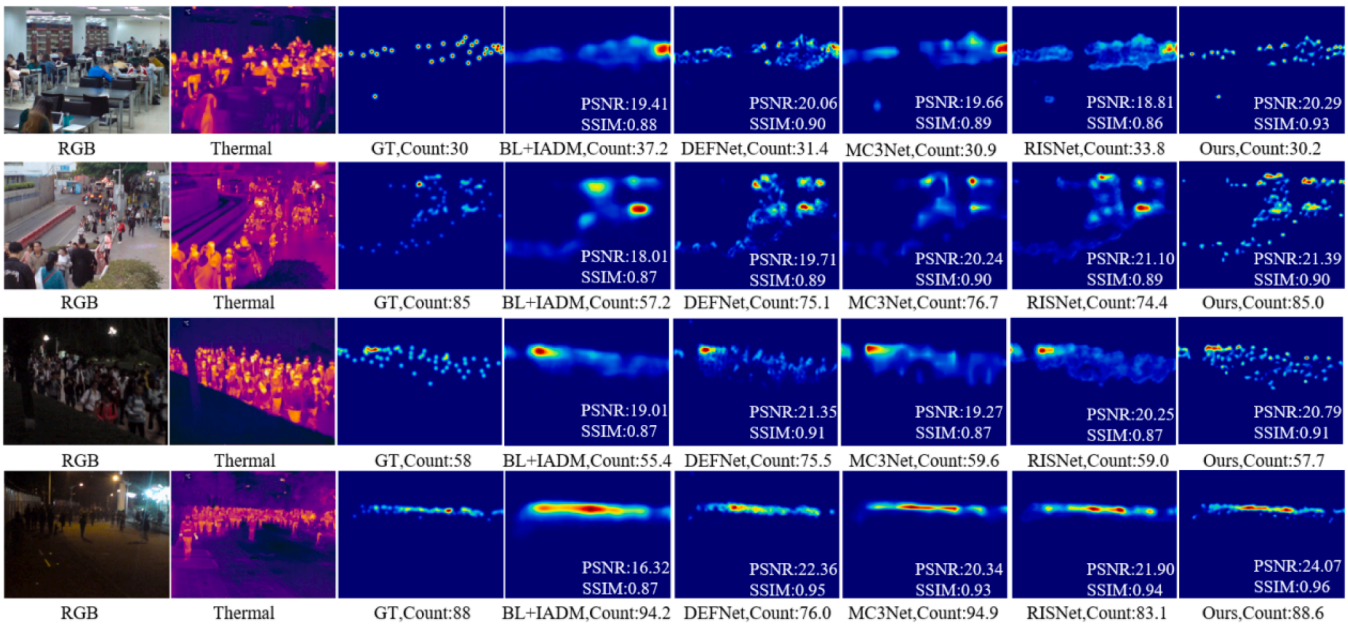


Fig. 9. Visualization results of different methods.

Table 12
Verification results of data distribution.

Fold	GAME(0)	GAME(1)	GAME(2)	GAME(3)	RMSE
Fold0	11.06	15.39	19.28	26.11	19.24
Fold1	10.98	**14.94**	19.31	**26.03**	**19.21**
Fold2	11.58	15.55	19.86	27.52	21.56
Fold3	**10.80**	15.19	**18.99**	26.05	19.36
Fold4	11.30	15.24	19.39	26.55	20.47
Fold5	11.25	15.33	19.62	26.14	19.38

6.3. Generalization study

To demonstrate the generalization capability of CMFX, we conducted performance testing on a new dataset, MICC (Bondi et al., 2014). The MICC dataset comprises 3358 pairs of RGB and depth images with a resolution of 480 × 640, encompassing three common indoor crowd scenarios: Flow, Groups, and Queue. In the Flow scenario, individuals move from one side of the room to the other; in the Groups scenario, individuals engage in conversation in two groups; and in the Queue scenario, individuals are queued and move slowly. For this dataset, we adopted the same data partitioning method as in previous research (Lian et al., 2019), using 20 % of the data for training and 80 % for testing. The results are presented in Table 13.

As shown in Table 13, CMFX outperforms MC^{3}Net across all metrics on the MICC dataset. Fig. 10 presents the visualization results of different methods, illustrating that CMFX excels in these three common indoor scenarios, effectively demonstrating its ability to learn in new environments and further validating its strong generalization capability.

Additionally, Table 13 highlights the performance of CMFX in various real-world settings. Specifically, we selected 500 images from the DroneRGBT dataset in the stadium scene (Scene B) as the testing set, while 500 images from the remaining scenes (collectively referred to as Scene A) were used for training. The results indicate that CMFX achieves superior performance metrics in the stadium scene compared to MC^{3}Net. This demonstrates that CMFX maintains robust generalization capability even in diverse real-world scenarios that were not previously captured.

6.4. Robustness study

In Fig. 11, we present the prediction results of CMFX under conditions of noise, occlusion, and darkness. Specifically, the first row displays a dense shopping mall scene characterized by significant lighting noise. The images in the second row are taken from a nighttime square, where trees on the right obscure part of the crowd. The third row shows a poorly lit scene, where the dense crowd in the distance is difficult to

Table 13
Research on generalization of new scenarios.

Methods	Train	Test	MAE	RMSE
MC^{3}Net	DroneRGBT (Scene A)	DroneRGBT (Scene B)	13.73	20.64
Ours			10.54	17.35
MC^{3}Net	MICC (20 %)	MICC (80 %)	0.66	0.81
Ours			0.23	0.36

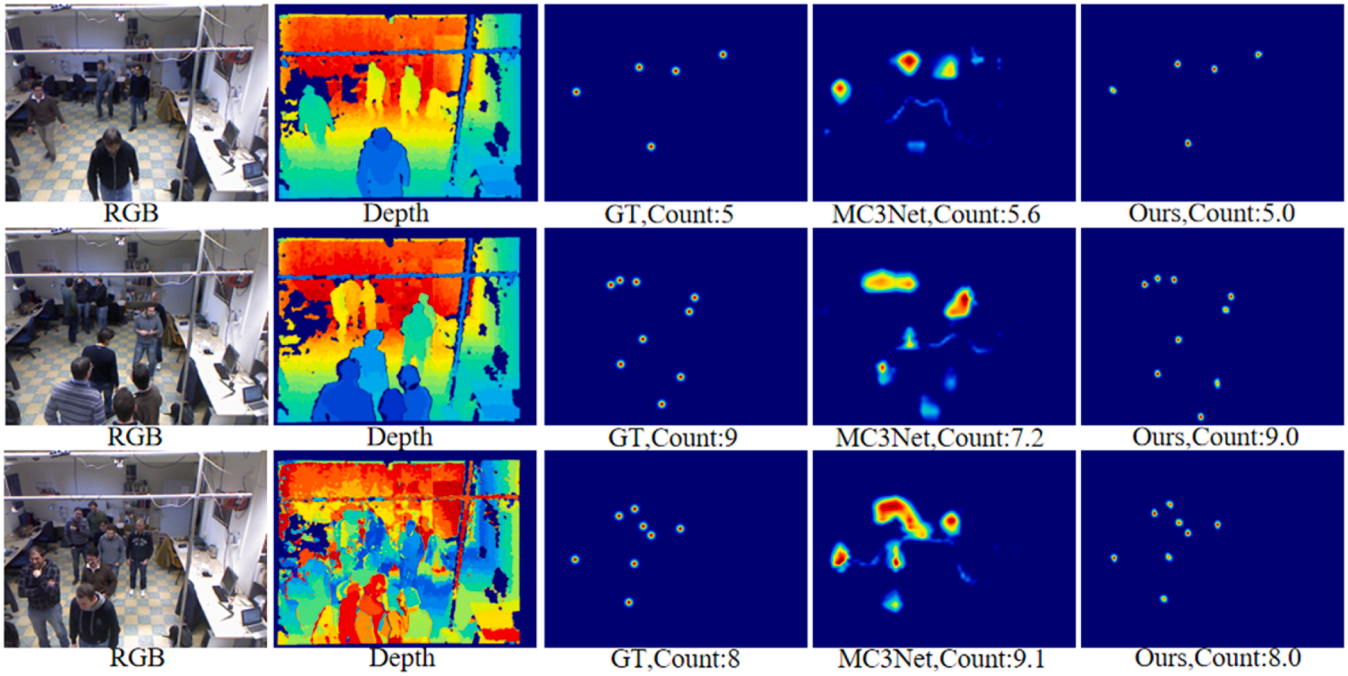


Fig. 10. Comparison of visualization results on the MICC dataset.

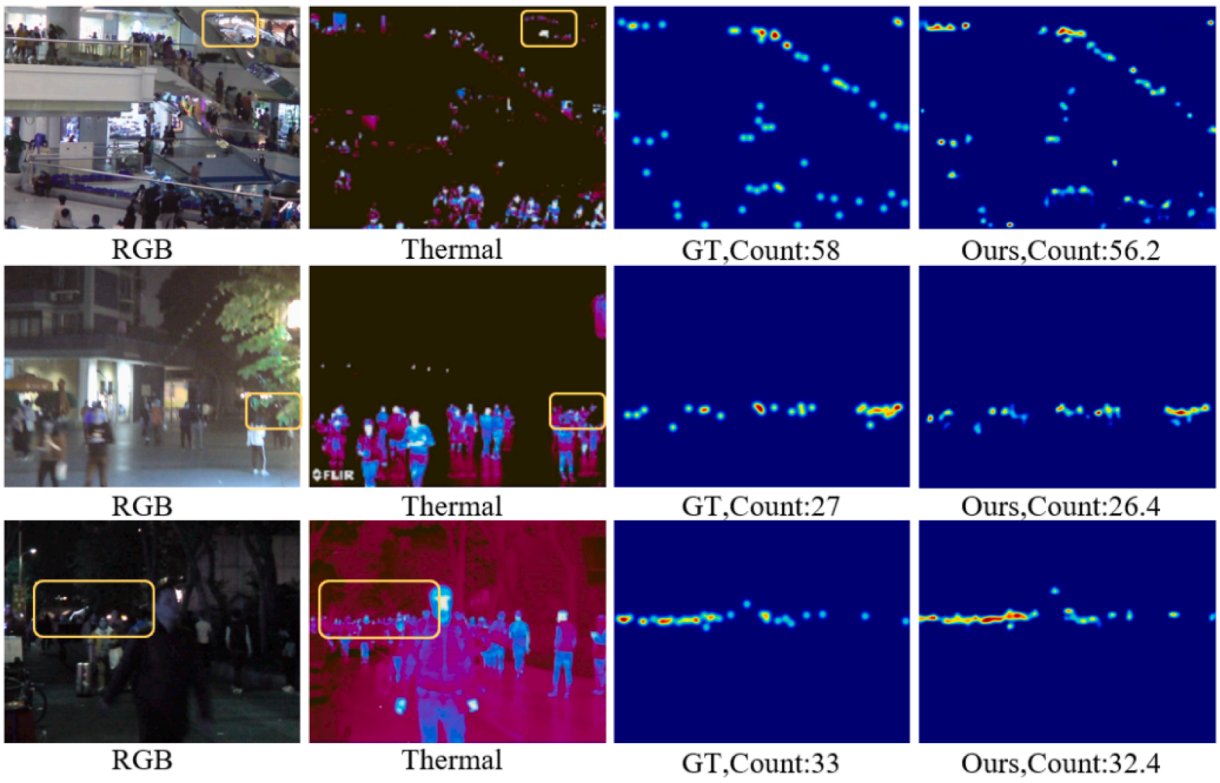


Fig. 11. Visualization results of robustness study.

identify using only the RGB images. Through these visualization results, it can be seen that CMFX can still accurately count in the face of adverse conditions such as thermal noise, occlusion, and insufficient lighting, proving its adaptability and reliability in various complex environments and demonstrating good robustness.

6.5. *Future work*

While CMFX demonstrates outstanding counting performance, its computational speed is somewhat reduced. Therefore, in future work, we plan to explore the application of knowledge distillation techniques, utilizing our model as a teacher model to extract a lightweight student model during the distillation process. This approach aims to enhance the

processing speed of the model while further optimizing counting performance.

7. Conclusion

To further improve the multi-modal crowd counting method, this study explores RGB-X crowd counting and proposes a unified framework called CMFX, which can be applied to various modal combinations. Specifically, this paper proposes three modules: FFAM, CFIM, and CFDM. FFAM is employed in the first two layers of the backbone network to efficiently accomplish the interaction and fusion of low-level features. CFIM comprises two stages: feature rectification and feature fusion. They rectify the information between different modalities and explore the potential correlations between them in the last three layers of the backbone network. CFDM employs graph convolution blocks to refine and preserve the feature maps produced by the preceding two modules to predict accurate crowd density maps. To validate the generality of CMFX, extensive experiments are conducted on one RGB-D dataset and two RGB-T datasets. The results indicate that the model exhibits significant robustness in the above two modal combinations. In future work, we plan to employ knowledge distillation technology to design a more lightweight network. In this way, our model is able to maintain lightweight while improving counting performance.

CRediT authorship contribution statement

Xiao-Meng Duan: Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Hong-Mei Sun:** Writing – review & editing, Supervision, Resources, Methodology. **Zeng-Min Zhang:** Visualization, Software, Investigation, Data curation. **Ling-Xiao Qin:** Visualization, Data curation. **Rui-Sheng Jia:** Writing – review & editing, Supervision, Methodology, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful for funding support from the Humanities and Social Science Fund of the Ministry of Education of the People's Republic of China (21YJAZH077).

Data availability

Data will be made available on request.

References

Bondi, E., Seidenari, L., Bagdanov, A. D., & Del Bimbo, A. (2014). Real-time people counting from depth imagery of crowded environments. In *2014 11th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 337–342). https://doi.org/10.1109/AVSS.2014.6918691

Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., & Chua, T. S. (2017). SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5659–5667). https://doi.org/10.1109/CVPR.2017.667

Chen, S., Tan, X., Wang, B., & Hu, X. (2018). Reverse attention for salient object detection. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 234–250). https://doi.org/10.1007/978-3-030-01240-3_15

Fan, D. P., Zhai, Y., Borji, A., Yang, J., & Shao, L. (2020). BBS-Net: RGB-D salient object detection with a bifurcated backbone strategy network. In *European conference on computer vision* (pp. 275–292). https://doi.org/10.1109/TIP.2021.3116793

Fu, K., Fan, D. P., Ji, G. P., Zhao, Q., Shen, J., & Zhu, C. (2021). Siamese network for RGB-D salient object detection and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(9), 5541–5559. https://doi.org/10.1109/TPAMI.2021.3073689

Han, K., Wang, Y., Guo, J., Tang, Y., & Wu, E. (2022). Vision GNN: An image is worth graph of nodes. *Advances in Neural Information Processing Systems, 35*, 8291–8303. https://doi.org/10.48550/arXiv.2206.00272

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141). https://doi.org/10.1109/CVPR.2018.00745

Hu, Q., & Li, G. (2023). Crowd counting study based on low light image enhancement. In *2023 4th international conference on computer engineering and application (ICCEA)* (pp. 792–796). https://doi.org/10.1109/ICCEA58433.2023.10135501

Hu, Y. X., Sun, Q., Jia, R. S., Li, Y. C., Liu, Y. B., & Sun, H. M. (2022). Le-SKT: Lightweight traffic density estimation method based on structured knowledge transfer. *Information Sciences, 607*, 947–960. https://doi.org/10.1016/j.ins.2022.06.047

Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014. https://doi.org/10.48550/arXiv.1412.6980.

Landrieu, L., & Simonovsky, M. (2018). Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4558–4567). https://doi.org/10.1109/CVPR.2018.00479

Li, H., Zhang, J., Kong, W., Shen, J., & Shao, Y. (2023). CSA-Net: Cross-modal scale-aware attention-aggregated network for RGB-T crowd counting. *Expert Systems with Applications, 213*, Article 119038. https://doi.org/10.1016/j.eswa.2022.119038

Li, H., Zhang, S., & Kong, W. (2022a). RGB-D crowd counting with cross-modal cycle-attention fusion and fine-coarse supervision. *IEEE Transactions on Industrial Informatics, 19*(1), 306–316. https://doi.org/10.1109/TII.2022.3171352

Li, Y., Zhang, X., & Chen, D. (2018). CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1091–1100). https://doi.org/10.1109/CVPR.2018.00120

Li, Y. C., Jia, R. S., Hu, Y. X., Han, D. N., & Sun, H. M. (2022b). Crowd density estimation based on multi scale features fusion network with reverse attention mechanism. *Applied Intelligence, 52*(11), 13097–13113. https://doi.org/10.1007/s10489-022-03187-y

Lian, D., Chen, X., Li, J., Luo, W., & Gao, S. (2021). Locating and counting heads in crowds with a depth prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(12), 9056–9072. https://doi.org/10.1109/TPAMI.2021.3124956

Lian, D., Li, J., Zheng, J., Luo, W., & Gao, S. (2019). Density map regression guided detection network for RGB-D crowd counting and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1821–1830). https://doi.org/10.1109/CVPR.2019.00192

Liang, D., Xu, W., Zhu, Y., & Zhou, Y. (2022). Focal inverse distance transform maps for crowd localization. *IEEE Transactions on Multimedia*. https://doi.org/10.1109/TMM.2022.3203870

Liu, L., Chen, J., Wu, H., Li, G., Li, C., & Lin, L. (2021). Cross-modal collaborative representation learning and a large-scale RGBT benchmark for crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4823–4833). https://doi.org/10.1109/CVPR46437.2021.00479

Liu, Y., Cao, G., Shi, B., & Hu, Y. (2023). CCANet: A collaborative cross-modal attention network for RGB-D Crowd counting. *IEEE Transactions on Multimedia*. https://doi.org/10.1109/TMM.2023.3262978

Ma, Z., Wei, X., Hong, X., & Gong, Y. (2019). Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6142–6151). https://doi.org/10.1109/ICCV.2019.00624

Michieli, U., Borsato, E., Rossi, L., & Zanuttigh, P. (2020). GMNet: Graph matching network for large scale part semantic segmentation in the wild. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28* (pp. 397–414). https://doi.org/10.1007/978-3-030-58598-3_24

Pan, Y., Zhou, W., Fang, M., & Qiang, F. (2024). Graph enhancement and transformer aggregation network for RGB-thermal crowd counting. *IEEE Geoscience and Remote Sensing Letters*. https://doi.org/10.1109/LGRS.2024.3362820

Pan, Y., Zhou, W., Qian, X., Mao, S., Yang, R., & Yu, L. (2023). CGINet: Cross-modality grade interaction network for RGB-T crowd counting. *Engineering Applications of Artificial Intelligence, 126*, Article 106885. https://doi.org/10.1016/j.engappai.2023.106885

Pang, Y., Zhang, L., Zhao, X., & Lu, H. (2020). Hierarchical dynamic filtering network for RGB-D salient object detection. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, proceedings, part XXV 16* (pp. 235–252). https://doi.org/10.1007/978-3-030-58595-2_15

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*. https://doi.org/10.48550/arXiv.1912.01703

Peng, T., Li, Q., & Zhu, P. (2020). RGB-T crowd counting from drone: A benchmark and MMCCN network. In *Proceedings of the Asian conference on computer vision*. https://doi.org/10.1007/978-3-030-69544-6_30

Rahman, M. M., & Marculescu, R. (2023). Medical image segmentation via cascaded attention decoding. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 6222–6231). https://doi.org/10.1109/WACV56688.2023.00616

Shi, M., Yang, Z., Xu, C., & Chen, Q. (2019). Revisiting perspective information for efficient crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7279–7288). https://doi.org/10.1109/CVPR.2019.00745

Tang, H., Wang, Y., & Chau, L. P. (2022). TAFNet: A three-stream adaptive fusion network for RGB-T crowd counting. In *2022 IEEE international symposium on circuits and systems (ISCAS)* (pp. 3299–3303). https://doi.org/10.1109/ISCAS48785.2022.9937583

Wang, L., Yang, J., Zhang, Y., Wang, F., & Zheng, F. (2024). Depth-aware concealed crop detection in dense agricultural scenes. In *Proceedings of the IEEE/CVF conference on*

computer vision and pattern recognition (pp. 17201–17211). https://doi.org/10.1109/CVPR52733.2024.01628

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11534–11542). https://doi.org/10.1109/CVPR42600.2020.01155

Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19). https://doi.org/10.1007/978-3-030-01234-2_1

Yang, S. D., Su, H. T., Hsu, W. H., & Chen, W. C. (2019). DECCNet: Depth enhanced crowd counting. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*. https://doi.org/10.1109/ICCVW.2019.00553

Zhang, B., Du, Y., Zhao, Y., Wan, J., & Tong, Z. (2021a). I-MMCCN: Improved MMCCN for RGB-T crowd counting of drone images. In *2021 7th IEEE international conference on network intelligence and digital content (IC-NIDC)* (pp. 117–121). https://doi.org/10.1109/IC-NIDC54101.2021.9660586

Zhang, S., Li, H., & Kong, W. (2021b). A cross-modal fusion based approach with scale-aware deep representation for RGB-D crowd counting and density estimation. *Expert Systems with Applications, 180*, Article 115071. https://doi.org/10.1016/j.eswa.2021.115071

Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 589–597). https://doi.org/10.1109/CVPR.2016.70

Zhou, W., Pan, Y., Lei, J., Ye, L., & Yu, L. (2022). DEFNet: Dual-branch enhanced feature fusion network for RGB-T crowd counting. *IEEE Transactions on Intelligent Transportation Systems, 23*(12), 24540–24549. https://doi.org/10.1109/TITS.2022.3203385

Zhou, W., Yang, X., Lei, J., Yan, W., & Yu, L. (2023). MC^{3}Net: Multimodality cross-guided compensation coordination network for RGB-T crowd counting. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2023.3321328

Zhou, W., Yu, L., Zhou, Y., Qiu, W., Wu, M. W., & Luo, T. (2018). Local and global feature learning for blind quality evaluation of screen content and natural scene images. *IEEE Transactions on Image Processing, 27*(5), 2086–2095. https://doi.org/10.1109/TIP.2018.2794207

Controlled Search: Building Inverted-Index PEKS With Less Leakage in Multiuser Setting

Guiyun Qin, Pengtao Liu, Chengyu Hu, Zengpeng Li, and Shanqing Guo

***Abstract*—The public key encryption with keyword search (PEKS) schemes are mostly applied to small data sets in mail forwarding systems. When retrieving large databases, the typical search mechanism makes them inefficient and impractical. When designing a PEKS scheme, except for remedying the vulnerability of keyword guessing attacks (KGAs), other leakage issues, such as multipattern privacy and forward/backward security are rarely considered, which may lead to information leakage. Moreover, most existing PEKS only consider applications in single-user scenarios, and cannot be directly transferred to multiuser scenarios, which undermines the value of data utilization. To cope with the above concerns, we propose a PEKS scheme based on an inverted index where the bitmap is used to build the index for the first time in PEKS to meet some seemingly conflicting yet desirable characteristics. First, it has high search efficiency under multiwriter and multiuser. Through linear transformation, users quickly retrieve data and control other users' access to their data without relying on a third party for authentication. Second, we prove its security in an enhanced security model that achieves multipattern privacy and forward and backward security. It can also resist KGA attacks without a designated tester, which makes it more practical. Finally, it can be extended to achieve search result verification. Compare to the scheme (Zhang et al. ICWS 2016), it has absolute advantages in security and computational cost where the search efficiency is improved by two orders of magnitude.**

***Index Terms*—Forward and backward security, inverted index, multisearchable management, public key encryption with keyword search (PEKS), result verification.**

I. Introduction

PUBLIC key encryption with keyword search (PEKS) [1], as an important branch of searchable encryption (SE) [2], focuses on solving the complex key management problems in symmetric SE (SSE). However, PEKS schemes currently suffer from the following deficiencies.

Due to the low efficiency of the typical PEKS schemes, they do not support the data owner sharing a large amount of data with the recipient, which limits the application scope of PEKS. The reason is that they always implement searching a keyword by testing all the keyword ciphertexts of all the files, i.e., the search efficiency depends on the total number of the keyword ciphertexts in all the files.

PEKS faces some security challenges and leakage problems emerge endlessly. Except for keyword guessing attacks (KGAs), the privacy of patterns and forward security are rarely considered which makes the adversary infer some sensitive information from trapdoors and search results (e.g., Leakage-Abuse Attack [3]). As far as we know, the trapdoor generation algorithms of many PEKS schemes (e.g., [4] and [5]) are deterministic, which reveals the search pattern. When searching for the same keywords, the server (tester) returns the same search results, which leads to the leakage of the access pattern. The size pattern privacy has not been considered in existing PEKS, which reveals the number of query results and the number of keywords contained in the update file. Forward security is mainly considered in SSE schemes to resist the file injection attack [6], but we show that this attack is easier to deploy in PEKS schemes. A common approach to resist KGA attacks is to require an additional designated tester, but it is not practical.

In addition, transferring PEKS to multiuser scenarios is conducive to improving data utilization. A straightforward approach is distributing the search key to other users. However, different recipients may have different search permissions (e.g., supervisors and employees, and doctors and patients). The naive approach cannot meet the above requirement and is prone to data abuse.

To sum up, a challenging and practical question we raise is: *How to break these limitations, improve efficiency and security and increase the usability of the PEKS systems?*

A. Motivations and Methodologies

To address the above issues in PEKS schemes, such as inefficiency, vulnerability to KGA attacks, patterns leakage, and the limitations of multiuser search, we construct an inverted-index-based PEKS scheme.

The data sender considers the "keyword-bitmap matrix" as the index structure and encrypts it through an integer vector homomorphic encryption (HE) algorithm, which facilitates the search/update and patterns hiding. Specifically: The elements in the index matrix represent the relationship between

Manuscript received 13 December 2022; revised 15 May 2023; accepted 1 June 2023. Date of publication 19 June 2023; date of current version 25 December 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFA1000600, and in part by the Shandong Provincial Natural Science Foundation under Grant ZR2022LZH013, Grant ZR2020LZH002, Grant ZR2020MF055, Grant ZR2021LZH007, and Grant ZR2020QF045. *(Corresponding authors: Chengyu Hu; Zengpeng Li.)*

Guiyun Qin is with the School of Cyber Science and Technology, Shandong University, Qingdao 266237, China (e-mail: qin1997@mail.sdu.edu.cn).

Pengtao Liu is with the School of Cyberspace Security, Shandong University of Political Science and Law, Jinan 250013, China (e-mail: ptwave@163.com).

Chengyu Hu, Zengpeng Li, and Shanqing Guo are with the School of Cyber Science and Technology, Shandong University, Qingdao 266237, China, also with the Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China, and also with Quan Cheng Laboratory, Jinan 250103, China (e-mail: hcy@sdu.edu.cn; zengpeng@email.sdu.edu.cn; guoshanqing@sdu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2023.3287353

keywords and files, where the element in the ith row and the jth column represent whether the jth file contains the ith keyword. If the file contains a keyword, the element in the corresponding position is set to 1, otherwise, it is set to 0. When the recipient wants to search a keyword w_i, a straightforward strategy is constructing a $1 \times m$ query matrix $\mathbf{B}$ where the element at the corresponding position of the keyword w_i is set to 1 and other elements are set to 0. Then, in the ciphertext state, query matrix $\mathbf{B}$ and index matrix are tested to obtain the bitmap corresponding to the keyword w_i. Since each keyword corresponds to the same number of bitmaps in the index matrix (i.e., the returned ciphertext results are the same), size pattern hiding is achieved. In fact, the search process is a linear transformation operation process. Due to the randomness of search tokens and ciphertext results, search and access patterns hiding are achieved. As for the update, the update of the encrypted index is achieved by utilizing a homomorphic addition operation, and during this process, forward and backward privacy is protected. See Section III for details. In addition, since there are no specific keyword ciphertexts, it can also resist KGA attacks without a designated tester.

Furthermore, through the linear transformation operation, it can establish an effective isolation mechanism among multiple users to prevent the interference of malicious users. Specifically, in multiuser scenarios, the sender (or recipient) can set different data access authorities (keywords or files) for users to achieve multisearchable management. More concretely, according to different retrieval needs, the sender (or recipient) constructs different control matrices $\mathbf{G}$. Using the control matrices $\mathbf{G}$ and the original index to perform calculation operations in the ciphertext state to obtain new index ciphertexts for different users. When specifying access authority to files, the sender (or recipient) also can further revoke the files they do not want to share to achieve file management.

Although it seems that the integer vector HE (IVHE) scheme can be trivially applied for constructing search schemes in the cloud, existing schemes only briefly describe how to construct a forward-index-based search scheme [7], [8]. To the best of our knowledge, there is no inverted-index-based PEKS scheme constructed based on the IVHE.

B. Our Contributions

In this article, we make affirmative progress in the following areas. The contributions are summarized as follows.

1) *Distinctive Index Construction:* Unlike traditional index construction, we use the bitmap structure to build the inverted index. The bitmap structure has the advantages of minimum search complexity, easy updating, etc., which is usually used in the SSE schemes. We apply it to PEKS for the first time and make up for its shortcoming of requiring to limit the number of files. Because the scheme is not necessary to test each keyword ciphertext of files, the keyword search is much more efficient. It also has "Noninteractive" search/update operations. No additional roundtrip between any parties is needed, except one inevitable roundtrip for the data owner to authorize the server to update the index, so it leaks less information.
2) *Multisearchable Management:* We realized a reliable isolation mechanism to ensure that individual users do not interfere with each other. In other words, it controls the retrieval authorities of the recipients neither requiring a trusted third party (or the data owner acting as a trusted third party) nor predetermining the number of users.
3) *Enhanced Security:* We present an enhanced security model—*Semantic Security*. It implies search/access/size pattern privacy, forward/backward privacy,[1,2] and resists KGA without a designed tester. We also show how to extend our scheme to verify the integrity of the search result.

C. Related Work

PEKS: Boneh et al. [1] proposed the first PEKS scheme via identity-based encryption (IBE) [9] for different data owners to send a small amount of data to a single user. Abdalla et al. [10] showed how to transform an anonymous IBE scheme into a PEKS scheme. Subsequently, various PEKS schemes are proposed.

Pattern Privacy: However, most PEKS schemes do not consider *search pattern* privacy and *access pattern* privacy, which leaks relevant information about queried keywords. The works [11], [12] claim that search pattern privacy can be preserved by introducing a security definition called *trapdoor indistinguishability* which makes the trapdoor generation algorithm probabilistic. It ensures that even if adversaries obtain a trapdoor, they cannot obtain the keyword information contained in the trapdoor. That is, there is a negligible advantage for any polynomial-time adversary A to correctly guess the trapdoor of keywords w_0 and w_1. Yet, search results for the same keyword are the same, so the adversary can still learn the search pattern by the access pattern. In addition, there is also *ciphertext indistinguishability* which prevents adversaries from obtaining keyword information contained in encrypted files without learning the keyword trapdoor. That is, there is a negligible advantage for any polynomial-time adversary A to correctly guess the ciphertext of keywords w_0 and w_1. Even if the *trapdoor and ciphertext indistinguishability* are achieved, the adversary can still distinguish the search results of different keywords by size pattern which is mainly considered in SSE.

KGA Attacks: Likewise, the typical PEKS schemes are vulnerable to KGAs [13] which can also be regarded as attack methods to get search pattern information in our opinion. To defend against KGAs, Rhee et al. [11] proposed a PEKS scheme with a designated tester. However, Wang et al. [14]

[1]Forward privacy ensures that the newly added file containing the keyword w cannot be linked to the trapdoor generated for the same keywords w in previous search operations. It is not considered in existing PEKS schemes, but we point out in Section III-B that it can be utilized by the server to get some private information about the outsourced data.

[2]Backward privacy ensures that searches for the keyword w cannot be linked to the files containing the keyword w which have been deleted. Although backward privacy needs not to be considered in traditional PEKS schemes, it is an important issue in the inverted-index-based schemes.

subsequently pointed out that it could not resist KGA attacks from internal malicious servers. Huang and Li [4] introduced an authenticated PEKS scheme that can resist internal KGA attacks. However, its trapdoor generation algorithm is deterministic, which reveals the search pattern. Qin et al. [5] presented the definition of multiciphertext indistinguishability (MCI), and improved Huang's scheme [4], while the improved scheme does not satisfy search pattern privacy either. Pan and Li [15] proposed a PEKS scheme that achieves *Multiciphertext/Trapdoor Indistinguishability*, but it also cannot hide the search pattern from the internal malicious server.

Efficiency: To improve the efficiency, some work attempts to build PEKS schemes using the inverted index with sublinear search efficiency which is commonly used in SSE schemes. Wang et al. [16] claimed to propose a PEKS scheme based on an inverted index. However, Wang et al. [17] proved that the scheme [16] cannot perform the keyword search correctly and modified it to be a new SSE scheme. Zhang et al. [12] proposed the first PEKS scheme based on the inverted index. Although the search efficiency has been greatly improved, its trapdoor generation efficiency is inefficient. In terms of security, although it claims to hide the search pattern, the search results reveal the access pattern. In addition, the file update method is interactive, which requires multiple rounds of communication between the user and the server.

Multiuser: To improve data utilization, multiuser PEKS schemes are proposed. The trivial method is that the owner creates the encrypted index for each receiver, which causes huge computational and communication overheads. There are currently two main paths [18]. One is relying on a third party (or the data owner to act as a third party (e.g., [19]) to generate search trapdoors for users (e.g., [20], [21], and [22]). The other path requires predetermine the number of users and precompute some parameters by all the users (e.g., [23] and [24]). The above two paths either have some leakage (eavesdropping and replay attacks) or are not flexible enough (requiring real-time online, etc.). How to break through them is still pending.

According to what we are informed, no PEKS schemes can achieve quick search efficiency while maintaining high security and can be extended to multiuser settings.

HE allows a third party (e.g., the cloud) to perform certain computational operations on encrypted data while retaining the functional and format characteristics of the encrypted data [25]. Rivest et al. [26] first proposed *homomorphism* as a possible solution to the problem of computing encrypted data without decryption. Subsequently, researchers have developed Partially HE algorithms that satisfy multiplication or addition (e.g., [27], [28], and [29]). Until, Gentry [30] proposed the first available fully HE (FHE) scheme, which supports both addition and multiplication operations. Brakerski et al. [31] proposed a parameterizable somewhat FHE scheme (BGV), whose security is based on LWE. It utilizes the key-switching technology to control the explosive growth of the ciphertext dimension during multiplication by restoring the expanded ciphertext dimension to the original dimension. At the same time, the modular-switching technology is used to replace the "Bootstrapping" process in [30] to control noise growth. Later, Brakerski [32] proposed the BFV scheme which only needs to solve the ciphertext dimension expansion problem through the key-switching technology. The computational cost and communication overhead of the above-mentioned technologies are still relatively high. To reduce the overhead, Peikert et al. [33] proposed a ciphertext packing technique (PVW). The cost of encrypting n-bit data in this scheme is basically the same as the cost of encrypting one-bit data in the Regev cryptosystem [34]. However, this scheme still requires decomposing the computational task into binary operations, which is expensive. In 2015, inspired by the above research, Zhou and Wornell [7] proposed an IVHE scheme. Unlike other general schemes, this scheme focuses on solving HE of a specific data type (integer vectors). Using the key-switching matrix to change the key-ciphertext pair, this scheme realizes complex operations in the encryption domain (e.g., linear transformation). Since there is no need to decompose the computational tasks into binary operations in the calculation process, it achieves better performance than FHE schemes, which is the main reason why we choose this algorithm.

Roadmap: In Section II, we introduce the preliminaries. In Section III, we present the framework and security definition of the scheme. In Section IV, we show the construction details and give how to implement multisearchable management. In Section V, we evaluate the experimental performance and make comparisons with typical schemes. Section VI discusses how to extend our solution to verify the integrity of the search result. Finally, we draw a conclusion in Section VII.

II. Preliminaries

A. Notation and Operation

$\mathrm{ID} = (id_1, id_2, \ldots, id_c)$ represents the file identifiers of c files, and $W = (w_1, w_2, \ldots, w_z)$ represents the keyword space of size z. $id_j = \{w_{j_1}, w_{j_2}, \ldots, w_{j_v}\}$ represents the file id_j and the keywords contained in it. In other words, each id_j is associated with a keyword list $W_j \subseteq W$, which contains all keywords in the file id_j. Below, we give the related definitions that will be used in the following parts.

1) For $a \in \mathbb{R}$, define $\lceil a \rfloor$ to round a to the nearest integer.
2) For vector $\mathbf{a} \in \mathbb{R}^n$, define $\lceil \mathbf{a} \rfloor$ to round each entry a_i of $\mathbf{a}$ to the nearest integer, and $|\mathbf{a}|$ represents the magnitude of vector $\mathbf{a}$: $|\mathbf{a}| = \max_i\{|a_i|\}$.
3) For matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, $|\mathbf{A}|$ represents the magnitude of matrix $\mathbf{A}$: $|\mathbf{A}| = \max_i\{|A_{ij}|\}$.
4) For $\mathbf{v} \in \mathbb{Z}_q^m$ (v_i is an entry of $\mathbf{v}$), $\mathbf{t}(t(\mathbf{v}))$ represents the "center" vector for $\mathbf{v}$, where $t(v_i) = \lfloor v_i \cdot q/p \rceil \in \mathbb{Z}_q$ is ith entry of $t(\mathbf{v})$.

Key-Switching Operation: According to [32], we introduce two decompositions.

1) *BitDecomp($\mathbf{y}$):* For $\mathbf{y} \in \mathbb{Z}_q^n$, let $\mathbf{x}_i \in \{0,1\}^n$ be such that $\mathbf{y} = \sum_{i=0}^{\lceil \log q\rceil -1} 2^i \cdot \mathbf{x}_i (\mathrm{mod} q)$, and output the vector $(\mathbf{x}_0, \ldots, \mathbf{x}_{\lceil \log q\rceil -1}) \in \{0,1\}^{n\lceil \log q\rceil}$.
2) *PowersOfTwo($\mathbf{u}$):* For $\mathbf{u} \in \mathbb{Z}_q^n$, output $[(\mathbf{u}, 2 \cdot \mathbf{u}, \ldots, 2^{\lceil \log q\rceil -1} \cdot \mathbf{u})]_q \in \mathbb{Z}_q^{n\cdot\lceil \log q\rceil}$.

Therefore, we can obtain

$$\langle \mathbf{y}, \mathbf{u}\rangle = \langle\ \mathrm{BitDecomp}\ (\mathbf{y}),\ \mathrm{PowersOfTwo}\ (\mathbf{u})\rangle\ (\mathrm{mod} q).$$

When the keys are all vectors, the key-switching matrix [32], [35] can switch the key-ciphertext pair to another key-ciphertext pair. To make it more practical, the work [7] gives a matrix version of key-switching composition to realize complex operations (e.g., linear transformation) of IVHE. For concrete, there is the following relation for the key $\mathbf{S} \in \mathbb{Z}_q^{f\times g}$ and the ciphertext $\mathbf{c} \in \mathbb{Z}_q^g$:

$$\mathbf{S}^*\mathbf{c}^* = \mathbf{S}\mathbf{c}.$$

Specifically, the intermediate ciphertext $\mathbf{c}^*$ means that each element c_j in $\mathbf{c}$ is represented as $c_j = \sum_{i=0}^{\ell-1} 2^i \cdot x_{j_i}$ according to *BitDecomp*. Therefore, $\mathbf{c}^* = [x_{1_0}, x_{1_1}, \ldots, x_{g_{\ell-2}}, x_{g_{\ell-1}}]^T \in \{0,1\}^{g\ell}$. According to *PowersOfTwo*, the intermediate key $\mathbf{S}^* \in \mathbb{Z}^{f\times g\ell}$ means that each element S_{ij} in $\mathbf{S}$ is replaces with $S_{ij} = [S_{ij}, 2\cdot S_{ij}, \ldots, 2^{\ell-1}\cdot S_{ij}]$.

Nextly, switch the original key $\mathbf{S} \in \mathbb{Z}_q^{f\times g}$ into the "target" key $\mathbf{S}' = [\mathbf{I}, \mathbf{T}] \in \mathbb{Z}^{f\times g'}$ where $\mathbf{I}$ is the identity matrix. To do this, a key-switching matrix $\mathbf{M} \in \mathbb{Z}^{g'\times g\ell}$ is constructed. It satisfies

$$\mathbf{M} = \begin{pmatrix} \mathbf{S}^* + \mathbf{E} - \mathbf{T}\mathbf{A} \\ \mathbf{A} \end{pmatrix} \bmod q$$

where $\mathbf{T}$ and $\mathbf{A} \in \mathbb{Z}^{(g'-f)\times g\ell}$ are the random matrices, and $\mathbf{E}$ is a noise matrix with small magnitude.

Finally, the ciphertext $\mathbf{c}$ is switched into the final ciphertext $\mathbf{c}'$ corresponding to the targe key $\mathbf{S}'$ which is defined as follows:

$$\mathbf{c}' = \mathbf{M}\mathbf{c}^* \bmod q \in \mathbb{Z}_q^{g'}.$$

B. Cryptographic Building Blocks

PEKS: There are four polynomial-time algorithms in a typical PEKS scheme.

1) *KeyGen*(λ) $\rightarrow$ (pk, sk): Input a security parameter λ and output the public-private key pair (pk, sk).
2) *PEKS*(pk, w) $\rightarrow$ C_w: Input a keyword w and the public key pk, and output the keyword ciphertext C_w.
3) *Trapdoor*(sk, w_i) $\rightarrow$ T_{w_i}: Input a keyword w_i and the private key sk, and output the trapdoor T_{w_i}.
4) *Test*(C_w, T_{w_i}) $\rightarrow$ c_i: Input the keyword ciphertext C_w and the trapdoor T_{w_i}, and output the result set c_i.

The data sender encrypts each keyword in the files using the public key of the recipient and sends the ciphertexts to the cloud server. Then, the recipient generates a search trapdoor based on its private key and the keyword to be searched. After receiving the trapdoor, the server executes the *Test* algorithm to match the keyword ciphertexts with the trapdoor.

IVHE: As an extension of the PVW scheme [33] from binary vectors to integer vectors, it is a public key cryptosystem.

Compared with the traditional HE schemes, IVHE is easier to implement complex operations, e.g., linear transformation. Their working mechanism is shown in Fig. 1.

The IVHE scheme consists of three polynomial time algorithms for encrypting the index matrix.

Key Generation: The secret key sk is $\mathbf{S} = [\mathbf{I}, \mathbf{T}] \in \mathbb{Z}_q^{m\times n}$, and the public key pk is $(\mathbf{A}, \mathbf{Q})$. $\mathbf{A} \in \mathbb{Z}_q^{(n-m)\times l}$ is a random matrix and $\mathbf{Q} = \mathbf{T}\mathbf{A} + \mathbf{E}$, where the elements in $\mathbf{E} \in \mathbb{Z}_q^{m\times l}$ come from the error distribution.

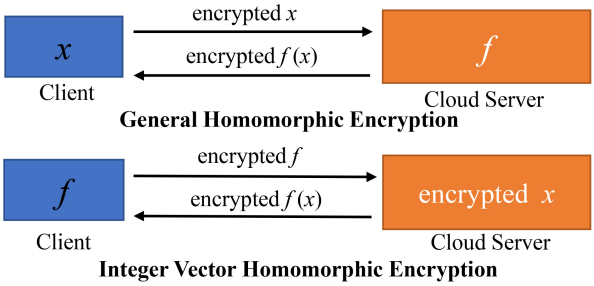


Fig. 1. Function and data.

Encryption: After encrypting the integer vector $\mathbf{v} \in \mathbb{Z}_p^m$, we get ciphertext $\mathbf{c} = (\mathbf{A}\mathbf{e}, \mathbf{Q}\mathbf{e} + \mathbf{t}(t(\mathbf{v}))) \in \mathbb{Z}_q^n$, where $q \gg p$ and $\mathbf{e} \leftarrow \{0,1\}^l$ is the noise vector.

Decryption: From [32], the vector $\mathbf{v} \in \mathbb{Z}_p^m$, the ciphertext $\mathbf{c}$, and the secret key $\mathbf{S}$ satisfy the equation

$$\mathbf{S}\mathbf{c} = w\mathbf{v} + \mathbf{e} \ (mod\ q) \tag{1}$$

where w is a large integer, that is, $w > 2|\mathbf{e}|$, $\mathbf{e}$ indicates that the noise vector belongs to the error term.

Therefore, the decryption process is represented as follows:

$$\mathbf{v} = \left\lceil \frac{\mathbf{S}\mathbf{c}}{w} \right\rfloor_q. \tag{2}$$

Addition Operation: If the ciphertexts $\mathbf{c_1}$ and $\mathbf{c_2}$ have the same key, $\mathbf{c_1} + \mathbf{c_2} = Enc(\mathbf{x_1} + \mathbf{x_2})$. When adding or deleting a file, we utilize the addition operation to achieve it.

Linear Transformation: In our solution, we achieve the keyword search and multisearchable management by the linear transformation operation $\mathbf{G}\mathbf{v}$. According to (1), for any matrix $\mathbf{G} \in Z^{m'\times m}$, it satisfies

$$(\mathbf{G}\mathbf{S})\mathbf{c} = w\mathbf{G}\mathbf{v} + \mathbf{G}\mathbf{e} \bmod q$$

$\mathbf{c}$ is regarded as the ciphertext of the plaintext $\mathbf{G}\mathbf{v}$ under the key $\mathbf{G}\mathbf{S}$. The client uses a key-switching matrix $\mathbf{M}$ to switch the key $\mathbf{G}\mathbf{S}$ into the new key $\mathbf{S}' = [\mathbf{I}', \mathbf{T}']$. After receiving $\mathbf{M}$, the server computes the new ciphertext $\mathbf{c}'$ under the new key $\mathbf{S}'$.

III. Framework and Security Definition

A. System Framework

To construct an inverted-index-based PEKS scheme that can realize retrieval control, we slightly modify the typical PEKS scheme, which consists of the following six algorithms. The system framework includes three entities: 1) the sender; 2) the cloud; and 3) the recipient (and other users), whose specific functions are shown in Fig. 2.

1) *KeyGen*(λ) $\rightarrow$ (pk, sk): Input a security parameter λ and output the public-private key pair (pk,sk).
2) *PEKS*(pk, H) $\rightarrow$ I: Input the index matrix H and the receiver's public key pk, and output a searchable encrypted index I.
3) *Trapdoor*(sk, w_i) $\rightarrow$ T_{w_i}: Input a keyword $w_i \in W$ and the private key sk, and output a trapdoor T_{w_i}.

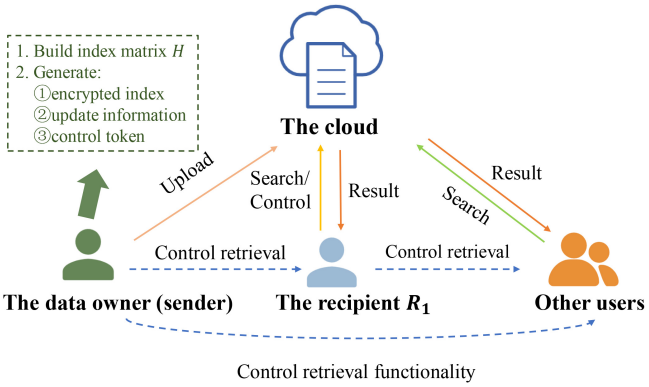


Fig. 2. System framework.

4) *Test*$(I, T_{w_i}) \rightarrow c_{w_i}$*:* Input the encrypted index I and the trapdoor T_{w_i}, and output the result set c_{w_i}.

Specifically, combined with auxiliary information, the owner constructs an index matrix H using file identifiers and keywords. Then, he encrypts H by using the recipient's public key pk to obtain the encrypted index I and sends the encrypted index to the cloud along with the encrypted data.

To make the scheme dynamic, we equip it with an *Update* protocol.

1) *Update*$(U, pk, I) \rightarrow I_{NEW}$*:* Input the update information U and the encrypted index I, and output the updated index ciphertext I_{NEW}.

In this phase, the data owner generates the update information U by keywords and identifier of the file to be updated and encrypts it under the public key pk. After receiving the ciphertexts, the server updates the encrypted index I. We call this "Noninteractive Update," i.e., the data owner does not have to implement the interactive "retrieve-decrypt-change-reencrypt-write back" protocol with the server.

In multiuser scenarios, to achieve controlled keyword search and file access, we also equip it with the *Control* protocol.

1) *Control*$(I, T_G) \rightarrow I_G$*:* Input the encrypted index I and the control token T_G, and output the controlled index ciphertext I_G.

B. Security Definition

Semantic Security: The basic security definition of typical PEKS is chosen-keyword attack (CKA) security. It requires that the adversary $\mathcal{A}$ cannot learn any information about the keyword from the encrypted index without seeing the related search trapdoor. In our dynamic PEKS scheme based on an inverted-index, we define *semantic security* for it based on *CKA* security definition. It ensures that $\mathcal{A}$ cannot learn any information about keywords from the index ciphertext and search results even if $\mathcal{A}$ can obtain the trapdoor of the keyword. Formally, *semantic security* is defined by the following *Game* between $\mathcal{A}$ and a challenger $\mathcal{C}$.

Step 1: The challenger $\mathcal{C}$ performs the following: Inputing a security parameter λ, $\mathcal{C}$ runs KeyGen$(\lambda) \rightarrow (pk, sk)$. Inputing the ordered keyword space $W = \{w_1, \ldots, w_m\}$ and the file identifiers set ID $= \{id_1, \ldots, id_h\}$, $\mathcal{C}$ runs $PEKS(pk, H) \rightarrow I$. Then, $\mathcal{C}$ sends pk, W, and ID to $\mathcal{A}$.

Step 2: $\mathcal{A}$ can adaptively make *Trapdoor* and *Test* queries to a challenger $\mathcal{C}$ for arbitrarily chosen keywords. $\mathcal{C}$ responds to the queries as follows:

1) *Trapdoor Query:* After receiving the trapdoor query for a keyword w_i from $\mathcal{A}$, $\mathcal{C}$ runs Trapdoor$(sk, w_i) \rightarrow T_{w_i}$ and sends T_{w_i} to $\mathcal{A}$.
2) *Test Query:* After receiving the search query for a trapdoor T_{w_i} from $\mathcal{A}$, $\mathcal{C}$ runs Test$(I, T_{w_i}) \rightarrow c_{w_i}$ and sends c_{w_i} to $\mathcal{A}$.
3) *Update Query:* After receiving the update information from $\mathcal{A}$, $\mathcal{C}$ uses it to update the encrypted index.

Step 3: $\mathcal{A}$ selects two keywords $w_0, w_1 \in W$ and sends to $\mathcal{C}$. $\mathcal{C}$ randomly selects a bit $b \in \{0, 1\}$, and generates trapdoor T_{w_b} for w_b. Then, $\mathcal{C}$ sends back to $\mathcal{A}$ the trapdoor T_{w_b} and the returned result c_{w_b} obtained by running $Test(I, T_{w_b}) \rightarrow c_{w_b}$.

Step 4: $\mathcal{A}$ can continue to adaptively make *Trapdoor*, *Test* and *Update* queries to $\mathcal{C}$ for arbitrarily chosen keywords as step 2.

Step 5: Finally, $\mathcal{A}$ outputs a guess b' that succeeds if $b' = b$.

The advantage of $\mathcal{A}$ in *Game* is defined as $\mathrm{Adv}^{\mathrm{Game}}_{\mathcal{A},DIIBPEKS}(\lambda) = |\Pr[b = b'] - 1/2|$.

Definition 1: A dynamic inverted-index-based PEKS scheme is *semantic security* iff the advantage of $\mathcal{A}$ is negligible in λ in the above game.

The search pattern reveals which queries are associated with the same keyword.

Theorem 1: If the adversary $\mathcal{A}$ outputs correct $b' = b$ with a negligible advantage, then it also implies satisfying search pattern privacy.

Proof: If the search pattern privacy is not satisfied, in step 3 of the *Game*, the adversary $\mathcal{A}$ can output correct $b' = b$ from the trapdoor T_{w_b} with nonnegligible advantage. Thus, *semantic security* implies search pattern privacy. ■

The access pattern reveals which queries return the same result.

Theorem 2: If the adversary $\mathcal{A}$ outputs correct $b' = b$ with a negligible advantage, then it also implies satisfying access pattern privacy.

Proof: If the access pattern privacy is not satisfied, in step 4, $\mathcal{A}$ issues a *Trapdoor* query for the keyword w_0 to obtain the trapdoor T_{w_0}. Then, $\mathcal{A}$ outputs correct $b' = b$ with a nonnegligible advantage based on the results returned by *Test* queries issued with the trapdoors T_{w_0} and T_{w_b}. Thus, *semantic security* implies access pattern privacy. ■

The size pattern reveals the volume of results matching the search trapdoor, as well as the number of keywords in the updated files.

Theorem 3: If the adversary $\mathcal{A}$ outputs correct $b' = b$ with a negligible advantage, then it also implies satisfying size pattern privacy.

Proof: If the size pattern privacy is not satisfied, the adversary $\mathcal{A}$ can output correct $b' = b$ with a nonnegligible advantage as follows:

In step 3, the adversary $\mathcal{A}$ can select and send to $\mathcal{C}$ two keywords $w_0, w_1 \in W$ which search results have different sizes. In step 4, $\mathcal{A}$ issues *Trapdoor* query for the keyword w_0 to get the trapdoor T_{w_0}. Then, $\mathcal{A}$ outputs correct $b' = b$ with a nonnegligible advantage based on the results returned by *Test* queries

issued with the trapdoors T_{w_0} and T_{w_b}. Therefore, *semantic security* implies size pattern privacy. ■

Forward privacy ensures that the newly added file containing the keyword w cannot be linked to the trapdoor generated for the same keywords w in previous search operations.

Forward security was first proposed in the construction of the Dynamic SSE schemes. If the DSSE schemes do not have forward security, the adversary can recover the keywords contained in files by the file injection attack [6]. Therefore, many researchers are studying how to construct a DSSE scheme that supports forward security, but this is not considered in typical PEKS schemes. In fact, the file injection attack is more likely to be applied in traditional PEKS scenarios. Because only the recipient's public key is required for a file injection attack, which is actually equivalent to a KGA. However, being resistant to KGAs does not imply forward security. This is because even if the adversary cannot actively conduct a file injection attack (as in the public key-authenticated SE scheme), the adversary can still judge whether the subsequent files have the same keywords as the previous search results based on the previous trapdoor. Therefore, when the keyword space is small, the adversary can infer the keyword information through prior knowledge (e.g., the search frequency) [6].

Backward privacy ensures that searches for the keyword w cannot be linked to the files containing the keyword w which have been deleted.

Theorem 4: If the adversary $\mathcal{A}$ outputs correct $b' = b$ with a negligible advantage, then it also implies satisfying forward privacy and backward privacy.

Proof: If the forward-backward privacy is not satisfied, in step 4, $\mathcal{A}$ issues an *Update* query for a file containing the keyword w_0. Then, $\mathcal{A}$ outputs correct $b' = b$ with a nonnegligible advantage based on the results returned by *Test* queries issued with the trapdoor T_{w_b}. Thus, *semantic security* implies forward privacy and backward privacy. ■

IV. Propose Scheme

A. Initial Design

It is known that the *Test* process is a linear transformation operation. However, the computation and communication costs of the retrieval process under the traditional HE schemes are intensive. To improve its practicability, we adopt IVHE.

In the index matrix, the bitmap of each row corresponding to a keyword is regarded as an integer, and all the integers form an integer vector which is encrypted via the public key of IVHE. A trapdoor for the keyword w_i is a random key-switching matrix generated according to $\mathbf{B} = [(0, \ldots, 0, 1, 0, \ldots, 0)]$ and the receiver's private key as described in Section II. According to the principle of a linear transformation, we obtain the search result by calculating the trapdoor and the ciphertexts of the integer vectors. At this time, the search result is the result of the linear transformation of matrix $\mathbf{B}$ and the integer vectors. After decrypting, the plaintext is regarded as a bit string where the *id* of the files containing the keyword w_i can be obtained. Furthermore, the randomness of the key-switching matrix makes the trapdoor generated for the same query matrix $\mathbf{B}$ random each time. Meanwhile, the ciphertext of the returned result is also random each time as it is the output of the linear transformation operation using the random key-switching matrix. That is, even the trapdoor and the result ciphertext of the same keyword are different and random. This makes our scheme have several good security features, such as hiding multiple patterns, etc.

In our scheme, it can update the file by homomorphic addition operation, and the size pattern (the number of keywords involved in the file to be updated) can be perfectly hidden in this process. It can also realize noninteractive update, which only requires a round of interaction between the client and the server, reducing information leakage.

However, we have to admit that when the element size or dimension of the vector is too large the initial idea is inefficient, so we have to further optimize this design.

B. Optimized Design

To support large databases, we utilize a packaging technique to handle the index settings to achieve reasonable computational and communication overheads. We set appropriate the vector element size and vector dimensions as needed. Specifically, we divide the rows into groups where each group is regarded as an integer of suitable size. Then, we form the integer vectors with suitable dimensions. In addition, the adversary may learn certain keywords of high query frequency through auxiliary information and then recovers the query keywords through the intersection operation in multiple queries [36]. To prevent this, inspired by [36], we evenly order the keywords according to the search frequency when constructing the index matrix, and store the sorted keywords in a table P. Finally, the index matrix is represented as a set of vectors with a relatively uniform frequency distribution.

We construct an optimized inverted-index-based PEKS scheme (*KeyGen*, *PEKS*, *Trapdoor*, *Test*, *Update*, *Control*) over the keyword space $\mathcal{W}$ as follows.

1) *KeyGen*(λ) $\rightarrow$ (pk, sk)*:* Taking a security parameter λ as input, and output public-private key pair (pk,sk) by running IVHE's key generation algorithm.
2) *PEKS*($PK, \{\vec{iv}\}$) $\rightarrow$ I*:* Take the IVHE's public key pk and the processed index vectors $\{\vec{iv}\}$ as input and output the encrypted index I. More concretely, let ID $= \{id_1, \ldots, id_h\}$ be the ordered shared file identifiers set, and $W = \{w_1, \ldots, w_m\}$ be the keywords set extracted from the shared files. Combined with the auxiliary information of different search frequencies of keywords, the data owner first constructs an index matrix H of size $m \times n$. The owner sets $H_{i,j} = 1$ if the jth file contains the keyword represented by the ith row, and $H_{i,j} = 0$ otherwise. Next, the owner divides the bit string of each row in the index matrix H into l_2 groups of the size of l_1 bits where $l_1 \times l_2 = n$, and each group is regarded as an integer. Now, the matrix can be regarded as a $m \times l_2$ integer matrix H_{int}. Then, each column of the integer matrix is divided into l_4 groups of the size of l_3, where $l_3 \times l_4 = m$, and each group is transformed into an integer vector of l_3 dimensions. Therefore, the index matrix is transformed into $l_4 \times l_2$ integer vectors $\{\vec{iv}\}$. Finally, each

Algorithm 1 *PEKS*

Input: The public key pk, ID, W
Output: The encrypted index I

Init Table P, two-dimensional array A
for the index matrix H of size $m \times n$ **do** ▷ Dependent on auxiliary information
  if $w_i \in id_j$ **then**
    $H_{(i,j)} = 1$
  else
    $H_{(i,j)} = 0$
  end if
end for
for $i = 1$ to m **do**
  Divide the bit string into l_2 groups of the size of l_1 bits which is regarded as integers ▷ Obtain integer matrix H_{int}
end for
for $j = 1$ to l_2 **do**
  each l_3 integers form a vector $\vec{iv}$
  $c_{sj} = IVHE.Enc(pk, \vec{iv})$
end for
for $i = 1$ to l_4 **do**
  for $j = 1$ to l_2 **do**
    $A[i][j] \leftarrow c_{ij}$ ▷ Send A to the server
  end for
end for

integer vector is encrypted via pk of IVHE's encryption algorithm to obtain the ciphertext set $\{c_{ij}\}$ of the integer vectors. An example is shown in Fig. 3(a). The data owner stores the ciphertext set in the 2-D array $A[i][j]$ where $1 \le i \le l_4, 1 \le j \le l_2$. See Algorithm 1 for details.

1) *Trapdoor*$(sk, w_i) \rightarrow T_{w_i}$*:* When the recipient wants to query the keyword w_i, he calculates the corresponding location of w_i in the vector by table P. First, the recipient queries the position p of w_i in table P and calculates the sequence number s and the corresponding position l through $s = 1 + (p-1)/l_3$ and $l = 1 + (p-1) \bmod l_3$. s represents the row number of the vector corresponding to w_i in the index array, and l represents the position of the element corresponding to w_i in the vector. Next, the recipient constructs a 1-D integer matrix $G_{1\times l_3}$ setting the lth element to 1, and the elements in other locations to 0. The recipient wants to perform a search operation in the ciphertext state. According to Section II, first, he randomly generates a new private key sk' based on the parameter settings of the matrix G. Then, he computes the key-switching matrix M using G, sk, and sk'. Finally, he sends M and the sequence number s to the server as a trapdoor T_{w_i}, refer to Algorithm 2.
2) *Test*$(I, T_{w_i}) \rightarrow c_{w_i}$*:* After receiving the trapdoor T_{w_i}, the server executes the linear transformation operation using M and all ciphertexts in the 2-D array $A[s][j]$, $1 \le j \le l_2$, and returns the result A' as the test result c_{w_i} to the recipient, refer to Algorithm 2.

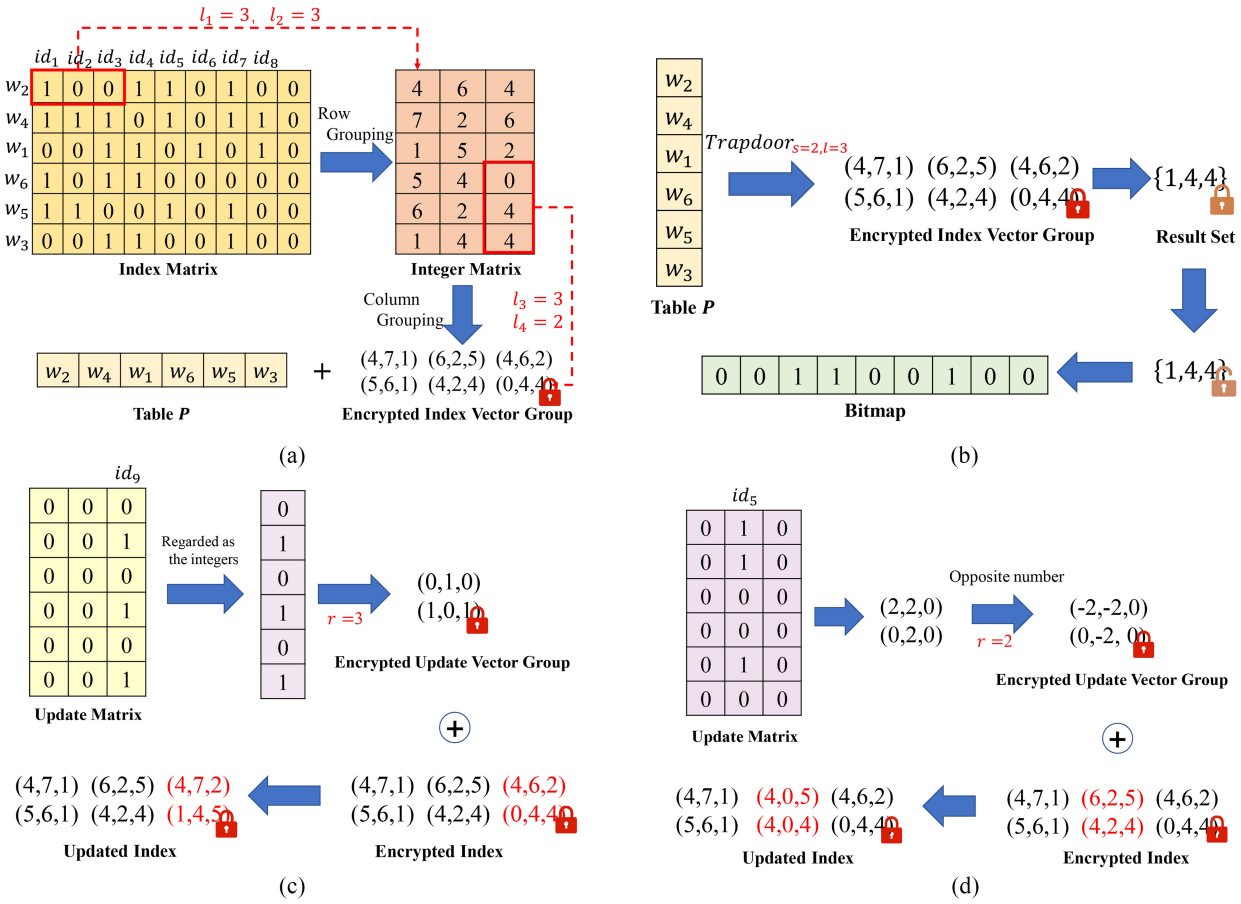
Algorithm 2 *Search* (*Trapdoor* − *Test*)

Input: w_i, (pk, sk), the encrypted index A
Output: The result identifiers $\{ID_{w_i}\}$

//Client:
Find p ▷ the position of w_i in table P
$s = 1 + (p-1)/l_3$ ▷ s represents the row number
$l = 1 + (p-1) \bmod l_3$ ▷ l represents the corresponding position in the vector
Init a one-dimensional zreo integer matrix $G_{1\times l_3}$
$Trapdoor(sk, w_i) \rightarrow T_{w_i}$
//Server:
for $j = 1$ to l_2 **do**
  $c_{w_j} = T_{w_i} \odot A[s][j]$ ▷ $\odot$ means linear transformation operation
  store $c_{w_j} \rightarrow A'$
end for
send A' to the client
//Client:
$\vec{ps} \leftarrow IVHE.Dec(sk, A')$
Form $\vec{ps}$ into bitmap bm in order (where each integer is treated as l_1 bits)

Adjustable Security: Different users or different fields may have different security requirements, such as sensitive databases needing high security, so we make a tradeoff between efficiency and security. In the above scheme, we match the trapdoor with the index ciphertext of the corresponding position (e.g., $A[s][j]$). The adversary can know the exact keyword to be searched with a small probability, but it has to be admitted that there is a certain leakage in this process.

Countermeasures: To achieve adjustable security, our remedy is to control the matching number of encrypted indexes and trapdoors in the *Test* phase. That is, the client can independently choose the number of the index to be matched according to the requirements of efficiency and security. For example, to achieve perfect search pattern hiding, we require trapdoors to match all ciphertexts in the encrypted index. Then, the matching results are stored in array A''. At this time, according to Section V, the computational cost of the search process is still acceptable (only 68 s for 1 million files), but the communication cost is relatively large. Therefore, to save communication overhead, we introduce a trusted execution environment (TEE) (e.g., Intel SGX). The TEE refers to a secure area built into the CPU through hardware and software methods. It guarantees the security, confidentiality, and integrity of the code and data loaded into that environment. In SGX, the environment in which an application runs is called an Enclave. SGX provides an attestation mechanism for the remote third party and Enclave, called SGX remote attestation. Through this technology, the client establishes a secure channel with the Enclave on the cloud server using the Diffie–Hellman key exchange protocol to achieve secret sharing. Therefore, the client achieves accurate filtering and transmission of search results through secure interaction with the Enclave. Specifically, the matching ciphertexts are read into the Enclave. According to the instructions of the client (i.e., the sequence number s), the



- Figure (a) indicates an example of index construction. Assuming that the index matrix is grouped by $l_1 = 3$ bits, each group is regarded as an integer, forming a $l_3 = 3$-dimensional vector. After encryption, an encrypted index and an auxiliary table P are obtained.
- Figure (b) shows the keyword search process. When searching for keyword w_3, the search trapdoor is first calculated according to table P, and then hand it over to the server for Test operation with A[s][j](i.e. A[2][j]).
- Figures (c)(d) indicate examples of an update operation. Based on the file to be updated (id_9 or id_5), the update matrix is constructed and the auxiliary parameter r is computed. The add and revoke operations are the same, except that the elements of the integer vector are taken as opposite numbers one by one in the revoke operation. The update ciphertext and parameter r are sent to the server to update the encrypted index.

Fig. 3. Illustration of operations. (a) Example of index construction. (b) Example of searching w_3 process. (c) Example of adding the file $id_9 = \{w_4, w_6, w_3\}$. (d) Example of deleting the file $id_5 = \{w_2, w_4, w_5\}$.

Enclave accurately filters out the result ciphertext corresponding to the keyword to be searched and sends them back to the client through the secure channel, where the computational and communication complexity are both $O(1)$. Note that the available memory size of Enclave is limited to (less than) 128 MB. Once exceeded, it will cause high-performance overhead.

After receiving the search result from the server, the recipient decrypts the ciphertext set with the new private key sk' to get the plaintext set ps. Each integer of ps is regarded as an l_1-bit string to which forms a bitmap bm. Finally, the recipient obtains the file identifiers $\{id_j\}$ containing the search keyword w_i by checking whether the jth bit in bm is 1. An example is shown in Fig. 3(b). When the recipient wants to query the keyword w_i again, he needs to recalculate a new search trapdoor (the key-switching matrix) M'.

1) *Update*: Our scheme supports updating operations to make itself dynamic.
Share/Add a New File: If the data owner wants to share a new file with the recipient, he proceeds as follows: He assigns to the new file an identifier $id_u \in ID$ which is not used, and then calculates the second dimension of the 2-D array (encrypted index) A corresponding to id_u as $r = 1 + (u-1)/l_1$. Then, he constructs a zero-bit matrix L of size $m \times l_1$. Suppose that the keywords set contained in the file id_u is W_u. If $w_i \in W_u$, the data owner finds the corresponding position p of w_i according to the table P and sets $L_{p,1+(u-1) \bmod l_1}$ to 1. Each row of L is regarded as an integer, and all the integers are divided in order into l_4 integer vectors of l_3-dimension. *Revoke/Delete a Shared File:* When the data owner wants to revoke a shared file from the recipient, he proceeds as follows: assuming that the file identifier is id_u, the data owner calculates the second dimension of the 2-D array (encrypted index) A corresponding to id_u as $r = 1 + (u-1)/l_1$. Then, he constructs a zero-bit matrix L of size $m \times l_1$. Suppose that the set of keywords contained in the file id_u is W_u. If $w_i \in W_u$, the data owner finds the corresponding position p of w_i according to the

table P and sets $L_{p,1+(u-1) \bmod l_1}$ to 1. Each row of L is regarded as an integer and then the integer is changed to its opposite number. Then, all the integers are divided in order into l_4 integer vectors of l_3-dimension.

These integer vectors are encrypted under the public key pk using IVHE, whose ciphertexts $uc[j]$ $(1 \le j \le l_4)$ and r are sent to the cloud server as the update token. Finally, the server performs a homomorphic addition operation between $uc[j]$ and $A[j][r]$ for $1 \le j \le l_4$, thereby realizing the addition/revoke of the file. The specific process is shown in Fig. 3(c) and (d).

It can be seen from the above update process that our scheme can also add and delete multiple files (corresponding files in the same vector) at the same time, i.e., batch update (partial), which can further improve the update efficiency. The batch update process is similar to the above process and will not be described in detail.

Expand Index: If a new shared file is assigned to the identifier id_{n+1}, i.e., the length of the bitmaps of keywords needs to increase. That is to say, the data owner needs to expand the encrypted index stored in the cloud server. Our scheme can expand the capacity of the encrypted index by $b \times l_1$ files when needed, which is not considered in the existing SE schemes using a bitmap. The data owner only needs to send to the cloud server the integer b and the ciphertext of the l_3-dimensional zero vector encrypted under the public key pk. Then, the server completes the index expansion by adding b zero vector ciphertexts at the end of each $A[i](1 \le i \le l_4)$.

Let's consider the following scenarios: A database is shared within the company, in which the search functionalities of the chairman and ordinary employees are different. The chairman has the authority to retrieve all data, while employees can only retrieve their related data. Similarly, in the medical systems, the doctors should be able to query all information about patients, such as disease/drug information, while patients can only access their own relevant information, and do not have the authority to retrieve the private information of other patients.

Similar scenarios are common in daily life. However, it is difficult to directly transfer the current PEKS schemes to achieve controlled retrieval authority under multiple users. In typical PEKS schemes, when the data owner $\mathcal{S}$ wants to share with multiple recipients U_k the search ability (Scenario 1), he must calculate the keyword ciphertexts for each file under the public key of each recipient. Especially, if the data owner $\mathcal{S}$ has already deleted the locally stored data, $\mathcal{S}$ must redownload all the data files and reextract the keywords for each file to calculate keyword ciphertexts under the public key of U_k. Consider another scenario: when a recipient $\mathcal{R}_1$ wants to share the data received from the data owner $\mathcal{S}$ with another recipient $\mathcal{R}_k$ (Scenario 2), the naive solution is also that $\mathcal{R}_1$ redownloads the data and recalculates the keyword ciphertexts under the public key of $\mathcal{R}_k$. In both scenarios, the common solution leads to heavy computing and communication overheads. "Proxy Re-encryption with keyword search" provides an alternative solution. It can transfer the keyword ciphertexts under the public key of $\mathcal{R}_1$ to that under the public key of $\mathcal{R}_k$ by the cloud server if needed. However, although there are some schemes proposed, they all need to transfer the keyword ciphertexts for all the data files one by one. They also have problems, such as no resistance to the KGA attacks [37], [38], [39], [40], deterministic trapdoor for the same keyword [37], [38], [39], [41], and using a designated tester [41].

Our scheme can solve these problems. To our knowledge, our scheme is the first PEKS scheme that can be directly extended to multiple users while ensuring high security. Since the owner can establish an isolation mechanism among multiple users by using only linear transformation operation, he does not need any trusted third party nor predetermine the users.

1) *Control:* Our scheme can realize multisearchable management.

Scenario 1: Data owner $\mathcal{S}$ constructs the encrypted index I via $PEKS(pk, W) \rightarrow I$ and sends it to the server. If $\mathcal{S}$ wants to share with recipients $U = \{U_1, U_2, \ldots\}$ the file access and the keyword search functionalities, the core is to build and store a new encrypted index I_k for each recipient $U_k \in U$ in the cloud. Since the key-switching matrix can change the key-ciphertext pair, the original index is switched to the index ciphertext under the new private key without revealing the keyword information. Specifically, $\mathcal{S}$ generates key pair (pk_k, sk_k) for U_k, switches the ciphertexts in I to the new ciphertexts under the public key pk_k in the cloud, and then gives the new private key sk_k to the recipient U_k.

① *Control the Search Functionality of Keywords:* During this process, the data owner $\mathcal{S}$ can control the keyword search functionality of each user $U_k \in U$, as follows:

The data owner $\mathcal{S}$ constructs a zero matrix $G_{l_3 \times l_3}$ (l_3 corresponds to the dimension of the integer vector introduced Section IV-B). Then, under the parameter settings of G, $\mathcal{S}$ randomly generates a new key pair (pk_{k_w}, sk_{k_w}) of the IVHE algorithm. By adjusting the elements in matrix $G_{l_3 \times l_3}$, $\mathcal{S}$ controls the keyword search authority of each user U: First, $\mathcal{S}$ finds out the keyword w_i to be authorized to the user U_k and judges the position l of the corresponding integer of w_i in the vector. Next, the element in the lth row and the lth column of the matrix G is set to 1. At this time, the integers corresponding to the unauthorized keywords in the result ciphertext obtained by linear transformation are 0.

The linear transformation process is as follows.

a) Based on the matrix G, the original private key sk and the new private key sk_{k_w}, the data owner S calculates the control token (key-switching matrix) T_{k_w} and sends it to the cloud.

b) The cloud generates a new encrypted index I_{k_w} for U_k by the token T_{k_w} and the corresponding ciphertext (i.e., the row(s) corresponding to the keyword(s) to be authorized) in the encrypted index, where I_{k_w} is the ciphertext under the new public key pk_{k_w}.

The data owner $\mathcal{S}$ can generate different G according to the keywords to be authorized, and then repeat the calculation process of linear transformation multiple times. Finally, $\mathcal{S}$ sends the new private key sk_{k_w} to the user U_k. Therefore, $\mathcal{S}$ can control the retrieval ability of other

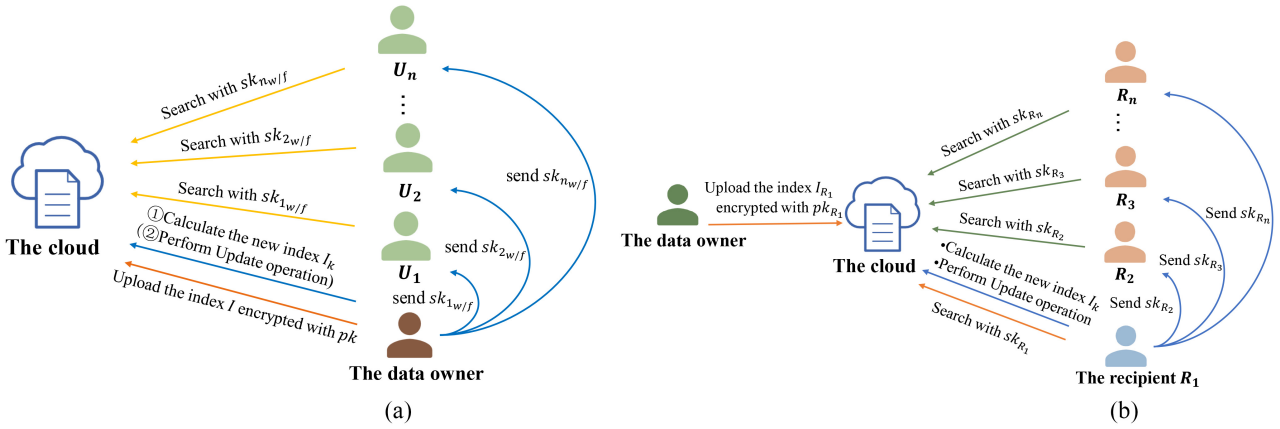


Fig. 4. Different scenarios. (a) Control the access authority of multiple recipients to files and keywords—Scenario 1. (b) Control the access authority of multiple recipients to files and keywords—Scenario 2.

recipients for his keywords without downloading the data files and the encrypted index. The user U_k can use the private key sk_{k_w} to generate trapdoors to search for keywords on I_{k_w}, but cannot access the initial encrypted index I. The process is shown in Fig. 4(a).

② *Control the Access Authority to Files:* When data $\mathcal{S}$ wants to control the access authority to the files by each user $U_k \in U$, the specific process is as follows.

The owner $\mathcal{S}$ constructs the matrix $G_{l_3 \times l_3}$, where G is the identity matrix. Similarly, $\mathcal{S}$ randomly generates a new key pair (pk_{k_f}, sk_{k_f}) of the IVHE algorithm. Based on the actual needs, $\mathcal{S}$ selects the corresponding columns (i.e., the columns corresponding to the files to be authorized) in the encrypted index and the matrix G to perform the linear transformation operation in the ciphertext state. As a result, a new encrypted index I_{k_f} is generated, which is based on the new public key pk_{k_f}.

The linear transformation process is as follows.

a) According to matrix G, the original private key sk and the new private key sk_{k_f}, S calculates the control token T_{k_f} which is sent to the cloud.

b) The cloud server generates a new encrypted index I_{k_f} for the user U_k by calculating the token T_{k_f} and the index part that the owner S wants to share.

If necessary, $\mathcal{S}$ can further delete/revoke certain files that he does not want to authorize in the encrypted index I_{k_f} by the *Update* operation. Finally, $\mathcal{S}$ sends the new private key sk_{k_f} to the user U_k. At this time, the user U_k can use the private key sk_{k_f} to generate trapdoors to search for keywords on I_{k_f}, but cannot access the initial encrypted index I. The process is shown in Fig. 4(a).

Scenario 2: Assume that the encrypted index built for the recipient $\mathcal{R}_1$ is I_{R_1}. $\mathcal{R}_1$ does the same operations as $\mathcal{S}$ does for U_k in Scenario 1. I.e., $\mathcal{R}_1$ can control other recipients' access authority to files and keywords through linear transformation and update operations. $\mathcal{R}_1$ generates key pair $(pk_{\mathcal{R}_k}, sk_{\mathcal{R}_k})$ for $\mathcal{R}_k$, builds the encrypted index I_{R_k} for $\mathcal{R}_k$ based on his own encrypted index I_{R_1}, and then gives the new private key $sk_{\mathcal{R}_k}$ to the recipient $\mathcal{R}_k$. $\mathcal{R}_k$ can use the private key $sk_{\mathcal{R}_k}$ to generate trapdoors to search for keywords on I_{R_k} rather than I_{R_1}. The process is shown in Fig. 4(b).

In our scheme, neither requiring a trusted third party nor predetermining the number of users, the data owner can securely control the access authority of other users to his data. That is, our scheme can be extended to a more demanding multiuser setting under fewer restrictions.

C. Proof of Security

The security of the proposed scheme depends on the security of the IVHE scheme. Please refer to [7], [33], and [34].

Theorem 5: Our scheme satisfies semantic security.

Proof: We analyze the semantic security of the scheme by a sequence of games.

Game$_0$: It is the same as *Game* except that in step 3, the challenger $\mathcal{C}$ obtains the returned result c_{w_b} as follows: Without loss of generality, assume that w_b is w_i. $\mathcal{C}$ first computes $l = 1 + (i-1) \bmod l_3$ and $h = (i-1)/l_3$. Then, $\mathcal{C}$ gets keywords set $W_r = \{w_{h \times l_3 + j}\}_{1 \le j \le l_3}$ where $w_{h \times l_3 + l} = w_i = w_b$, and generates the bitmap for each keyword in W_r. $\mathcal{C}$ builds a matrix using these bitmaps and divides each bitmap into l_2 integers which can be regarded as an $l_3 \times l_2$ integer matrix. Then, each column of the integer matrix is transformed into an integer vector of l_3 dimension. Thus, the matrix is transformed into $1 \times l_2$ vector array. Finally, the public key pk of IVHE is used to encrypt each vector in the integer vector array in order, thereby obtaining the ciphertext set $\{c'_t\}_{1 \le t \le l_2}$. Then, $\mathcal{C}$ replaces corresponding elements in the 2-D array A with $\{c'_t\}_{1 \le t \le l_2}$, i.e., $A[h+1][t] = c'_t$.

Observe that the process to generate the corresponding part of the encrypted index for W_r is a normal process in the PEKS algorithm of our scheme. Consequently, *Game$_0$* is the same as *Game*.

For $1 \le j \le l_3$, the following operations are available.

Game$_j$: is the same as *Game$_{j-1}$* except that the bitmap of $w_{h \times l_3 + j}$ is a random bit string.

We can obtain that $Game_{j+1}$ and $Game_j$ are indistinguishable for $0 \leq j \leq l_3 - 1$ by the semantic security of the IVHE scheme.

Finally, in $Game_{l_3}$, all the bitmaps corresponding to W_r are random bit strings. In our scheme, since the trapdoor (the key-switching matrix) generation algorithm is probabilistic, the generated search trapdoors are different even if search the same keyword. In addition, since the search frequency among index groups is indistinguishable, it is difficult for the adversary $\mathcal{A}$ to distinguish search trapdoor of w_b with the aid of auxiliary information. Similarly, according to the linear transformation calculation process, the returned results by the *Test* algorithm are also random and indistinguishable. Even if the same keyword is searched, the result ciphertexts are different. Therefore, the semantic security of IVHE prevents $\mathcal{A}$ from guessing the right challenge keyword by the test results with nonnegligible advantage. Since the number of identifiers corresponding to all keywords in the bitmap is the same, $\mathcal{A}$ cannot distinguish keywords based on the size of the result ciphertext. If the adversary $\mathcal{A}$ only has trapdoors, it can only distinguish w_b with negligible advantage. Likewise, even if the adversary $\mathcal{A}$ issues *Update* queries and changes the encrypted index, the result returned by the Test algorithm on the changed encrypted index is indistinguishable from the result returned by the Test algorithm on an encrypted index for the random bitmap matrix. Actually, even if the encrypted index is not changed, the results returned by the *Test* algorithm with different trapdoors are indistinguishable even if for the same keyword. Since all keywords participate in the update process, the adversary $\mathcal{A}$ cannot determine the number of keywords contained in the update file. Thus, the advantage of the adversary in $Game_{l_3}$ to output correct b' is negligible. ■

From the search pattern privacy and size pattern privacy, the adversary $\mathcal{A}$ cannot determine the specific search trapdoor and the specific keyword involved in the update process. Since it is difficult to link the trapdoor with the keyword to be updated, so the forward-backward security is realized.

In summary, the advantage of $\mathcal{A}$ winning the *Game* is negligible, so our scheme achieves adaptive semantic security.

Theorem 6: Our scheme can resist offline KGAs.

Proof: Our scheme constructs a (keyword, identifier bitmap) matrix and transforms it into integer vectors as the index. We use the key-switching matrix as the trapdoor and utilize the linear transformation operation of IVHE to realize the *Test* operation. Because there is no specific keyword ciphertext in our scheme, it can prevent the adversary from generating ciphertexts for known keywords to test a trapdoor as what the adversary does in KGA attacks for the typical PEKS schemes. In addition, the location of the real search result for a keyword is only known by the receiver itself and the search result plaintext must be obtained by decrypting the ciphertexts of the vectors set of the real search result using the receiver's private key. This guarantees that the adversary cannot determine which specific keyword is related to a trapdoor, thus achieving stronger security than the typical PEKS schemes. It completes the proof. ■

TABLE I
PERFORMANCE COMPARISON FOR DIFFERENT DIMENSIONS

	10	20	40	50
Construction of index (vectors/s)①	4000	1910	1040	588
Pre-computation (vectors/s)①	3225	1587	793	636
Trapdoor generation (ms)	0.018	0.1	0.13	0.24
Trapdoor (KB)	19.6	39.3	78.7	98.4

① $vectors/s$ represents the number of vectors running unit time

Different from the previous PEKS schemes resisting offline KGAs, our scheme neither needs a designed tester nor need to use private keys for signature.

Theorem 7: Our scheme achieves search pattern privacy, access pattern privacy, size pattern privacy, and forward and backward privacy.

Proof: From Theorems 1–5, we can straightly prove that our scheme achieves search/access/size pattern privacy, and forward and backward security. ■

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance concerning the computational and communication overheads of the *PEKS*, *Trapdoor*, *Test*, and *Update* operations, and analyze the additional overheads of *Control* protocol for the new application scenarios. We use the Numpy library in Python to implement the scheme on the Windows platform of Intel Core i7-8700 CPU @ 3.20 GHz 3.19 GHz, and 16.0 GB (15.8 GB is available) RAM. In the experiment, we modify the IVHE algorithm [7],[3] where the parameters $\ell = 63$ and $w = 2^{31}$ are set.

First, we determine the initial storage capacity of the index matrix as 1000×1240, which represents 1000 keywords and 1240 files.[4] To optimize efficiency as much as possible, we take the parameter $l_1 = 31$ bits, so each row in the index matrix is regarded as $l_2 = 40$ integers. Subsequently, the encrypted index can be expanded in real-time according to the files and keywords to be added. To further reduce the computational cost, the client has stored some zero-vector ciphertexts in advance. As a result, when the client requests to encrypt the zero vector, he does not need to encrypt it again but directly uploads the stored zero vector ciphertext.[5] Considering the special property of the IVHE algorithm, the ciphertext of a zero vector is still a zero vector. To address the problem, we replace the first element of the vector with a random 31-bit nonzero integer so that it is a nonzero integer vector.

Nextly, we evaluate the performance when the vector takes different dimensions, i.e., l_3 takes 10, 20, 40, and 50. Note that l_3 is restricted by the number of keywords (that is, m). All the reported costs are the average of 100 experiments. The results are shown in Table I.

[3]https://github.com/secureldata/Efficient-Integer-Vector-Homomorphic-Encryption

[4]This size is set to facilitate subsequent packing.

[5]Homomorphic addition is much faster than encryption. In the experiment, the client can continuously generate new zero vector ciphertext by homomorphic addition among the stored zero vector ciphertext to meet different needs.

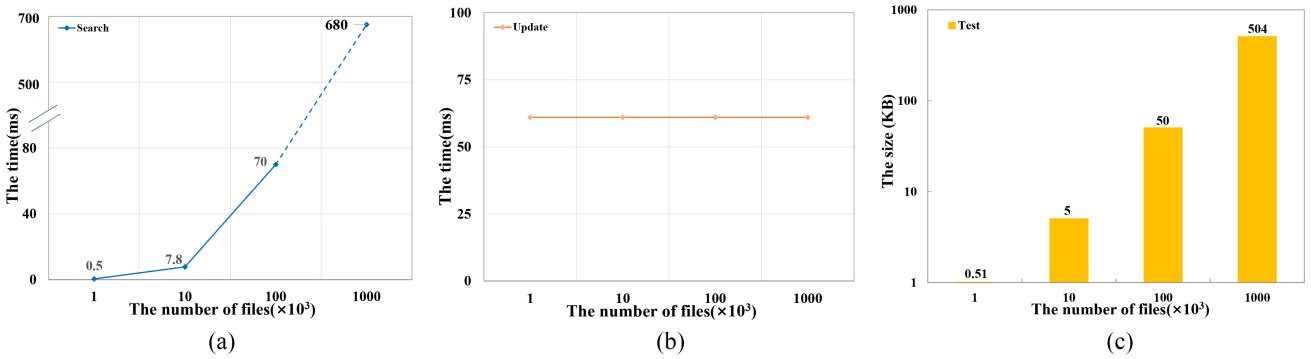


Fig. 5. Performance evaluation. (a) Computational cost of *Search* protocol. (b) Computational cost of *Update* protocol. (c) Communication overhead of *Search* protocol.

According to Section IV-B, the size and generation time of the trapdoor are independent of the number of files and keywords but are linear to the vector dimension. We can interpret the result as follows. In the *Test* process, it is required to perform linear transformation using the trapdoor and the vector ciphertexts in the encrypted index, so the computational cost of trapdoor generation is only dominated by the dimension of the query matrix, i.e., vector dimension l_3. When l_3 is fixed, the computational cost and the communication overhead of trapdoor generation remain constant. As shown in Table I, as the number of dimensions increases, the generation time also increases. After comprehensive consideration, we choose the parameter $l_3 = 10$ to continue the experiment. At this time, the initial index matrix can be indicated as 4000 vectors.

A. Computational Cost

Precalculation: To avoid redundant calculations, we require the server to perform some precalculation. That is, the server converts the index ciphertext c in A into the intermediate bit representation c^*. This operation is only performed once in the *Setup* stage and can accelerate the *Test* process. Therefore, this consumption is acceptable. Refer to Table I for the overhead of precomputation operations.

Test Phase: Unlike traditional PEKS schemes, in our scheme, we do not need to match the trapdoor to every keyword ciphertext in each file, so it can be applied to large databases. The specific cost is shown in Fig. 5(a) which includes testing and decrypting the results.

As shown in Fig. 5(a), the test time is linear with the number of files. The reason is that when the number of files increases, the number of vectors representing the index also increases. Therefore, more vector ciphertexts need to be linearly transformed with the trapdoor and then decrypted, which leads to a larger computational cost.

SGX Evaluation: In the tradeoff between efficiency and security, to save communication overhead and securely transmit the result ciphertext to the receiver, we introduce a TEE (e.g., SGX). Through the Intel SGX remote attestation mechanism, users can establish a secure channel with the Enclave on the cloud server using the Diffie–Hellman key exchange protocol, which takes 3.3 s. Since the process is only executed once during the entire system life cycle, we consider it acceptable.

Update Phase: From Section IV-B, the sender encrypts the update vectors[6] and sends them to the server. After receiving the ciphertexts, the server performs precomputation to convert them into the intermediate ciphertext (additional overhead), and then performs a homomorphic addition operation with the index ciphertext. Next, we consider the worst-case scenario where the update process does not involve zero-vector encryption, and the specific cost is 61 ms. As demonstrated in Fig. 5(b), the cost is independent of the number of files but is only related to the number of vectors involved in the keywords to be updated. Among them, the conversion overhead (additional overhead) is 31 ms.

B. Communication Cost

Since we adopt a unique index-building method and search method, we should analyze the specific communication overheads. In the *Test* phase, the communication overhead for the recipient to upload a search trapdoor is 19.6 kB, and the communication overhead of the cloud sending the search result to the recipient is shown in Fig. 5(c). It can be seen that the overhead is increasing linearly with the number of files. In addition, due to the introduction of SGX, when the client increases the number of encrypted index to be matched, the communication cost remains unchanged (same communication overhead as searching only for A[s][j]). In the *Update* phase, the sender uploads the ciphertexts of 100 integer vectors to the cloud server, for which communication overhead is 15.6 kB. According to Section IV-B, when the size of the keyword space remains unchanged, the communication overhead of the *Update* process also remains unchanged.

C. Cost of Control Protocol

In addition, we test the additional overheads of the *Control* protocol that implements controlled keyword retrieval in new application scenarios. It is known that traditional PEKS schemes cannot be directly extended to achieve multisearchable management. The *Control* protocol of our scheme perfectly solves this problem.

[6]Since the zero vector ciphertexts are stored in advance, when encryption of the zero vector is involved, the sender does not need to encrypt again.

TABLE II
COMPARISON OF COMPLEXITY ANALYSIS

	Build Inde/PEKS	Trapdoor generation	Search/Test	Comm_complexity (Search)
[12]	$2 \times m(\ell' \times \ell' \ C_E + \ell' \times \ell' \ C_{mul}) + mC_{sig}$	$2(\ell' \times \ell' \ C_E + \ell' \times \ell' \ C_{mul}) + C_{mul'} + C_{add} + C_{E'}$	$(m \times 2 \cdot \ell' + 1) \ C_p$	O $(2\ell' \cdot \lvert G \rvert + C_{sig} + \sigma)$
[4]	$3 \ C_E + C_H$	$C_E + C_H + C_p$	$2n \ C_p$ at best and $2mn \ C_p$ at worst	$O(\sigma)$
Our	$l_2 \times l_4 \ C_{enc}$	C_{ke-sw}	$l_2(C_{li-tr} + C_{dec})$	$O(n)$

The symbols C_E, C_{mul}, and C_H denote the complexity of modular exponentiation, multiplication operation on Bilinear group G and hash operation. The symbols C_{sig}, C_p, C_{add}, and $C_{mul'}$ denote the complexity of signature, pairing operation, addition operation, and multiplication operation in $\mathbb{Z}_q$, respectively. C_{enc}, C_{dec}, C_{ke-sw} and C_{li-tr} denote the complexity of encryption and decryption operations of IVHE, key switching matrix, and linear transformation operation. The symbol σ represents matching file information.

TABLE III
COMPARISON OF PERFORMANCE

	Build Index/PEKS	Trapdoor generation	Search/Test	Communication volume
[12]	When $\ell' = 3$, it takes 6 seconds①	6 milliseconds	6.001 seconds ①	about 1300 Byte ④
[4]	80 seconds at best and 800s at worst②	6 milliseconds	20 seconds at best and 200s at worst③	⑤
Our	When $\ell = 63$, $w = 2^{31}$, it takes 18.5 seconds.	18 microseconds	7.8 milliseconds (search+decrypt)	5×10^3 Byte

① The encryption time of file id information is not taken into account. It takes 1 millisecond to test a pairing operation.
② It takes 8 milliseconds to perform PEKS once. When constructing the index during Build Index, it is necessary to perform the PEKS operation on each keyword in the files, with a total of performing $\sum_{i=1}^{n} m_i$ PEKS (where n is the number of files, and m_i the number of keywords in the i-th file). In the best case, where the file contains only one keyword, it takes 80 seconds to perform 10000 PEKS operations. Assuming that a file has 10 keywords, it takes 800s in the worst case.
③ It takes 2 milliseconds to perform the Test once. When Searching, it is necessary to perform the Test operation on each keyword ciphertext in the files, and a total of performing $\sum_{i=1}^{n} m_i$ Test. In the best case, the Test is performed 10,000 times, which takes 20s. In the worst case, the Test is performed 100,000 times, which takes 200s.
④ If the file information is stored in bitmap format, the fixed communication overhead is 1300 bytes. When stored in file identifier format, a file identifier takes 2 bytes (not practical when matching files exceeding 700).
⑤ After testing the encrypted index, the results are not returned directly returned to the client but are notified to the server of the matching results.

TABLE IV
COMPARISON OF SECURITY PROPERTIES

	Forward/Backward Security	Search Pattern Privacy	Access Pattern	Size Pattern	Multi-Searchable Management	Non-Interactive Update
[12]	×	✓	×	×	×	×
[4]	×	×	×	×	×	✓
Our	✓	✓	✓	✓	✓	✓

We tested the computational cost and communication overhead of authorizing other users to query keywords or files. According to Section IV-B, combined with the specific needs, the computational cost of the user generating the control token is 0.59 ms, which is independent of the number of files and only related to the vector dimension. The control token is sent to the server, whose communication overhead is 196 kB. After receiving the control token, the server selects the corresponding encrypted index to perform a linear transformation operation with the actual needs. When $l_3 = 10$, the computational cost of this process is 22 000 (vectors/s).

Next, we compare the performance and security of our scheme with these of two typical schemes, including a PEKS scheme [12] based on the inverted index and an authenticated PEKS scheme [4] resistant to inside KGAs. Specifically, we have theoretically analyzed the computational and communication volumes of the three schemes, as shown in Table II. Due to the significant differences in the construction primitives between our scheme and [4], [12], we give the specific computational time and communication volume of each scheme under our parameter settings in Table III. We obtain their performances in the index construction, trapdoor generation, and search phases for 10 000 files and 1000 keywords. Table IV provides a comparison of security between solutions.

It can be seen that our solution has absolute advantages in security and computational cost which supports large databases with millions of files.

VI. INTEGRITY VERIFICATION OF SEARCH RESULT

In this section, we extend the above scheme to verify search results as follows.

In the $PEKS(pk, H) \rightarrow I$ algorithm, besides constructing the encrypted index I described in Section IV-B, a sequence of integers sqv is built for all keywords, where the integer corresponding to the keyword w represents the sum of the l_2 integers formed from the row in the matrix H corresponding to the keyword w. Next, the sequence of integers sqv is divided orderly into l_4 integer vectors of l_3-dimension for integrity verification of the results. For example, as shown in Fig. 3(a), the newly added l_3-dimensional verifiable integer vectors are required to be $\vec{V}_1 = (v_2, v_4, v_1)$ amd $\vec{V}_2 = (v_6, v_5, v_3)$, where $v_2 = 4+6+4 = 14$, $v_4 = 7+2+6 = 15$, $v_1 = 8$, $v_6 = 9$, $v_5 = 12$, and $v_3 = 9$. Then, these l_4 integer vectors are encrypted via the IVHE scheme, whose ciphertexts $vc[j]$ $(1 \leq j \leq l_4)$ called *verification vectors* are sent to the cloud along with I.

In the $\text{Test}(I, T_{w_i}) \rightarrow c_{w_i}$ algorithm, besides the ciphertexts in the 2-D array $A[s][j]$, $1 \leq j \leq l_2$, the trapdoor T_{w_i} also needs to execute the linear transformation calculation with the

ciphertext of each integer vector in the *verification vectors vc*. Then, the server returns the verification values along with the search result. The receiver gets and decrypts the ciphertext of the verification value corresponding to the searched keyword, and then compares the plaintext with the sum of the integers in the result set *ps*. If they are equal, the search result is complete and correct. Otherwise, it means that some errors occur in the returned result. As demonstrated in Fig. 3(b), the verification value obtained by decrypting the ciphertext of v_3 is equal to 9, and the sum of integers of the corresponding search result is $1+4+4=9$, which means that the returned result is complete.

In the *Update* process, in addition to updating the encrypted index, the ciphertexts $vc[j]$ $(1 \leq j \leq l_4)$ also need to be modified by performing homomorphic addition operation with the update vector ciphertexts $uc[j]$ $(1 \leq j \leq l_4)$ in order. As shown in Fig. 3(c), when the file id_9 is added, $\vec{V}_1 = (v_2, v_4, v_1)$ and $\vec{V}_2 = (v_6, v_5, v_3)$ are modified to be $\vec{V}_1 = (v_2+0, v_4+1, v_1+0)$ and $\vec{V}_2 = (v_6+1, v_5+0, v_3+1)$, i.e., $v_2 = 14$, $v_4 = 16$, $v_1 = 8$, $v_6 = 10$, $v_5 = 12$, and $v_3 = 10$. Similarly, as shown in Fig. 3(d), when the file id_5 is deleted, $\vec{V}_1 = (v_2, v_4, v_1)$ and $\vec{V}_2 = (v_6, v_5, v_3)$ are modified to be $\vec{V}_1 = (v_2-2, v_4-2, v_1+0)$, $\vec{V}_2 = (v_6+0, v_5-2, v_3+0)$, i.e., $v_2 = 12$, $v_4 = 13$, $v_1 = 8$, $v_6 = 9$, $v_5 = 10$, and $v_3 = 9$.

For implementation, it is verified that our integrity verification method of the returned result in databases of appropriate size is completely feasible and the extra overhead is small compared with the basic construction.

VII. Conclusion

In this article, we focus on how to improve the practicability and security of PEKS schemes in the scenario that a data owner outsources a large number of data files to the cloud server and shares them with multiple recipients. We design a dynamic inverted-index-based PEKS scheme utilizing index matrix structure and IVHE. Compared with the existing schemes, our construction has distinct advantages in security and computational cost. Using only the linear transformation operation, the data owner (the recipients) achieves multisearchable management, which increases the application value of PEKS. We also consider the verification of the search result which guarantees the correctness and completeness of the returned result and makes it supported by our scheme. To support more file identifiers, we have to consider some specific measures in designing the scheme, which may cause some additional costs, but we trade acceptable overhead for extremely high search efficiency. We believe that in the future, with the development of the IVHE technology, our scheme can continue to be optimized to achieve the goal of more efficiency.

References

[1] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology (EUROCRYPT)*. Interlaken, Switzerland: Springer, 2004, pp. 506–522.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Security Privacy*, 2000, pp. 44–55.

[3] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation," in *Proc. NDSS*, 2012, pp. 1–15.

[4] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Inf. Sci.*, vols. 403–404, pp. 1–14, Sep. 2017.

[5] B. Qin, Y. Chen, Q. Huang, X. Liu, and D. Zheng, "Public-key authenticated encryption with keyword search revisited: Security model and constructions," *Inf. Sci.*, vol. 516, pp. 515–528, Apr. 2020.

[6] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: The power of file-injection attacks on searchable encryption," in *Proc. 25th USENIX Security Symp.*, 2016, pp. 707–720.

[7] H. Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications," in *Proc. ITA Workshop*, 2014, pp. 1–9.

[8] H. Yang, B. Jin, C. Chen, and X. Wu, "Efficient homomorphic encryption and its application," *J. Cryptol. Res.*, vol. 4, no. 6, pp. 611–619, 2017.

[9] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO)*. Santa Barbara, CA, USA: Springer, 2001, pp. 213–229.

[10] M. Abdalla et al., "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in *Advances in Cryptology (CRYPTO)*. Santa Barbara, CA, USA: Springer, 2005, pp. 205–222.

[11] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *J. Syst. Softw.*, vol. 83, no. 5, pp. 763–771, 2010.

[12] R. Zhang, R. Xue, T. Yu, and L. Liu, "PVSAE: A public verifiable searchable encryption service framework for outsourced encrypted data," in *Proc. IEEE ICWS*, 2016, pp. 428–435.

[13] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Secure Data Management*. Berlin, Germany: Springer, 2006.

[14] B. Wang, T. Chen, and F. Jeng, "Security improvement against malicious server's attack for a dPEKS scheme," *Int. J. Inf. Educ. Technol.*, vol. 1, pp. 350–353, Jan. 2011.

[15] X. Pan and F. Li, "Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 102075.

[16] B. Wang, W. Song, W. Lou, and Y. T. Hou, "Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee," in *Proc. IEEE INFOCOM*, 2015, pp. 2092–2100.

[17] Y. Wang, S.-F. Sun, J. Wang, J. K. Liu, and X. Chen, "Achieving searchable encryption scheme with search pattern hidden," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 1012–1025, Mar./Apr. 2022.

[18] C. Hu, P. Liu, R. Yang, and Y. Xu, "Public-key encryption with keyword search via obfuscation," *IEEE Access*, vol. 7, pp. 37394–37405, 2019.

[19] Q. Tang, "Nothing is for free: Security in searching shared and encrypted data," *IEEE Trans. Inf. Forensics Security*, vol. 9, pp. 1943–1952, 2014.

[20] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," in *Data and Applications Security XXII*. London, U.K.: Springer, 2008, pp. 127–143.

[21] A. Kiayias, O. Oksuz, A. Russell, Q. Tang, and B. Wang, "Efficient encrypted keyword search for multi-user data sharing," in *Computer Security (ESORICS)*. Heraklion, Greece: Springer, 2016, pp. 173–195.

[22] D. Sharma and D. C. Jinwala, "Multiuser searchable encryption with token freshness verification," *Security Commun. Netw.*, vol. 2017, pp. 1–16, Nov. 2017.

[23] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Pairing-Based Cryptography Pairing*. Tokyo, Japan: Springer, 2007, pp. 2–22.

[24] H. Wang, X. Dong, and Z. Cao, "Secure and efficient encrypted keyword search for multi-user setting in cloud computing," *Peer-to-Peer Netw. Appl.*, vol. 12, pp. 32–42, Jan. 2019.

[25] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," Apr. 2017, *arXiv:1704.03578*.

[26] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," Ph.D. dissertation, Dept. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 1978.

[27] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[28] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 469–472, Jul. 1985.

[29] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology (EUROCRYPT)*. Prague, Czech Republic: Springer, 1999, pp. 223–238.

[30] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. Symp. Theory Comput.*, 2009, pp. 169–178.
[31] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory*, vol. 6, no. 3, pp. 309–325, Jul. 2014.
[32] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Advances in Cryptology (CRYPTO)*. Santa Barbara, CA, USA: Springer, 2012, pp. 868–886.
[33] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in *Advances in Cryptology (CRYPTO)*. Santa Barbara, CA, USA: Springer, 2008, pp. 554–571.
[34] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. STOC*, 2005, pp. 84–93.
[35] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. FOCS*, 2011, pp. 97–106.
[36] C. Liu, L. Zhu, M. Wang, and Y.-A. Tan, "Search pattern leakage in searchable encryption: Attacks and new construction," *Inf. Sci.*, vol. 265, pp. 176–188, May 2014.
[37] J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Inf. Sci.*, vol. 180, no. 13, pp. 2576–2587, 2010.
[38] L. Fang, W. Susilo, C. Ge, and J. Wang, "Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search," *Theor. Comput. Sci.*, vol. 462, pp. 39–58, Nov. 2012.
[39] Z. Chen, S. Li, Q. Huang, Y. Wang, and S. Zhou, "A restricted proxy re-encryption with keyword search for fine-grained data access control in cloud storage," *Concurrency Comput. Pract. Exp.*, vol. 28, no. 10, pp. 2858–2876, 2016.
[40] Y. Yang, X. Zheng, V. Chang, and C. Tang, "Semantic keyword searchable proxy re-encryption for postquantum secure cloud storage," *Concurrency Comput. Pract. Exp.*, vol. 29, no. 19, p. e4211, 2017.
[41] W. Zhang, B. Qin, X. Dong, and A. Tian, "Public-key encryption with bidirectional keyword search and its application to encrypted emails," *Comput. Stand. Interfaces*, vol. 78, Oct. 2021, Art. no. 103542.

Guiyun Qin received the master's degree from Shandong University, Qingdao, China, in 2023.

Her research interests include searchable encryption, cloud data security, and ciphertext database.

Pengtao Liu received the master's degree from Shandong University, Jinan, China, in 2006.

She is currently a Professor with the School of Cyberspace Security, Shandong University of Political Science and Law, Jinan. Her main research interests include searchable encryption and leakage-resilient cryptography.

Chengyu Hu received the Ph.D. degree from Shandong University, Jinan, China, in 2008.

He is currently an Associate Professor with the School of Cyber Science and Technology, Shandong University, Qingdao, China. His main research interests include cloud system security, public key cryptography, and network security.

Zengpeng Li received the Ph.D. degree from Harbin Engineering University, Harbin, China, in 2017.

He is currently an Associate Researcher with the School of Cyber Science and Technology, Shandong University, Qingdao, China. He has worked on lattice-based cryptography, password-based cryptography, and cryptographic protocol.

Shanqing Guo received the Ph.D. degree from Nanjing University, Nanjing, China, in 2006.

He is currently a Professor with the School of Cyber Science and Technology, Shandong University, Qingdao, China. His main research interests include network security and software security.

Received: 10 August 2022 | Revised: 16 October 2022 | Accepted: 2 December 2022

DOI: 10.1002/ett.4712

RESEARCH ARTICLE

WILEY

Blockchain-based secure deduplication of encrypted data supporting client-side semantically secure encryption without trusted third party

Guiyun Qin[1] | **Limin Li**[1] | **Pengtao Liu**[2] | **Chengyu Hu**[1,3,4] | **Shanqing Guo**[1,3,4]

[1]School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China

[2]School of Cyberspace Security, Shandong University of Political Science and Law, Jinan, Shandong, China

[3]Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, Shandong, China

[4]Quancheng Laboratory, Jinan, Shandong, China

Correspondence
Chengyu Hu, School of Cyber Science and Technology, Shandong University, 72 Binhai Rd, Qingdao, China.
Email: hcy@sdu.edu.cn

Funding information
Shandong Provincial Natural Science Foundation, Grant/Award Numbers: ZR2020LZH002, ZR2020MF055, ZR2021LZH007, ZR2020QF045; The Open Project of Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences, Grant/Award Number: KFKT2019-002

Abstract

To accommodate the new demand for the deduplication of encrypted data, secure encrypted data deduplication technologies have been widely adopted by cloud service providers. At present, of particular concern is how deduplication can be applied to the ciphertexts encrypted by semantically secure symmetric encryption scheme. Avoiding disadvantages of the existing methods, in this article, we propose a blockchain-based secure encrypted data deduplication protocol supporting client-side semantically secure encryption. In the proposed protocol, the smart contracts are deployed by the first file uploader, and then the subsequent uploaders implement an interactive proof of ownership for the same file with the help of the smart contracts executing a cloud data integrity auditing protocol. The smart contracts play the role of the trusted third party and therefore make up for the poor feasibility for the existence of a trusted third party in real scenario. In addition, in the proposed protocol, there is no need for other clients who have uploaded the same file to be online to help the current uploader obtain the encryption key. We also prove its security and evaluate its performance.

1 | INTRODUCTION

The development of cloud storage technology prompts the demand for outsourcing massive data to the cloud servers. Cloud storage brings obvious benefits to data owners, such as saving users from the management and maintenance of local storage, saving local resources, and allowing users to use any device such as smartphones, laptops, and desktop computers anytime and anywhere to access the outsourced data in the cloud server, regardless of time and location. Subsequently, cloud storage brings many problems. In actual applications, cloud service providers are not completely trusted, and data leakage often occurs. For example, Facebook disclosed the user's contact information in 2013;[1] Icloud disclosed users' private photos in 2014;[2] Cathay Pacific Airways Limited leaked passenger personal information, including the passenger's name, passport, and identity information, telephone number, etc., in 2019.[3] To ensure data privacy,

users usually encrypt data before uploading it to the cloud server. As a result, the same data is encrypted into different ciphertexts and stored repeatedly by different users, resulting in a huge waste of storage space. According to the "Data Age 2025" whitepaper of International Data Corporation (IDC), the global data volume is expected to reach 175 Zettabytes by 2025.[4] Another IDC survey shows that 75% of the data are duplicate.[5] Because redundant data wastes an abundance of storage resources, researchers have proposed many different data deduplication protocols based on different encryption algorithms. However, the various deduplication protocols currently on the market have also been proven to have a large number of security risks and inefficiencies: the protocols based on convergent encryption are vulnerable to offline brute-force attacks; the requirement that there exists a trusted third party for some protocols is difficult to meet in reality; the protocols supporting semantically secure client-side encryption without the need for additional independent servers may be suffering some additional attacks or overheads as a result of the requirement for other clients who have uploaded the same encrypted file to implement a key distribution protocol with the current uploader. Therefore, how to securely and efficiently delete duplicate encrypted data[6] has become one of the current research hotspots in the field of cloud storage security.

1.1 | Related work

To protect data privacy, the researchers proposed some methods to deduplicate encrypted data stored on the cloud server. The client uses convergent encryption (CE)[7] and message locked encryption (MLE) scheme[8] to encrypt data, which provides users with practical data privacy options while using the deduplication function. CE was proposed by Douceur et al[7] which computes and uses the hash value of the data as the encryption key, and ensures that the same data always corresponds to the same ciphertext. Although CE is efficient, it does not achieve semantic security and is vulnerable to offline brute-force attacks.[9] Bellare et al proposed a MLE scheme encrypting data under the MLE key which comes from the encrypted hash value of data. In this way, MLE ensures that the same data will be encrypted into the same ciphertext. Compared with CE, the core idea of MLE has not changed so that it is also unable to achieve semantic security.[10,11] Bellare et al[12] proposed *DupLESS*, in which different owners of the same data and the trusted key server execute the obfuscated pseudorandom function (OPF) to generate the same encryption key. To improve the security, Duan[13] proposed a scheme in which the key server is eliminated and a distributed key generation protocol is implemented by the clients before uploading the data. Both Bellare et al's and Duan's protocols[12,13] cannot resist the online exhaustive attack of cloud servers. Puzio et al[14] put forward the first double-encryption based data deduplication protocol ClouDedup in which the inner layer uses efficient CE and the outer layer outsources the encryption and decryption work to a trusted third party. Although the security is improved, double-layer encryption technology brings high computing and communication costs. In addition, CloudeDup cannot prevent collusion attacks between cloud service providers and the third parties. Recently, three encrypted data deduplication methods achieving semantic security[15-17] have been proposed. However, these methods need a trusted third party to generate for each user some auxiliary information such as the key for broadcast encryption (BE)/attribute-based encryption (ABE) and parameters encrypted by BE/ABE scheme. This makes that each user can obtain parameters of other users. Therefore these methods cannot resist collision attacks, that is, if the cloud server compromises one or more users, it can obtain the data of other users. As is known to all, in a real scenario, it is difficult to deploy a fully trusted third party. Therefore, Liu et al designed the protocols[9,18] of encrypted data deduplication without the participation of the trusted third party, using password authenticated key exchange (PAKE) to transfer the encryption key. The server requires the client with the same short hash value to execute the PAKE protocol with the file hash value as the input password, and finally output the session key. After that, clients with the same data will get the same session key. Through this key, a subsequent uploader of the same data can get the encryption key used by the previous uploader. Although the protocol can achieve semantic security, the requirement that some participants must be online leads to less practical.

1.2 | Our contribution

As we mentioned above, existing secure deduplication protocols of encrypted data supporting client-side semantically secure encryption without trusted third-party[9,18] require some clients who have uploaded the same data to be online and run PAKE protocol when a client wants to upload data. To solve this problem, in this article, we propose a

blockchain-based secure deduplication protocol of encrypted data. Based on the advantages of blockchain,[19] our protocol supports client-side semantically secure encryption without the trusted third party and does not need other uploaders to be online. Different from the traditional credit endorsement mechanism relying on the trusted third party, blockchain technology is deeply integrated through P2P network, cryptography technology, consensus mechanism, etc., to solve the problem of establishing a trust mechanism between nodes in a decentralized system.[20,21] Therefore, based on the blockchain, our deduplication protocol of encrypted data solves the third-party trust mechanism problem. In our protocol, smart contracts are used to implement the proof of data ownership. Before uploading the file, we first check whether the file already exists in the cloud server. That is, the current uploader uploads the short hash value of the file to the cloud server to find whether there is an equivalent short hash value in the cloud server. If it exists, the cloud server sends the address set of smart contracts T_0 and T_1 deployed by the original uploader to the current uploader and the current uploader executes the smart contracts. If it is confirmed that the current uploader does fully own the file F, he can obtain the encryption key used by the original uploader. Otherwise, the current uploader is the first uploader. Then he sends the ciphertext of the file and the addresses of the constructed smart contracts T_0 and T_1 to the server. In this way, our protocol does not require other uploaders to be online.

We proved the security of the protocol in the adversarial model. We also analyzed the performance of our protocol. We implemented three experiments by controlling the data size, the block size, and the number of selected checked blocks to study the durations of tag, challenge and proof generation. In addition, it is that our solution does not need to call other uploaders to participate in the protocol, that prevents the collusion attacks of malicious uploaders and the cloud server, and improves the security. On the whole, compared with the existing encrypted data deduplication protocols, our protocol is much more practical.

1.3 | Organization

The rest of this article is organized as follows: Section 2 introduces some preliminaries of our work and gives an overview of our solution. Section 3 describes our blockchain-based deduplication protocol in detail and analyzes the security of the protocol. In Section 4, some experiments are implemented to evaluate the performance of our protocol. Some privacy-preserving technologies for the data encryption key stored in the blockchain are discussed in Section 5. Finally, Section 6 concludes the article.

2 | PRELIMINARIES

In this section, we describe some preliminaries.

2.1 | Cloud data integrity auditing protocol

A cloud data integrity auditing protocol consists of the following four algorithms (SysSetup, AuthGen, ProofGen, ProofVerify):[22]

- SysSetup$(1^\lambda)\rightarrow (PK, SK)$: The system setup algorithm is executed by the client and takes as input a security parameter λ, and generates a public key PK and the client's secret key SK.
- AuthGen$(PK, SK, F)\rightarrow (\Phi)$: The authenticator generation algorithm is executed by the client and takes as input the public key PK, the client's current secret key SK and a file F, and generates the set of authenticators Φ for F.
- ProofGen$(PK, Chal, F, \Phi)\rightarrow (P)$: The proof generation algorithm is run by the cloud server and takes as input the public key PK, a challenge $Chal$ which is randomly selected by the client and sent to the cloud, a file F and the set of authenticators Φ, and generates a proof P that the cloud has correctly preserved F.
- ProofVerify$(PK, Chal, P)\rightarrow$("True" or "False"): The proof verification algorithm is executed by the client to verify the proof generated by ProofGen$(PK, Chal, F, \Phi)$. It takes as input the public key PK, the same challenge $Chal$ used in ProofGen and the proof P, and outputs "True" or "False."

2.2 | Public key encryption

The public key encryption (PKE) scheme[23] consists of three polynomial-time algorithms (KeyGen, Enc, Dec) as follows:

- KeyGen(1^λ)$\rightarrow$ (pk, sk): The randomized key generation algorithm takes as input a security parameter λ and outputs the public key/private key pair (pk, sk).
- Enc$(pk, m)\rightarrow c$: The probabilistic encryption algorithm takes as input a message $m \in \mathcal{M}$ and a public key pk, then outputs a ciphertext c.
- Dec$(sk, c)\rightarrow m$: The deterministic decryption algorithm takes as input a ciphertext c and a private key sk, either outputs a message m or an error symbol $\perp$.

2.3 | Blockchain and smart contract

Blockchain is a new distributed infrastructure and computing paradigm based on transparent and trusted consensus rules.[24] It is composed of data blocks in the form of ordered chain in the peer-to-peer network environment. It uses the consensus algorithm to update data and uses cryptographic technologies to ensure tamper-resilience, unforgeability, and traceability of its data. Although researchers have proposed some methods[25-28] to attack blockchain, these attacks mainly focus on bitcoin. Blockchain is still considered as an effective technology to build trust in distributed nodes.[21,29,30]

In 1994, the concept of smart contract was first proposed by Szabo[31] which is defined as a kind of computer protocol that executes contract terms through information technology. Blockchain can supply a platform for supporting the automatical operation of programmable smart contracts by consensus nodes without any trusted third party once predefined rules have been met.[32]

2.4 | Overview of our solution

In the secure deduplication of encrypted data supporting client-side semantically secure encryption without the trusted third party, there are two types of the participants, that is, cloud storage server $\mathcal{S}$ and data owners $\mathcal{C}$. The data owners are divided into two categories as the original uploader and the subsequent uploader of the data. When an uploader $\mathcal{C}$ wants to upload to $\mathcal{S}$ a data file F which has been originally encrypted and uploaded to $\mathcal{S}$ by another uploader $\mathcal{C}_i$, there must be some mechanisms to ensure that $\mathcal{C}$ can obtain the encryption key K_F used by $\mathcal{C}_i$. Meanwhile, other users without F cannot obtain K_F. In traditional client-side deduplication, there may be an additional independent server to assist the encryption key transfer from $\mathcal{C}_i$ to $\mathcal{C}$ which is unrealistic in practice.

As blockchain is naturally a decentralized system maintained by all nodes in the network, the core idea of our solution of transferring the encrypted key from $\mathcal{C}_i$ to $\mathcal{C}$ is to replace the trusted third party with smart contracts. The smart contracts play the roles of an encryption key holder and a checker for the integrity of the data stored in $\mathcal{C}$, while $\mathcal{C}$ plays the role of an integrity prover to prove that he really holds the data completely.

As shown in Figure 1, the original uploader of the data file F encrypts the data using a semantically secure symmetric encryption scheme, generates and writes into a smart contract the set of authenticators Φ for the plaintext of F by AuthGen algorithm of a cloud data integrity auditing protocol **Audit**. When another data owner of the data file F wants to upload F to the cloud storage, the cloud storage server redirects him to the smart contracts so that a data integrity auditing protocol is implemented between the uploader and the smart contracts. The uploader proves to the smart contracts that he really has the complete data of F as a proof of ownership. Then, he can obtain the data encryption key used by the original data uploader from the smart contracts.

Specifically, to upload a data file F, the data owner first calculates a short hash value h of the file F and sends it to the cloud storage server. The cloud storage server uses the short hash value h to determine whether the data has already been stored in the cloud storage.

If the data file F has not been uploaded, the data owner is the original uploader of this file. He encrypts the data to get the ciphertext C of F by a semantically secure symmetric encryption scheme **SE** under a randomly selected key K. Then, he generates the set of authenticators Φ for F by AuthGen algorithm of a cloud data integrity auditing protocol **Audit**.

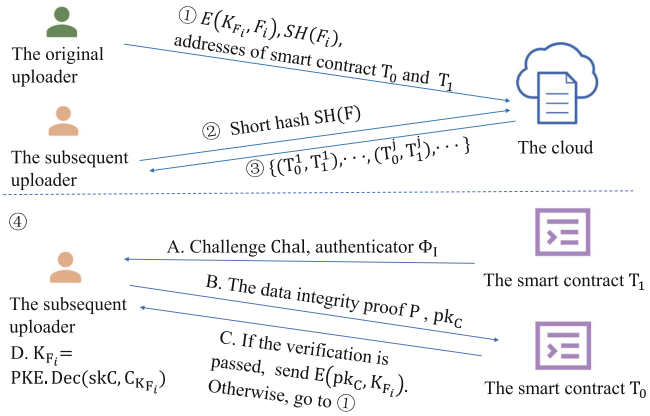


FIGURE 1 The uploading processes example

The ciphertext C of F is outsourced to the cloud storage server. The encryption key K and the code of ProofVerify algorithm of Audit are written into a data-privacy-preserving smart contract T_0 which can be programmed by some coding language for data-privacy-preserving smart contracts such as zkay.[33] The data owner also writes the set of authenticators Φ for F and his public key of **Audit** into a smart contract T_1. Then the data uploading process is done.

If the data file F has been uploaded, the data owner is a subsequent uploader of this file. Then, the cloud storage server redirects him to execute the smart contract T_1. Smart contract T_1 generates challenges *Chal* according to its current state and sends *Chal* and corresponding authenticators in Φ to the uploader. The uploader generates the data integrity proof P of F and executes the smart contract T_0 taking P as its inputs. T_0 implements data integrity verification by the ProofVerify algorithm of **Audit**. If the verification is passed, it means that the uploader actually preserves the data file F. Then, T_0 encrypts the data encryption key K using a PKE scheme under the public key of the uploader and sends the ciphertext C_K of K to the uploader. At last, the uploader can obtain the data encryption key K of the data file F by decrypting C_K under his own private key.

3 | THE PROPOSED PROTOCOL

In this section, we describe our blockchain-based deduplication protocol in detail. Let **PKE** be a public-key encryption scheme, and **Audit** be a cloud data integrity auditing protocol which is described in Section 2. Let *pFunc* be a pseudorandom function and *hFunc* be a hash function.

Step 1. File existence check. The idea of file existence check comes from Liu et al's work,[9] and we modify the data structure maintained by the storage server in their protocol[9] to make it applicable to our blockchain-based protocol. Specifically, when C wants to upload a data file $F = \{m_1, \ldots, m_n\}$ to S, it first sends a short cryptographic hash $sh = SH(F)$ of F to S so that it can check the existence of F in S storage. Note that due to the high collision rate of $SH()$, S cannot use sh to reliably guess the content of F offline.[9] Actually, since different files may have the same short hash, S can only check the existence of the ciphertext of the files $(F_1, F_2, \ldots, F_i, \ldots)$ whose plaintexts have the same short hash value due to the high collision rate of the hash function SH. S should record the index of files whose plaintext maps to the same short hash, the list of clients who have stored the same data file in the cloud, and the original uploader of a file. The data structure maintained by the storage server S is shown in Figure 2 which is similar to that in Liu et al's work.[9] For example, "C_1(Original)" in Figure 2 is the uploader who originally uploaded the ciphertext of the file F_1, which is the only encrypted copy of F_1 stored in the cloud storage. "T_0 address" and "T_1 address" in Figure 2 are the addresses of the two smart contracts programmed by "C_1(Original)." Our protocol should ensure that if $F = F_i$, the encryption key K_{F_i} can be securely transferred to C, where F_i is encrypted by another client C_j using a semantic-secure symmetric encryption scheme under the key K_{F_i}.

Step 2. According to the short hash value sh, S finds the original uploaders of the files in the set of $F_{sh} = \{F_1, F_2, \ldots, F_i, \ldots\}$ which sh is associated with. If $F_{sh} = \emptyset$, C is the original uploader of the data file F and goes to Step 4. Otherwise, it sends to C the set of all the addresses of the smart contracts programmed by these original uploaders denoted as $SC = \{(T_0^1, T_1^1), \ldots, (T_0^j, T_1^j), \ldots\}$, and goes to Step 3.

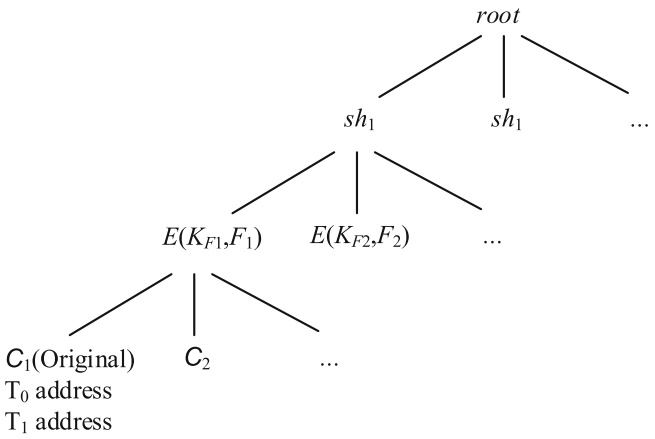


FIGURE 2 The record structure of server S

Step 3. For each $(T_0^j, T_1^j) \in SC$, C does the following operations:

C executes the smart contract T_1 deployed in T_1^j. Smart contract T_1 implements the following operations:

1. T_1 generates challenges *Chal* according to its current state τ as follows:
   - T_1 fixes an integer c.
   - T_1 constructs a set $I = \{s_1, \ldots, s_c\}$, where $s_i = pFunc(\tau||i)$, $i \in \{1, \ldots, c\}$.
   - T_1 generates the set of coefficients $\{v_1, \ldots, v_c\}$, where $v_i = hFunc(\tau||s_i)$, $s_i \in I$. Then, $Chal = \{(s_i, v_i)\}_{s_i \in I}$.
2. T_1 selects corresponding authenticators from Φ according to I. Let Φ_I be the set of corresponding authenticators.
3. T_1 outputs $Chal, \Phi_I$.

C generates data integrity proof P with the help of its own data file F by executing **Audit**.ProofGen$(PK, Chal, F, \Phi_I) \rightarrow (P)$. Let (pk_C, sk_C) be the public/private key pair generated by **PKE**.KeyGen$(1^\lambda) \rightarrow (pk_C, sk_C)$. C executes the smart contract T_0 deployed in T_0^j, and sends the data integrity proof P and pk_C to T_0 as its input.

Smart contract T_0 implements data integrity verification by executing **Audit**.ProofVerify$(PK, Chal, P)$. If the verification is passed which means that C actually preserves the data file F, T_0 generates the ciphertext $C_{K_{F_j}}$ of the encryption key K_{F_j} using **PKE**.Enc(pk_C, K_{F_j}), then sends $C_{K_{F_j}}$ to C.

C decrypts $C_{K_{F_j}}$ under its own private key to obtain K_{F_j}, that is, $K_{F_j} =$ **PKE**.Dec$(sk_C, C_{K_{F_j}})$. Go to Step 5.

If for all $(T_0^j, T_1^j) \in SC$, no integrity verification is passed, go to Step 4.

Step 4. In this step, it means that there doesn't exist a file satisfying that $F_i = F$ and $F_i \in F_{sh}$. C will be the original uploader of the data file F and does the following operations:

C executes **Audit**. SysSetup$(1^\lambda) \rightarrow (PK_C, SK_C)$.

C generates the set of authenticators Φ for F by executing **Audit**.AuthGen$(PK, SK, F) \rightarrow (\Phi)$.

C selects a random data encryption key K_F for F, then generates F's ciphertext $E(K_F, F)$.

C writes K_F and the code in Figure 3 into a smart contract T_0.

C writes Φ, PK_C and the code in Figure 4 into a smart contract T_1.

C uploads $E(K_F, F)$ and the addresses of smart contracts T_0 and T_1 to the cloud storage server S. Go to Step 5.

Step 5. The uploading process is finished.

Security proof. We will show that the execution of the deduplication protocol in the real model is computationally indistinguishable from the execution of F_{dedup} in the ideal model[9] as shown in Figure 5. We construct a simulator which can not only access F_{dedup} in the ideal model but also obtain messages that the corrupt parties would send in the real model. The simulator will generate a message transcript of the ideal model execution (IDEAL) which is computationally indistinguishable from that of the real model execution (REAL). In the proof, we assume that the smart contract T_0 and T_1 is implemented as an oracle to which the parties send their inputs and receive their outputs.

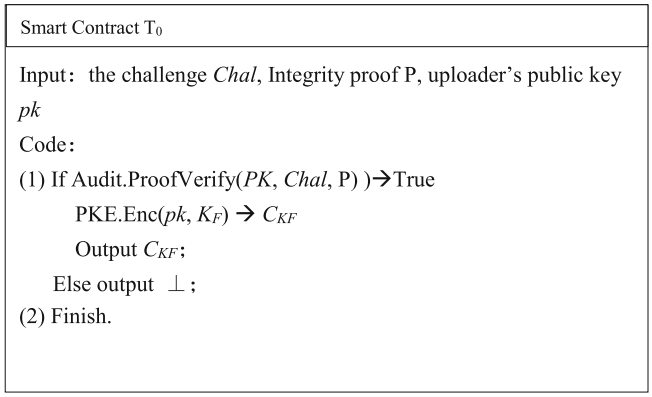


FIGURE 3 The smart contract T_0

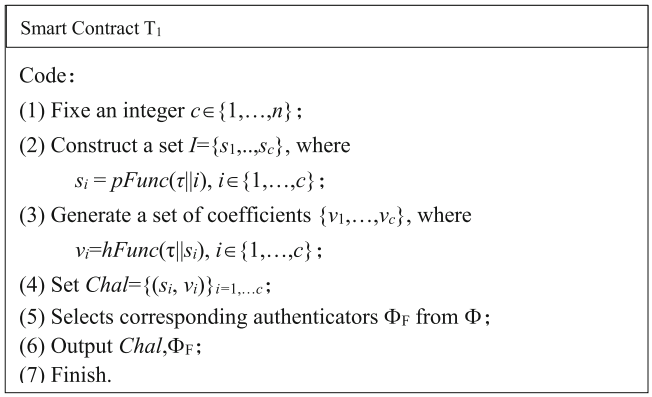


FIGURE 4 The smart contract T_1

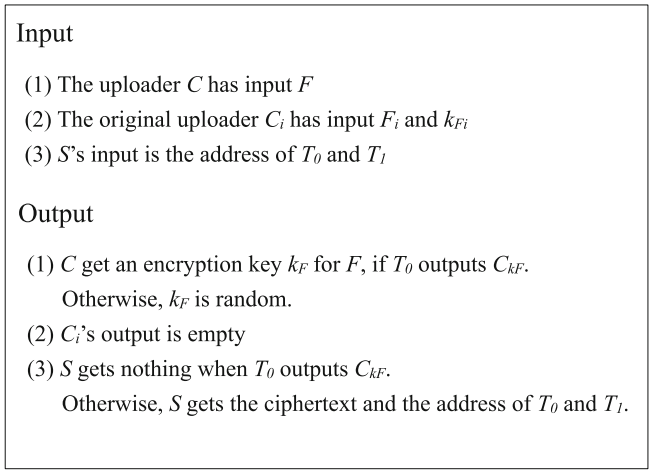


FIGURE 5 The ideal functionality F_{dedup} of deduplicating encrypted data

A corrupt subsequent uploader C: We first assume that the cloud storage server S and the original uploader C_i are honest and construct a simulator for uploader C. On receiving sh from C, the simulator observes the calls that C makes to the smart contracts T_0 and T_1. The simulator also records the output set $\{(P, F, sh)\}$. If C uses in that call the value P that appears in a set together with sh, the simulator invokes F_{dedup} with the file F appearing in that tuple. Otherwise, it invokes F_{dedup} with a random value. In either case, the simulator will obtain a key K_F(If F has been uploaded by any C_j, K_F is the key k_{F_j} for that file, otherwise K_F is a random value selected by C).

Suppose C is the subsequent uploader, that is, the file F already exists in the server. The simulator also records the output set $\left\{\Phi'_j\right\}$ that C receives from smart contract T_1. C executes **Audit**.ProofGen(PK, $Chal$, F, Φ) with the help of its own file F to get the proof $\{P_j\}$ and public key encryption **PKE**.KeyGen$\left(1^\lambda\right)$ to get (pk_C, sk_C). On receiving the P and pk_C from C, it chooses to check if the smart contract T_0^i's final result is C_{K_F}. If so, it calculates $e' = Enc\left(pk_c, C_{K_F}\right)$ and sends it back to C. Otherwise, it randomly selects an encryption key k, then sends it to the uploader C.

We now show that $IDEAL_C = \langle\left\{\Phi'_j\right\}, e'\rangle$ and $REAL_C = \langle\left\{\Phi'_j\right\}, e\rangle$ are identically distributed. (1) If the smart contract T_0's result is true and C behaves honestly, then $K_F = K_{F_j}$ and consequently e' and e are computationally indistinguishable. (2) If T_0's result is false, then K_F is a random value and the structure of e and e' are similar. (3) If C deviates from the protocol then the only action it can take, except for changing its input, is to replace some elements of $\{P_i\}$ that it sends to T_0. In the real model the execution will change, because the P_i is replaced by C. As a result, C will get a random value even though it inputs an existing file. In the ideal model, the same result will be caused by the event that a replaced element is chosen by the simulator. Based on (1) to (3), we can conclude that $IDEAL_c$ and $REAL_c$ are identically distributed.

A corrupt original uploader C_i: We prove security with relation to the relaxed functionality, where C_i also learns whether the uploaded file has the same short hash as F_i. The simulator needs to extract C_i's input:$\left(K_{F_i}, \Phi_i, address\right)$.

The simulator first observes whether sh matches the short hash of C_i. If not, it randomly chooses K_F and sends it to the C. C uses the key K_F to encrypt file F, getting ciphertext E. C writes K_F and **Audit**.ProofVerify(PK, $Chal$, P) to smart contract T_0, Φ and PK_C to T_1, $E\left(K_F, F\right)$ and the addresses of and T_0 and T_1 to the server S. The corrupt C_i may send the wrong Φ to smart contract T_1. As a result, when the subsequent uploader executes **Audit** to prove the integrity of his own F, the result of T_0 is still false. If so, the subsequent uploader will randomly choose the encryption key.

To show that the simulation is accurate, we observe that (1) if C_i behaves honestly, then $IDEAL_{C_i}$ and $REAL_{C_i}$ are obviously indistinguishable; (2) if C_i deviates from the protocol, the only operation it can do is to send wrong values to T_1, as a result, the proof P generated in the subsequent process is also wrong in both the real and ideal model. If P is wrong, T_0's final result must be false and K_F is assigned a random value. Based on (1) and (2), we can conclude that $IDEAL_{C_i}$ and $REAL_{C_i}$ are identically distributed.

A corrupt server S: The simulator first sends a short hash to the server S. Then S selects and sends to the uploader C a set of addresses. The set contains the addresses of the smart contracts T_0 and T_1 deployed by the original uploaders whose uploaded files are with the same short hash values. After executing the smart contracts, if the result of T_0 is true, the ciphertext C_{K_F} of encryption key K_F is sent to C, otherwise, an error is sent. The corrupt S may send to C a set of random values $SC = \left\{\left(T_0^1, T_1^1\right), \ldots, \left(T_0^j, T_1^j\right), \ldots\right\}$ and observes the outputs.

Next, the simulator invokes F_{dedup}. It detects the result of T_0, if it receives $E\left(pk_C, K_F\right)$ it sets $b = 1$; otherwise, it sets $b = 0$. If $b = 1$, it means T_0 passes **Audit**.ProofVerify(PK, $Chal$, P) and passes the result C_{K_F} to C, and the simulator receives $T_0's$ output C_{K_F}. If $b = 0$, the simulator receives $Enc(k_F, F)$, and the addresses of T_0 and T_1.

We now show that $IDEAL_S = \langle sh, \left\{\left(T_0^i, T_1^i\right), C_{K_{F_i}}, F\right\}\rangle$ and $REAL_S = \langle sh, \left\{\left(T_0^i, T_1^i\right), C'_{K_{F_i}}, E\left(K_F, F\right)\right\}\rangle$ are identically distributed. (1) Because $\left\{C_{K_{F_i}}, C'_{K_{F_i}}\right\}$ is generated by the smart contract T_0, $C_{K_{F_i}}$ cannot be distinguished from $C'_{K_{F_i}}$ in both ideal and real models. (2) If F exists and S behaves honestly, it will get the same $C_{K_{F_i}}$ in the real model and the ideal model. (3) If F does not exist in S, then K_F is the chosen random value in the real model, so S cannot distinguish $Enc(K_F, F)$ from a random string. Also, in the ideal model, the simulator gets from F_{dedup} the encryption of F under a random key. (4) Since S does not have the private key sk_C of an uploader C, it is impossible for S to decrypt the output C_{K_F} of T_0 in both of the ideal and the real models. Based on (1)-(4) we can conclude that $IDEAL_S$ and $REAL_S$ are identically distributed.

A collusion between a corrupt uploader and a corrupt server: The simulator acts to be C and S to call F_{dedup} inputting C's file F and public key PK_c. Here, S has no input to F_{dedup}. Then, the simulator receives the outputs of C and S. This situation is similar to the situation where the uploader is corrupted, except that S can choose a series of $\left\{\left(T_0^i, T_1^i\right)\right\}$ to implement the cloud data integrity audit protocol.

4 | RESULTS AND ANALYSIS

In this section, we analyze the performance of the protocol. The smart contracts on the Hyperledger Fabric are written in the Go language, and they are packaged, installed, and invoked in the test environment to implement our protocol. Figure 6 summarizes the basic information of our experiments.

4.1 | Experiment deployment

In this subsection, we describe how to implement the experiments. In our experiments, we use the auditing protocol proposed by Shacham and Waters[34] as it is a very simple and efficient protocol which is suitable to be used in the smart contract.

- First, we divide the file into N blocks, each of which is M kB in size.
- Randomly select a 256-bit prime number p to perform modulo operation with each block, and finally get an integer set $\{m_i\}_{i\in[1,N]}$ within 256 bits.
- Calculate file block corresponding authenticator $\{\Phi_i\}_{i\in[1,N]}$, where $\Phi_i = f_k(i) + \alpha m_i \in \mathbb{Z}_p$, $\alpha \in Z_p$. The function $f()$ is HMAC-SHA256, and k is its random key.
- Randomly select S blocks from $\{m_i\}_{i\in[1,N]}$. In the smart contract T_1, the corresponding $Chal(Q) = \{(s_i, v_i)\}_{i\in[1,S]}$ is calculated using the pseudorandom function *pFunc* and the hash function *hFunc*.
- The client generates the proof $P = (\Phi, \mu)$, where $\Phi \leftarrow \sum_{(s_i,v_i)\in Q} v_i \cdot \Phi_i$ and $\mu \leftarrow \sum_{(s_i,v_i)\in Q} v_i \cdot m_i$.
- Pass P and the random key k to the smart contract for verification, the verification formula is $\Phi \stackrel{?}{=} \alpha \cdot \mu + \sum_{(s_i,v_i)\in Q} v_i \cdot f_k(i)$. The final verification result is true or false.
- When the result is true, we encrypt the data encryption key K_F in the smart contract T_0 using RSA public-key encryption algorithm under the public key given by the uploader.

We implemented three experiments.

1. In the first experiment, we studied the duration of the authenticator and proof generation when files of different sizes were uploaded. We selected a total of 10 groups of file data from 10M to 100M and fixed the size of each file block to 100k, and the number of selected checked data blocks was 100. The results are shown in Figure 7.
2. In the second experiment, we evaluated the duration of authenticator and proof generation when dividing files into blocks of different sizes. Let the file size be 100M and the number of selected checked data blocks be 100. We set the block size from 100k to 1000k, that is, the file is divided into 1000 down to 100 blocks. The results are shown in Figure 8.
3. In addition, we tested the duration of challenge generation in the both of the above experiments.
4. In the last experiment, we evaluated the duration of proof verification of different numbers of checked data blocks in smart contracts, wherein we fixed the file size to 100M, the block size to 100k, and selected 100 to 1000 blocks.

CPU Series	Intel(R) Core(TM) i7-8700
RAM	16.0 GB (15.8 GB is available)
Operate System	Windows 10
	Ubuntu 18.04.2
Compiler Version	Microsoft Visual Studio 2019
	Fabric version 2.2.1
Environment	Go version: Go 1.14.4 linux/amd64
	Docker version 20.10.2
	Docker-compose version 1.17.1

FIGURE 6 Basic information

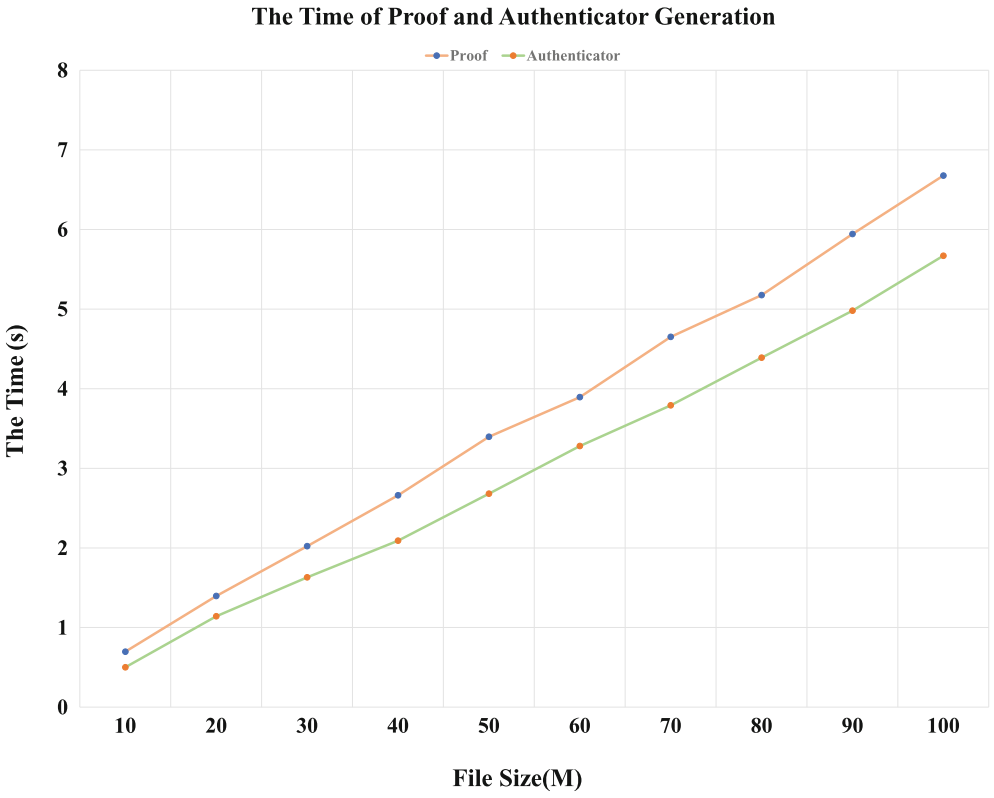


FIGURE 7 The time of proof and authenticator generation with different size of files

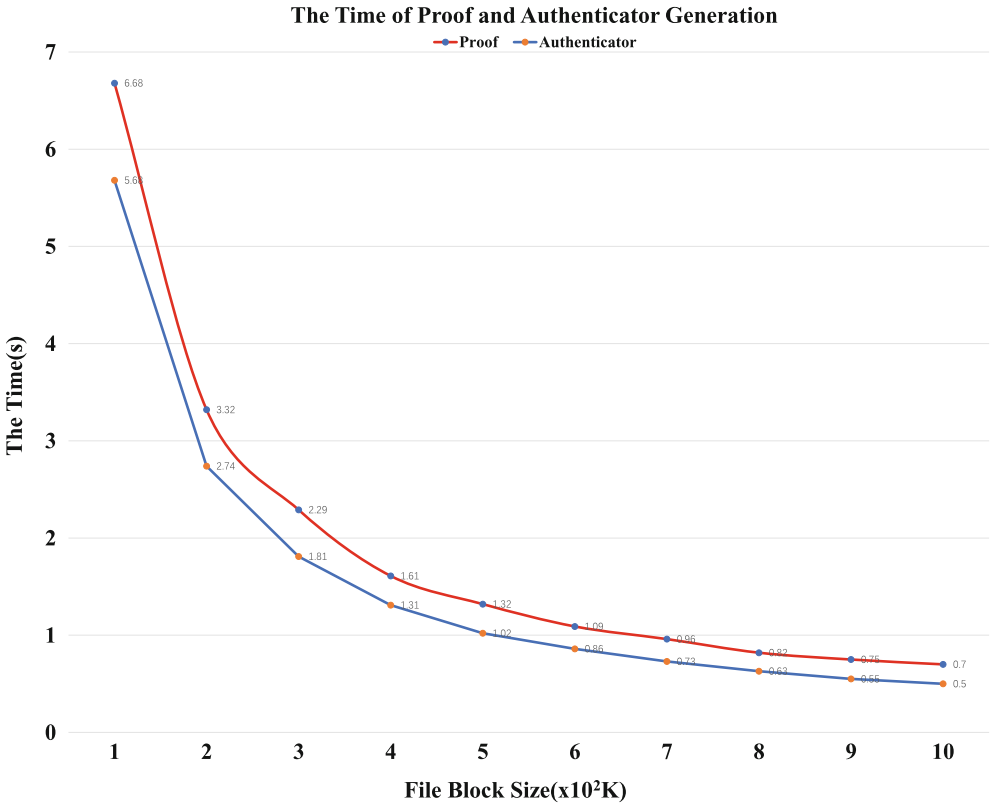


FIGURE 8 The time of proof and authenticator generation with different size of file blocks

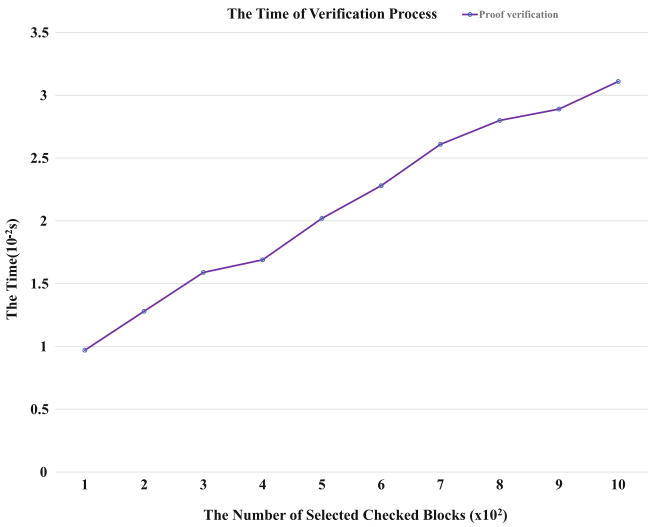


FIGURE 9 The time of verification process with different number of selected checked blocks

4.2 | Performance analysis

The duration of authenticator generation is the sum of the time to calculate corresponding authenticator for each block, and the duration of proof generation is the sum of the time to calculate Φ and μ.

From Figure 7, we can demonstrate that the durations of the authenticator and proof generation are linearly related to the size of the uploaded file.

The most interesting aspect of Figure 8 is that the duration of the authenticator and proof generation is negatively correlated with the block size. Through the analysis of the experimental data, it can be seen that the larger the file block and the fewer the number of blocks, the less time to generate the authenticator and the less time it eventually consumes. In addition, the fewer the number of blocks, the less time of the proof generation.

From the description of the proposed protocol, it can be obtained that the duration of the challenge generation is linear with the number of selected checked data blocks rather than the file size or the block size. By the implementation, we can get that an element $(s_i, v_i) \in Chal(Q)$ can be generated in the smart contract in about 15 milliseconds.

In the last experiment, we added a timestamp to the chain code to record the start time of the contract call and calculated the duration of the verification process according to the end time of the contract call and the time stamp. The experimental results are shown in Figure 9.

We also analyze the running time overhead of the PKE process in smart contract T_0 which takes about 26 milliseconds when using the RSA public-key encryption algorithm under a 1024-bit length key.

Finally, we make a comparison in terms of efficiency between our proposed protocol and the current deduplication protocols that support client-side semantically secure encryption and require the trusted third party or the previous uploader to be online. We focus on the scenario that the data to be uploaded is duplicated, and we compare the time required by each protocol when uploading a duplicated data of size 100 MB, that is, the time of obtaining the data encryption key. For the protocols requiring the trusted third party (key server),[12,13] the time of obtaining the data encryption key is independent of the data size. Using the benchmarks of the protocols[12,13] as references, to obtain the data encryption key, the protocol proposed in Bellare et al's work[12] needs about 82 milliseconds and the protocol proposed in Duan's work[13] needs about 500 milliseconds for the worst case due to the use of distributed key servers. For our solution, it relies on the original uploader and the smart contracts. The original uploader generates the authenticator when uploading the file, and the following uploader calculates the proof P and submits it to the smart contract for verification. According to the performance analysis, when the file size is 100 MB, the block size is 1000 KkB, and the number of selected check blocks is 100, the uploading process costs about extra 2.7 seconds besides the normal uploading process. Note that the original uploader needs to generate the authenticator Φ and deploy two smart contracts which costs about more than 0.5 seconds. Table 1 shows the differences between our protocol and the protocols in the compared works.[12,13] Therefore, we

TABLE 1 The comparison

Protocol	No third trusted party	Key request time (ms)
12	×	82
13	×	500
Our	✓	2700

can conclude that our protocol neither requires any trusted third party nor needs other uploaders to be online, and moreover it has reasonable running time overhead and high practicability, although we have to admit that our protocol is less efficient than the protocols require the trusted third party.[12,13]

5 | DISCUSSION

Privacy protection of the data encryption key K_F in the smart contract. Privacy protection has attracted more and more attention in various fields, such as machine learning,[35,36] data storage,[37] blockchain,[33] social networks,[38] etc. As is known to all, Hyperledger Fabric is a permissioned blockchain architecture, which provides a consistently distributed ledger, shared by a set of peers. The core principle of Hyperledger Fabric is that all the peers have the same view of the shared ledger, which makes it a challenge to support private data for the different peers.[39] It can be seen from Section 3 that the private data K_F is placed in the smart contract T_0. In this scenario, how can we ensure data privacy in the smart contract? There are many existing solutions.

Privacy protection can be achieved by specifying the privacy dataset. It is pointed out that for the implementation of the privacy database, if some organizations on the channel want to protect the data privacy and keep it secret to other organizations on the channel, a new channel can be created to allow only organizations with access rights to private data to join.[40] However, this method will incur additional management overhead. Zkay[33] is a system for specifying and executing data privacy in smart contracts. To implement zkay contract, it is proposed to automatically convert it into the contract that can be deployed on the public blockchain and is equivalent in terms of privacy and function. The implementation of the original prototype of zkay proves the feasibility of the method, but its proof of concept is seriously restricted, such as insecure encryption and lack of important language features. Later, the zkay v0.2[41] made up for the deficiency of zkay, which can support the most advanced asymmetric encryption and hybrid encryption. It introduces many new language features (such as function call, private control flow, and extension type support), allows different zk-SNARKs backends, and reduces compilation time and on-chain cost.

For Hyperledger Fabric, it can achieve data isolation by introducing multiple channels and private data collections. The multi-channel method separates the information between different channels. In theory, all nodes in the same channel share the data, but the data access permissions can be controlled by policies, which define the fabric system operation and data access permissions.[40]

From the above discussion, it can be seen that the existing blockchain privacy protection technologies can fully realize the privacy protection for the data encryption key K_F in the smart contract. The specific technologies will not be discussed in detail here.

6 | CONCLUSION

In this article, we mainly focus on how to use smart contracts to complete integrity auditing work, thereby realizing secure encrypted data deduplication. We deploy the proof verification process of the integrity auditing protocol in the smart contract and make the client generate the integrity proof of the data to be uploaded and send it to the smart contract for checking. Based on the characteristics of the blockchain, such as decentralization, non-tampering, traceability, multi-party maintenance, openness, and transparency,[42] it realizes the data ownership proof relying on the smart contract, which in turn enables encrypted data deduplication. Our protocol avoids the limitations of the existing encrypted data deduplication protocols which need a trusted third party or require that some previous uploaders be online to run the PAKE protocol. In addition, the run-time overhead of our protocol is reasonable, and it is much more practical.

It should be pointed out that the privacy-preserving smart contract plays an important role in our proposed protocol for the protection of data encryption key, which has not been implemented in our protocol. We leave it a future work.

ACKNOWLEDGMENTS

This research was supported by Shandong Provincial Natural Science Foundation (Nos. ZR2020LZH002, ZR2020MF055, ZR2021LZH007, ZR2020QF045), The Open Project of Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences (No. KFKT2019-002).

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. Guarini D. Experts say Facebook leak of 6 million users'data might be bigger than we thought; 2013.
2. iCloud leaks of celebrity photos; 2014.
3. Recent major data leakage incidents at home and abroad; 2019.
4. Reinsel D, Gantz J, Rydning J. The digitization of the world from edge to core - data age 2025; 2018.
5. Jiang T, Chen X, Wu Q, Ma J, Susilo W, Lou W. Secure and efficient cloud data deduplication with randomized tag. *IEEE Trans Inf Forens Secur*. 2017;12(3):532-543.
6. Xu LJ, Hao R, Yu J, Vijayakumar P. Secure deduplication for big data with efficient dynamic ownership updates. *Comput Electr Eng*. 2021;96:107531. doi:10.1016/j.compeleceng.2021.107531
7. Douceur JR, Adya A, Bolosky WJ, Simon P, Theimer MM. Reclaiming space from duplicate files in a serverless distributed file system. ICDCS'02; 2002:617-624; IEEE, Vienna, Austria.
8. Bellare M, Keelveedhi S, Ristenpart T. Message-locked encryption and secure deduplication. *Advances in Cryptology – EUROCRYPT'15*. Athens, Greece: Springer; 2013:296-312.
9. Liu J, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional independent servers. *CCS'15*. Denver, Colorado: ACM; 2015:874-885.
10. Abadi M, Boneh D, Mironov I, Raghunathan A, Segev G. Message-locked encryption for lock-dependent messages. *Advances in Cryptology – CRYPTO'13*. Santa Barbara, CA: Springer; 2013:374-391.
11. Bellare M, Keelveedhi S. Interactive message-locked encryption and secure deduplication. *PKC'15*. Gaithersburg, MD: Springer; 2015:516-538.
12. Bellare M, Keelveedhi S, Ristenpart T. DupLESS: server-aided encryption for deduplicated storage. *Sec'13*. Washington, DC: USENIX Association; 2013:179-194.
13. Duan Y. Distributed key generation for encrypted deduplication: achieving the strongest privacy. *CCSW'14*. Scottsdale, Arizona: ACM; 2014:57-68.
14. Puzio P, Molva R, Önen M, Loureiro S. ClouDedup: secure deduplication with encrypted data for cloud storage. *CloudCom'13*. Bristol, UK: IEEE; 2013:363-370.
15. Zhang S, Xian H, Wang Y, Liu H, Hou R. Secure encrypted data deduplication method based on offline key distribution. *J Softw*. 2018;29(7):1909-1921.
16. Zhang S, Xian H, Wang L, Liu H. Secure cloud encrypted data deduplication method. *J Softw*. 2019;30(12):3815-3828.
17. Xian H, Liu H, Zhang S, Hou R. Verifiable secure data deduplication method in cloud storage. *J Softw*. 2020;31(2):455-470.
18. Liu J, Duan L, Li Y, Asokan N. Secure deduplication of encrypted data: refined model and new constructions. In: 393, ed. *CT-RSA'18*. San Francisco, CA: Springer; 2018:374.
19. Chen H, Yu J, Zhou H, Zhou T, Liu F, Cai Z. SmartStore: a blockchain and clustering based intelligent edge storage system with fairness and resilience. *Int J Intell Syst*. 2021;36(9):5184-5209.
20. Sarkar A, Maitra T, Neogy S. *Blockchain in Healthcare System: Security Issues, Attacks and Challenges*. Cham: Springer International Publishing; 2021:113-133.
21. Guo J, Wang Y, An H, Liu M, Zhang Y, Li C. IIDQN: an incentive improved DQN algorithm in EBSN recommender system. *Sec Commun Netw*. 2022;2022:1-12. doi:10.1155/2021/7502248
22. Hu C, Xu Y, Liu P, Yu J, Guo S, Zhao M. Enabling cloud storage auditing with key-exposure resilience under continual key-leakage. *Inf Sci*. 2020;520:15-30. doi:10.1016/j.ins.2020.02.010
23. Naor M, Yung M. Public-key cryptosystems provably secure against chosen ciphertext attacks. *STOC'90*. Baltimore, Maryland: ACM; 1990:427-437.
24. Zhang D, Le J, Lei X, Xiang T, Liao X. Exploring the redaction mechanisms of mutable blockchains: a comprehensive survey. *Int J Intell Syst*. 2021;36(9):5051-5084.

25. Singh S, Hosen AS, Yoon B. Blockchain security attacks, challenges, and solutions for the future distributed IoT network. *IEEE Access*. 2021;9:13938-13959. doi:10.1109/ACCESS.2021.3051602
26. Wang Y, Wang Z, Zhao M, et al. BSMEther: bribery selfish mining in blockchain-based healthcare systems. *Inf Sci*. 2022;601:1-17. doi:10.1016/j.ins.2022.04.008
27. Li T, Wang Z, Chen Y, Li C, Jia Y, Yang Y. Is semi-selfish mining available without being detected? *Int J Intell Syst*. 2021:22. doi:10.1002/int.22656
28. Li T, Chen Y, Wang Y, et al. Rational protocols and attacks in blockchain system. *Sec Commun Netw*. 2020;2020(44):1-11.
29. Wang W, Xu H, Alazab M, Gadekallu TR, Han Z, Su C. Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Trans Ind Inform*. 2022;18(10):7059-7067. doi:10.1109/TII.2021.3084753
30. Deebak BD, Memon FH, Khowaja SA, et al. Lightweight blockchain based remote mutual authentication for AI-empowered IoT sustainable computing systems. *IEEE Internet Things J*. 2022;1:9. doi:10.1109/JIOT.2022.3152546
31. Szabo N. Formalizing and securing relationships on public networks. *First Monday*. 1997;2(9):1-27.
32. Buterin V. A next generation smart contract and decentralized application platform. Ethereum white paper; 2014.
33. Steffen S, Bichsel B, Gersbach M, Melchior N, Tsankov P, Vechev M. Zkay: specifying and enforcing data privacy in smart contracts. *CCS'19*. New York, NY: Association for Computing Machinery; 2019:759-1776.
34. Shacham H, Waters B. Compact proofs of retrievability. *J Cryptol*. 2013;26:442-483.
35. Zhu T, Zhou W, Ye D, Cheng Z, Li J. Resource allocation in IoT edge computing via concurrent federated reinforcement learning. *IEEE Internet Things J*. 2022;9(2):1414-1426. doi:10.1109/JIOT.2021.3086910
36. Shokri R, Shmatikov V. Privacy-preserving deep learning. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security; 2015:1310-1321.
37. Gao X, Yu J, Chang Y, Wang H, Fan J. Checking only when it is necessary: enabling integrity auditing based on the keyword with sensitive information privacy for encrypted cloud data. *IEEE Trans Depend Sec Comput*. 2022;19(6):3774-3789. doi:10.1109/TDSC.2021.3106780
38. Li J, Hu X, Xiong P, Zhou W. The dynamic privacy-preserving mechanisms for online dynamic social networks. *IEEE Trans Knowl Data Eng*. 2022;34(6):2962-2974. doi:10.1109/TKDE.2020.3015835
39. Benhamouda F, Halevi S, Halevi T. Supporting private data on hyperledger fabric with secure multiparty computation. *IBM J Res Develop*. 2019;63(2/3):3:1-3:8. doi:10.1147/JRD.2019.2913621
40. Ma C, Kong X, Lan Q, Zhou Z. The privacy protection mechanism of hyperledger fabric and its application in supply chain finance. *Cybersecurity*. 2019;2(1):1-9. doi:10.1186/s42400-019-0022-2
41. Baumann N, Steffen S, Bichsel B, Tsankov P, Vechev M. zkay v0.2: practical data privacy for smart contracts. Technical report; 2020.
42. Guo S, Wang R, Zhang F. Overview of the principles and applications of blockchain technology. *Comput Sci*. 2021;48:271-281.

How to cite this article: Qin G, Li L, Liu P, Hu C, Guo S. Blockchain-based secure deduplication of encrypted data supporting client-side semantically secure encryption without trusted third party. *Trans Emerging Tel Tech*. 2024;35(4):e4712. doi: 10.1002/ett.4712

Information Sciences 520 (2020) 15–30

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Enabling cloud storage auditing with key-exposure resilience under continual key-leakage

Chengyu Hu[a,b,c,*], Yuqin Xu[d], Pengtao Liu[e], Jia Yu[f], Shanqing Guo[a,b], Minghao Zhao[g]

[a] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, 250100, China
[b] School of Cyber Science and Technology, Shandong University, Qingdao, 266237, China
[c] Key laboratory of Network Assessment Technology, CAS (Institute of Information Engineering, Chinese Academy of Sciences), Beijing, 100093, China
[d] School of Software, Shandong University, Jinan, 250101, China
[e] College of Cyberspace Security, Shandong University of Political Science and Law, Jinan, 250014, China
[f] College of Computer Science and Technology, Qingdao University, Qingdao, 266071, China
[g] School of Software, Tsinghua University, Beijing, China

ARTICLE INFO

Article history:
Received 7 April 2019
Revised 2 February 2020
Accepted 5 February 2020
Available online 5 February 2020

Keywords:
Data storage
Cloud storage auditing
Continual key-leakage resilience
Forward security

ABSTRACT

Cloud storage auditing is a service that is usually provided to enable clients to verify the integrity of their data stored in the cloud. However, clients risk exposing their secret key. To address the problem of key exposure, researchers have provided "Forward Security" by dividing the entire lifetime of the secret key into several periods and updating the secret key within each of these periods. Forward security can ensure the validity of authenticators before the period in which the secret key is fully exposed. However, the security of these protocols can be broken by launching side-channel attacks to leak the secret key partially rather than fully. In this study, we focus on implementing measures in cloud storage auditing to protect against side-channel attacks in practice. We formalize the definition and security model of a cloud storage auditing protocol, which supports forward security under continual key-leakage, and construct the first protocol. Our protocol remains secure even if an adversary obtains partial leakage of the secret key during a period. In addition, if the secret key were to be fully disclosed in a certain period, our protocol would maintain forward security. Therefore, the proposed protocol provides stronger security compared with existing protocols.

1. Introduction

Cloud storage is an emerging technology that provides clients with convenient data-related services. Recently, many world-leading IT companies have released cloud storage products, such as Google Cloud Storage, Microsoft Azure, and Amazon S3. Clients who utilize these cloud storage services rent the storage capacity and network bandwidth in a pay-as-you-go manner. Accordingly, they can outsource their data to the cloud and access the data anytime, anywhere through the internet, and enjoy other storage services based in the cloud (e.g., data analysis or image processing) if required. This obviates

* Corresponding author at: School of Cyber Science and Technology, Shandong University, Qingdao, 266237, China.
E-mail address: hcy@sdu.edu.cn (C. Hu).

https://doi.org/10.1016/j.ins.2020.02.010
0020-0255/

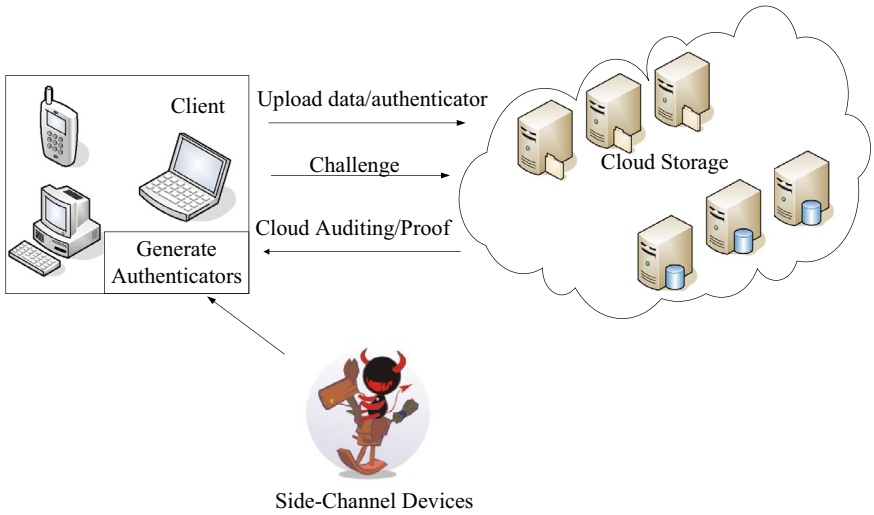


Fig. 1. System model of our cloud storage auditing.

the need for clients to maintain basic storage infrastructure, and the storage service provider can concentrate on the quality of the service themselves. Many individuals and institutions have adopted cloud storage to maintain their data. Since their inception, cloud storage services have become a lucrative industry, with the global cloud storage market estimated to reach $65.41 billion by the year 2020 [1].

Despite the numerous advantages of using a cloud storage service, data integrity has always been a significant problem that has prevented prospective clients from adopting this service. When users upload their data to the cloud, they lose complete control of their data and rely entirely on the cloud to maintain them. Although cloud service providers adopt a variety of advanced techniques (eg. replication [2] or erasure code [3]) to ensure data reliability and robustness, data corruption still frequently occurs [4]. In addition, a dishonest cloud server may conceal the incident of data loss to the users, or even maliciously delete users' data. Accordingly, from a user's perspective, the service provider should convince the user that the data they saved in the cloud will remain intact.

Unfortunately, unlike traditional settings in which hash functions and signatures can be utilized for integrity insurance, in a cloud storage scenario, the clients seldom retain a local copy of their data. In addition, it would be unrealistic to require the clients to download the entire dataset. Thus, it is necessary to compose an appropriate integrity auditing mechanism in a cloud storage scenario that can remotely verify the intactness of the data without reliance on local copies [5,6]. In this regard, cloud storage auditing protocols are cryptographic protocols that can efficiently and effectively prove the intactness of the data stored in the cloud. They normally adopt a spot-checking technique and thus the auditors are only required to access a fraction of the data to verify the integrity of the entire dataset. Consequently, cloud storage auditing has become a tool of significant importance for cloud data security.

Most of the cloud auditing protocols that have been developed to date assume that the client's secret key for auditing is securely maintained. However, in practice, newly emerging side-channel attacks may invalidate this assumption. Traditional techniques that were used to launch side-channel attacks such as a power analysis attack [7,8], timing attack [9], and electromagnetic analysis [10] were expensive to carry out and sometimes led to observable physical damage to the affected device. Modern side-channel attacks (eg. [11–13]) can grab users' secret key inexpensively and imperceptibly. For example, as shown by Genkin [14], who jointly analyzed different traces (e.g., the far end of a cable, human touch, electromagnetic, and power consumption), it is feasible to extract 4096-bit RSA keys and 3072-bit ElGamal keys from laptops with little effort. Once the secret key for auditing is leaked to the cloud service provider, all the present cloud auditing protocols would fail.

Schemes concentrating on secret key leakage in cloud auditing have been proposed [15–17], with all of them addressing the problem of cloud auditing as a result of *key disclosure*. These schemes consider the client's key to be fully leaked rather than partially leaked in a side-channel attack. For example, the client may inadvertently and carelessly download malware that reads the client's key and sends it to the attacker. In these previous studies, the entire lifetime of the secret key was separated into several time periods and *forward security* for a cloud auditing protocol was provided by updating the secret keys among the periods. As a result, these auditing protocols still remain secure in those periods that occur before the secret key is fully exposed. In practice, however, the adversary can obtain pieces of information about the secret key between two updates by launching a side-channel attack, which can obviously help it breach the security of the auditing protocol.

In this study, we focus on enabling leakage-tolerant cloud storage auditing to overcome the problem of partial key leakage between two key updates in the forward-secure cloud auditing protocols. Specifically, our proposed cloud storage auditing method achieves both "forward security" and "key-leakage resilience" simultaneously. Fig. 1 shows the scenario on which our work is based. The two participants are: the client (file owner) and the cloud. The client partitions each of his

files to blocks and uploads the blocks and the corresponding authenticators to the cloud. The client can use a service based in the cloud to verify whether their files are correctly stored in the cloud. An adversary can obtain partial information about the client's secret key by using side-channel attacks.

In this regard, leakage-resilience has attracted considerable attention in theoretical cryptography as an algorithmic countermeasure (contrary to engineering countermeasures such as hiding [18] and masking [19]) against side-channel attacks. In leakage-resilient cryptography, leakage models are generalized to capture the features of multiple types of side-channel attacks. Among these models, the *continual memory leakage model* is generally considered to be the most powerful model, which assumes the secret key in the memory can be (partially) acquired by the adversary.

1.1. Contribution

To make the auditing protocol support both of "forward security" and "key-leakage resilience" simultaneously, we first propose an auditing protocol with continual key-leakage resilience. Then, we extend the scheme to achieve our goal. The main contributions are as follows:

1. First, we attempted to provide the storage auditing protocol with continual key-leakage resilience, a capability previous auditing protocols did not have. Our design enables malicious operations on the client's cloud data to be detected, even if the malicious cloud obtains partial information about the client's current secret key for cloud storage auditing. We define *continual key-leakage resilience* for the cloud auditing protocol and propose the first concrete protocol for cloud storage.
2. We developed a cloud storage auditing protocol to support "*forward security*" and "*continual key-leakage resilience*" simultaneously. This protocol makes it possible to detect malicious operations on the client's cloud data in previous time periods, even if the malicious cloud server were to obtain the client's current secret key for cloud storage auditing and partial information about the secret keys of previous time periods. Specifically, we employ a binary tree structure [20,21] to update the clients secret keys in different time periods. We apply an existing technique [20] to our continual key-leakage resilient auditing protocol and propose the first auditing protocol with the above-mentioned two security properties.

1.2. Related work

Data Auditing for cloud storage. Remote data integrity verification has its origins in integrity protection memory management systems [22], which enable a client to verify whether read/write operations are correctly executed in unreliable memory. With the proliferation of cloud storage, proof of retrievability (POR) [23] and proof of data possession (PDP) [5,24] were proposed to efficiently verify the integrity of archival datasets. Specifically, a POR scheme stores each encrypted file in the cloud server along with a set of pseudorandom blocks. Subsequently, the client can examine the data integrity by verifying whether the server retains the pseudorandom blocks. PDP follows a different approach by allowing the client to verify the integrity by challenging the server with some randomly selected block numbers to determine whether the server generates valid proofs.

Later, multiple PDP and POR schemes were proposed to extend the performance or functionality of traditional schemes. For example, dynamic PDP [6,25,26] enables the client's file archive to be dynamically updated (e.g., via file upload or delete). PDP or POR with public verifiability (e.g., [27–29]) enables a third party, rather than the client, to verify the data integrity. Other solutions (e.g.,[30–32]) took privacy into consideration and ensured that neither the cloud nor the auditor could acquire the user's data.

The aforementioned studies (including ours) adopted the *single-server* model, which regards the cloud storage platform as a whole entity. Accordingly, they only focus on integrity verification in the cloud but cannot recover the original data when an inconsistency is found. It is worth mentioning that another approach was to adopt the *multi-server* model with the aim of reconstructing the compromised data by using a redundancy (e.g., replication or coding) technique. For example, a replication technique was adopted for data-recovery [33], whereas the high-availability and integrity layer (HAIL) [2] utilizes erasure coding, and a third approach involved regenerating codes in recovering corrupted data [34,35].

Leakage-resilient cryptographic protocols for the cloud. Secure multiparty computation (SMPC) [36,37] is a generic cryptographic protocol that enables distributed parties to jointly compute a functionality, while ensuring that each party's input and output remains secret. Generally, SMPC first transforms the targeted functionality into arithmetic or logic circuits for subsequent evaluation in a secure manner. Theoretically, the goal of leakage resilience SMPC is to secure circuit evaluation against an adversary who probes the values of internal wires. Several researchers (e.g., [38–40]) conducted in-depth research in this field.

Likewise, secret sharing [41] is a kind of cryptographic protocol that enables a user to randomly split a secret into multiple shares, such that certain subsets of the shares can be used to reconstruct the secret and others do not reveal any particulars of the secret. Secret sharing is also a significant tool for constructing secure cloud applications [42]. The leakage resilience of secret sharing was formalized by the work of Benhamouda et al. [43], after which several leakage resilient secret sharing schemes were proposed [44,45]. In terms of application-level secure cryptographic schemes for cloud computing, Hu et al. [46] and Dai et al. [47] considered leakage resilience for searchable encryption [48] to enable secure search in the cloud.

Studies that are the most closely related to this one are [15,16], all of which focused on the problem of cloud auditing *under key disclosure*. However, as mentioned previously, these solutions only provide "forward security" and do not consider the problem of partial key leakage between two key-updates.

1.3. Organization

In Section 2, we introduce the necessary preliminaries. Then, in Section 3, we propose a concrete auditing protocol with continual key-leakage resilience and analyze its security and performance. In Section 4, we extend the protocol in Section 3 such that it supports "forward security" and "continual key-leakage resilience" simultaneously. Finally, we conclude the paper in Section 5.

2. Preliminaries

2.1. Composite order bilinear groups

Our protocols are constructed on the composite order bilinear groups of order N where $N = p_1p_2p_3p_4$ is a product of four distinct primes [49]. Let G, G_T be cyclic groups of order N. Let e: $G \times G \to G_T$ be a map satisfying the following properties:

1. Bilinearity: For all $u, v \in G$ and any $a, b \in Z_N$, $e(u^a, v^b) = e(u, v)^{ab}$;
2. Non-degeneracy: For all generators $g \in G$, $e(g, g) \neq 1_{G_T}$;
3. Computability: $e(u, v)$ can be computed efficiently for all $u, v \in G$;

Following the explanation in [49], the composite order bilinear groups have some properties. Let G_{abc} be the subgroup of order abc for $a, b, c \in \{1, p_1, p_2, p_3, p_4\}$. There is an isomorphism $G_{mn} \cong G_m \times G_n$ if gcd(m,n)=1. Let G_{p_i} be subgroups of order p_i. Any element of G is in the form of $g_{p_1}^{x_1} g_{p_2}^{x_2} g_{p_3}^{x_3} g_{p_4}^{x_4}$ and $\prod_{i\in S} g_{p_i}^{x_i}$ is an element of subgroup $G_{\prod_{i\in S} p_i}$, where $S \subseteq \{1, 2, 3, 4\}$, g_{p_i} is a generator of subgroup G_{p_i} and $x_i \in Z_{p_i}$. The orthogonal property guarantees that if u, v are group elements of different order, then $e(u, v) = 1_{G_T}$.

In [49], some assumptions were proposed which can be used to prove the security. The security of our protocol mainly relies on the following assumption. Let $I = (N = p_1p_2p_3p_4, G, G_T, e)$ be a random bilinear setting.

Assumption 1: Pick $g_1, U_1 \leftarrow G_{p_1}$, $U_2, V_2 \leftarrow G_{p_2}$, $V_3, g_3 \leftarrow G_{p_3}$, $g_4 \leftarrow G_{p_4}$, $C_1 \leftarrow G_{p_1p_2p_3}$, $C_2 \leftarrow G_{p_1p_3}$ and set $E = (I, g_1, g_3, g_4, U_1U_2, V_2V_3)$. The advantage of an algorithm $\mathcal{A}$ to break Assumption 1 is defined to be

$$\mathrm{Adv}^{\mathcal{A}}_{Asmp1}(\lambda) = |\Pr[\mathcal{A}(E, C_1) = 1] - \Pr[\mathcal{A}(E, C_2) = 1]|.$$

It is proved in [49] that the above assumption holds in the generic group model, i.e., for all PPT algorithms $\mathcal{A}$, $\mathrm{Adv}^{\mathcal{A}}_{Asmp1}(\lambda)$ is a negligible function in λ.

2.2. Signature

A signature scheme typically consists of three algorithms (KeyGen,Sig,Ver). To ensure it becomes continual key-leakage resilient, it should be equipped with an additional algorithm KeyUpdate as follows:

1. KeyGen(1^λ) $\to$ (vk, sk_0): The key generation algorithm takes as input a security parameter λ and outputs the public verification key/private signing key pair (vk, sk_0).
2. Sig(sk_i, m) $\to \sigma$: The signing algorithm takes as input a private signing key sk_i and a message m, and outputs a signature σ.
3. Ver(vk, m, σ) $\to$ 0/1: The verification algorithm takes as input the public verification key vk, a message m, and a signature σ. If the output is 1, σ is a valid signature of m; otherwise the output is 0.
4. KeyUpdate(sk_{i-1}) $\to sk_i$: The key update algorithm takes as input the private signing key sk_{i-1}. It outputs a re-randomized key sk_i for the same verification key. sk_i has the same length as sk_{i-1} and a distribution that is indistinguishable from that of sk_{i-1}.

***l*-Leakage resilience security.** The security of "*l-leakage resilience against continual leakage on memory and computation* (CLR)" for signatures has been defined [49]. However, this definition of the signature [49] requires the signing algorithm to first update the signing key to a new one, and then sign the message. In our definition, this requirement is unnecessary. We define the security of *l-continual leakage resilience* (CLR) for our signatures based on the following game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

1. Setup phase. The challenger $\mathcal{C}$ takes a security parameter 1^λ and executes the KeyGen algorithm to return the verification key vk to $\mathcal{A}$ while keeping the signing key sk_0 itself. Set $i = 1$. Let L_{sk} be the numbers of leaked bits with the signing key in the current time period. No leakage is allowed in this phase.
2. Query phase. The adversary $\mathcal{A}$ adaptively issues the following three kinds of queries: *Signing Queries*. $\mathcal{A}$ supplies a message m to $\mathcal{C}$. The challenger signs the message and returns the resulting signature.
   Leak queries. Let $sk = sk_{i-1}$ be the current signing key. $\mathcal{A}$ supplies to $\mathcal{C}$ a polynomial-time computable arbitrary function f: $\{0, 1\}^* \to \{0, 1\}^*$, and receives $f(sk)$ or $\perp$ from $\mathcal{C}$ depending on whether the amount of leaked bits exceeds the

leakage bound, where sk is the signing key in the current time period. In addition, $\mathcal{C}$ updates the number of leaked bits with sk by adding $|f(sk)|$ or 0 to it. *Update Queries*. $\mathcal{A}$ asks the challenger C to update the signing key. C updates the signing key from sk_{i-1} to sk_i. Set $i = i + 1$.

3. Forgery phase. The adversary $\mathcal{A}$ supplies the challenger with a message/signature pair, (m^*, σ^*), with the restriction that m^* has not been previously queried. The adversary wins the game if Ver(vk, m^*, σ^*)=1.

Definition 1. A signature scheme SIG=(KeyGen,Sig,Ver,KeyUpdate) is *l-continual leakage resilient*(CLR) if the advantage $\mathrm{Adv}^{CLR}_{SIG,\mathcal{A}}(\lambda)$ of any probabilistic polynomial-time adversary $\mathcal{A}$ to win the above game is a negligible function in λ.

A signature scheme with *l-leakage resilience security*, which is also *l-continual leakage resilient* under the above definition, was developed [49]. We describe the scheme as follows and show that multiple signatures can be aggregated to ensure that it satisfies the requirement of the auditing protocol.

1. KeyGen(λ) $\rightarrow$ (vk, sk_0). It chooses a composite order bilinear group G as described in Section 2.1. It randomly selects $g, u, h \leftarrow_R G_{p_1}$ and $R, R', R'', R''' \leftarrow_R G_{p_4}$. The verification key is set to be $vk = \{N, G, R, gR', uR'', hR'''\}$. Then it randomly selects $g_2 \leftarrow G_{p_2}$, $g_3 \leftarrow G_{p_3}$, and random vectors $\vec{r} = (r_1, \cdots, r_n), \vec{c} = (c_1, \cdots, d_n), \vec{d} = (d_1, \cdots, d_n), \vec{f} = (f_1, \cdots, f_n), \vec{x} = (x_1, \cdots, x_n), \vec{y} = (y_1, \cdots, y_n), \vec{z} = (z_1, \cdots, z_n) \in Z_N^n$. Denote $g^{\vec{r}}$ to be the vector with n group elements $(g^{r_1}, \cdots, g^{r_n})$ and $g^{\vec{r}}g_2^{\vec{c}}$ to be the vector with n group elements formed by componentwise multiplication $(g^{r_1}g_2^{c_1}, \cdots, g^{r_n}g_2^{c_n})$. We let $\vec{S_0} = (S_{1,0}, \cdots, S_{n,0}), \vec{U_0} = (U_{1,0}, \cdots, U_{n,0})$ and $\vec{H_0} = (H_{1,0}, \cdots, H_{n,0})$ be vectors with n group elements defined as follows:
$\vec{S_0} = g^{\vec{r}}g_2^{\vec{c}}g_3^{\vec{x}}$, $\vec{U_0} = u^{\vec{r}}g_2^{\vec{d}}g_3^{\vec{y}}$, $\vec{H_0} = h^{\vec{r}}g_2^{\vec{f}}g_3^{\vec{z}}$
The signing key is $sk_0 = \{\vec{S_0}, \vec{U_0}, \vec{H_0}\}$ (this contains $3n$ group elements).
2. Sig(m, sk_i) $\rightarrow \sigma$. The signing algorithm produces the signature σ under current signing key sk_i as:
$\sigma = (\sigma_1, \sigma_2) = (U_{1,i}^m H_{1,i}, S_{1,i})$
where *name* is arbitrary here and can be treated as the file identifier when this scheme is used in auditing protocol.
3. Ver(vk, m, σ) $\rightarrow$ {"True" or "False"}. The verification algorithm checks that
$e(\sigma_1, gR'') = e(\sigma_2, (uR'')^m(hR''')) \neq 1$, and $e(\sigma_1, R) = e(\sigma_2, R) = 1$.
4. KeyUpdate(sk_{i-1}) $\rightarrow sk_i$.

The signature scheme presented above can be used to generate authentication values in the auditing protocol. The structure of the signature allows multiple signatures to be aggregated into a linear combination as follows. We must point that, to make the aggregation possible, blocks of a file identified by *name* should be signed under the same signing key sk_i.

Aggregation. Suppose there are n message/signature pairs $\{(m_1, \sigma_1), \cdots, (m_n, \sigma_n)\}$. We can construct a query which is an s-element set $Q = \{(i, v_i)\}$ by first randomly choosing an s-element subset I of $[1, \cdots, n]$ which denotes the block indexes and then choosing an element v_i for each $i \in I$. We can aggregate the signatures by computing the value $\mu \leftarrow \Sigma_{i \in I} v_i m_i$ and an aggregated signature $\tilde{\sigma} = (\tilde{\sigma}_1, \tilde{\sigma}_2) = (\prod_{i \in I} \sigma_{1,i}^{v_i}, \sigma_{2,*})$, where $\sigma_{1,i}$ is the first part of σ_i. Note that the second part of σ_i equals that of σ_j, we denote $\sigma_{2,*}$ as the second part of all the σ_i for all $i \in I$. The aggregated signature can be verified by checking that $e(\tilde{\sigma}_1, gR'') = e(\tilde{\sigma}_2, (uR'')^{\mu}(hR''')^{\sum_{i \in I} v_i}) \neq 1$, and that $e(\tilde{\sigma}_1, R) = e(\tilde{\sigma}_2, R) = 1$.

3. Auditing protocol with continual key-leakage resilience

In this section, we propose our first auditing protocol and prove that it achieves continual key-leakage resilience security. We also present an analysis of the performance of our protocol.

3.1. Definition and security model

To resist side-channel attacks, the client's secret key for auditing should be updated periodically while leaving the public key unchanged. Therefore, an auditing protocol with continual key-leakage resilience security consists of the following five algorithms (SysSetup, AuthGen, ProofGen, ProofVerify, KeyUpdate):

- SysSetup(1^{λ}) $\rightarrow$ (PK, SK_0): the system setup algorithm is executed by the client and takes as input a security parameter λ, and generates a public key PK and the client's initial secret key SK_0.
- AuthGen(PK, SK_{i-1}, F) $\rightarrow$ (Φ): the authenticator generation algorithm is executed by the client and takes as input the public key PK, the client's current secret key SK_{i-1} and a file F, and generates the set of authenticators Φ for F.
- ProofGen(PK, $Chal$, F, Φ) $\rightarrow$ (P): the proof generation algorithm is run by the cloud server and takes as input the public key PK, a challenge $Chal$ which is randomly selected by the client and sent to the cloud, a file F and the set of authenticators Φ, and generates a proof P that the cloud has correctly preserved F.
- ProofVerify(PK, $Chal$, P) $\rightarrow$ ("True" or "False"): the proof verification algorithm is executed by the client to verify the proof generated by ProofGen(PK, $Chal$, F, Φ). It takes as input the public key PK, the same challenge $Chal$ used in ProofGen and the proof P, and outputs "True" or "False".
- KeyUpdate(SK_{i-1}) $\rightarrow$ (SK_i): the key update algorithm is executed by the client and takes as input the client's current secret key SK_{i-1}, and returns a re-randomized secret key SK_i with the same length as SK_{i-1} and a distribution that is indistinguishable from that of SK_{i-1}.

Security model. We consider the leakage resilience security [49] and data possession property [50] in the security model. An adversary (possibly the cloud itself) can obtain partial information about the client's secret key for auditing between two key-update operations. This means that key leakage may occur in the model. The security of *l-continual leakage resilience*(CLR) for the auditing protocol is based on the following game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

1. Setup phase. The challenger $\mathcal{C}$ takes a security parameter λ and implements the SysSetup algorithm to return the public key PK to $\mathcal{A}$ while retaining the client's initial secret key SK_0 itself. Set $i = 1$. Let L_{SK} be the number of leaked bits with the current secret key. No leakage is allowed in this phase.
2. Query phase. The adversary $\mathcal{A}$ adaptively issues the following queries:
   Authenticator queries. $\mathcal{A}$ adaptively selects and sends a series of blocks $m_1, \cdots, m_n$ to the challenger $\mathcal{C}$. $\mathcal{C}$ computes and sends to $\mathcal{A}$ the authenticators for $m_k (k = 1, \cdots, n)$ under the current secret key. $\mathcal{A}$ stores all blocks $F = (m_1, \cdots, m_n)$ and their corresponding authenticators.
   Leak queries. Let $SK = SK_{i-1}$ be the current secret key. $\mathcal{A}$ supplies to $\mathcal{C}$ a polynomial-time computable arbitrary function f: $\{0, 1\}^* \to \{0, 1\}^*$, and receives $f(SK)$ or $\perp$ from $\mathcal{C}$ depending on whether the number of leaked bits exceeds the leakage bound, where SK is the client's current secret key. Then, $\mathcal{C}$ updates the number of leaked bits with SK by adding $|f(SK)|$ or 0 to it.
   Update queries. $\mathcal{A}$ asks the challenger $\mathcal{C}$ to update the secret key SK_{i-1}. $\mathcal{C}$ updates the secret key from SK_{i-1} to a re-randomized secret key SK_i with the same length as SK_{i-1} and a distribution that is indistinguishable from that of SK_{i-1}.
3. Challenge phase. The challenger $\mathcal{C}$ sends $\mathcal{A}$ a challenge *Chal* and asks $\mathcal{A}$ to provide a proof of the correct preservation of the blocks $m_{s_1}, \cdots, m_{s_c}$ of file $F = (m_1, \cdots, m_n)$ under *Chal*, where $1 \le s_l \le n$, $1 \le l \le c$, and $1 \le c \le n$.
4. Forgery phase. The adversary $\mathcal{A}$ outputs a proof P for the preservation of the blocks $m_{s_1}, \cdots, m_{s_c}$ under *Chal*. $\mathcal{A}$ wins the game if ProofVerify(*PK, Chal, P*)="True".

Definition 2 (*Continual Key-leakage Resilience*)**.** An auditing protocol is *l-continual leakage resilience*(CLR) if for any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage of the adversary to win the above game is a negligible function in λ.

Definition 3. (*Detectability*) [15]: An auditing protocol is (ρ, δ)-*detectable* ($0 < \rho, \delta < 1$) if the probability to detect a fraction ρ of corrupted blocks is at least δ.

3.2. Proposed protocol

Let Π be the above signature scheme in Section 2.

- SysSetup(1^λ) $\to$ (*PK, SK*$_0$): Let λ be the security parameter. It runs Π.KeyGen(λ) and sets $PK = \Pi.vk$, $SK_0 = \Pi.sk_0$. Let $Hash : \{0, 1\}^* \to G_{p_1}$ be a cryptographic hash function.
- AuthGen(PK, SK_{i-1}, F) $\to$ (Φ): Let the file $F = \{m_1, \cdots, m_n\}$ be identified by *name*, where $m_i \in Z_N (i = 1, \cdots, n)$. The client first computes $\{\sigma_i\}_{1 \le i \le n}$, where $\sigma_i = \Pi.Sig(m_i, SK_{i-1})$. Then the client computes $U = g^r g_2^{r_2} g_3^{r_3}$ where r, r_2, $r_3 \in Z_N$ is randomly selected. The authenticator $\Phi = (U, \{(\sigma_{1,i} \cdot Hash(name \parallel i \parallel U)^r \cdot u^{rm_i}, \sigma_{2,i})\}_{1 \le i \le n})$
- ProofGen(*PK, Chal, F,* Φ) $\to$ (*P*): The client randomly selects and sends to the cloud a challenge $Chal = \{(i, v_i)\}_{i \in I}$, where $I = \{s_1, \cdots, s_c\}$ is a c-element subset of set $[1, n]$ and $v_i \in Z_N$. Let file $F = \{m_1, \cdots, m_n\}$ be identified by *name*. The cloud computes an aggregated authenticator $\Phi = (U, \tilde{\sigma})$, where $\tilde{\sigma} = (\tilde{\sigma}_1, \tilde{\sigma}_2) = (\prod_{i \in I} \sigma_{1,i}^{v_i} \prod_{i \in I} Hash(name \parallel i \parallel U)^{rv_i} \prod_{i \in I} u^{rm_i v_i}, \sigma_{2,*})$ as above. It also computes the linear combination of randomly selected blocks $\mu \leftarrow \Sigma_{i \in I} v_i m_i$. It then sends $P = \{\Phi, \mu\}$ to the client along with the file tag as the response proof of correct file storage.
- ProofVerify(*PK, Chal, P*) $\to$ ("True" or "False"): Let the challenge *Chal* be $\{(i, v_i)\}_{i \in I}$. By receiving a proof *P*, the client checks whether the following equations hold:
  $e(\tilde{\sigma}_1, gR'') = e(\tilde{\sigma}_2, (uR'')^\mu (hR''')^{\sum_{i \in I} v_i}) \prod_{i \in I} e(Hash(name \parallel i \parallel U)^{v_i}, U) e(u^\mu, U) \neq 1$,
  and
  $e(\tilde{\sigma}_1, R) = e(\tilde{\sigma}_2, R) = 1$.
  If they all hold, it returns "True"; otherwise it returns "False".
- KeyUpdate(SK_{i-1}) $\to$ (SK_i): It runs $SK_i \leftarrow \Pi.KeyUpdate(SK_{i-1})$ to update the secret key.

Correctness. The ProofVerify algorithm returns "True" if the valid proof *P* is generated under the random challenge *Chal*. Note that for any secret key SK_i, the G_{p_1} parts of $\vec{S_i}, \vec{U_i}, \vec{H_i}$ have the form of $g^{\vec{r'}}, u^{\vec{r'}}, h^{\vec{r'}}$ for some $\vec{r'} \in Z_N^n$. Thus, $\sigma_{1,i}$ can be written as $\sigma_{1,i} = (u^{m_i} h)^{r'} g_2^{s_2} g_3^{s_3}$, and $\sigma_{2,i}$ can be written as $\sigma_{2,i} = g^{r'} g_2^{t_2} g_3^{t_3}$ for some value r', s_2, s_3, t_2, $t_3 \in Z_N$. Then:

$e(\tilde{\sigma}_1, gR'') = e(\prod_{i \in I} \sigma_{1,i}^{v_i}, gR'') e(\prod_{i \in I} Hash(name \parallel i \parallel U)^{rv_i}, gR'') e(\prod_{i \in I} u^{rm_i v_i}, gR'') = e((u^\mu h^{\sum_{i \in I} v_i})^{r'}, g) \prod_{i \in I} e(Hash(name \parallel i \parallel U)^{v_i}, U) e(u^\mu, U) \neq 1$

$e(\tilde{\sigma}_2, (uR'')^\mu (hR''')^{\sum_{i \in I} v_i}) = e(g^{r'} g_2^{t_2} g_3^{t_3}, (uR'')^\mu (hR''')^{\sum_{i \in I} v_i}) = e((u^\mu h^{\sum_{i \in I} v_i})^{r'}, g) \neq 1$ i.e. $e(\tilde{\sigma}_1, gR'') = e(\tilde{\sigma}_2, (uR'')^\mu (hR''')^{\sum_{i \in I} v_i}) \prod_{i \in I} e(Hash(name \parallel i \parallel U)^{v_i}, U) e(u^\mu, U)$. and

$e(\tilde{\sigma}_1, R) = e(\tilde{\sigma}_2, R) = 1$.

Table 1
Basic information.

Ellipse Curve Type	Type A1
Ellipse Curve	$y^2 = x^3 + x$
Symmetry or not	Symmetry
Order	$N = p_1 p_2 p_3 p_4$
Security Level	$\log p_i$=192
Platform	Personal Computer
CPU Series	Intel Core i5-6300
RAM	8GB
Operate System	Windows 10
JDK Version	JDK 1.8.0
jPBC Version	2.0.0

3.3. Security analysis

Theorem 1 (Continual Key-leakage Resilience)**.** *If **Assumption 1** holds, then the above auditing protocol is continual key-leakage resilient.*

Proof. First, as the signature scheme on which our auditing protocol is based is *l-continual leakage resilient*, the leakage in the authenticator generation algorithm cannot improve the advantage of the adversary to break the security of our auditing protocol. Next, we show that, if the adversary can compute a proof P for the blocks $m_{s_1}, \cdots, m_{s_c}$ integrity under *Chal*, and pass the ProofVerify check, the challenger can break **Assumption 1** with a non-negligible advantage. □

Suppose the forged proof for the query $Q = \{(i, v_i)\}$ is $P' = \{U', \tilde{\sigma}', \mu'\}$ and the expected valid proof generated by an honest prover is $P = \{U, \tilde{\sigma}, \mu\}$ where $\tilde{\sigma}.\tilde{\sigma_1} = \prod_{(i,v_i)\in Q} \sigma_{1,i}^{v_i} \prod_{(i,v_i)\in Q} Hash(name \parallel i \parallel U)^{rv_i} \prod_{(i,v_i)\in Q} u^{rm_i v_i}$ and $\mu = \sum_{(i,v_i)\in Q} v_i m_i$.

As $\tilde{\sigma}'$ can pass the ProofVerify check, i.e.

$e(\tilde{\sigma}_1', gR'') = e(\tilde{\sigma}_2', (uR'')^{\mu'}(hR''')^{\sum_{(i,v_i)\in Q} v_i}) \prod_{(i,v_i)\in Q} e(Hash(name \parallel i \parallel U')^{v_i}, U') e(u^{\mu'}, U')$, and $\prod_{(i,v_i)\in Q} e(Hash(name \parallel i \parallel U')^{v_i}, U') e(u^{\mu'}, U')$ only comes from $e(\tilde{\sigma}_1', gR'')$, the $\tilde{\sigma}_1'$'s G_{p_1} part from the signature Π should be equal to that of $\tilde{\sigma_1}$. That is, $u^{\mu'} h^{\sum_{(i,v_i)\in Q} v_i} = u^{\mu} h^{\sum_{(i,v_i)\in Q} v_i} \bmod N$, i.e. $u^{\mu'} = u^{\mu} \bmod N$. Note that $\mu \neq \mu' \bmod N$ which means that $\mu \equiv \mu' \bmod p_1$. Now, we can compute $a = gcd(\mu' - \mu, N)$ and $b = N/a$ satisfying that one of a, b is equal to p_1. Without loss of generality, let $a = p_1$ and $b = p_2 p_3 p_4$. The challenger can break **Assumption 1** by checking whether $e((U_1 U_2)^a, C)$ equals 1. As $(U_1 U_2)^a = U_2^a \bmod N$, if $e((U_1 U_2)^a, C) = 1$, which means C does not contain part of G_{p_2}, then $C \in G_{p_1 p_3}$; otherwise, $C \in G_{p_1 p_2 p_3}$.

Theorem 2 (Detectability)**.** *Our auditing protocol is* $(\frac{t}{n}, 1 - (\frac{n-t}{n})^c)$ *detectable if the cloud stores a file with n blocks, and deletes or modifies t blocks.*

Proof. According to the definitions, n blocks of the file are stored in the cloud, including t corrupted blocks (by deletion or modification). Therefore, if the challenged blocks selected randomly by the client contain at least one corrupted block, then the corrupted blocks can be detected. The probability that a randomly selected block is not among the corrupted t blocks is $1 - \frac{t}{n}$. Therefore, none of c randomly selected blocks in the corrupted t blocks is $(1 - \frac{t}{n})^c$. Now, we can obtain the detectable probability as $1 - (1 - \frac{t}{n})^c = 1 - (\frac{n-t}{n})^c$. □

3.4. Performance evaluation

In this section, we present a performance analysis of our protocol. We implement our protocol based on jPBC Library [http://libeccio.di.unisa.it/projects/jpbc/]. Table 1 summarizes the basic information of our implementation.

The order of the group has $192 \times 4 = 768$ bits. Accordingly, the sizes of an element in Z_N and G are 96 bytes and 196 bytes, respectively. We divide the data file into 1,000,000 blocks, which is approximately 91.55M bytes.

We demonstrate the duration of authenticator generation in Fig. 2 by varying the number of file blocks. We also illustrate the duration of the challenge generation, the proof generation, and the proof verification for a different number of checked data blocks in Fig. 3.

Focusing on the communication messages in our auditing protocol, we evaluate the size of the challenge and the proof messages in bytes in the proof generation process. Fig. 4 shows the linearity of the size of the challenge message with the number of checked blocks. Fig. 5 indicates that the size of the proof message is constant, i.e., 488 B.

4. Extension to forward secure protocol under continual key-leakage

In practice, the client's secret key of the auditing protocol may be fully exposed. Usually, clients prefer to use software-based key management to manage their different keys for different security goals. The limitation of software-based key management and careless mistakes by the client make it possible for the key to be exposed. In addition, if data loss incidents

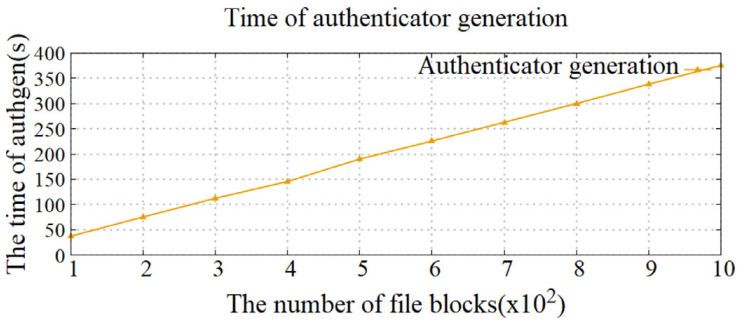


Fig. 2. The time of authenticator generation with different number of blocks.

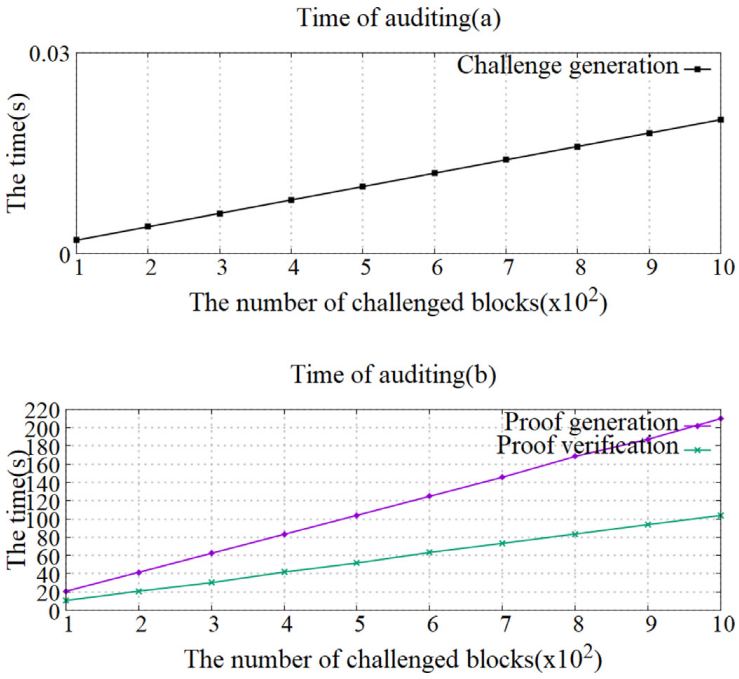


Fig. 3. The time of auditing procedures with different number of checked blocks.

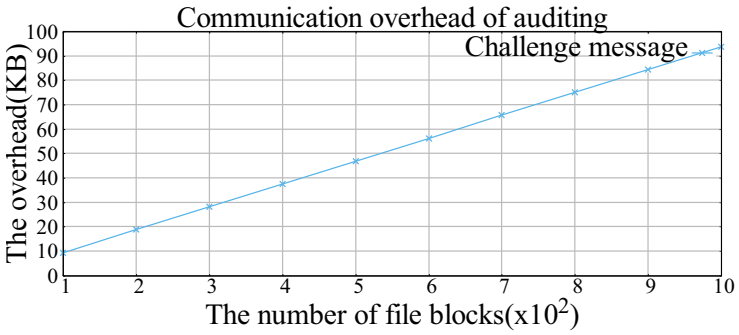


Fig. 4. Communicational Cost. (a) The size of the challenge message with different number of checked blocks.

were to occur on the cloud server side or, for storage cost reasons, the cloud server discards data the client rarely accesses, then the cloud service provider might want to obtain the clients secret keys for auditing purposes to forge authenticators and conceal the fact. Full exposure of their auditing secret key would clearly be disastrous for a client of cloud storage applications. Therefore, solving this problem to prevent exposure of the client's auditing secret key is of critical importance. However, to the best of our knowledge, although auditing protocols, which provide forward security under key exposure by updating the auditing secret key periodically, have been developed [15–17], they do not consider the severity of the consequences caused by partial secret key leakage between key-updates as a result of side-channel attacks. In this section, we describe the extension of our auditing protocol (as described in Section 3) with security against continual key-leakage attacks such that it supports both "key-exposure resilience" and "continual key-leakage resilience" simultaneously.

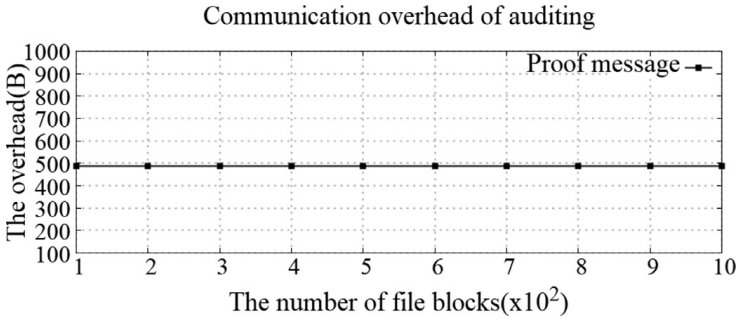


Fig. 5. Communicational Cost. (b) The size of the proof message with different number of checked blocks.

A method to securely construct a forward-secure signature scheme under continual key-leakage from any signature scheme under continual key-leakage has been described [20]. In this section, we apply their tree-based construction method to the continual key-leakage resilient auditing protocol we present in Section 3 and construct a forward-secure auditing protocol under continual key-leakage. In the following subsections, we first present the "forward security under continual leakage" or "key-exposure resilience under continual leakage" definition for auditing protocol, and then show the concrete construction of our protocol.

4.1. The model

To equip an auditing protocol with forward security (key-exposure resilience) under continual leakage, we modify an existing auditing protocol model with forward security [15]. An auditing protocol with forward security under continual leakage consists of the following five algorithms (SysSetup, AuthGen, ProofGen, ProofVerify, KeyUpdate):

- SysSetup(1^λ, T) $\rightarrow$ (PK, $SK_{0,0}$): The system setup algorithm is executed by the client and divides the entire lifetime into T periods. It takes a security parameter λ and T as input and outputs a public key PK and the client's initial secret key $SK_{0,0}$. Note that $SK_{j,i}$ is denoted as the current secret key in time period j which has been updated i times in this time period, i.e. the secret key in a time period can be updated to a new one for the same time period to resist continual key-leakage.
- AuthGen(PK, j, $SK_{j,i-1}$, F) $\rightarrow$ (Φ): The authenticator generation algorithm is executed by the client and takes as input the public key PK, a time period j, a client's current secret key $SK_{j,i-1}$ and a file F, and generates the set of authenticators Φ for F.
- ProofGen(PK, j, $Chal$, F, Φ) $\rightarrow$ (P): The proof generation algorithm is executed by the cloud and takes as input the public key PK, a time period j, a random challenge $Chal$ selected by the client and sent to the cloud, a file F, and the set of authenticators Φ, and generates a proof P for the clouds correct preservation of F.
- ProofVerify(PK, j, $Chal$, P) $\rightarrow$ ("True" or "False"): The proof verification algorithm is executed by the client to verify the proof generated by ProofGen(PK, j, $Chal$, F, Φ). It takes as input the public key PK, a time period j, the same challenge $Chal$ used in ProofGen and a proof P, and returns "True" or "False".
- KeyUpdate(PK, j, $SK_{j,i-1}$) $\rightarrow$ ($SK_{j+1,0}$): The key update algorithm is executed by the client and takes as input the public key PK, the current time period j and the client's current secret key $SK_{j,i-1}$, and outputs a new secret key $SK_{j+1,0}$ for the next time period $j+1$.

Forward security(key-exposure resilience) under continual leakage. The security definition of forward security under continual leakage for the auditing protocol is based on the following game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

1. Setup phase. Set time period $j=0$. The challenger $\mathcal{C}$ takes a security parameter λ and implements the SysSetup algorithm to return the public key PK to $\mathcal{A}$ while retaining the client's initial secret key $SK_{j,0}$. Set $i=1$. Let L_{SK} be the number of leaked bits with the current secret key in time period j. No leakage is allowed in this phase.
2. Query phase. The adversary $\mathcal{A}$ adaptively issues the following queries:
   Authenticator queries. $\mathcal{A}$ adaptively selects and sends a series of blocks $m_1, \cdots, m_n$ to the challenger $\mathcal{C}$. $\mathcal{C}$ computes and sends to $\mathcal{A}$ the authenticators for $m_k (k=1,\cdots,n)$ under the current secret key in time period j. $\mathcal{A}$ stores all blocks $F=(m_1,\cdots,m_n)$ and their corresponding authenticators.
   Leak queries. Let $SK=SK_{j,i-1}$ be the current secret key in time period j. $\mathcal{A}$ supplies to $\mathcal{C}$ a polynomial-time computable arbitrary function f: $\{0,1\}^* \rightarrow \{0,1\}^*$, and receives $f(SK)$ or $\perp$ from $\mathcal{C}$ depending on whether the amount of leaked bits exceeds the leakage bound, where SK is the client's current secret key. Then, $\mathcal{C}$ updates the number of leaked bits with SK by adding $|f(SK)|$ or 0 to it.
   Update queries. This query models the key update in period j for the resilience of continual key-leakage. $\mathcal{A}$ asks the challenger $\mathcal{C}$ to update the secret key $SK_{j,i-1}$ in time period j. $\mathcal{C}$ updates the secret key from $SK_{j,i-1}$ to a re-randomized secret key $SK_{j,i}$ with the same length as $SK_{j,i-1}$ and a distribution indistinguishable from that of $SK_{j,i-1}$.

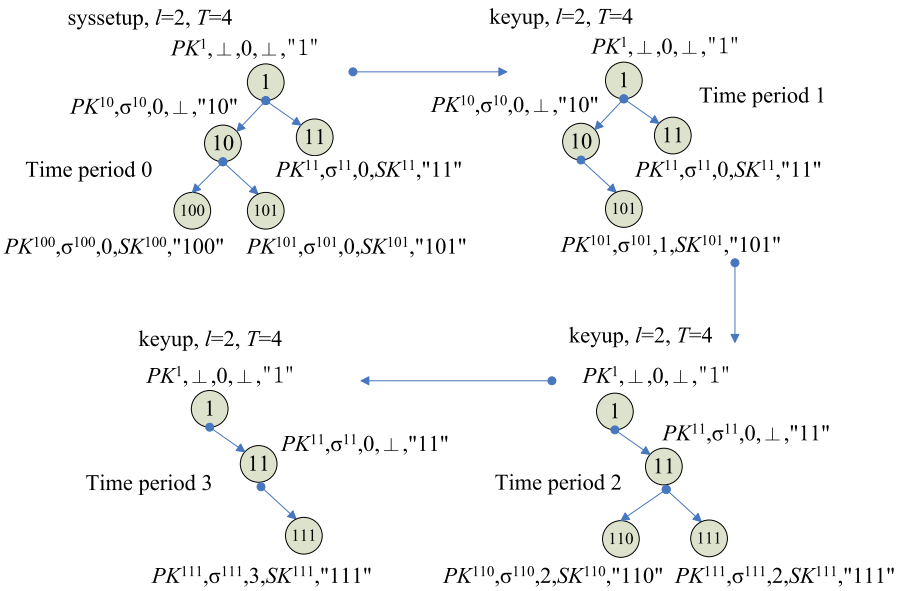


Fig. 6. Structure of thebinary tree in the lifetime.

At the end of each time period j, $\mathcal{A}$ can decide to remain in the query phase for the next time period $j = j + 1$ or proceed to the break-in phase.

3. Break-in phase. In this phase, the adversary $\mathcal{A}$ sets the break-in time period $b = j$, which means that key exposure occurs during this time period. The challenger $\mathcal{C}$ generates the secret key $SK_{b,0}$ by using the KeyUpdate algorithm and sends it to $\mathcal{A}$.
4. Challenge phase. The challenger $\mathcal{C}$ sends $\mathcal{A}$ a challenge *Chal*, a time period $j^*(j^* < b)$ and asks $\mathcal{A}$ to provide a proof for the correct preservation of the blocks $(m_{s_1}, \cdots, m_{s_c})$ of file $F = (m_1, \cdots, m_n)$ under *Chal* in time period j^*, where $1 \le s_l \le n$, $1 \le l \le c$, and $1 \le c \le n$.
5. Forgery phase. Adversary $\mathcal{A}$ outputs a proof P for the preservation of the blocks $(m_{s_1}, \cdots, m_{s_c})$ under *Chal* in time period j^*. $\mathcal{A}$ wins the game if ProofVerify(PK, j^*, *Chal*, P)="True".

Definition 4 (Forward Security(Key-Exposure Resilience) under Continual Leakage)**.** An auditing protocol is *l-Forward Secure(Key-Exposure Resilient) under Continual Leakage* if for any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage of the adversary to win the above game is a negligible function in λ.

4.2. Our protocol

Let Ψ be the auditing protocol that is secure under continual key-leakage, as described in Section 3. Let Ξ be the continual key-leakage resilient signature scheme used in Ψ. We construct a new binary-tree-based auditing protocol from Ψ by following an existing approach [20].

1) Notations and Structures: We first describe the structures of our protocol at a high level. The entire lifetime of data is divided into discrete time periods $0, \cdots, T-1$. Without loss of generality, we assume $T = 2^l$. We employ a binary tree structure to designate these time periods. The leaf nodes of the tree from the leftmost one to the rightmost one represent the independent auditing protocol Ψ that is used to audit the data integrity from time period 0 to $T-1$. That is, the depth of the binary tree is $l = log_2 T$. The depth of the root node is 0. The leftmost two leaf nodes representing time periods 0 and 1 are generated in the *system setup process*. Other leaf nodes are generated by the *key update algorithm* as follows: we generate two leaf nodes representing two time periods $j = 2t$ and $j+1$ $(t = 1 \sim \frac{T}{2} - 1)$ with the same parent node at a time when the time period changes from $j-1$ to j for the following time periods. A leaf node of the tree corresponding to time period j is erased when the time period changes from j to $j+1$, which is implemented by the key update procedure. As [20], each node of the tree contains an independently generated Ψ public key and the corresponding secret key, and the signature of the Ψ public key under the secret key of its parent node is also included, except for the root node. The secret key is erased immediately once both child nodes of the node have been generated, thereby ensuring the forward security of the constructed auditing protocol.

Each node is appointed to a binary string w to represent the path from the root node to the node. Let the binary string of the root node be $w = 1$. The binary string of a node can be constructed as follows: let w be the binary string representing its parent node. If it is the left child node, its representing binary string is $w||0$; otherwise it is $w||1$. This construction indicates that the maximum length of w is $l + 1$. We provide an example of a binary tree with a depth of 2 for multiple time periods in Fig. 6.

An item of Tree[], i.e. structure of a node:

PK^w	Public key of this node
σ	Signature of PK^w under its parent's secret key
ts	Time period
SK^w	Secret key of this node
w	Binary string representing the node

Auth[]: l+n spaces

(PK_1, σ_1)	Public key of the first node in the path and the signature under root node's secret key
(PK_2, σ_2)	Public key of the second node in the path and the signature under the first node's secret key
……	……
(PK_l, σ_l)	Public key of the leaf node in the path and the signature under its parent node's secret key
ϕ	The authenticator generated by ψ

P[]: l+2 spaces

(PK_1, σ_1)	Public key of the first node in the path and the signature under root node's secret key
(PK_2, σ_2)	Public key of the second node in the path and the signature under the first node's secret key
……	……
(PK_l, σ_l)	Public key of the leaf node in the path and the signature under its parent node's secret key
μ	The linear combination of the challenged blocks
ρ	The aggregated authenticator generated by ψ

Fig. 7. Structures.

We use an array named *Tree*[] to store the entire binary tree with 2^{l+1} storage space. *Tree*[w] stores the node of which the representing binary string is w, which is a tuple as (PK^w, σ^w, ts^w, SK^w, w). Here, PK^w is the independently generated Ψ public key, and SK^w is the corresponding secret key. Further, σ^w is the signature of PK^w signed by Ξ under the secret key of its parent node, and ts^w is the time period.

The public key of the auditing protocol is the Ψ public key of the root node which is constant during the entire lifetime. The secret key $SK_{j,*}$ corresponding to time period j consists of the current state of the entire binary tree and the representing string of the leaf node corresponding to the current time period.

The authenticator of a message m includes an authenticator of m generated by Ψ.AuthGen under the secret key of the current leaf node, along with a chain of pairs(*public key,signature of public key under the node's parent's secret key*) of each non-root node on the path from the root node to this leaf. This ensures that the authenticator can be verified with only the public key of the root node. We use $\Phi = (ts, Auth[])$ to store the authenticator of F, where ts is the time period and *Auth*[] is an array in which to store the chain and the authenticator generated by Ψ. As the secret key used to generate authenticator is in a leaf node, the number of non-root nodes on the path from the root node to this leaf is l. Let $F = \{m_1, \cdots, m_n\}$, then *Auth*[] needs $l + n$ spaces. We also use an array P with $l + 2$ spaces in which to store the proof. The first l spaces store the above chain of pairs(*public key,signature of public key under the node's parent's secret key*). The last two spaces are used to store the linear combination of m_i, i.e., $\mu = \Sigma_{i\in I} v_i m_i$, and the aggregated authenticator generated by Ψ.AuthGen. We provide an example of the structure of the secret key, the authenticator, and the proof of our forward-secure auditing protocol in Fig. 7.

Algorithm 1 syssetup.

Input: input λ, T
Output: output public key and secret key

Set an array $Tree[0 \cdots 2T-1] = \{(\perp, \perp, \perp, \perp, \perp)\}$; $w =$"1"; $l = log_2 T$;
$(PK, SK) \leftarrow \Psi.\text{SysSetup}(1^\lambda)$;
$Tree[\text{StrToInt}(w)]=(PK, \perp, 0, SK, w)$;
$TmpSk = SK$;
while $|\ w\ | < l+1$ **do**
  $(PK_1, SK_1) \leftarrow \Psi.\text{SysSetup}(1^\lambda)$;
  $(PK_2, SK_2) \leftarrow \Psi.\text{SysSetup}(1^\lambda)$;
  $Tree[\text{StrToInt}(w \parallel 0)]=$
    $(PK_1, \Xi.\text{Sig}(PK_1, TmpSk), 0, SK_1, w \parallel 0)$;
  $Tree[\text{StrToInt}(w \parallel 1)]=$
    $(PK_2, \Xi.\text{Sig}(PK_2, TmpSk), 0, SK_2, w \parallel 1)$;
  EraseSk($Tree[\text{StrToInt}(w)]$);
  $TmpSk = SK_1$; $w = w \parallel 0$;
return PK, $(Tree, w)$;

Algorithm 2 keyup.

Input: input $SK_{j,i-1}$, j
Output: output the initial secret key $SK_{j+1,0}$ in time period $j+1$

Parse $SK_{j,i-1}$ as $(Tree, w^j)$; $w = w^j$;
if $j == \frac{T}{2} - 1$ **then**
  while $|\ w\ | > l$ **do**
    EraseNode($Tree, w$);
    $w = w[0, |\ w\ | -1]$;
else
  $w = w[0, |\ w\ | -1]$;
  if $Tree[\text{StrToInt}(w \parallel 0)] \neq \perp$ **then**
    EraseNode($Tree, w \parallel 0$);
    $Tree[\text{StrToInt}(w \parallel 1)].\text{ts}=j+1$;
    return $(Tree, w \parallel 1)$;
  while $Tree[\text{StrToInt}(w \parallel 0)]==\perp$ **do**
    EraseNode($Tree, w \parallel 1$);
    $w = w[0, |\ w\ | -1]$;
  EraseNode($Tree, w \parallel 0$);
$w = w \parallel 1$;
$TmpSk = Tree[\text{StrToInt}(w)].SK^w$;
while $|\ w\ | < l+1$ **do**
  $(PK_1, SK_1) \leftarrow \Psi.\text{SysSetup}(1^\lambda)$;
  $(PK_2, SK_2) \leftarrow \Psi.\text{SysSetup}(1^\lambda)$;
  $Tree[\text{StrToInt}(w \parallel 0)]=$
    $(PK_1, \Xi.\text{Sig}(PK_1, TmpSk)$,j+1,$SK_1, w \parallel 0)$;
  $Tree[\text{StrToInt}(w \parallel 1)]=$
    $(PK_2, \Xi.\text{Sig}(PK_2, TmpSk)$,j+1,$SK_2, w \parallel 1)$;
  EraseSk($Tree[\text{StrToInt}(w)]$);
  $TmpSk = SK_1$; $w = w \parallel 0$;
$Tree[\text{StrToInt}(w)].\text{ts}=j+1$;
return $(Tree, w)$;

2) Description of Our Protocol: Let w^j be the string representing the leaf node corresponding to time period j, and $|w|$ be the length of w. Let $w[q, h]$ denote the substring $w[q] \cdots w[h]$.

- SysSetup(1^λ, T) $\rightarrow$ (PK, $SK_{0,0}$): Let λ be the security parameter and T be the total number of time periods. It runs (PK, $SK_{0,0}$) $\leftarrow$ **syssetup**(1^λ, T) described in Algorithm 1.
- KeyUpdate(PK, j, $SK_{j,i-1}$) $\rightarrow$ ($SK_{j+1,0}$): It runs $SK_{j+1,0} \leftarrow$**keyup**($SK_{j,i-1}$, j) described in Algorithm 2 to update the secret key.

- AuthGen($PK, j, SK_{j,i-1}, F$) $\rightarrow$ (Φ): Let the file $F = \{m_1, \cdots, m_n\}$ be identified by *name*, where $m_i \in Z_N (i = 1, \cdots, n)$. The client computes the authenticator $\Phi \leftarrow$ **authgen**($PK, SK_{j,i-1}, F$) described in Algorithm 3.

Algorithm 3 authgen.

Input: input $PK, SK_{j,i-1}, F$
Output: output the authenticator of F

Parse $SK_{j,i-1}$ as $(Tree, w^j)$; $w = w^j$;
Parse $F = \{m_1, \cdots, m_n\}$;
$Auth[0 \sim l + n - 1] = \{0\}$;
$Auth[l \sim l + n - 1]$=
    Ψ.AuthGen($PK, Tree$[StrToInt(w)].SK^w, F);
$k = 1$;
while $|\ w\ | > 1$ **do**
    $Auth[l - k]$=
        ($Tree$[StrToInt(w)].PK, $Tree$[StrToInt(w)].σ);
    $w = w[0, |\ w\ | - 1]$;
    $k = k + 1$;
return $\Phi = (j, Auth)$;

- ProofGen(*PK, j, Chal, F,* Φ) $\rightarrow$ (P): The client randomly selects and sends to the cloud a challenge $Chal = \{(i, v_i)\}_{i \in I}$, where $I = \{s_1, \cdots, s_c\}$ is a c-element subset of set $[1, n]$ and $v_i \in Z_N$. The cloud generates a proof $P \leftarrow$ **proofgen**(*PK, j, Chal, F,* Φ) described in Algorithm 4.

Algorithm 4 proofgen.

Input: input $PK, j, Chal, F, \Phi$
Output: output a proof P

Parse $Chal = \{(i, v_i)\}_{i \in I}$;
Parse $F = \{m_1, \cdots, m_n\}$;
Parse $\Phi = (j, Auth)$;
if $j \neq \Phi.j$ **then**
    return error;
$\mu = \sum_{i \in I} v_i m_i$;
$P[0 \sim l - 1] = Auth[0 \sim l - 1]$;
$P[l] = \Psi$.ProofGen($PK, Chal, F, \Phi$);
$P[l + 1] = \mu$;
return P;

- ProofVerify(*PK, j, Chal, P*) $\rightarrow$ ("True" or "False"): By receiving the proof P, the client can verify the proof by executing **proofver**(*PK, j, Chal, P*) described in Algorithm 5.

Algorithm 5 proofver.

Input: input $PK, j, Chal, P$
Output: output "True" or "False"

Parse $Chal = \{(i, v_i)\}_{i \in I}$;
$TmpPk = PK$; $i = 0$;
for $i < l$ **do**
    if Ξ.Ver($TmpPk, P[i].PK, P[i].\sigma$)=="False" **then**
        return error;
    $TmpPk = P[i].PK$;
    $i = i + 1$;
return Ψ.ProofVerify($TmpPk, Chal, P[l \sim l + 1]$);

In these algorithms, StrToInt is a function that transforms a binary string into its value, EraseSk is used to delete the secret key associated with a tree node and EraseNode is a function to delete a node from the tree. We can prove the security by applying the same technique used in [20]. The continual key-leakage resilience of Ψ ensures the forward security(key-exposure resilience) under continual leakage of the above auditing protocol. We omit it here.

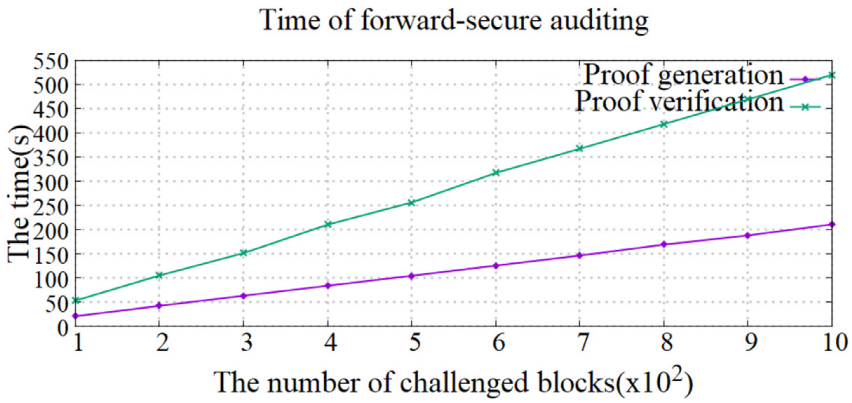


Fig. 8. The time of auditing procedures with different number of checked blocks.

4.3. Performance analysis

We use the same parameters as in the protocol in Section 3, i.e., the order of the curve group has $N = 192 \times 4 = 768$ bits, and the same testing environment. For simplification, we set the total time period $T = 16$ and the depth of the binary tree $l = 4$. The protocol is based on the continual key-leakage resilient protocol Ψ described in Section 3. The description of the protocol indicates that the size of the challenge message is the same as that in Ψ. The size of the proof message equals that in Ψ plus the size of l signatures signed by Ξ, which is the continual key-leakage signature in Section 2, i.e., $488 + 392 \times l$ bytes. The duration of the process to generate the authenticator is what in Ψ plus the time required for memory copying and *string-to-integer* converting operations. The duration of the process to generate the challenge is the same as that in Ψ, and the duration of the proof verification process is $l + 1$ times that in Ψ. The duration of the proof generation process is what in Ψ plus the time required for memory copying operations. Fig. 8 shows the duration of the proof generation and proof verification processes with different number of checked data blocks.

5. Conclusion

In this paper, we focus on providing a cloud auditing protocol with forward security under continual key-leakage. We feed a new security definition named "key-exposure resilience under continual leakage" to the auditing protocol and initiate the first attempt to construct an auditing protocol with this definition of security. This protocol enables the integrity of the data uploaded to the cloud to be successfully verified during the time period before that in which the client's current key exposure occurred even if the client's secret keys were partially leaked during previous periods. To this end, we first defined the formal security model of the auditing protocol with continual key-leakage resilience, and proposed the first concrete protocol. Then, we used an existing technique [20] to extend this protocol such that it provides key-exposure resilience under continual key-leakage.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Chengyu Hu: Conceptualization, Methodology, Software, Investigation, Writing - original draft, Writing - review & editing. **Yuqin Xu:** Software, Investigation. **Pengtao Liu:** Investigation, Writing - original draft. **Jia Yu:** Methodology, Writing - original draft. **Shanqing Guo:** Methodology, Writing - review & editing. **Minghao Zhao:** Software, Writing - original draft.

Acknowledgments

This project is supported in part by National Natural Science Foundation of China (no.61602275, 61632020, 61772311), Major Scientific and Technological Innovation Projects of Shandong Province, China (no.2019JZZY010132), Shandong Province Higher Educational Science and Technology Program (no.J15LN01), the Open Project of Key Laboratory of Network Assessment Technology, Institute of information engineering, Chinese Academy of Sciences (no.KFKT2019-002), the Open Project of Co-Innovation Center for Information Supply & Assurance Technology, Anhui University (no.ADXXBZ201702).

References

[1] Markets, Markets, Cloud Storage Market by Type, Deployment Model, Organization Size, Vertical, and Region - Global Forecast to 2022, 2018, https://www.marketsandmarkets.com/Market-Reports/cloud-storage-market-902.html.
[2] N. Bonvin, T.G. Papaioannou, K. Aberer, A Self-organized, Fault-tolerant and Scalable Peplication Scheme for Cloud Storage, in: Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC), ACM, 2010, pp. 205–216.
[3] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, S. Yekhanin, et al., Erasure Coding in Windows Azure Storage., in: Proceedings of the Usenix Annual Technical Conference (ATC), Boston, MA, 2012, pp. 15–26.
[4] Z. Whittaker, Amazon web services suffers partial outage, 2012,
[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, Provable Data Possession at Untrusted Stores, in: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS), ACM, 2007, pp. 598–609.
[6] C.C. Erway, A. Küpçü, C. Papamanthou, R. Tamassia, Dynamic provable data possession, ACM Trans. Inf. Syst. Secur. (TISSEC) 17 (4) (2015) 15.
[7] P. Kocher, J. Jaffe, B. Jun, Differential Power Analysis, in: Proceedings of the Annual International Cryptology Conference (CRYPTO), Springer, 1999, pp. 388–397.
[8] E. Brier, C. Clavier, F. Olivier, Correlation power analysis with a leakage model, in: Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES), Springer, 2004, pp. 16–29.
[9] P.C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, in: Proceedings of the Annual International Cryptology Conference (CRYPO), Springer, 1996, pp. 104–113.
[10] K. Gandolfi, C. Mourtel, F. Olivier, Electromagnetic analysis: concrete results, in: Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES), Springer, 2001, pp. 251–261.
[11] S. Jana, V. Shmatikov, Memento: Learning Secrets from Process Footprints, in: Proceedings of the 2012 IEEE Symposium on Security and Privacy (S&P), IEEE, 2012, pp. 143–157.
[12] M. Vuagnoux, S. Pasini, Compromising electromagnetic emanations of wired and wireless keyboards., in: Proceedings of the USENIX Security Symposium, 2009, pp. 1–16.
[13] R. Raguram, A.M. White, D. Goswami, F. Monrose, J.-M. Frahm, iSpy: Automatic Reconstruction of Typed Input from Compromising Reflections, in: Proceedings of the 18th ACM Conference on Computer and Communications Security, ACM, 2011, pp. 527–536.
[14] D. Genkin, I. Pipman, E. Tromer, Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs, in: Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES), Springer, 2014, pp. 242–260.
[15] J. Yu, K. Ren, C. Wang, V. Varadharajan, Enabling cloud storage auditing with key-exposure resistance, IEEE Trans. Inf. Forensics Secur. 10 (6) (2015) 1167–1179.
[16] J. Yu, H. Wang, Strong key-exposure resilient auditing for secure cloud storage, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1931–1940.
[17] X. Zhang, H. Wang, C. Xu, Identity-based key-exposure resilient cloud storage public auditing scheme from lattices, Inf. Sci. 472 (2019) 223–234.
[18] R.P. McEvoy, C.C. Murphy, W.P. Marnane, M. Tunstall, Isolated wddl: a hiding countermeasure for differential power analysis on fpgas, ACM Trans. Reconfigurable Technol. Syst. (TRETS) 2 (1) (2009) 3.
[19] J.D. Golić, C. Tymen, Multiplicative masking and power analysis of aes, in: International Workshop on Cryptographic Hardware and Embedded Systems (CHES), Springer, 2002, pp. 198–212.
[20] M. Bellare, A. O'Neill, I. Stepanovs, Forward-security under continual leakage, in: Proceedings of the 16th International Conference on Cryptology and Network Security (CANS), 2017, pp. 3–26.
[21] J. Yu, R. Hao, H. Xia, H. Zhang, X. Cheng, F. Kong, Intrusion-resilient identity-based signatures: concrete scheme in the standard model and generic construction, Inf. Sci. 442–443 (2018) 158–172.
[22] M. Naor, G.N. Rothblum, The Complexity of Online Memory Checking, in: Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, 2005, pp. 573–582.
[23] A. Juels, B.S. Kaliski Jr, PORs: Proofs of Retrievability for Large Files, in: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS), Acm, 2007, pp. 584–597.
[24] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, D. Song, Remote data checking using provable data possession, ACM Trans. Inf. Syst. Secur. (TISSEC) 14 (1) (2011) 12.
[25] E. Shi, E. Stefanov, C. Papamanthou, Practical dynamic proofs of retrievability, in: Proceedings of the 2013 ACM SIGSAC conference on Computer and Communications Security, ACM, 2013, pp. 325–336.
[26] Z. Ren, L. Wang, Q. Wang, M. Xu, Dynamic proofs of retrievability for coded cloud storage systems, IEEE Trans. Serv. Comput. 11 (4) (2018) 685–698.
[27] C. Wang, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for data storage security in cloud computing, in: Proceedings of the International Conference on Computer Communications (INFOCOM), IEEE, 2010, pp. 1–9.
[28] F. Armknecht, J.-M. Bohli, G.O. Karame, Z. Liu, C.A. Reuter, Outsourced proofs of retrievability, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 831–843.
[29] H. Tian, Y. Chen, C.-C. Chang, H. Jiang, Y. Huang, Y. Chen, J. Liu, Dynamic-hash-table based public auditing for secure cloud storage, IEEE Trans. Serv. Comput. 10 (5) (2017) 701–714.
[30] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage, IEEE Trans. Inf. Forensics Secur. 14 (2) (2019) 331–346.
[31] C. Wang, S.S. Chow, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for secure cloud storage, IEEE Trans. Comput. 62 (2) (2013) 362–375.
[32] Y. Yu, M.H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G. Min, Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage, IEEE Trans. Inf. Forensics Secur. 12 (4) (2017) 767–778.
[33] M. Etemad, A. Küpçü, Transparent, distributed, and replicated dynamic provable data possession, in: International Conference on Applied Cryptography and Network Security (ACNS), Springer, 2013, pp. 1–18.
[34] H.C. Chen, P.P. Lee, Enabling data integrity protection in regenerating-coding-based cloud storage: theory and implementation, IEEE Trans. Parallel Distrib. Syst. 25 (2) (2014) 407–416.
[35] J. Liu, K. Huang, H. Rong, H. Wang, M. Xian, Privacy-preserving public auditing for regenerating-code-based cloud storage, IEEE Trans. Inf. Forensics Secur. 10 (7) (2015) 1513–1528.
[36] S. Goldwasser, Multi party computations: past and present, in: Proceedings of the sixteenth annual ACM Symposium on Principles of Distributed Computing (PODC), ACM, 1997, pp. 1–6.
[37] A.C. Yao, Protocols for secure computations, in: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS), IEEE, 1982, pp. 160–164.
[38] E. Boyle, S. Goldwasser, A. Jain, Y.T. Kalai, Multiparty Computation Secure Against Continual Memory Leakage, in: Proceedings of the 44h Annual ACM Symposium on Theory of Computing (STOC), ACM, 2012, pp. 1235–1254.
[39] E. Boyle, S. Garg, A. Jain, Y.T. Kalai, A. Sahai, Secure Computation Against Adaptive Auxiliary Information, in: Advances in Cryptology–CRYPTO 2013, Springer, 2013, pp. 316–334.
[40] N. Bitansky, D. Dachman-Soled, H. Lin, Leakage-tolerant Computation with Input-independent Preprocessing, in: Proceedings of the International Cryptology Conference (CRYPTO), Springer, 2014, pp. 146–163.
[41] A. Shamir, How to share a secret, Commun. ACM 22 (11) (1979) 612–613.
[42] V. Attasena, J. Darmont, N. Harbi, Secret sharing for cloud data security: a survey, VLDB J. –Int. J. Very Large Data Bases (VLDBJ) 26 (5) (2017) 657–681.

[43] F. Benhamouda, A. Degwekar, Y. Ishai, T. Rabin, On the local leakage resilience of linear secret sharing schemes, in: Annual International Cryptology Conference, Springer, 2018, pp. 531–561.
[44] A. Srinivasan, P.N. Vasudevan, Leakage resilient secret sharing and applications(2018). https://eprint.iacr.org/2018/1154.pdf.
[45] D. Aggarwal, I. Damgard, J.B. Nielsen, M. Obremski, E. Purwanto, J. ao Ribeiro, M. Simkin, Stronger leakage-resilient and non-malleable secret-sharing schemes for general access structures, IACR eprint (2018). https://eprint.iacr.org/2018/1147.pdf
[46] C. Hu, Z. Li, P. Liu, R. Yang, S. Guo, H. Zhang, Verifiable public-key encryption with keyword search secure against continual memory attacks, Mobile Networks and Applications (2018) 1–11.
[47] S. Dai, H. Li, F. Zhang, Memory leakage-resilient searchable symmetric encryption, Future Generation Computer Systems 62 (2016) 76–84.
[48] X. Ge, J. Yu, H. Zhang, C. Hu, Z. Li, Z. Qin, R. Hao, Towards achieving keyword search over dynamic encrypted cloud data with symmetric-key based verification, IEEE Transactions on Dependable and Secure Computing (2019). 10.1109/TDSC.2019.2896258
[49] A. Lewko, M. Lewko, B. Waters, How to leak on key updates, in: Proceedings of the 43rd ACM Symposium on Theory of Computing, (STOC 2011), ACM, 2011, pp. 725–734.
[50] A. Ateniese, R. Burns, R. Curtmola, J. Herring, Provable data possession at untrusted stores, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, (CCS 2007), ACM, 2007, pp. 598–609.

Knowledge-Based Systems 255 (2022) 109727

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

WSNet: A local–global consistent traffic density estimation method based on weakly supervised learning

Ying-Xiang Hu [a], Rui-Sheng Jia [a,*], Yan-Bo Liu [b], Yong-Chao Li [a], Hong-Mei Sun [a]

[a] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[b] College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210096, China

ARTICLE INFO

Article history:
Received 19 April 2022
Received in revised form 29 July 2022
Accepted 16 August 2022
Available online 23 August 2022

Keywords:
Traffic density estimation
Weak supervision
Local–global consistency
Feature fusion

ABSTRACT

The available traffic density estimation methods mainly rely on label maps generated by manual annotation to train a network model and enhance the ability of the network model to extract traffic flow feature information. This fully supervised approach requires a lot of manual annotation, which is a time-consuming and laborious task. To solve this problem, this paper proposes a local–global consistent traffic density estimation method based on weakly supervised learning (WSNet). This method extracts the global traffic flow feature information through a trans-traffic module, solves the problem that the traffic flow feature cannot be fully extracted due to the limited receptive fields of convolutional neural networks (CNNs), extracts the local traffic flow feature information through a feedback module, enhances the local representation ability of the traffic flow feature information, and solves the problem of CNNs lacking inductive bias capability due to the use of a transformer only. This method extracts the global traffic flow feature information through the trans-traffic module and the local traffic flow feature information through the feedback module. In addition, a local–global consistency loss function (L_c) is added into the training process and combined with the L_1 loss function to strengthen the constraints on traffic density estimation, which effectively improves the accuracy of traffic density estimation. The experimental results show that this method significantly reduces the gap between fully supervised traffic density estimation and weakly supervised traffic density estimation, and the MAE and MSE values of this method are reduced to 4.33 and 5.82, respectively, on the TRANCOS dataset and to 3.90 and 5.8 on the VisDrone2019 Vehicle dataset.

1. Introduction

With the rapid development of society, an increasing number of people are choosing private cars as a means of transportation. The emergence of many private cars makes urban traffic congested, which is not conducive to the development of intelligent transportation. Therefore, to alleviate traffic congestion, accurate traffic density estimation has become increasingly important. To train a robust and reliable traffic density estimation network model, most traffic density estimation network models adopt a fully supervised sample training method, that is, the network model is trained on a label map generated by manual annotation. This approach requires considerable manpower and material and financial resources. For example, vehicle labeling for the TRANCOS dataset [1], which contains annotations for 46 786 vehicles, took 1000 h to complete. Therefore, the main goal of this paper is to omit manual annotation while maintaining good traffic density estimation performance. Specifically, this paper uses a weakly supervised method to estimate the traffic density and uses unmarked data to improve the robustness of the network model.

Traffic density estimation methods typically analyze the traffic information in video images. The available traffic density estimation methods mainly include methods based on full supervision [2–26], methods based on semisupervision [27–36] and methods based on weak supervision [37–44]. The methods based on full supervision mainly manually label the vehicles in video images to obtain a label map and then train a network model through the label map to improve the network model traffic density estimation performance. Although these methods show high performance in traffic density estimation, they require considerable manpower and material and financial resources to label the vehicles in the video images. Semisupervised methods can be divided into two forms: those that train a network model by labeling all the vehicles in some video images in the training set and those that train by labeling some vehicles in the video images in the training set. The performance of these methods in traffic density estimation is close to that of methods based on full supervision, and they show good robustness. However, these

* Corresponding author.
E-mail address: jrs716@163.com (R.-S. Jia).

https://doi.org/10.1016/j.knosys.2022.109727
0950-7051/

methods still need to label the vehicles in the video images, and the training process is very cumbersome.

To further reduce the workload of manual labeling, researchers have proposed a method based on weak supervision. This method does not need to manually label the vehicles in the video images and only needs annotations with count-level supervision to train a traffic density estimation model with good performance. With the rapid development of deep learning, an increasing number of scholars are committed to research on traffic density estimation methods using deep learning frameworks. To completely eliminate the cumbersome step of manual labeling, many traffic density estimation methods based on weak supervision have been proposed one after another for traffic density estimation method. In 2016, Borstel et al. [38] proposed a weakly supervised vehicle flow density estimation method based on a Gaussian process. In their work, all samples were annotated with count-level supervision on the training set. In 2019, Sam et al. [41] proposed a traffic density estimation method that uses sparse features to train a traffic density estimation model under almost unsupervised conditions. In 2021, Yang et al. [42] proposed a soft label sorting network, which can effectively return the traffic flow and obtain the result of traffic density estimation without location annotation. In 2022, Zheng et al. [37] proposed a new weakly supervised setting. They used the binary sorting of two images with high-contrast crowd counts for training guidance and converted the density estimation regression problem into a problem of sorting potential predictions in the process of training. Although the above methods have achieved good performance in traffic density estimation, there is still a performance gap compared with fully supervised traffic density estimation methods, and the above methods are all CNN-based traffic density estimation methods. In CNN-based traffic density estimation methods, the global context information cannot be modeled due to the limited receptive field of the convolution kernel; in addition, the CNN cannot establish the interactions between image blocks, so it cannot effectively learn the comparative characteristics between the traffic flow and background. Therefore, the neutral energy of the CNN's weakly supervised model in traffic flow density estimation is poor. In contrast to CNNs, transformers have a self-attention mechanism in image recognition and classification, so transformers can fully learn the comparative features between different image blocks and can effectively distinguish traffic flow feature information and background information. With the application of transformers in computer vision, Liang et al. [45] proposed a transformer-based crowd counting method (TransCrowd) for obtaining global feature information. However, Tian et al. [46] believe that TransCrowd only extracts limited features from one stage and performs counting regression using additional regression tokens and global average pooling. This design cannot make full use of the advantages of global attention. Therefore, Tian et al. proposed using the pyramid transformer trunk to capture different levels of global feature information and make full use of the advantages of global attention to obtain more comprehensive feature information. However, the above weakly supervised traffic density estimation method based on a transformer has no location annotation and lacks a local supervision signal, so the local feature representation is weakened. At the same time, due to the lack of inductive bias capability of CNNs, only using a transformer requires more parameters to achieve superior performance. In summary, transformer-based weakly supervised traffic density estimation methods and CNN-based weakly supervised traffic density estimation methods are complementary.

In view of the problems with the above methods, this paper proposes a local–global consistent traffic density estimation method based on weakly supervised learning. Compared with the existing traffic flow density estimation methods based on weak supervision, this method effectively combines the advantages of CNNs and transformers, enhances the representation of local features while giving attention to the global context information, and obtains higher accuracy in traffic flow density estimation. The main contributions of this paper are as follows:

(1) Because it is expensive to create a large-scale traffic flow dataset with location annotations, this paper designs a traffic density estimation method based on weakly supervised learning (WSNet). This method only needs counting-level supervised annotation to train a robust traffic density estimation network model, and the estimation accuracy is close to those of fully supervised methods.
(2) To solve the problem of variable vehicle scale in video images, we use the trans-traffic module to extract different levels of global traffic flow feature information and fuse the obtained global traffic flow feature information to make full use of the advantage of global attention. In addition, in the process of training, a local–global consistency loss function (L_c) is designed and added to enhance the feature extraction ability of the network model.
(3) To enhance the ability of the network model to extract traffic flow feature information and alleviate the problem of background interference, a new feedback module is designed in this paper. The module calculates the similarity between the extracted traffic flow feature information and unbiased feature estimates to obtain a probability map and then feeds the probability map back to the backbone network to improve the regional perception ability of the network model and accelerate the convergence of the network model.

2. Related work

Before deep learning was widely used in computer vision, the methods that were used for traffic density estimation mainly included the use of probe vehicles equipped with on-board kinematic instruments to detect the passage of vehicles with electronic labels through detectors installed at key locations in the road network and the use of satellite-based systems (such as the global positioning system (GPS)) [47–50]. The above methods can provide useful real-time information about vehicle location with high counting accuracy and are widely used in the fields of vehicle tracking, vehicle monitoring, vehicle scheduling and traffic flow estimation. However, these methods can only be applied to the location where the receiver is installed, and in most cases, they cannot monitor traffic flow information in real time. In addition, the various instruments used to interpret information are expensive and not suitable for wide application. With the rapid development of computer vision, to solve the above problems, many vehicle flow density estimation methods based on deep learning have been proposed.

2.1. Traffic density estimation methods based on full supervision

With the rapid development of deep learning, many traffic density estimation methods have emerged. In 2016, Zhang et al. [3] proposed a traffic flow detection method based on Fast R-CNN. This method takes the whole image and a group of object suggestions as input, generates the boundary box position, and outputs probability estimates of the object class [3]. In a scene with sparse traffic flow, it can quickly and accurately detect the traffic flow, but the scene counting performance under vehicle congestion is poor. With the wide application of density maps in the task of target counting, many methods based on density estimation have been proposed. Zhang et al. [11] proposed a multi-column convolution structure (MCNN) to solve the

problem of variable target scale in video images. Li et al. [16] designed a traffic density estimation network model (CSRNet) with VGG16 [51] as the backbone network, where the back end of the network model uses dilated convolution to expand the receptive field to extract detailed traffic flow characteristic information. Wang et al. [17] proposed a segmentation-guided attention network (SGANet) with Inception-v3 as the backbone network for the first time, which uses segmentation-guided attention and curvature loss functions to enhance the feature extraction ability. Liu et al. [20] proposed a multifaceted attention network (MAN) for extracting vehicle feature information of different scales.

The above fully supervised traffic density estimation methods have achieved high estimation accuracy in density estimation, but such methods rely on label maps generated by manual annotation to train the network model, which consumes many human, material and financial resources.

2.2. Traffic density estimation method based on weak supervision

To alleviate the consumption of human, material and financial resources caused by a large number of manual annotations, L2R [52] simplifies the counting task by sorting image patches. Wang et al. [46] introduced a synthetic group dataset called GCC, on which models are pretrained before being fine-tuned on real data. Although the above two methods do not use manual annotation, they still rely on point-level annotation, which is a fully supervised traffic density estimation method, and [52] needs to collect additional data information. [46] uses synthetic crowd scenes to train the network model, which has low robustness. Therefore, Meng et al. [35] proposed a semi supervised traffic density estimation method based on spatial uncertainty perception based on regularized agent tasks (binary segmentation). This method uses the spatial uncertainty perception teacher–student framework to focus on the information of high confidence areas and processes the noise supervision from unmarked data in an end-to-end manner. Lei et al. [34] trained the network model on a small number of point-level annotations (fully supervised) and a large number of count-level annotations (weakly supervised), that is, by using multiple density map estimation branches and asymmetric training strategies to enforce the consistency between the predicted density map and the total number of objects to improve the accuracy of traffic flow density estimation. However, this semi supervised traffic density estimation method only reduces the number of location annotations and still requires point-level annotations, and the training process is complex. To completely eliminate the cumbersome step of manual annotation, Sam et al. [41] used a grid winner take all (GWTA) automatic encoder to train a traffic density estimation model under almost unsupervised conditions. Yang et al. [42] proposed a density estimation network based on weak supervision, which can directly return the traffic flow without location annotation and train a network model to estimate the traffic flow density through the relationship between images. Liang et al. [43] first proposed a transformer-based density estimation method (TransCrowd), which uses a transformer to obtain the global context traffic flow characteristic information while significantly distinguishing between traffic flow information and background information. Tian et al. [52] proposed using the pyramid transformer trunk to capture different levels of global crowd feature information, making full use of the advantages of global attention to make up for TransCrowd's inability to obtain global traffic flow feature information in only one stage.

Benefiting from the above research, this paper designs a local–global consistent traffic flow density estimation method based on weakly supervised learning for the traffic flow density estimation task. Whereas CNNs have the ability to extract local features and use inductive bias, transformers have the advantages of extracting global features and significantly distinguishing traffic flow feature information and background information. This paper uses the advantages of CNNs and transformers to extract local and global traffic flow feature information, respectively, to enhance the accuracy of traffic flow density estimation. To enhance the representation ability of the network model, a feedback module is designed, which feeds back a probability map with traffic flow information to the backbone network, improves the accuracy of traffic density estimation, and accelerates the convergence of the network model. In addition, this paper designs and adds a local–global consistency loss function (L_c), which uses the consistency between global and local results to improve the performance of weakly supervised traffic density estimation.

3. Method

To reduce the consumption of human, material and financial resources due to the performance of many manual annotations, a vehicle density estimation method based on weakly supervised learning is designed in this paper. Specifically, to solve the problem of variable vehicle scale in video images, the trans-traffic module is used, which obtains the global traffic flow characteristic information of different stages and then fuses the global traffic flow characteristic information of each stage, making full use of the advantage of global attention. To enhance the ability of the network model to extract vehicle feature information and alleviate the problem of background interference, a feedback module is designed. The module calculates the similarity between the extracted vehicle flow feature information and unbiased feature estimation results to obtain a probability map and then feeds back the probability map to the backbone network to enhance the vehicle feature information and weaken the background information. The existing density estimation methods based on weak supervision often only consider the global loss and ignore the local characteristic information. Therefore, in the training process, this paper designs and adds a local–global consistency loss function (L_c) to strengthen the constraints on traffic density estimation and effectively improve the accuracy of traffic density estimation. The network structure is illustrated in Fig. 1.

As shown in Fig. 1, WSNet is mainly composed of a backbone network, trans-traffic module and feedback module. First, a video image is input into the backbone network to extract different levels of traffic flow feature information. Second, these different levels of traffic flow characteristic information are processed by the trans-traffic module to obtain the global traffic flow characteristic information of each stage, and the global traffic flow characteristic information of each stage is fused to obtain more comprehensive global traffic flow characteristic information f_1. At the same time, the traffic flow feature information extracted by the backbone network is processed by the feedback module to obtain a probability map P. P is fed back to the backbone network to enhance the feature extraction ability of the network model. In addition, P is multiplied pixel by pixel with the traffic flow feature information f_0 to obtain the enhanced local traffic flow feature information f_2. Then, f_1 and f_2 are fused to give attention to the correlation between local and global traffic flow feature information, and the traffic flow feature information f_3 is obtained. Finally, we flatten f_3 into a linear regression vector to construct the input of the linear regression network.

3.1. Trans-traffic module

In the task of weakly supervised traffic density estimation based on a CNN, the global context information cannot be modeled due to the limited receptive field of the convolution kernel.

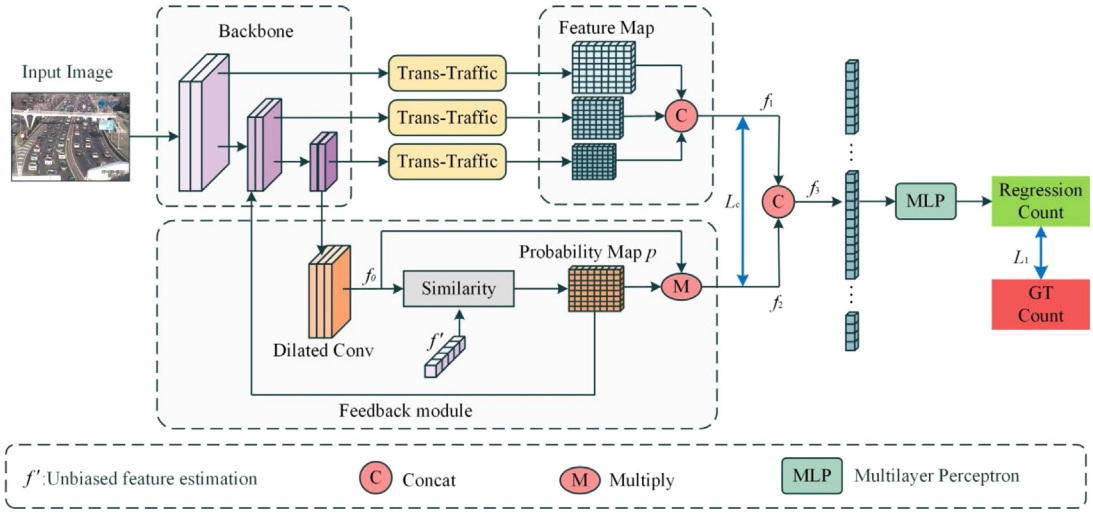


Fig. 1. WSNet. In the network, Concat denotes fusion of the traffic flow features according to the channel connection, and Multiply denotes multiplication of the traffic flow feature map and probability map pixel by pixel.

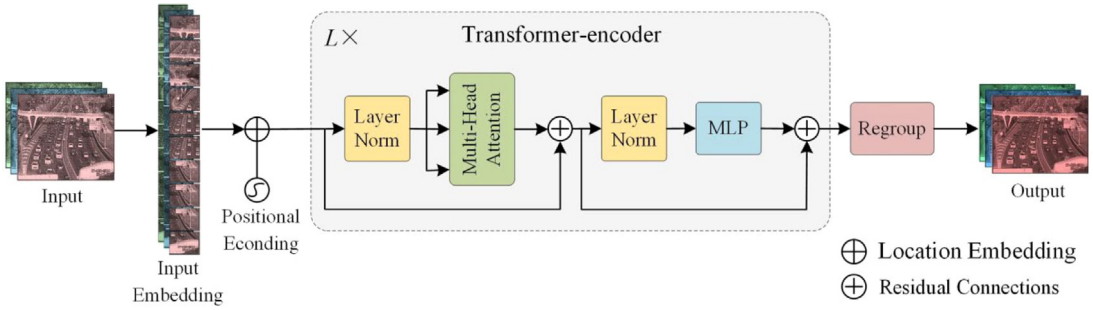


Fig. 2. Trans-traffic module. In this module, $L\times$ indicates that L cycles have passed.

Therefore, we use the trans-traffic module to extract the global traffic flow feature information of different stages at the same time and make full use of the advantages of global attention to overcome the disadvantage of extracting limited feature information from only one stage in ViT [45]. Its structure is illustrated in detail in Fig. 2.

As shown in Fig. 2, the main working steps of the trans-traffic module are as follows:

Step 1: The input feature map $I \in R^{H\times W\times D}$ is cut into N patches of the same size to obtain a patch set $\left\{x_p^i \in R^{K^2\cdot 3} | i = 1, \ldots, N\right\}$, where D represents the number of channels of the input feature map and $N = \frac{H}{K} \times \frac{W}{K}$ and K are the predefined patch sizes.

Step 2: We map x to potential D-dimensional embedded features through a learnable projection to obtain a feature map e:

$$e = [e_1; e_2; \cdots; e_N] = [x_p^1 E; x_p^2 E; \cdots x_p^N E], E \in R^{(K^2\cdot 3)}, \quad (1)$$

where E is a learnable matrix and $e \in R^{N\times D}$. Then, we embed position $\left\{p_i \in R^D | i = 1, \ldots, N\right\}$ into e to obtain the input Z_0 of the transformer-encoder. Z_0 is defined as

$$Z_0 = [Z_0^1; Z_0^2; \cdots Z_0^N] = [e_1 + p_1; e_2 + p_2; \cdots; e_N + p_N] \quad (2)$$

Step 3: Z_0 is input into the transformer-encoder module, which contains L layers. Each layer consists of layer normalization (LN), multihead self-attention (MSA), residual connections and multilayer perception (MLP) blocks. Therefore, the output of the transformer-encoder module can be expressed as

$$Z_l' = MSA(LN(Z_{l-1})) + Z_{l-1}, \quad (3)$$

$$Z_l = MLP(LN(Z_l')) + Z_l' \quad (4)$$

where Z_{l-1} is the input of layer l; Z_l' is the intermediate result of layer l, that is, the result of passing through the first LN and MSA and the remaining connection with the input of layer l; and Z_l is the output of layer l. MLP contains two linear layers with the GELU [53] activation function. The first linear layer expands the dimension of the feature embedding from D to 4D, and the second layer reduces the dimension from 4D to D.

3.2. Feedback module

It is not enough to only use transformers to model the global context traffic flow feature information. Due to the CNN's lack of local feature extraction ability and inductive bias ability, using only transformers requires more parameters to achieve better performance. At the same time, to suppress the influence of complex backgrounds in video images on the traffic density estimation results and enhance the ability of the network model to extract traffic flow feature information, a feedback module is designed. In this module, we propose an adaptive similarity learning strategy to enhance the representation of local traffic flow feature information. The feedback module mainly enhances the representation ability of traffic flow characteristic information through two steps.

Step 1: The traffic flow in a video image is classified into a positive category, and its unbiased feature estimate can be

expressed as

$$f' = \underset{f}{argmax} \sum_{i=1}^{M} p_i sim(f_i, f), \tag{5}$$

where M is the number of pixels; f_i and p_i are the eigenvectors and the probability of being classified into a positive class at the ith position, respectively; and $sim()$ is a feature similarity function. However, in the weak supervision-based traffic density estimation method without local annotation, the value of p_i is difficult to determine. Therefore, in this network model, we set the unbiased feature estimate f' as a learnable vector and update it through gradient descent and optimization of the network parameters. p_i is defined as

$$p_i = \lambda_1 sim(f_i, f'), \tag{6}$$

where λ_1 is a positive hyperparameter. We use *cosine* similarity as the feature similarity function and normalize it to the range of [0,1], which is calculated as follows:

$$sim(f_i, f') = \frac{f_i \odot f'}{2\,|f_i|\,|f'|} + 0.5, \tag{7}$$

where $\odot$ is the inner product of the vector. After obtaining the unbiased feature estimate f' and calculating the similarity of the traffic flow feature map f_0, a probability map P is obtained.

Step 2: The probability map P is fed back to the backbone network, and the traffic flow characteristic map of the backbone network is multiplied by the probability map P pixel by pixel to enhance the traffic flow characteristic representation ability of the backbone network. The enhanced traffic flow characteristics can be expressed as

$$f_z = \underset{f}{argmax} \sum_{i=1}^{M} p_i f_i, \tag{8}$$

where M is the number of pixels, p_i represents the probability that the ith position in the probability graph P is divided into positive classes, and f_i represents the traffic flow characteristic information of the ith position in the characteristic graph. Similarly, the traffic flow feature f_0 is multiplied by the probability map P pixel by pixel to obtain the traffic flow feature f_2 to enhance the local traffic flow feature information. f_2 can be expressed as

$$f_2 = \underset{f}{argmax} \sum_{i=1}^{M} p_i f_i \tag{9}$$

3.3. Loss function

The count loss (L_n) focuses only on the difference between the predicted traffic flow and the real traffic flow, but in practical applications, the phenomena of miscalculation and missed estimation will inevitably occur. However, L_n cannot give attention to the occurrence of these phenomena. Therefore, this paper designs a new loss function — the local–global consistency loss function (L_c), which normalizes the network model by calculating the difference between the traffic flow estimated by the feedback module and the traffic flow estimated by the trans-traffic module. When calculating L_c, this paper divides the characteristic map obtained by the feedback module into 4^l regions of equal size (according to [1], l is set to 2 in this paper.), calculates the estimated traffic flow of each region, and then sums the traffic flows over all regions to obtain the total traffic flow. Finally, it compares the total traffic flow with the traffic flow estimated by the trans-traffic module to standardize the network model.

To give close attention to the correlation between local and global traffic flow characteristic information, this paper uses both the L_n loss function and the local–global consistency loss function (L_c). The L_n loss function is defined as follows:

$$L_n = \frac{1}{N} \sum_{i=1}^{N} |pc_i - gt_i|, \tag{10}$$

where N represents the number of input video images, pc_i represents the predicted traffic flow in the ith video image, and gt_i represents the real traffic flow in the ith video image.

When calculating the L_c loss function, we divide the characteristic map processed by the feedback network into 4^l regions, calculate the traffic flow of each region and add the traffic flow of each region. At the same time, we calculate the total traffic flow processed by the trans-traffic module. Finally, we calculate the difference between the two. The L_c loss function is defined as follows:

$$L_c = \frac{1}{N} \sum_{n=1}^{N} \left\| \sum_{l=1}^{4^l} pc_{l,i}^1 - pc_i^2 \right\|^2, \tag{11}$$

where N represents the number of input video images, $pc_{l,i}^1$ represents the predicted traffic flow of the lth area in the ith video image processed by the feedback network, and pc_i^2 represents the predicted traffic flow in the ith video image processed by the trans-traffic module.

Therefore, the total loss function (L) of this method is defined as follows:

$$L = L_n + \lambda L_c, \tag{12}$$

where λ is a parameter, which is set to 1 in this paper.

4. Experimental

4.1. Experimental setup

Dataset: In this work, we perform many experiments on the TRANCOS dataset [1] and the VisDrone2019 Vehicle dataset [54]. Different from fully supervised methods, we only use count-level labels as supervision information in the training process. The video images in the TRANCOS dataset were captured by a road surveillance camera, and they exhibit drastic scale changes compared with the images in the VisDrone2019 Vehicle dataset. The video images in the VisDrone2019 Vehicle dataset were captured by an unmanned aerial vehicle (UAV) and have severe background interference compared with the images in the TRANCOS dataset. Video images from the two datasets are shown in Fig. 3.

Training details and hyperparameters: In the training process, we use the Adam optimizer. The batch size is set to 1, the initial learning rate is set to $5*10^{-4}$, which is reduced by 0.5 times every 50 stages, and the total number of training cycles is set to 300. The experimental environment is presented in Table 1.

Table 1
Experimental environment.

Name	Parameter
System	Windows 10
Frame	TensorFlow
Language	Python
CPU	Intel(R) Core(TM) i5-9300H CPU @ 2.40 GHz
GPU	NVIDIA GEFORCE GTX 1050
RAM	8.00 GB

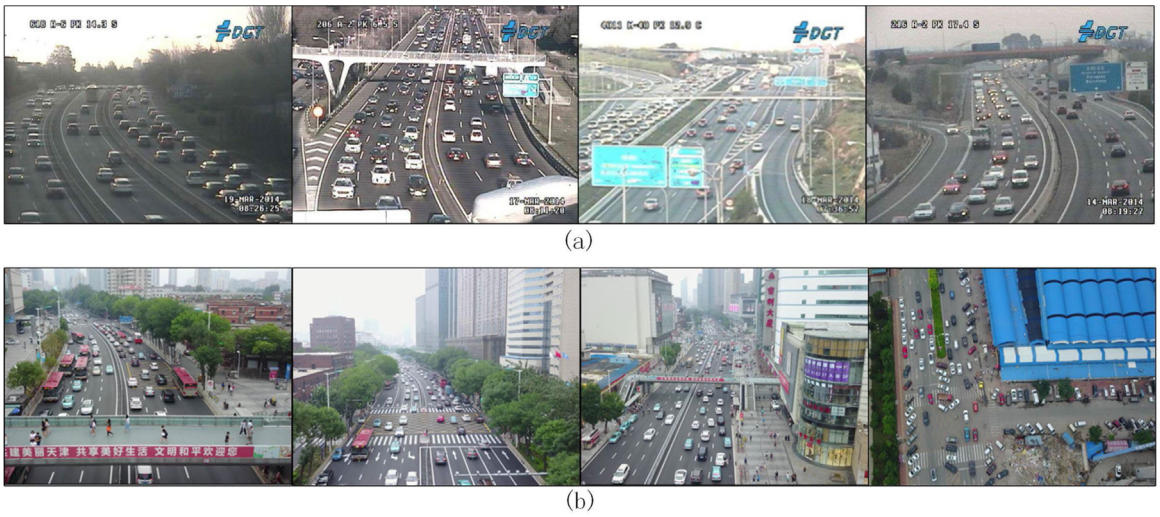


Fig. 3. Images from the (a) TRANCOS dataset and (b) VisDrone2019 Vehicle dataset.

4.2. Evaluation indicators

This paper uses the mean absolute error (MAE) and mean square error (MSE) as evaluation indices to evaluate the performance of the network model. MAE reflects the accuracy of the network model, and MSE reflects the stability of the network model. MAE and MSE are defined as follows:

$$MAE = \frac{1}{N}\sum_{i}^{N}|pc_i - gt_i|, \tag{13}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i}^{N}|pc_i - gt_i|^2}, \tag{14}$$

where N represents the number of input video images, pc_i represents the estimated traffic flow in the ith video image, and gt_i represents the actual traffic flow in the ith video image.

4.3. Experimental results

To show the accuracy of this method in estimating traffic density, we convert the final output of the network model into a density map. As shown in Fig. 4, the density map generated by this method is almost consistent with the density distribution of the traffic flow in the real scene, and there is little difference between the predicted traffic flow and the actual traffic flow. In the second video image of the TRANCOS dataset, the traffic flow predicted by this method is the same as the actual traffic flow. In the second video image of the VisDrone2019 Vehicle dataset, the difference between the predicted traffic flow and the actual traffic flow is 1. Therefore, this method accurately estimates the traffic flow in the surveillance video image, and there is almost no difference between the estimated traffic flow and the real traffic flow.

To verify the effectiveness of the method proposed in this paper, we perform many experiments on the TRANCOS dataset and VisDrone2019 Vehicle dataset and compare the experimental results with those of the most advanced methods at present. The optimal results are shown in bold and red. The experimental results are presented in Tables 2 and 3.

Table 2 presents the comparison results between this method and the most advanced vehicle density estimation methods on the TRANCOS dataset. Among them, the methods using both point-level annotation and count-level annotation are the vehicle density estimation methods based on full supervision, and the methods using only count-level annotation are the vehicle density estimation methods based on weak supervision. The comparison results show that on the TRANCOS dataset, compared with the optimal MAE and MSE values of the current advanced traffic density estimation methods based on full supervision, the MAE and MSE values of this method are only 0.53 and 0.94 higher. The MAE and MSE values of this method are better than those of MCNN and CSRNet. The MAE and MSE values of this method are better than those of the current advanced traffic density

Table 2
Experimental results on the TRANCOS dataset, where location refers to training the network model with point-level annotation, and traffic number refers to training the network model with count-level annotation.

Method	Year	Training labels		TRANCOS	
		Location	Traffic number	MAE	MSE
MCNN [11]	CVPR16	✓	✓	10.71	13.55
CSRNet [16]	CVPR18	✓	✓	6.01	7.78
ASNet [12]	CVPR20	✓	✓	**3.80**	4.92
DM-Count [18]	NeurIPS20	✓	✓	3.89	5.22
P2PNet [19]	ICCV21	✓	✓	3.81	**4.88**
STNet [26]	TMM22	✓	✓	3.82	4.96
Ours	–	×	✓	4.33	5.82
MATT [34]	PR21	×	✓	4.77	6.66
TransCrowd [43]	arxiv21	×	✓	4.52	6.47
CCTrans [52]	arxiv21	×	✓	4.47	6.38
Ours	–	×	✓	**4.33**	**5.82**

Table 3
Experimental results on the VisDrone2019 Vehicle dataset.

Method	Year	Training labels		VisDrone2019 Vehicle	
		Location	Traffic number	MAE	MSE
MCNN [11]	CVPR16	✓	✓	8.41	10.55
CSRNet [16]	CVPR18	✓	✓	6.91	8.89
ASNet [12]	CVPR20	✓	✓	3.88	5.72
DM-Count [18]	NeurIPS20	✓	✓	3.91	5.69
P2PNet [19]	ICCV21	✓	✓	**3.83**	**4.92**
STNet [26]	TMM22	✓	✓	3.82	4.96
Ours	–	×	✓	3.90	5.81
MATT [34]	PR21	×	✓	4.63	6.96
TransCrowd [43]	arxiv21	×	✓	4.37	6.36
CCTrans [52]	arxiv21	×	✓	4.13	6.18
Ours	–	×	✓	**3.90**	**5.81**

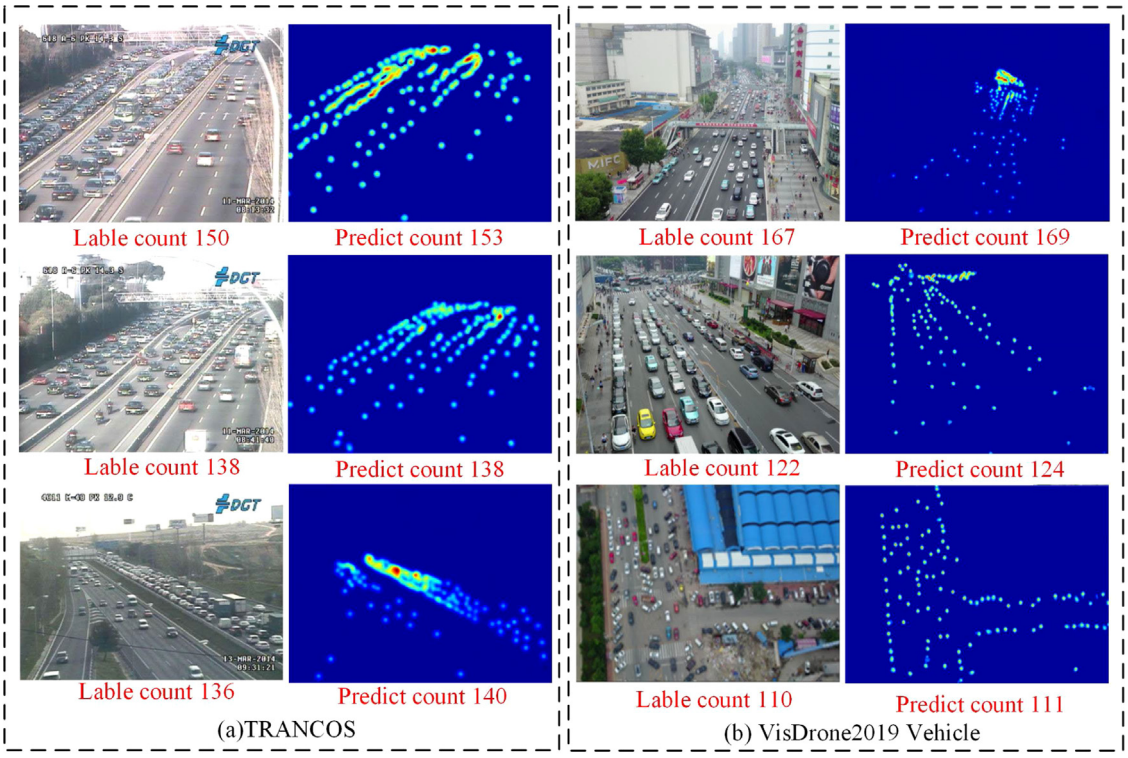


Fig. 4. Predicted density maps. (a) The density maps generated for the TRANCOS dataset and (b) the density maps generated for the VisDrone2019 Vehicle dataset.

estimation methods based on weak supervision. Therefore, this method can accurately estimate the traffic density and effectively solve the problem of variable vehicle scale in video images. Its estimation results are close to those of the traffic density estimation methods based on full supervision.

Table 3 presents the comparison results between this method and the most advanced vehicle density estimation methods on the VisDrone2019 Vehicle dataset. According to the comparison results, on the VisDrone2019 Vehicle dataset, compared with the optimal MAE and MSE values of the current advanced traffic density estimation methods based on full supervision, the MAE and MSE values of this method are only 0.07 and 0.89 higher. The MAE value of this method is better than those of the MCNN, CSRNet and DM-Count, and the MSE value of this method is better than those of the MCNN and CSRNet. Compared with the optimal MAE and MSE values of the current advanced traffic density estimation methods based on weak supervision, the MAE and MSE values of this method are the best. Therefore, this method can effectively estimate the traffic density, and the estimation results are close to those of the traffic density estimation methods based on full supervision. This method effectively alleviates the problem of reduced traffic density estimation accuracy due to background interference.

5. Discussion

5.1. Ablation experiment on each module

To verify the effectiveness of the trans-traffic module and feedback module and the acceleration of the convergence of the network model by the feedback module, we conduct ablation experiments on the TRANCOS dataset and VisDrone2019 Vehicle dataset. The experimental results are presented in Table 4 and Fig. 5.

According to Table 4, when only the backbone network is used, the MAE and MSE values of the method in this paper reach 14.24 and 18.33, respectively, on the TRANCOS dataset and 14.13 and 18.21 on the VisDrone2019 Vehicle dataset, and the traffic density estimation results are not ideal. When the backbone network and trans-traffic module are used, MAE and MSE on the TRANCOS dataset and VisDrone2019 Vehicle dataset are significantly reduced, which proves the effectiveness of the trans-traffic module in traffic density estimation. When the backbone network and feedback module are used, MAE and MSE on the TRANCOS dataset and VisDrone2019 Vehicle dataset are significantly lower than those of only the backbone network, which proves the effectiveness of the feedback module in traffic flow density estimation. When the backbone network, trans-traffic module and feedback module are used at the same time, MAE and MSE on the TRANCOS dataset and VisDrone2019 Vehicle dataset are the best. Therefore, the backbone network, trans-traffic module and feedback module can be used at the same time to effectively estimate the traffic density.

Table 4
Experimental results on effectiveness of the trans-traffic module and feedback module.

Method	TRANCOS		VisDrone2019 Vehicle	
	MAE	MSE	MAE	MSE
Backbone	14.24	18.33	14.13	18.21
Backbone+Trans-Traffic	6.34	8.47	5.98	8.03
Backbone+Feedback	9.25	12.23	9.31	13.10
Backbone+Trans-Traffic +Feedback	**4.33**	**5.82**	**3.90**	**5.81**

In Fig. 5, (a) represents the results of the experiment on the TRANCOS dataset, and (b) represents the results of the experiment on the VisDrone2019 Vehicle dataset. The experimental results show that when the trans-traffic module is added to the network model, the MAE value decreases by half after training for approximately 50 epochs and gradually smooths after training for approximately 100 epochs. When the trans-traffic module is not added to the network model, the MAE value decreases by

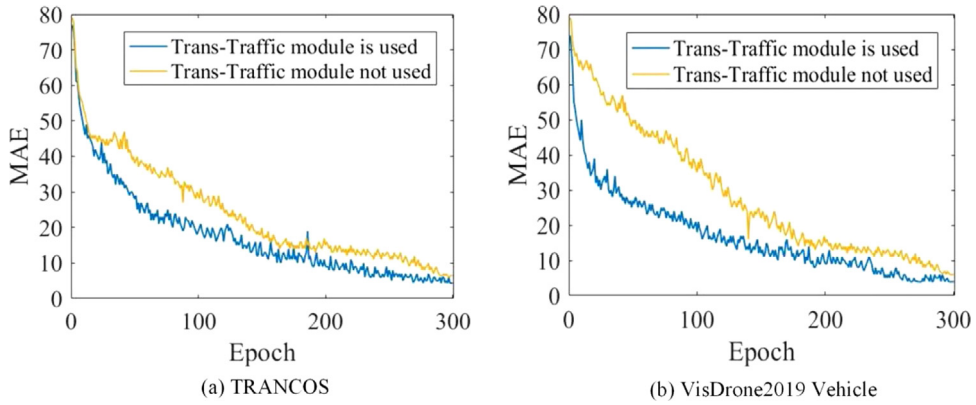


Fig. 5. Convergence trend of the network model. In the figure, the abscissa represents the training time, the ordinate represents the value of MAE, the yellow curve represents the convergence trend of MAE without the trans-traffic module, and the blue curve represents the convergence trend of MAE after using the trans-traffic module.

Table 5
Experimental results on the effectiveness of the loss function (the best method in the results is marked in bold.).

Method	TRANCOS		VisDrone2019 Vehicle	
	MAE	MSE	MAE	MSE
L_n	11.34	14.45	11.31	13.91
L_c	8.16	8.65	7.53	8.61
$L_n + L_c$	**4.33**	**4.82**	**3.90**	**4.81**

half after training for approximately 110 epochs and gradually smooths after training for approximately 200 epochs. Therefore, adding the trans-traffic module to the network model accelerates the convergence of the network model, and the final MAE value is relatively small.

5.2. Ablation experiment on the loss function

To verify the effectiveness of using the sum of the counting loss function (L_n) and the local–global consistency loss function (L_c) as the loss function of this method, an ablation experiment is conducted in which only the loss function is changed, while other conditions remain unchanged. The experimental results on the TRANCOS dataset and the VisDrone2019 vehicle dataset are presented in Table 5.

6. Conclusions and future work

The fully supervised traffic density estimation method requires a large number of label graphs generated by manual annotation to train the network model, which consumes considerable manpower, material and financial resources. To solve this problem, this paper designs a local–global consistent traffic density estimation method based on weakly supervised learning. In this method, the global traffic flow feature information in different stages is extracted through a trans-traffic module, and the local traffic flow feature information is extracted through a feedback module. Then, the extracted global traffic flow feature information and local traffic flow feature information are fused to obtain more comprehensive traffic flow feature information. In addition, when training the network model, we use both the L_1 loss and the local global consistency loss function (L_c) to obtain the correlation of local and global traffic flow characteristic information and strengthen the regional perceptive ability of the network model.

The task of traffic density estimation is mostly applied in embedded platforms such as surveillance cameras and UAVs. In our next work, we will focus on the development of a lightweight traffic density estimation method based on weak supervision.

CRediT authorship contribution statement

Ying-Xiang Hu: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Rui-Sheng Jia:** Supervision, Methodology, Resources, Writing – review & editing. **Yan-Bo Liu:** Data curation, Investigation, Visualization. **Yong-Chao Li:** Data curation, Investigation, Visualization. **Hong-Mei Sun:** Supervision, Methodology, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful for collaborative funding support from the Humanities and Social Science Fund of Ministry of Education of the People's Republic of China (Research on abnormal behavior perception, recognition and early warning model of dense population from the perspective of video surveillance, No. 21YJAZH077).

References

[1] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, D. Onoro-Rubio, Extremely overlapping vehicle counting, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, Cham, 2015, pp. 423–431, http://dx.doi.org/10.1007/978-3-319-19390-8_48.

[2] V. Lempitsky, A. Zisserman, Learning to count objects in images, Adv. Neural Inf. Process. Syst. (2010) 23.

[3] Z. Zhang, K. Liu, F. Gao, X. Li, G. Wang, Vision-based vehicle detecting and counting for traffic flow analysis, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 2267–2273, http://dx.doi.org/10.1109/IJCNN.2016.7727480.

[4] M.A. Abdelwahab, Fast approach for efficient vehicle counting, Electron. Lett. 55 (1) (2019) 20–22, http://dx.doi.org/10.1049/el.2018.6719.

[5] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen, Q. Liu, A comparative study of state-of-the-art deep learning algorithms for vehicle detection, IEEE Intell. Transp. Syst. Mag. 11 (2) (2019) 82–95, http://dx.doi.org/10.1109/MITS.2019.2903518.

[6] H. Song, H. Liang, H. Li, Z. Dai, X. Yun, Vision-based vehicle detection and counting system using deep learning in highway scenes, Eur. Transp. Res. Rev. 11 (1) (2019) 1–16, http://dx.doi.org/10.1186/s12544-019-0390-4.

[7] W. Li, Z. Wang, X. Wu, J. Zhang, Q. Peng, H. Li, CODAN: Counting-driven attention network for vehicle detection in congested scenes, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 73–82, http://dx.doi.org/10.1145/3394171.3413945.

[8] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 833–841, http://dx.doi.org/10.1109/CVPR.2015.7298684.
[9] X. Shi, X. Li, C. Wu, S. Kong, J. Yang, L. He, A real-time deep network for crowd counting, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 2328–2332), http://dx.doi.org/10.1109/ICASSP40776.2020.9053780.
[10] W. Xie, J.A. Noble, A. Zisserman, Microscopy cell counting and detection with fully convolutional regression networks, Comput. Methods Biomech. Biomed. Eng.: Imaging Vis. 6 (3) (2018) 283–292, http://dx.doi.org/10.1080/21681163.2016.1149104.
[11] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (, 2016, pp. 589–597, http://dx.doi.org/10.1109/CVPR.2016.70.
[12] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou …, Y. Pang, Attention scaling for crowd counting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4706–4715, http://dx.doi.org/10.1109/CVPR42600.2020.00476.
[13] D.Babu. Sam, S. Surya, R.Venkatesh. Babu, Switching convolutional neural network for crowd counting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5744–5752, http://dx.doi.org/10.48550/arXiv.1708.00199.
[14] D. Onoro-Rubio, R.J. López-Sastre, Towards perspective-free object counting with deep learning, in: European Conference on Computer Vision, Springer, Cham, 2016, pp. 615–629, http://dx.doi.org/10.1007/978-3-319-46478-7_38.
[15] S. Zhang, G. Wu, J.P. Costeira, J.M. Moura, Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras, in: Proceedings of the IEEE International Conference on Computer Vision (, 2017, pp. 3667–3676, http://dx.doi.org/10.1109/iccv.2017.39.6.
[16] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1091–1100, http://dx.doi.org/10.1109/CVPR.2018.00120.
[17] Q. Wang, T.P. Breckon, Crowd counting via segmentation guided attention networks and curriculum loss, IEEE Trans. Intell. Transp. Syst. (2022) http://dx.doi.org/10.48550/arXiv.1911.07990.
[18] Y.B. Liu, R.S. Jia, J.T. Yu, R.N. Yin, H.M. Sun, Crowd density estimation via a multichannel dense grouping network, Neurocomputing 449 (2021) 61–70, http://dx.doi.org/10.1016/j.neucom.2021.03.078.
[19] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang ., Y. Wu, Rethinking counting and localization in crowds: A purely point-based framework, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3365–3374, http://dx.doi.org/10.48550/arXiv.2107.12746.
[20] H. Lin, Z. Ma, R. Ji, Y. Wang, X. Hong, Boosting crowd counting via multifaceted attention, 2022, http://dx.doi.org/10.48550/arXiv.2203.02636, arXiv preprint arXiv:2203.02636.
[21] M.H. Oh, P. Olsen, K.N. Ramamurthy, Crowd counting with decomposed uncertainty, in: Proceedings of the AAAI Conference on Artificial Intelligence Vol, vol. 34, 2020, pp. 11799–11806, http://dx.doi.org/10.1609/aaai.v34i07.6852, (07).
[22] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, L. Shao, Crowd counting and density estimation by trellis encoder–decoder networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6133–6142, http://dx.doi.org/10.48550/arXiv.1903.00853.
[23] S. Zhang, G. Wu, J.P. Costeira, J.M. Moura, Understanding traffic density from large-scale web camera data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5898–5907, http://dx.doi.org/10.48550/arXiv.1703.05868.
[24] Z. Ma, X. Wei, X. Hong, Y. Gong, Bayesian loss for crowd count estimation with point supervision, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6142–6151, http://dx.doi.org/10.48550/arXiv.1908.03684.
[25] M. Hossain, M. Hosseinzadeh, O. Chanda, Y. Wang, Crowd counting using scale-aware attention networks, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 1280–1288, http://dx.doi.org/10.1109/WACV.2019.00141.
[26] M. Wang, H. Cai, X. Han, J. Zhou, M. Gong, Stnet: Scale tree network with multi-level auxiliator for crowd counting, IEEE Trans. Multimed. (2022) http://dx.doi.org/10.48550/arXiv.2012.10189.
[27] X. Liu, J. Van De Weijer, A.D. Bagdanov, Exploiting unlabeled data in cnns by self-supervised learning to rank, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1862–1878, http://dx.doi.org/10.1109/TPAMI.2019.2899857.
[28] Q. Wang, J. Gao, W. Lin, Y. Yuan, Learning from synthetic data for crowd counting in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8198–8207, http://dx.doi.org/10.1109/CVPR.2019.00839.
[29] G. Olmschenk, J. Chen, H. Tang, Z. Zhu, Dense crowd counting convolutional neural networks with minimal data using semi-supervised dual-goal generative adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition: Learning with Imperfect Data Workshop, 2019, http://dx.doi.org/10.5220/0009156201850195.
[30] Q. Wang, J. Gao, W. Lin, Y. Yuan, Pixel-wise crowd understanding via synthetic data, Int. J. Comput. Vis. 129 (1) (2021) 225–245, http://dx.doi.org/10.1007/s11263-020-01365-4.
[31] Z. Zhao, M. Shi, X. Zhao, L. Li, Active crowd counting with limited supervision, in: European Conference on Computer Vision, Springer, Cham, 2020, pp. 565–581, http://dx.doi.org/10.1007/978-3-030-58565-5_34.
[32] Y. Liu, L. Liu, P. Wang, P. Zhang, Y. Lei, Semi-supervised crowd counting via self-training on surrogate tasks, in: European Conference on Computer Vision, Springer, Cham, 2020, pp. 242–259, http://dx.doi.org/10.1007/978-3-030-58555-6_15.
[33] V.A. Sindagi, R. Yasarla, D.S. Babu, R.V. Babu, V.M. Patel, Learning to count in the crowd from limited labeled data, in: European Conference on Computer Vision, Springer, Cham, 2020, pp. 212–229, http://dx.doi.org/10.1007/978-3-030-58621-8_13.
[34] Y. Lei, Y. Liu, P. Zhang, L. Liu, Towards using count-level weak-supervised for crowd counting, Pattern Recognit. 109 (2021) 107616, http://dx.doi.org/10.1016/j.patcog.2020.107616.
[35] Y. Meng, H. Zhang, Y. Zhao, X. Yang, X. Qian, X. Huang, Y. Zheng, Spatial uncertainty-aware semi-supervised crowd counting, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15549–15559.
[36] S. Khaki, H. Pham, Y. Han, A. Kuhl, W. Kent, L. Wang, Deepcorn: A semi-supervised deep learning method for high-throughput image-based corn kernel counting and yield estimation, Knowl.-Based Syst. 218 (2021) 106874, http://dx.doi.org/10.1016/j.knosys.2021.06874.
[37] Z. Xiong, L. Chai, W. Liu, Y. Liu, S. Ren, S. He, Glance to count: Learning to rank with anchors for weakly-supervised crowd counting, 2022, http://dx.doi.org/10.48550/arXiv.2205.14659, arXiv preprint arXiv:2205.14659.
[38] M.V. Borstel, M. Kandemir, P. Schmidt, M.K. Rao, K. Rajamani, F.A. Hamprecht, Gaussian process density counting from weak-supervised, in: European Conference on Computer Vision, Springer, Cham, 2016, pp. 365–380, http://dx.doi.org/10.1007/978-3-319-46448-0_22.
[39] M.V. Borstel, M. Kandemir, P. Schmidt, M.K. Rao, K. Rajamani, F.A. Hamprecht, Gaussian process density counting from weak-supervised, in: European Conference on Computer Vision, Springer, Cham, 2016, pp. 365–380, http://dx.doi.org/10.1093/nsr/nwx106.
[40] E. Bellocchio, T.A. Ciarfuglia, G. Costante, P. Valigi, Weak-supervised fruit counting for yield estimation using spatial consistency, IEEE Robot. Autom. Lett. 4 (3) (2019) 2348–2355, http://dx.doi.org/10.1109/LRA.2019.2903260.
[41] D.B. Sam, N.N. Sajjan, H. Maurya, R.V. Babu, Almost unsupervised learning for dense crowd counting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8868–8875, http://dx.doi.org/10.1609/aaai.v33i01.33018868, (01).
[42] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, N. Sebe, Weakly-supervised crowd counting learns from sorting rather than locations, in: European Conference on Computer Vision, Springer, Cham, 2020, pp. 1–17, http://dx.doi.org/10.1007/978-3-030-58598-3_1.
[43] D. Liang, X. Chen, W. Xu, Y. Zhou, X. Bai, Transcrowd: Weakly-supervised crowd counting with transformer, 2021, http://dx.doi.org/10.48550/arXiv.2104.09116, arXiv preprint arXiv:2104.09116.
[44] Y. Tian, X. Chu, H. Wang, Cctrans: Simplifying and improving crowd counting with transformer, 2021, http://dx.doi.org/10.48550/arXiv.2109.14483, arXiv preprint arXiv:2109.14483.
[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, 2020, http://dx.doi.org/10.48550/arXiv.2010.11929, arXiv preprint arXiv:2010.11929.
[46] Q. Wang, J. Gao, W. Lin, Y. Yuan, Learning from synthetic data for crowd counting in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8198–8207, http://dx.doi.org/10.1109/CVPR.2019.00839.
[47] R. Zito, G. D'este, M.A.P. Taylor, Global positioning systems in the time domain: How useful a tool for intelligent vehicle-highway systems? Transp. Res. C 3 (4) (1995) 193–209, http://dx.doi.org/10.1016/0968-090X(95)00006-5.
[48] R.S. Nerem, K.M. Larson, Global positioning system, theory and practice, 2001, http://dx.doi.org/10.1029/01EO00224.
[49] G. Jarjees, C. Drane, Detection of congestion using information theoretic techniques, in: Proceedings of the International Conference on Application of New Technology to Transport Systems, May 1995, Melbourne, Australia: Volume 1, 1995.
[50] M.A.P. Taylor, Exploring the nature of urban traffic congestion: concepts, parameters, theories and models, in: Proceedings, 16th Arrb Conference, 9-13 November 1992, Perth, Western Australia; Volume 16, Part 5, 1992.
[51] H. Qassim, A. Verma, D. Feinzimer, Compressed residual-VGG16 CNN model for big data places image recognition, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2018, pp. 169–175, http://dx.doi.org/10.1109/CCWC.2018.8301729.

[52] X. Liu, J. Van De Weijer, A.D. Bagdanov, Exploiting unlabeled data in cnns by self-supervised learning to rank, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1862–1878, http://dx.doi.org/10.1109/TPAMI.2019.2899857.
[53] D. Hendrycks, K. Gimpel, Bridging nonlinearities and stochastic regularizers with gaussian error linear units, 2016.
[54] L. Wen, P. Zhu, D. Du, X. Bian, H. Ling, Q. Hu, Z. Tong, Visdrone-mot2019: The vision meets drone multiple object tracking challenge results, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, http://dx.doi.org/10.1109/ICCVW.2019.00028.

Expert Systems With Applications 238 (2024) 122069

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A lightweight dense crowd density estimation network for efficient compression models

Yong-Chao Li, Rui-Sheng Jia *, Ying-Xiang Hu, Hong-Mei Sun *

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

ARTICLE INFO

Keywords:
Dense crowd density estimation
Lightweight networks
Model compression
Channel attention
Pyramid feature aggregation

ABSTRACT

Crowd density estimation is a task of intelligent applications, and its operation efficiency is very important. However, to obtain a better density estimation performance, most of the existing works often design larger and more complex network structures, which will result in them occupying considerable memory, time and other resources at runtime, and require the support of high-performance hardware platforms, which are difficult to apply in practice. In this paper, to overcome the above problems, we propose a lightweight dense crowd estimation method based on channel attention multi-scale feature fusion. Specifically, in the process of feature extraction, an efficient and lightweight convolution module (L-weight) is designed to extract crowd features in stages, which reduces the amount of network parameters and computing costs, and we capture multi-scale crowd information through the feature extraction network of pyramid structure, which solves the problem of uneven crowd scale in video images. In the process of feature fusion, a channel attention fusion module is designed, which weights and fuses the feature information of different scales, effectively fuses multi-scale information and suppresses useless information. In addition, we design a new loss function, which enhances the sensitivity of the crowd through the pixel space loss (L_2), counting loss (L_C) and structural similarity loss (L_S), to ensure the counting accuracy. Extensive experiments on four mainstream datasets demonstrate that compared with other state-of-the-art methods, our method achieves an optimal trade-off between counting performance and running speed, and is suitable for low-performance computing platforms such as embedded.

1. Introduction

With the rapid growth of our country's population and the continuous acceleration of urbanization, large-scale crowd gatherings such as sports events, transportation, and tourism are increasing, and crowd gatherings often bring potential dangers, such as stampede incidents. Therefore, crowd density estimation has become an important research topic in the field of public safety (Li et al., 2014).

In recent years, more and more researchers have devoted themselves to the study of crowd density estimation methods based on deep neural networks, and have proposed a large number of high-quality crowd density estimation methods (Onoro-Rubio, & López-Sastre, 2016; Zhang et al., 2017). Among them, most of the state-of-the-art methods use a heavy backbone network to extract crowd features, although these methods can improve the accuracy of crowd density estimation, these methods consume a lot of resources during the process of training (such as computing time, computing memory, etc.).At the same time, due to the large amount of training model parameters, it is necessary to carefully adjust the parameters for different datasets, and the calculation efficiency is low, which is difficult to apply in practice. As shown in Table 1, comparison of the mean absolute error (MAE), Param (Parameters), FLOPs (Floating-point operations) and running time of our method and the most state-of-the-art methods on the ShanghaiTech dataset.

It can be seen that STNet (Wang et al., 2022) takes 10.8 s to process an image of size 576 × 857 and 19.3 s to process an image of size 768 × 1024 on Intel(R) Core(TM) i5-10300H CPU, which will severely limit their deployment range, especially on edge embedded devices with limited computing resources. Therefore, how to trade-off between estimation accuracy and calculation speed, and further improve the accuracy of crowd density estimation on the basis of ensuring the lightweight of the algorithm has become an important research content.

In recent years, a series of lightweight crowd density estimation methods have been proposed by researchers (Shi et al., 2020; Chen, Xiu, Chen, Guo, & Xie, 2021; Gao, Wang, & Li, 2019; Wu et al., 2019; Wang et al., 2020; Liu et al., 2022). These methods improve the computational

* Corresponding authors.
E-mail addresses: jrs716@163.com (R.-S. Jia), shm0221@163.com (H.-M. Sun).

https://doi.org/10.1016/j.eswa.2023.122069
Received 15 September 2022; Received in revised form 26 September 2023; Accepted 6 October 2023
Available online 13 October 2023

efficiency of the network, but there are still some shortcomings, mainly in the following two points:

1. Most of the existing lightweight methods reduce the amount of parameters of the network by directly deleting a large number of convolutional layers, but the accompanying problem is that a small number of convolutional layers cannot extract enough crowd features at all, which leads to the decline of network feature extraction ability and network performance, and affects the accuracy of crowd density estimation;
2. Secondly, lightweight networks generally extract multi-scale information and perform feature fusion to improve the expression ability of the crowd scale. However, most existing lightweight crowd density estimation methods focus more on enhancing the ability of crowd feature extraction, while ignoring the problem of feature loss during feature fusion, which inevitably leads to the loss of the accuracy of crowd density estimation.

The existing methods do not have a good trade-off between estimation accuracy and calculation speed. On the premise of meeting the real-time performance, the accuracy of crowd density estimation can still be further improved. Therefore, according to the problems of the above crowd density estimation methods, this paper proposes a lightweight dense crowd density estimation method based on channel attention multi-scale feature fusion. This method makes a trade-off between estimation accuracy and calculation speed, and uses pyramid feature aggregation and channel attention feature fusion modules to further improve the estimation accuracy of dense crowds on the basis of ensuring the lightweight of the algorithm. The main contributions of this paper are as follows:

1. To reduce the amount of parameters and running time of the network, this paper designs a lightweight network, using the L-weight module, through operations such as point convolution, channel-by-channel convolution and feature stacking, extract crowd characteristics with less parameters and lower computational cost, effectively compress the network model, and enhance the real-time performance of the network.
2. To solve the problem of uneven crowd scale, this paper designs a pyramid feature aggregation module to capture multi-scale feature information, and uses the channel attention mechanism to fuse the multi-scale information, avoid the loss of crowd feature during feature fusion, thereby enhancing the network's ability to express multi-scale crowd characteristics.
3. To improve the sensitivity of the network to the crowd distribution area, a new loss function is designed, the counting loss (L_C) and structural similarity loss (L_S) are added on the basis of the previous pixel space loss (L_2), which effectively strengthens the global perception ability of the network and improves the accuracy of crowd density estimation.

2. Related work

Crowd density estimation has been developed for many years, because it plays an important role in the field of public safety, so it has been a long-term concern of researchers in the field of computer vision. The existing methods can be divided into three categories: detection-based, regression-based and CNN-based. Detection-based and regression-based methods belong to traditional counting methods. In recent years, CNN-based methods have shown better performance than traditional methods in crowd density estimation task.

Detection-based methods: Most of the early studies are focused on the detection-based methods, which use a detector similar to moving window to detect the people in the image, and then calculate the number of people using statistics. This kind of method should train a detector to extract the features of human head or body from the image, the cost of training this detector is colossal, and it needs to scan the image globally, in the past studies, many kinds of detectors have been used to detect individuals in the image, but the effect is not superb. In 2001, Lin et al. used Haar wavelet classifier to detect the head characteristics of the population (Lin et al., 2001). In 2005, Navneet Dalal et al. replaced Haar wavelet classifier with HOG (histogram oriented gradient) classifier to detect head features of people (Dalal & Triggs, 2005). In 2008, Li et al. proposed a head shoulder detector with foreground segmentation (Li et al., 2008). In 2009, Pedro F Felzenszwalb et al. tried to detect some typical features of the body, rather than the whole body, because in crowd scenes, the human body is always covered (Felzenszwalb et al., 2009). In 2011, Dollar et al. used a detector similar to a moving window to detect the human body and calculate the number of people in the image (Dollar et al., 2011). However, detection-based counting is limited by the occlusion between people in the crowded scene. As the crowd becomes dense, the detection performance decreases rapidly.

Regression-Based methods: Because the detection-based methods can't adapt to the scene of high crowd density estimation, the regression-based methods are proposed. The regression-based methods directly map between the image features and the crowd density estimation, which avoid the complex process of training the detector. These methods mainly have two steps: First, extract the feature information of the crowd image, including head for body features, foreground features, texture features and so on, these features are used to generate low-level information; Second, use regression function to calculate the number of people, the common regression methods are ridge regression (An et al., 2007), Bayesian Poisson regression (Chan & Vasconcelos, 2009), random forest (Pham et al., 2015), etc. In 2009, Chan et al. used foreground and texture features to generate low-level information, and calculated the number of people after learning the relationship between the crowd and the extracted features (Chan & Vasconcelos, 2009). Although this method better solves the shortcomings of the detection-based crowd counting methods, the mapping relationship between the number of crowds and the extracted features is difficult to be learned directly, so there is still a lot of room to improve the counting accuracy of this crowd counting method. In 2010, Lempitsky et al. proposed a density map estimation method, which uses linear mapping between the

Table 1
Comparison of the MAE, Param, FLOPs and running time of our method and the most advanced methods on the ShanghaiTech dataset. The running times are measured on Intel(R) Core(TM) i5-10300H CPU @ 2.5 GHz 16G RAM and NVIDIA GTX 2060 GPU. The unit of Param is millions(M), the unit of FLOPs is GIGA(G), the running time of GPU is milliseconds(MS), the running time of CPU is seconds(S). Red indicates the best performance.

Method	Reference	Venue	Param	Part A (576 × 864)				Part B (768 × 1024)			
				MAE	FLOPs	GPU	CPU	MAE	FLOPs	GPU	CPU
CSRNet	Li et al. (2018)	CVPR18	16.26	68.2	205.9	57.8	10.2	10.6	325.3	87.2	26.4
DUBNet	Oh et al. (2020)	AAAI20	18.05	64.4	562.3	316.4	37.3	7.7	888.5	415.6	54.3
SFCN	Wang, Gao et al. (2021)	IJCV21	38.60	64.8	69.8	65.7	14.6	7.6	110.3	84.3	31.7
DKPNet	Chen, Yan et al. (2021)	ICCV21	30.36	55.6	105.6	68.7	13.7	6.6	166.8	91.3	30.9
STNet	Wang et al. (2022)	TMM22	15.56	**52.9**	186.6	39.4	10.8	**6.3**	294.9	67.8	19.3
Ours	–	–	**0.070**	63.8	**1.88**	**7.3**	**1.9**	7.1	**2.9**	**9.8**	**3.2**

features in a local region and its density map to integrate saliency problems (Lempitsky & Zisserman, 2010). Since then, the crowd density estimation methods based on density maps have been developed and have gradually become a mainstream counting method. In 2012, Chan et al. proposed the Bayesian model of discrete regression for crowd density estimation, which connects the proposed approximate Bayesian Poisson regression with Gaussian kernel to realize the crowd density estimation from low-level features (Chan & Vasconcelos, 2011). This method improves the accuracy of crowd density estimation, however, this method is easy to miss identifying some crowds with inconspicuous features during foreground segmentation, which affects the accuracy of the final density estimation. In 2013, Idrees et al. proposed a model, which introduced Fourier analysis and SIFT (Scale invariant feature transform) to extract foreground and texture features of interested points to complete the crowd density estimation (Idrees et al., 2013). The method uses traditional machine learning as well as frequency domain analysis to greatly improve the accuracy of crowd density estimation. In 2015, to solve the gain problem of ideal linear mapping, Pham et al. proposed using random forest regression to learn nonlinear mapping instead of linear mapping (Pham et al., 2015). They used a patch-based density estimation method that learns a mapping between patch features and the relative positions of all objects within each patch, which helps to generate density maps through Gaussian kernel density estimation, further improving the accuracy of crowd density estimation. In the same year, Ryan et al. used size, shape, edges, key-points and texture as regression features, and used K-fold cross validation on multiple datasets and achieved better performance (Ryan et al., 2015). In 2019, Gao et al. introduced both spatial and channeled attention mechanisms into a traditional regressive convolutional neural network, its attention mechanism attenuates the interference of background information to improve the accuracy of crowd density estimation (Gao, Wang, & Yuan, 2019). 2022 Zheng et al. proposed a crowd density estimation method based on the Transformer framework, which proposed a location recovery module to recover the loss of location information and a new regression strategy to fully utilize the multi-scale information without the need of up-sampling operation during the fusion process (Zheng et al., 2022). Regression-based methods are effective in some specific scenarios, such as low-density crowd or fixed monitoring perspective. However, for the presence of perspective distortion and dense crowd, the counting ability is insufficient.

CNN-Based methods: With the development of convolutional neural networks, a large number of CNN-based crowd density estimation methods have been proposed, and has shown good performance. Basic CNN is first applied to crowd density estimation, such as in 2015, Wang et al. first applied convolutional neural network to crowd density estimation, the model used basic CNN layers, including convolution layer, pooling layer, and full-connection layer (Wang et al., 2015). It did not need additional feature information and was easy to implement, but the model did not consider the problem of uneven crowd size, and the counting accuracy and robustness were poor. Recently, in view of the problems of large crowd density, uneven scale, and background interference that make crowd density estimation difficult, Multi-column CNN-based crowd density estimation methods have been developed, such as in 2016, Zhang et al. proposed a multi-column convolutional network structure (MCNN), which extracted multi-scale feature information through three different convolutional network structures, which improved the accuracy of crowd counting, but the model structure is complex, the number of parameters is huge, and the model training is more difficult (Zhang et al., 2016). In 2017, Sindage et al. proposed an end-to-end cascaded multi-task learning method (CMTL), which simultaneously learns the classification of crowd density estimation and the estimation of density maps, which improves the efficiency of crowd density estimation, but in the training process, this method relies too much on the classifier, and pre-trains the network before training the classifier, which takes a long time (Sindagi & Patel, 2017a), in 2017, Sindagi et al. used the context pyramid structure to capture global semantic information, by combining the global and local context information of crowd images to improve the accuracy of crowd density estimation, but the structure of this method is redundant, and it also takes more training time (Sindagi & Patel, 2017b). To solve the problem of redundant feature extraction by Multi-column CNN, reduce the number of parameters of the network and speed up the training speed of the network, Single-column CNN crowd density estimation methods have been developed, such as in 2018, Li et al. proposed a crowded scene recognition network (CSRNet) (Li et al., 2018), the model uses cascaded dilated convolutions (Yu & Koltun, 2015) to fit the multi-scale information of the crowd, which reduces the number of model parameters, but a large amount of detailed information is castrated, resulting in complex background information that interferes with the accuracy of counting, in the same year, Cao et al. proposed SANet, which extracts multi-scale features through multiple convolution kernels of different sizes, which resulted in a redundant feature extraction and affected the accuracy of counting (Cao et al., 2018). In 2019, Liu et al. proposed a context-aware network (CANet), which can adaptively predict the scale of contextual information required for crowd density, although this improves the accuracy of crowd counting, it increases the complexity of the model (Liu, Salzmann, & Fua, 2019). At the same time, to solve the interference of the complex background on the crowd density estimation, the attention mechanism was introduced into the crowd density estimation network. The attention mechanism can supplement the features extracted by the backbone network or the head network by providing the capability to encode distant dependencies or heterogeneous interactions to highlight the head position, such as in 2019, ADCrowdNet designed an attention deformable convolutional network, which improves the global context modeling ability of the network through attention-aware training and multi-scale deformable convolution, but the method has a complex calculation process and numerous parameter adjustments, which require careful parameter adjustment for different datasets, which wastes many computing resources (Liu, Long, Zou, Niu, Pan, & Wu, 2019). In 2020, Jiang et al. integrated Density Attention Network (DANet) and Attention Scaling Network (ASNet) to resolve the difference in crowd counting performance in different regions of the image, but the number of feature parameters after fusion is large (Jiang et al., 2020). In 2021, Liu et al. proposed a crowd density estimation method based on self-attention mechanism to improve the global feature extraction ability of the network, but this will increase the number of parameters of the model, the computational cost is high and the real-time performance is poor, which is not suitable for embedding platform application (Liu et al., 2021).

Most of the above methods require heavy network parameters, but in practical applications, most of the application platforms for crowd density estimation are embedded devices, and heavy networks are difficult to apply on embedded devices. Therefore, this paper designs a lightweight crowd density estimation method to balance the estimation accuracy and calculation speed, on the basis of ensuring the lightweight of the algorithm, the estimation accuracy of dense population is further improved.

3. Proposed solutions

In order to efficiently estimate the crowd density, reduce the computational cost and enhance the real-time performance of the network, at the same time, solve the problem of uneven crowd scale, this paper designs a lightweight dense crowd density estimation method based on channel attention multi-scale feature fusion. The backbone of the network uses a lightweight pyramid feature aggregation module to extract multi-scale crowd features to reduce the amount of network parameters, and uses the channel attention mechanism to weight and fuse multi-scale feature information to avoid unnecessary feature loss during the fusion process, solve the problem of uneven crowd scale, the proposed network structure is shown in Fig. 1.

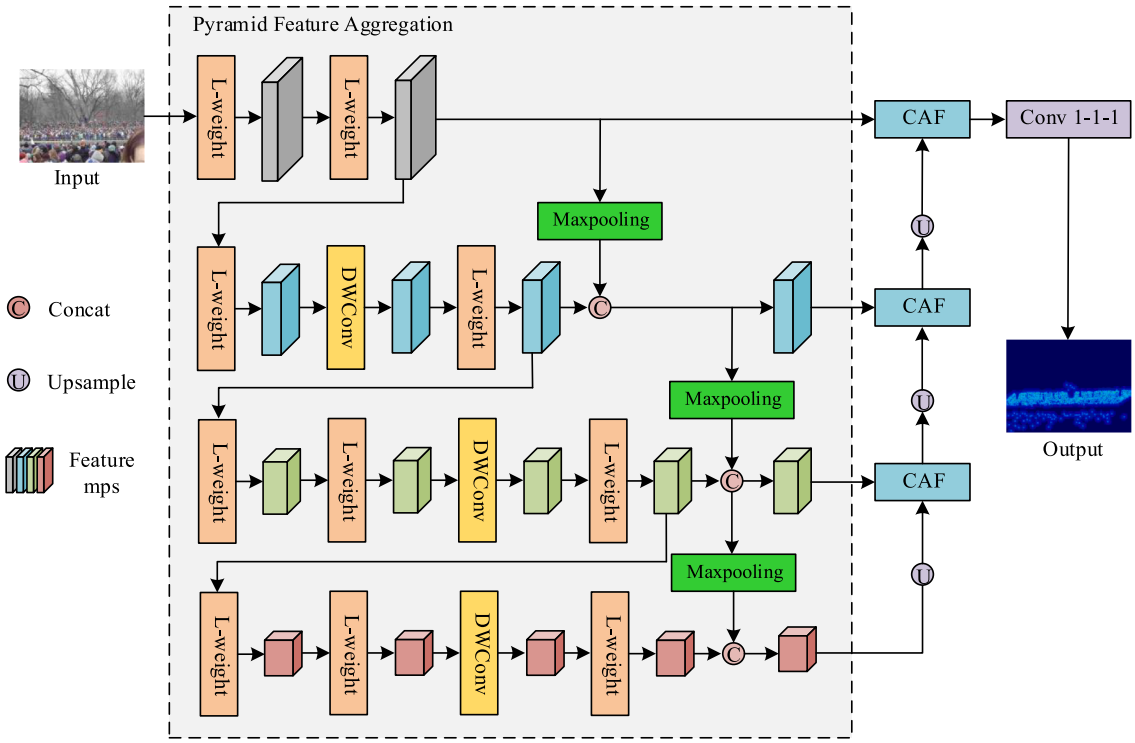


Fig. 1. Overview of the proposed architecture.

3.1. L-weight module

A deep convolutional neural network is usually composed of numerous convolutional blocks to avoid the loss of feature information, but this will lead to numerous parameter calculations, moreover, there will be many similar feature map pairs in the feature map extracted by a well-trained deep convolutional neural network, and a similar feature map of a feature map can be generated by convolution of the feature map, that is to say, the deep feature map extracted by the network can be obtained from the shallow feature map using convolution, and the shallow feature map can be obtained from point convolution. Therefore, this paper designs a lightweight convolution module L-weight, such as shown in Fig. 2.

First, the input of the L-weight module is the feature map F, through channel split, F is divided into two feature maps F1 and F2, and the point convolution operation is performed on F_1 to generate a shallow feature map F3, then, the channel-by-channel convolution operation is performed on F_3 to generate a deep feature map F4, at the same time, F_3 and F_2 are added pixel by pixel to obtain a shallow feature map F5, and finally superimpose the shallow feature map F5 and the deep feature map F4 to obtain the output feature map O, O is similar to the feature map generated by the deep convolutional neural network, on the basis of ensuring the full extraction of crowd characteristics, the calculation amount of parameters is greatly reduced, and the calculation speed of the network is improved. Among them, the size of the feature map F is $H \times W \times C$, and the size of the feature maps F1, F2, F3, F4 and F5 is $H \times W \times C/2$, the size of the feature map O is $H \times W \times C$.

The ratio of L-weight and standard convolution calculations is as follows:

$$\frac{C_l}{C_s} = \frac{I \times \frac{O}{2} \times H \times W + S_k \times S_k \times \frac{O}{2} \times \frac{O}{2} \times H \times W}{S_k \times S_k \times I \times O \times H \times W} = \frac{1}{2S_k^2} + \frac{O}{4I} \tag{1}$$

In the formula, C_l is the calculation amount of the L-weight module, C_s is the calculation amount of the standard convolution, H is the height of the feature map, W is the width of the feature map, $S_k \times S_k$ is the size of the convolution kernel, I is the number of input channels, O is the

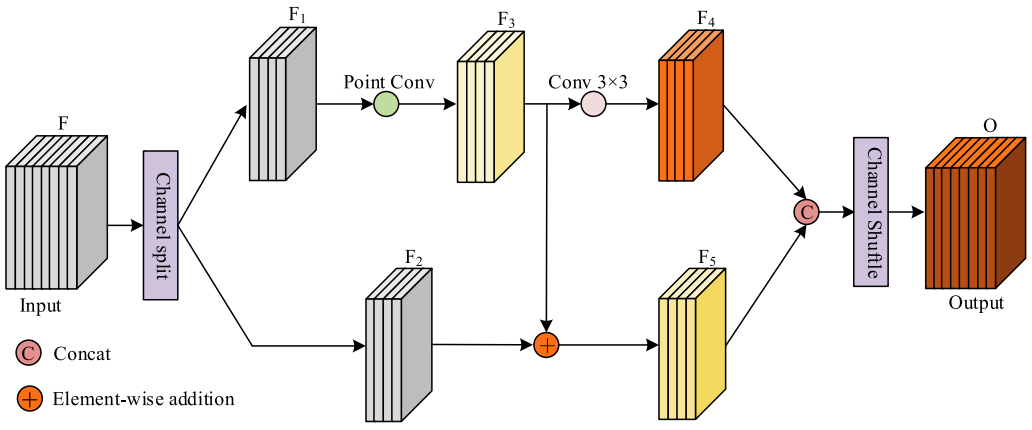


Fig. 2. The structure of L-weight module.

number of output channels. It can be seen from formula (1) that the calculation amount of the L-weight module is $\frac{1}{2S_k^2}+\frac{O}{4I}$ of that of the ordinary convolution. When the convolution kernel size is 3 × 3, the parameter amount of the L-weight module is about 1/18 of ordinary convolution, which simplifies the scale of the model and improves the efficiency of the model.

3.2. Pyramid feature aggregation module

In large-scale dense crowds, due to the different distances between people and cameras in the same scene, the crowd scale diversity is caused. To extract crowd characteristics of different scales and solve the problem of uneven scale of dense crowds, this paper designs a pyramid feature aggregation module, the network parameters are shown in Table 2.

As shown in Fig. 1, the pyramid feature aggregation module includes 4 layers of convolutional layers. In each layer, L-weight module and depthwise separable convolution (DWConv) with stride 2 and kernel size 3 × 3 are used to reduce the amount of model parameters. Specifically, in the first layer, two L-weight modules are used to extract features, and the output is sent to the second layer; the second layer first uses one L-weight module to extract features, and then using a depthwise separable convolution with stride 2 to take the feature map down-sampled to 1/4 of the input image, and then uses an L-weight module to further extract features, finally uses residual connections to add the input feature map and the output feature map; the third layer has the same structure as the fourth layer, first two L-weights modules are used to extract features, the feature maps are down-sampled to 1/16 and 1/64 of the input image using depthwise separable convolution with stride 2, respectively, then uses an L-weight module to further extract features, finally uses residual connections to add the input feature map and the output feature map. The feature map output by the 4-layer pyramid convolutional layers contains the crowd feature information of different scales, the feature maps output by the four convolutional layers are respectively up-sampled to the size of the input image, and input into the channel attention fusion module for fusion of multi-scale crowd features.

3.3. Channel attention fusion module

In order to solve the problem of uneven crowd scale and avoid the loss of some crowd feature information during the fusion process, which affects the accuracy of crowd density estimation. A channel attention fusion module is designed, which calculates the correlation between pixels in different feature maps through matrix multiplication, and then uses the correlation as the weight vector of the feature map to effectively fuse the crowd information in the feature maps of different scales, the calculation process is shown in Fig. 3.

Table 2
Network parameters of pyramid feature aggregation module. S represents the size of the input image.

Pyramid Layer	Convolutional	Output channel	Output size
Layer 1	L-weight	16	S
	L-weight	16	S
Layer 2	L-weight	32	S
	DWConv	32	1/4S
	L-weight	32	1/4S
Layer 3	L-weight	64	1/4S
	L-weight	64	1/4S
	DWConv	64	1/16S
	L-weight	64	1/16S
Layer 4	L-weight	32	1/16S
	L-weight	32	1/16S
	DWConv	32	1/64S
	L-weight	32	1/64S

As shown in Fig. 3, first, the feature map of size $H \times W \times C$ is input, the number of channels of the input feature map is reduced to $C/2$ through a 3 × 3 convolution layer to reduce the number of computation. Then, reshape the feature map into a 2D feature matrix of size $N \times C/2$, where $N = H \times C$, transpose one of the feature matrices, and compute the correlation between pixels in two feature matrices by matrix multiplication:

$$W_{ij} = \frac{\exp(P_i \cdot Q_j)}{\sum_{i=1}^{N} \exp(P_i \cdot Q_j)} \tag{2}$$

In the formula, P and Q are the feature matrix that needs to be fused, P_i is the i^{th} position of the feature matrix P, Q_j is the j^{th} position of the feature matrix Q, W_{ij} measures the correlation between the P_i and the Q_j, N represents the number of pixels, the higher the similarity between two pixels, the greater their correlation, and the greater the corresponding weight W_{ij} value. Then, use W_{ij} to perform matrix multiplication with the feature matrix, and perform feature weighting processing on the feature matrix to reduce the noise interference in the feature map. Finally, the feature matrix is reshaped into a feature map, and the feature maps are superimposed to obtain the output feature map.

3.4. Loss function

At this stage, most crowd density estimation methods use Euclidean loss (L_2) on pixels to train the network, but the Euclidean loss function only reflects the pixel-level accuracy of the estimated density map, ignoring the global difference between the estimated density map and the Ground Truth image and the goal of crowd counting, which is to count the total number of people in the image. Therefore, this paper proposes a new loss function, which adds counting loss (L_C) and structural similarity loss (L_S) to the pixel space loss to enhance the sensitivity of the network to counting. As shown in formula (3):

$$L = (1-\alpha)(L_2 + L_C) + \alpha L_S \tag{3}$$

In the formula, L is the comprehensive loss function, L_2 is the Euclidean loss function, L_C is the counting loss function, L_S is the structural similarity loss function, and α is the weight value of the loss function. Among them, the Euclidean loss function can be expressed as:

$$L_2(\theta) = \frac{1}{N}\sum_{i=1}^{N} \left| Z(X_i, \theta) - Z_i^{GT} \right|^2 \tag{4}$$

In the formula, N represents the number of images in the training set, X_i represents the i^{th} image of the input, θ represents a set of parameters that can be learned, $Z(X_i, q)$ represents the prediction result of the network, and Z_i^{GT} represents the ground truth of the input image X_i. The counting loss function can be expressed as:

$$L_C(\theta) = \frac{1}{N}\sum_{i=1}^{N} \left| \frac{F_Y(X_i, \theta) - Y_i}{Y_i} \right|^2 \tag{5}$$

In the formula, N represents the number of images in the training set, X_i represents the i^{th} image of the input, θ represents a set of parameters that can be learned, $F_Y(X_i, \theta)$ represents the estimated number of people obtained by integrating and summing the estimated density map of the X_i ($i = 1, \ldots\ldots, N$); Y_i is the real number of people in the X_i ($i = 1, \ldots, N$).

Structural similarity is an index to measure the global difference of images, it can calculate the global difference between two images according to the local information of the image, such as mean, variance, covariance. Its value range is [−1, 1], and the more similar the comparison images, the larger the value. The structural similarity loss function can be expressed as:

$$SSIM(P) = \frac{(2\mu_Z\mu_G + C)(2\sigma_{ZG} + C)}{(\mu_Z^2 + \mu_G^2 + C)(\sigma_Z^2 + \sigma_G^2 + C)} \tag{6}$$

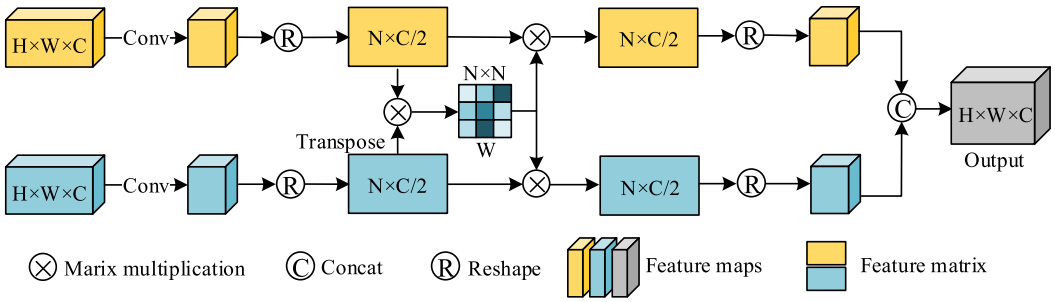


Fig. 3. The structure of CAF.

$$L_S = 1 - \frac{1}{M}\sum_{P}^{M} SSIM(P) \tag{7}$$

First, calculate the local mean, variance, and covariance of each head position P in the predicted crowd density map and Ground Truth, respectively. In formula (6), μ_Z, σ_Z^2 represents the mean and variance of the predicted crowd density map, μ_G,σ_G^2 represents the mean and variance of the Ground Truth, σ_{ZG} represents the covariance between the predicted crowd density map and Ground Truth, C is a very small constant to avoid division by 0; secondly, calculate *SSIM* point by point, and substitute it into formula (7) to calculate L_S, where M represents the total number of pixels in the density map.

3.5. Ground truth generation

The existing datasets generally provide the original image and the coordinate position and total number of the corresponding crowd in the image. We use the Focal Inverse Distance Transform (FIDT) map to generate the ground truth of large-scale dense crowds. If there are B labeled points in an image, perform the following processing on the feature image:

$$P(x,y) = \min_{(x',y')\in B} \sqrt{(x-x')^2 + (y-y')^2} \tag{8}$$

$$I = \frac{1}{P(x,y)^{(\alpha P(x,y)+\beta)} + C} \tag{9}$$

In formula (8), B represents the set of all marked points, for any marked point (x, y), calculate the Euclidean distance $P(x, y)$ of the nearest marked point, (x', y') denotes the coordinates of the nearest marked point. Because the distance between the marked points varies greatly, it is difficult to perform distance regression directly, so the inverse function is used for regression, as shown in formula (9), I is the focal inverse distance transform map, C is an additional constant, usually set to 1, to avoid the occurrence of the calculation process divide by 0, and exponential processing is performed on $P(x, y)$ to slow down the decay speed of crowd head information. In the experiment, set $\alpha = 0.02$, $\beta = 0.75$.

4. Experiment

4.1. Experimental environment and parameter settings

The experiments were performed using the Pytorch deep learning framework on a computer with an Intel(R) Core(TM) i5-10300H CPU @ 2.5 GHz 16G RAM, and accelerated using NVIDIA GTX1080 and NVIDIA GTX 2060 graphics cards for acceleration.

Before training, we initialize the network parameters with Gaussian initialization with a standard deviation of 0.01. In this paper, the Adam optimization method is used to optimize the parameters and the training batch size is set to 8, the initial learning rate of the network is set to 1e-5, the decay rate is set to 1e-4, and the activation function uses the ReLU6 function to improve the training speed and effectively avoid the vanishing and exploding gradients.

4.2. Dataset

ShanghaiTech (Zhang et al., 2016): Contains 1198 annotated images with a total of 330,165 people. The dataset consists of two parts, Part A and Part B. Part A includes 482 highly crowded crowd images, of which 300 form the training sample set and the remaining 182 form the test sample set; Part B includes 716 relatively sparse crowd images, of which 400 images form the training sample set, and the remaining 316 images form the test sample set.

UCF_CC_50 (Idrees, et al, 2013): Contains 50 images from different perspectives and different resolutions from the Internet. The number of annotations per image ranged from 94 to 4543, with an average of 1280. Due to the limited number of images in this dataset and the large span of labeled people in the image, five-fold cross-validation is used in this dataset.

UCF_QNRF (Idrees et al., 2018**)**:Contains 1535 crowd images, a total of 12,500 people, of which 1201 images form the training sample set, 334 images form the test sample set, and the number of people included in each image ranges from 49 to 12,865, with an average of 815 people.

This paper conducts experiments on 4 datasets, and selects a representative crowd image in each dataset, as shown in Fig. 4.

4.3. Evaluation metric

In this paper, the mean absolute error (MAE), the mean square error (MSE) and the mean absolute percentage error (MAPE) are used as the evaluation metrics of the algorithm performance. Their expressions are as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\widehat{C}_i - C_i| \tag{10}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{C}_i - C_i)^2} \tag{11}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\widehat{C}_i - C_i}{C_i}\right| \tag{12}$$

In the formula, N represents the number of test images, C_i represents the actual number of people in the i^{th} image, and $\widehat{C}_i$ represents the estimated number of people in the i^{th} image. When the values of MAE, MSE and MAPE are smaller, the error between the estimated number of people and the actual number of people is smaller, indicating that the effect of the experiment is better.

We also use the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) to evaluate the quality of the output density map. The PSNR is a mathematical method based on image pixel statistics. It uses

Fig. 4. Representative examples from four crowd density estimation datasets. (a) from the ShanghaiTech Part A dataset, (b) from the ShanghaiTech Part B dataset, (c) from the UCF_CC_50 dataset, (d) from the UCF_QNRF dataset.

statistical methods to measure the quality of the resulting image by calculating the difference between the greyscale values of the pixels of the resulting image corresponding to the original image. Their expressions are as follows:

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{MSE}\right) \tag{13}$$

The larger the *PSNR* value, the smaller the difference between the output density map and the Ground Truth image, which means the better the image quality. The calculation process of *SSIM* is shown in Eq. (6), its value range is [−1, 1], and the larger the SSIM value, the more similar output density map and the Ground Truth image.

4.4. Experiment: Comparisons with state-of-the-art methods

ShanghaiTech dataset is a crowded and variable scale dataset, to evaluate the performance of our method, the experiment was carried out on this dataset, the comparison of MAE, MSE and MAPE with the state-of-the-art methods is given in columns 3 and 4 of Table 3. UCF_CC_50 dataset includes 50 grayscale images with different viewing angles and resolutions, which is a very challenging dataset with various crowd scenes and a limited total number of images, therefore, this paper performs 5-fold cross-validation to maximize the use of samples, the dataset is randomly divided into five equal parts, four of which are used as the training set, and the remaining one is used as the test set, a total of five times of training and testing are carried out, and finally the average value of the error index is taken as the final result of the experiment, the comparison of MAE, MSE and MAPE with the state-of-the-art methods is given in column 5 of Table 3. The UCF_QNRF dataset is also a crowded dataset with diverse crowd scales. It is collected from three different datasets and contains a wide variety of scenes, the total number of images, and the total number of people far exceeding the first two datasets. On this dataset, we have done a lot of experiments, the comparison of MAE, MSE and MAPE with the state-of-the-art methods is given in columns 6 of Table 3. In order to prove the efficiency of our method, a comparison experiment of parameter quantity was carried out with the current mainstream deep convolutional neural network and lightweight crowd density estimation methods, and the comparison results are given in the 7 column of Table 3. The compared methods include: Switching-CNN (Babu Sam et al., 2017), CSRNet (Li et al., 2018), DUBNet (Oh et al., 2020), ASNet (Jiang et al., 2020), RPNet (Yang et al., 2020), AMSNet (Hu et al., 2020), NoiseCC (Wan, & Chan, 2020), LSC-CNN (Sam et al., 2020), UOT (Ma et al., 2021), SUA-Fully (Meng et al., 2021), UEPNet (Wang, Song, Zhang, Wang, Tai, Hu, & Wu, 2021), P2PNet (Song et al., 2021), DKPNet (Chen, Yan, Li, Li, Wang, Zuo, & Zhang, 2021), GL (Wan et al., 2021), SSR-HEF (Chen et al., 2022), STNet (Wang et al., 2022), MCNN (Zhang et al., 2016), TDF-CNN (Sam, & Babu, 2018), SANet (Cao et al., 2018), PCC-Net (Gao, Wang, & Li, 2019), LCNet (Ma et al., 2019), CCNN (Shi et al., 2020).

Performance on the ShanghaiTech dataset: As shown in Table 3, we divided the compared methods into two parts. The methods in the first part all rely on heavy backbone networks with a large amount of

Table 3
Comparison with state-of-the-art methods on ShanghaiTech Part A, ShanghaiTech Part B, UCF_CC_50 and UCF_QNRF datasets. Red and blue indicate the first and the second best performances, respectively. The unit of Param is millions (M).

Method	Venue	Part A			Part B			UCF_CC_50			UCF_QNRF			Params (M)
		MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	
Switching-CNN	ICCV17	90.4	135.0	22.2 %	21.6	33.4	14.8 %	318.4	439.2	33.1 %	–	–	–	15.11
CSRNet	CVPR18	68.2	115.0	15.8 %	10.6	16	7.8 %	266.1	397.5	26.3 %	135.4	207.4	19.9 %	16.26
DUBNet	AAAI20	64.4	106.8	14.8 %	7.7	12.5	5.8 %	234.8	329.3	22.5 %	105.6	180.5	14.9 %	18.05
ASNet	CVPR20	57.8	90.0	13.1 %	–	–	–	174.8	**251.6**	15.9 %	91.6	159.7	12.7 %	30.39
RPNet	CVPR20	61.2	96.9	13.9 %	8.1	116	6.1 %	–	–	–	–	–	–	23.89
AMSNet	ECCV20	56.7	93.4	12.6 %	6.7	10.2	5.1 %	208.4	297.3	19.5 %	101.8	163.2	14.3 %	3.79
NoiseCC	NIPS20	61.9	99.6	14.1 %	7.4	11.3	5.5 %	–	–	–	85.8	150.6	11.8 %	20.02
LSC-CNN	TPAMI20	66.5	101.8	15.6 %	7.7	12.7	5.9 %	–	–	–	120.5	218.2	17.4 %	42.23
UOT	AAAI21	58.1	95.9	12.9 %	6.5	10.2	5.0 %	–	–	–	83.3	142.3	11.4 %	21.50
SUA-Fully	ICCV21	66.9	125.6	15.7 %	12.3	17.9	9.1 %	–	–	–	119.2	213.3	17.1 %	15.85
UEPNet	ICCV21	54.6	91.2	12.1 %	6.4	10.9	4.9 %	**165.2**	275.9	**14.8 %**	**81.1**	**131.7**	**11.1 %**	26.21
P2PNet	ICCV21	**52.7**	**85.1**	**11.8 %**	**6.3**	**9.9**	**4.8 %**	172.7	256.1	15.6 %	85.3	154.5	11.7 %	18.34
DKPNet	ICCV21	55.6	91.0	12.5 %	6.6	10.9	5.1 %	–	–	–	81.4	147.2	**11.1 %**	30.63
GL	CVPR21	61.3	95.4	14.5 %	7.3	11.7	5.6 %	–	–	–	84.3	147.5	11.5 %	21.51
SSR-HEF	TII22	55.0	88.3	12.3 %	**6.1**	**9.5**	**4.7 %**	173.3	260.4	15.7 %	**70.2**	**128.6**	**9.4 %**	2.36
STNet	TMM22	**52.9**	**83.6**	**11.7 %**	**6.3**	10.3	4.9 %	**162.0**	**230.4**	14.5 %	87.9	166.4	12.1 %	15.56
MCNN	CVPR16	110.2	173.2	28.2 %	26.4	41.3	17.5 %	377.6	509.1	41.7 %	277.0	426.0	51.7 %	0.13
TDF-CNN	AAAI18	97.5	145.1	24.2 %	20.7	32.8	14.3 %	354.7	491.4	38.4 %	–	–	–	0.13
SANet	ECCV18	75.3	122.2	17.7 %	10.5	17.9	7.8 %	258.4	334.9	25.3 %	152.6	247.0	23.1 %	0.91
PCC-Net	TCSVT19	73.5	124.0	17.2 %	11.0	19.0	8.5 %	240.0	315.5	23.1 %	148.7	247.3	22.3 %	0.55
LCNet	ICIP19	93.3	149.0	22.7 %	15.3	25.2	10.9 %	326.7	430.6	34.3 %	–	–	–	0.86
CCNN	ICASSP20	88.1	141.7	21.4 %	14.9	22.1	10.7 %	–	–	–	–	–	**–**	**0.073**
Ours	**–**	**63.8**	**110.1**	**14.6 %**	**7.1**	**12.1**	**5.4** %	**239.6**	**332.9**	**23.0** %	**90.4**	**217.8**	**12.3 %**	**0.070**

parameters, while the methods in the second part are all lightweight networks with a small amount of parameters. Compared with the methods in the first part, the MAE, MSE and MAPE of our method are lower than the traditional counting methods such as CSRNet and DUBNet, compared with the most advanced methods such as SSR-HEF and STNet, although the MAE, MSE and MAPE are higher than them, the parameter amount of our method is only 1/220 of these methods, or even less, which is more suitable for edge devices such as embedded than them, and more time-effective than them. Compared with the methods in the second part, our method is more competitive, not only the MAE, MSE and MAPE are lower than the comparison methods, but also the parameter quantity is lower than them, achieving higher performance. On the ShanghaiTech Part A dataset, compared with the state-of-the-art lightweight method PCC-Net, our method improves the performance of MAE, MSE, and MAPE by 13.2 %, 11.2 %, and 15.1 %, respectively. On the ShanghaiTech Part B dataset, compared with the state-of-the-art lightweight method SANet, our method improves the performance of MAE, MSE, and MAPE by 32.4 %, 32.4 %, and 30.7 %, respectively. Moreover, the number of parameters in our method is only 1/8 of that of PCC-Net and 1/13 of that of SANet. Therefore, our method achieves a better balance between counting performance and running speed on the ShanghaiTech dataset, and achieves accurate counting accuracy with fewer parameters. Fig. 5 shows the partial visualization results of our method on the ShanghaiTech dataset, and we use PSNR and SSIM to evaluate the quality of the output density map.

It can be seen that our method has good performance on these two datasets, generates density maps with accurate distribution, and the resolution is also high, and the prediction results are close to the real values. Compare (a) and (b) in Fig. 5, the ShanghaiTech Part A dataset is extremely crowded and the crowd scale variation is small, while the ShanghaiTech Part B dataset is relatively sparse but the crowd scale varies greatly, which shows that our method can be well fitted for different degrees of crowd scale changes. Simultaneously, we can see that the above four images have background interferences, such as houses and trees, and the generated density map avoids these interferences, which fits the crowd well and further verifies the anti-interference of the model in this paper. But on extremely crowded datasets, more images are needed for training to improve the accuracy of the model.

Performance on the UCF_CC_50 dataset: We also divide the comparison method into two parts. Compared with the methods in the first part, the performance of our method is close to that of DUBNet, but the parameter quantity is only 1/260 of that of DUBNet. Compared with the methods in the second part, our method is also lower than the comparison method in MAE, MSE, MAPE and parameter quantity, compared with the state-of-the-art lightweight method SANet, our method improves the performance of MAE, MSE, and MAPE by 7.3 %, 0.6 %, and 9.1 %, respectively. Moreover, the number of parameters in our method is only 1/13 of that of SANet, which proves that the pyramid structure feature extraction network proposed in this paper can extract multi-scale crowd information on the basis of saving the amount of parameters, which is helpful for the extraction of smaller features in large-scale dense crowds, so as to obtain more accurate crowd numbers. Fig. 6 shows the partial visualization results of our method on the UCF_CC_50 dataset, and we use PSNR and SSIM to evaluate the quality of the output density map.

Our method has a good fitting ability for the crowd in the first image in Fig. 6, and generates an accurate and high-resolution density map. It also generates an accurate density map for the denser crowd in the second image, but the estimated value has a certain error relative to the actual value, which is a small number of images with large errors in the test of the method in this paper. The next step requires more training to improve the robustness of the model and exclude large errors.

Performance on the UCF_QNRF dataset: We also divide the comparison method into two parts. A large number of experiments were carried out on this dataset, compared with the methods in the first part, the performance of our method is close to that of most methods, but the amount of parameters is significantly lower than them. Compared with the second part, our method has achieved the best results in terms of MAE, MSE, MAPE and parameter quantity, compared with the state-of-the-art lightweight method SANet, our method improves the performance of MAE, MSE, and MAPE by 40.8 %, 11.9 %, and 46.8 %, respectively, compared with the state-of-the-art lightweight method PCC-Net, our method improves the performance of MAE, MSE, and

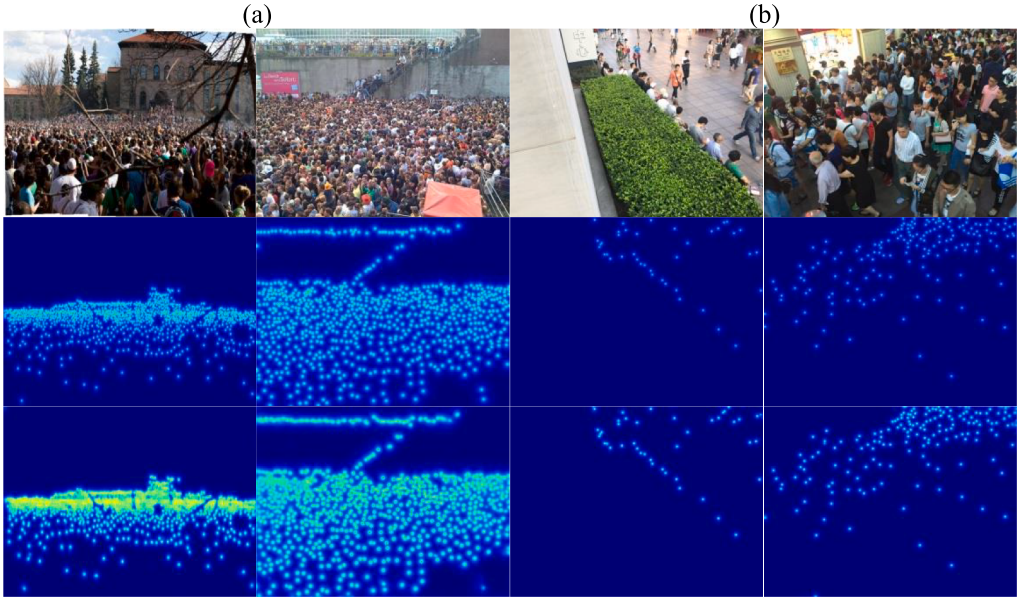


Fig. 5. Visualization results of the density maps on (a) ShanghaiTech Part A, (b) ShanghaiTech Part B. The first row is the original image, the second row is the ground truth, and the third row is the estimated density map.

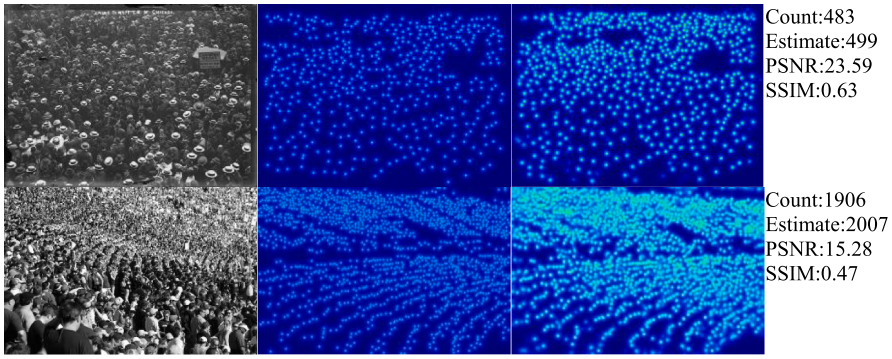


Fig. 6. Visualization results of the density maps on UCF_CC_50. The first column is the original image, the second column is the ground truth, and the third column is the estimated density map.

MAPE by 39.2 %, 11.9 %, and 44.8 %, respectively. Moreover, the number of parameters in our method is only 1/13 of that of SANet and 1/8 of that of PCC-Net, which further proves the rationality of our method. Fig. 7 shows the partial visualization results of our method on the UCF_QNRF dataset, and we use PSNR and SSIM to evaluate the quality of the output density map.

As shown in Fig. 7, our method can adapt to the light interference of different intensities, but there is still a certain error in the extremely dark environment, such as, the part marked in the first image in Fig. 7, the crowd characteristics are blurred, which affects the counting performance, it is a small number of images with large errors in the test of our method. The second image is the crowd image with better performance in the dataset, it can be seen that the Ground Truth is basically similar to the prediction map, and the predicted number of crowd is basically close to the actual number of crowd. At the same time, it is proved that a large number of experiments can improve the performance of our method on this dataset.

4.5. Speed comparison

To verify the advantages of our method in terms of running speed, compared of our method and the state-of-the-art lightweight counting methods on ShanghaiTech Part A, ShanghaiTech Part B, UCF_CC_50 and UCF_QNRF datasets, using NVIDIA GTX 2060 and NVIDIA GTX 1080 graphics cards for acceleration, respectively. Table 4 gives the comparison results of running time and frames per second (FPS).

On the ShanghaiTech Part A dataset, the average size of the input images is 576 × 864, on the ShanghaiTech Part B dataset, the average size of the input images is 768 × 1024, on the UCF_CC_50 dataset, the average size of the input images is 656 × 1024, on the UCF_QNRF dataset, the average size of the input images is 2013 × 2902. As shown in Table 4, using different types of graphics cards for acceleration, the FPS of our method is higher than that of the comparison methods, and the running time of our method is significantly less than that of the comparison methods. Using GTX2060 graphics card for acceleration, the running time of our method can be controlled within 20 ms, it is 2–3 times faster than methods such as MCNN, SANet, CCNN, etc. This fully proves the real-time performance of our method, which has the advantage of fast computing speed, and is suitable for embedded computing platforms with low performance.

5. Discuss

5.1. Study of pyramid feature aggregation module

The feature aggregation module proposed in this paper consists of 4 layers of convolutional layers composed of L-weight, to verify its rationality, ablation experiments were performed on four datasets. In the experiment, the channel attention fusion module was kept the same, and the performance of the pyramid feature aggregation module under different configurations was tested, the results are shown in Table 5.

As shown in Table 5, the ablation experiments were conducted for 2 layers of convolution, 3 layers of convolution and 4 layers of convolution. The experimental results show that our method is better than the comparison methods, on the Part A dataset, compared with the 2 layers of convolution architecture, MAE, MSE and MAPE are improved by 14.7 %, 12.5 % and17.0 %, respectively, compared with the 3 layers of convolution architecture, MAE, MSE and MAPE are improved by 7.8 %, 3.1 % and 8.8 %, respectively; on the Part B dataset, compared with the 2 layers of convolution architecture, MAE, MSE and MAPE are improved by 50.7 %, 36.6 % and 50.7 %, respectively, compared with the 3 layers of convolution architecture, MAE, MSE and MAPE are improved by 33.0 %, 22.4 % and 31.6 %, respectively, on the UCF_CC_50 dataset, compared with the 2 layers of convolution architecture, MAE, MSE and MAPE are improved by 10.5 %, 9.9 % and 13.2 %, respectively, compared with the 3 layers of convolution architecture, MAE, MSE and

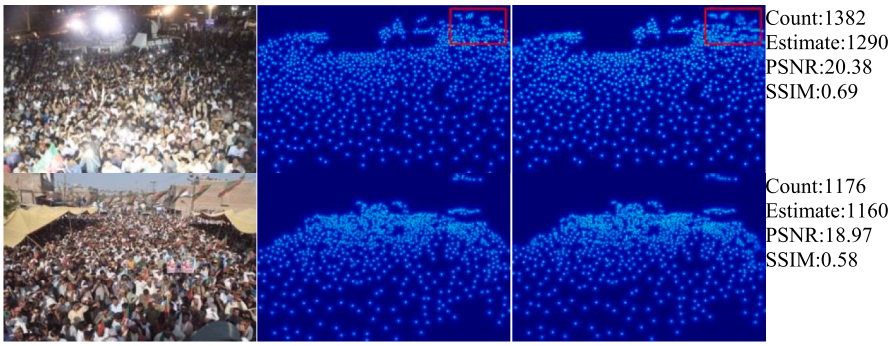


Fig. 7. Visualization results of the density maps on UCF_QNRF. The first column is the original image, the second column is the ground truth, and the third column is the estimated density map.

Table 4
Comparison results of our method with other lightweight methods in terms of running time and FPS. The unit of time is milliseconds(MS). Red indicates the best performances.

Method	Part A (576 × 864)				Part B (768 × 1024)				UCF_CC_50(656 × 1024)				UCF_QNRF(2013 × 2902)			
	GTX1080		GTX2060		GTX1080		GTX2060		GTX1080		GTX2060		GTX1080		GTX2060	
	FPS	Time	FPS	Time	FPS	Time	FPS	Time	FPS	Time	FPS	Time	FPS	Time	FPS	Time
MCNN	13.2	115.4	71.2	19.3	9.7	122.4	65.8	22.4	10.3	119.3	67.3	21.9	4.3	209.3	31.4	43.7
SANet	6.4	275.6	19.6	76.7	2.6	295.6	15.9	83.7	3.9	281.4	16.8	80.2	1.2	543.3	7.3	154.6
CCNN	23.1	59.3	119.6	11.4	18.9	63.3	111.2	13.4	20.1	61.9	114.3	12.9	8.7	134.2	56.9	29.3
Ours	**29.4**	**49.3**	**147.8**	**7.3**	**26.1**	**52.1**	**140.5**	**9.8**	**27.2**	**51.3**	**143.4**	**9.4**	**12.9**	**92.8**	**65.4**	**17.9**

Table 5
Results of ablation experiments on four datasets. L* represents the convolutional layer. Red indicates the best performances.

Structure	Part A			Part B			UCF_CC_50			UCF_QNRF		
	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE
L1 + L2	74.8	125.8	17.6 %	14.4	19.1	10.4 %	267.9	369.3	26.5 %	104.2	249.6	14.7 %
L1 + L2 + L3	69.2	113.6	16.0 %	10.6	15.6	7.9 %	254.4	361.2	24.8 %	98.1	228.5	13.7 %
L1 + L2 + L3 + L4	**63.8**	**110.1**	**14.6 %**	**7.1**	**12.1**	**5.4 %**	**239.6**	**332.9**	**23.0 %**	**90.4**	**217.8**	**12.3 %**

MAPE are improved by 5.8 %, 7.8 % and 7.3 %, respectively; on the UCF_CC_QNRF dataset, compared with the 2 layers of convolution architecture, MAE, MSE and MAPE are improved by 13.2 %, 12.7 % and 16.3, respectively, compared with the 3 layers of convolution architecture, MAE, MSE and MAPE are improved by 7.8 %, 4.7 % and 10.2 %, respectively. This proves that the pyramid feature aggregation module proposed in this paper can extract crowd features reasonably and effectively, and improve the accuracy of crowd density estimation.

5.2. Study of channel attention fusion module

The attention module has been widely used in image processing. We designed a channel attention module fusion module to weight and fuse feature maps of different levels, and verified the module through extensive experiments on four datasets. The results are shown in Table 6.

As shown in Table 6, the ablation experiments were conducted on the presence or absence of the channel attention fusion module and the number of fused feature maps, the experimental results show that when the channel attention fusion module is added, the effect is significantly better than that without the channel attention fusion module. After adding the channel attention fusion module, on the Part A dataset, MAE, MSE and MAPE are increased by 24.8 %, 18.9 % and 28.4 %, respectively, on the Part B dataset, MAE, MSE and MAPE are increased by 63.9 %, 55.4 % and 60.3 %, respectively, on the UCF_CC_50 dataset, MAE, MSE and MAPE are increased by 10.9 %, 9.5 % and 13.5 %, respectively, on the UCF_QNRF dataset, MAE, MSE and MAPE are increased by 20.4 %, 17.3 % and 24.1 %, respectively, and MAE, MSE and MAPE are optimal when fusing three-layer features of different scales. Therefore, we designed the channel attention fusion module can be effectively applied to crowd density estimation and improve the counting accuracy of crowd density estimation.

5.3. Study of loss function

In the training process of the network, the use of different loss functions will have a direct impact on the performance of the model. In this paper, we use a comprehensive loss function consisting of Euclidean loss, counting loss and structural similarity loss, and through the loss weight α to adjust their proportions. To get the best results, experiments were performed on the ShanghaiTech dataset, and the results are shown in Fig. 8.

As shown in Fig. 8, it shows the changes of MAE and MSE on the ShanghaiTech dataset as α changes. When α, it means that the structural similarity loss (L_S) is not used, it can be seen that both MAE and MSE have increased significantly, which proves the necessity of the structural similarity loss (L_S). When α, MAE and MSE achieve the minimum value, which indicates that the method has the best performance.

6. Conclusion and future work

In this work, a lightweight dense crowd density estimation network for efficient compression models is proposed to achieve the best balance between counting performance and running speed, the network is mainly composed of feature extraction module and feature fusion module. We designed a lightweight convolution module (L-weight) to reduce the amount of network parameters, which solves the problems of long computing time and high hardware requirements, at the same time, the feature extraction module of pyramid structure extracts multi-scale feature information, which solves the problem of uneven crowd scale, finally, the feature fusion module effectively fuses multi-scale information and suppresses useless information. In addition, the spatial correlation and crowd sensitivity of density map are enhanced by pixel space loss, counting loss and structural similarity loss, and Focal Inverse Distance Transform map is used to accurately express the crowd location information, which improves the quality of the crowd density map. A large number of experiments are conducted on the ShanghaiTech

Table 6
Results of ablation experiments on four datasets. None represents that the channel attention fusion module is not used, C* represents the feature maps of different scales output by the L* layer in the pyramid feature aggregation module. Red indicates the best performances.

Structure	Part A			Part B			UCF_CC_50			UCF_QNRF		
	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE
None	84.8	135.8	20.4 %	19.4	27.1	13.6 %	268.9	367.9	26.6 %	113.5	263.6	16.2 %
C3 + C4	70.2	123.1	16.3 %	9.6	13.6	7.2 %	246.7	341.8	23.9 %	94.5	224.7	13.1 %
C2 + C3 + C4	**63.8**	**110.1**	**14.6 %**	**7.1**	**12.1**	**5.4 %**	**239.6**	**332.9**	**23.0 %**	**90.4**	**217.8**	**12.3 %**

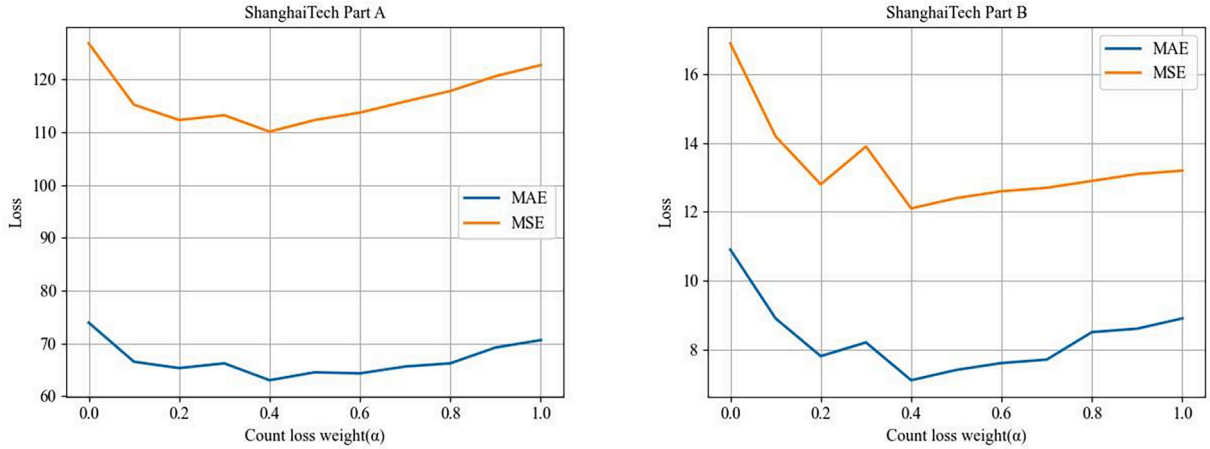


Fig. 8. MAE and MSE on the ShanghaiTech dataset under different loss weights.

dataset, UCF_CC_50 dataset and UCF_QNRF dataset, the results show that our method has the advantages of fewer parameters and faster calculation speed without losing the estimation accuracy, and effectively compress the network model, which is more suitable for low-performance computing platforms.

In the future work, we will carry out further research on the basis of the methods proposed in this paper, achieve more advanced density estimation accuracy with less parameter quantity, and train and test on more public population density datasets.

CRediT authorship contribution statement

Yong-Chao Li: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Rui-Sheng Jia:** Supervision, Methodology, Resources, Writing – review & editing. **Ying-Xiang Hu:** Data curation, Investigation, Visualization. **Hong-Mei Sun:** Supervision, Methodology, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors are grateful for collaborative funding support from the Humanities and Social Science Fund of the Ministry of Education of the People's Republic of China (21YJAZH077).

References

An, S., Liu, W., & Venkatesh, S. (2007). Face recognition using kernel ridge regression. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1–7). IEEE.

Babu Sam, D., Surya, S., & Venkatesh Babu, R. (2017). Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5744–5752).

Cao, X., Wang, Z., Zhao, Y., & Su, F. (2018). Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 734–750).

Chan, A. B., & Vasconcelos, N. (2009). In *Bayesian poisson regression for crowd counting* (pp. 545–551). IEEE.

Chan, A. B., & Vasconcelos, N. (2011). Counting people with low-level features and Bayesian regression. *IEEE Transactions on Image Processing, 21*(4), 2160–2177.

Chen, B., Yan, Z., Li, K., Li, P., Wang, B., Zuo, W., et al. (2021). Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16065–16075).

Chen, J., Xiu, S., Chen, X., Guo, H., & Xie, X. (2021). Flounder-Net: An efficient CNN for crowd counting by aerial photography. *Neurocomputing, 420*, 82–89.

Chen, J., Wang, K., Su, W., & Wang, Z. (2022). SSR-HEF: Crowd counting with multi-scale semantic refining and hard example focusing. *IEEE Transactions on Industrial Informatics*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (pp. 886–893). IEEE.

Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2011). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(4), 743–761.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1627–1645.

Gao, J., Wang, Q., & Li, X. (2019). PCC-Net: Perspective crowd counting via spatial convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology, 30*(10), 3486–3498.

Gao, J., Wang, Q., & Yuan, Y. (2019). SCAR: Spatial-/channel-wise attention regression networks for crowd counting. *Neurocomputing* (pp.363: 1-8).

Hu, Y., Jiang, X., Liu, X., Zhang, B., Han, J., Cao, X., et al. (2020). Nas-count: Counting-by-density with neural architecture search. In *European conference on computer vision* (pp. 747–766). Cham: Springer.

Idrees, H., Saleemi, I., Seibert, C., & Shah, M. (2013). Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2547–2554).

Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., et al. (2018). Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 532–546).

Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., et al. (2020). Attention scaling for crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4706–4715).

Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. *Advances in Neural Information Processing Systems, 23*, 1324–1332.

Li, M., Zhang, Z., Huang, K., & Tan, T. (2008). Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *2008 19th international conference on pattern recognition* (pp. 1–4). IEEE.

Li, T., Chang, H., Wang, M., Ni, B., Hong, R., & Yan, S. (2014). Crowded scene analysis: A survey. *IEEE Transactions on Circuits and Systems for Video Technology, 25*(3), 367–386.

Li, Y., Zhang, X., & Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1091–1100).

Lin, S. F., Chen, J. Y., & Chao, H. X. (2001). Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 31*(6), 645–654.

Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., & Wu, H. (2019). Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3225–3234).

Liu, W., Salzmann, M., & Fua, P. (2019). Context-aware crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5099–5108).

Liu, Y. B., Jia, R. S., Liu, Q. M., Zhang, X. L., & Sun, H. M. (2021). Crowd counting method based on the self-attention residual network. *Applied Intelligence, 51*, 427–440.

Liu, Y., Cao, G., Shi, H., & Hu, Y. (2022). Lw-count: An effective lightweight encoding-decoding crowd counting network. *IEEE Transactions on Circuits and Systems for Video Technology, 32*(10), 6821–6834.

Ma, X., Du, S., & Liu, Y. (2019). A lightweight neural network for crowd analysis of images with congested scenes. In *2019 IEEE international conference on image processing (ICIP)* (pp. 979–983). IEEE.

Ma, Z., Wei, X., Hong, X., Lin, H., Qiu, Y., & Gong, Y. (2021). Learning to count via unbalanced optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 3, pp. 2319-2327).

Meng, Y., Zhang, H., Zhao, Y., Yang, X., Qian, X., Huang, X., et al. (2021). Spatial uncertainty-aware semi-supervised crowd counting. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 15549–15559).

Oh, M. H., Olsen, P., & Ramamurthy, K. N. (2020). Crowd counting with decomposed uncertainty. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, pp. 11799-11806).

Onoro-Rubio, D., & López-Sastre, R. J. (2016). Towards perspective-free object counting with deep learning. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14* (pp. 615-629). Springer International Publishing.

Pham, V. Q., Kozakaya, T., Yamaguchi, O., & Okada, R. (2015). Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE international conference on computer vision* (pp. 3253–3261).

Ryan, D., Denman, S., Sridharan, S., et al. (2015). An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding* (pp. 130:1-17).

Sam, D. B., & Babu, R. V. (2018). Top-down feedback for crowd counting convolutional neural network. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

Sam, D. B., Peri, S. V., Sundararaman, M. N., Kamath, A., & Babu, R. V. (2020). Locate, size, and count: Accurately resolving people in dense crowds via detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(8), 2739–2751.

Shi, X., Li, X., Wu, C., Kong, S., Yang, J., & He, L. (2020). A real-time deep network for crowd counting. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2328–2332). IEEE.

Sindagi, V. A., & Patel, V. M. (2017a). Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1–6). IEEE.

Sindagi, V. A., & Patel, V. M. (2017b). Generating high-quality crowd density maps using contextual pyramid cnns. In *Proceedings of the IEEE international conference on computer vision* (pp. 1861–1870).

Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., et al. (2021). Rethinking counting and localization in crowds: A purely point-based framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 3365–3374).

Wan, J., & Chan, A. (2020). Modeling noisy annotations for crowd counting. *Advances in Neural Information Processing Systems, 33*, 3386–3396.

Wan, J., Liu, Z., & Chan, A. B. (2021). A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1974–1983).

Wang, C., Zhang, H., Yang, L., Liu, S., & Cao, X. (2015). Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on multimedia* (pp. 1299–1302).

Wang, C., Song, Q., Zhang, B., Wang, Y., Tai, Y., Hu, X., et al. (2021). Uniformity in heterogeneity: Diving deep into count interval partition for crowd counting. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3234–3242).

Wang, M., Cai, H., Han, X., Zhou, J., & Gong, M. (2022). Stnet: Scale tree network with multi-level auxiliator for crowd counting. *IEEE Transactions on Multimedia*.

Wang, P., Gao, C., Wang, Y., Li, H., & Gao, Y. (2020). MobileCount: An efficient encoder-decoder framework for real-time crowd counting. *Neurocomputing, 407*, 292–299.

Wang, Q., Gao, J., Lin, W., & Yuan, Y. (2021). Pixel-wise crowd understanding via synthetic data. *International Journal of Computer Vision, 129*(1), 225–245.

Wu, X., Xu, B., Zheng, Y., Ye, H., Yang, J., & He, L. (2019). Video crowd counting via dynamic temporal modeling. *CoRR*.

Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., & Sebe, N. (2020). Reverse perspective network for perspective-aware object counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4374–4383).

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zhang, S., Wu, G., Costeira, J. P., & Moura, J. M. (2017). Understanding traffic density from large-scale web camera data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5898–5907).

Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 589–597).

Zheng, Z., Ni, N., Xie, G., et al. (2022). HARNet: Hierarchical adaptive regression with location recovery for crowd counting. *Neurocomputing* (pp. 513: 329-340).

A Weakly-Supervised Crowd Density Estimation Method Based on Two-Stage Linear Feature Calibration

Yong-Chao Li, Rui-Sheng Jia, Ying-Xiang Hu, and Hong-Mei Sun

***Abstract*—In a crowd density estimation dataset, the annotation of crowd locations is an extremely laborious task, and they are not taken into the evaluation metrics. In this paper, we aim to reduce the annotation cost of crowd datasets, and propose a crowd density estimation method based on weakly-supervised learning, in the absence of crowd position supervision information, which directly reduces the number of crowds by using the number of pedestrians in the image as the supervised information. For this purpose, we design a new training method, which exploits the correlation between global and local image features by incremental learning to train the network. Specifically, we design a parent-child network (PC-Net) focusing on the global and local image respectively, and propose a linear feature calibration structure to train the PC-Net simultaneously, and the child network learns feature transfer factors and feature bias weights, and uses the transfer factors and bias weights to linearly feature calibrate the features extracted from the Parent network, to improve the convergence of the network by using local features hidden in the crowd images. In addition, we use the pyramid vision transformer as the backbone of the PC-Net to extract crowd features at different levels, and design a global-local feature loss function (L_2). We combine it with a crowd counting loss (L_C) to enhance the sensitivity of the network to crowd features during the training process, which effectively improves the accuracy of crowd density estimation. The experimental results show that the PC-Net significantly reduces the gap between fully-supervised and weakly-supervised crowd density estimation, and outperforms the comparison methods on five datasets of ShanghaiTech Part A, ShanghaiTech Part B, UCF_CC_50, UCF_QNRF and JHU-CROWD++.**

Manuscript received June 22, 2023; revised August 12, 2023; accepted September 10, 2023. This work was supported by the Humanities and Social Science Fund of the Ministry of Education of China (21YJAZH077). Recommended by Associate Editor Xin Luo. *(Corresponding authors: Rui-Sheng Jia and Hong-Mei Sun.)*

Citation: Y.-C. Li, R.-S. Jia, Y.-X. Hu, and H.-M. Sun, "A weakly-supervised crowd density estimation method based on two-stage linear feature calibration," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 4, pp. 965–981, Apr. 2024.

Y.-C. Li was with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590. He is now with the Faculty of Information Science and Engineering, Ocean University of China, Qingdao 266000, China (e-mail: 202082060030@sdust.edu.cn).

R.-S. Jia and H.-M. Sun are with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China (e-mail: skd990551@sdust.edu.cn; skd991925@sdust.edu.cn).

Y.-X. Hu was with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590. She is now with the College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210000, China (e-mail: 202082060017@sdust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123960

***Index Terms*—Crowd density estimation, linear feature calibration, vision transformer, weakly-supervision learning.**

I. Introduction

With the increase of the global population and human social activities, large crowds often gather in public places, which brings huge hidden dangers to public safety. Therefore, determining how to accurately estimate crowd density has become an important research topic in the field of public safety. To train a robust and reliable network for accurate crowd density estimation, most existing crowd density estimation networks use a fully-supervised or semi-supervised training method, the network model is trained through the ground truth generated by manual annotation, which requires a lot of manpower, material and financial resources, and in large-scale dense crowd images, interference factors such as low resolution, object occlusion, and scale changes make it difficult to label each pedestrian in the crowd. Therefore, determining how to trade off the accuracy of crowd density estimation and dataset labeling cost, and save the dataset labeling cost without losing counting accuracy becomes a challenge.

The crowd density estimation method mainly obtains the number of crowds by extracting crowd information from the image. Existing crowd density estimation training methods mainly include fully-supervised methods [1]−[25] and semi-supervised methods [26]−[39]. The fully-supervised method is to obtain the ground truth by manually labeling each pedestrian in the image, and then training the network model through the ground truth, although this method shows high performance in crowd density estimation, it requires significant manpower, material and financial resources to label people in the image; the ground truth for the semi-supervised method is mainly divided into two types, that is, we mark all pedestrians in some images and mark some pedestrians in all images; this method is close to the fully-supervised method in crowd density estimation and shows good robustness, but this method still needs to label crowds in the image, and the training process is very cumbersome. Moreover, the problem faced by both fully-supervised and semi supervised methods is the limitation of the dataset. Then, the method for obtaining the distribution of the crowd changes, such with a change in the shooting perspective or the spatial distribution characteristics of the crowd, the ground truth obtained under the current labeling method, needs to be re-labeled, and the labeled ground truth will not be used to evaluate the counting perfor-

mance during the test process. This means that the ground truth labeled for each pedestrian is redundant. To reduce the cost of manual labeling, weakly-supervised training methods are proposed, and the main difference between these methods and the fully-supervised and semi-supervised methods are that the weakly-supervised methods do not require any manual annotation of the crowd location information at all, while the fully-supervised and semi-supervised methods require manual annotation of all or part of the crowd location information. In fact, without the demand for locations, the crowd numbers can be obtained in other economical ways. For instance, with an already collected dataset, the crowd numbers can be obtained by gathering the environmental information, e.g., detection of disturbances in spaces, or estimation of the number of moving crowds. Chan *et al.* [40] segment the scene by crowd motions and estimate the crowd number by calculating the area of the segmented regions. To collect a novel counting dataset, we can employ sensor technology to obtain the crowd number in constrained scenes, such as mobile crowd sensing technology [41]. Moreover, Sheng *et al.* [42] propose a GPS-less energy-efficient sensing scheduling to acquire the crowd number more economically. On the other hand, several approaches [43]−[46] prove that, with the estimated results, there is no tight bond between the crowd number and the location. The weakly-supervised labeling data in this paper, all of which were obtained from already collected datasets, use only the crowd's quantity labeling and drop the location labeling information.

However, although such weakly-supervised methods save the cost of dataset labeling, the ensuing problem is that the network does not know the characteristics of pedestrians at the beginning of the training process, due to the lack of the location information of the crowd as the training label, and the characteristics of pedestrians are learned only after several iterations, which leads to reduced sensitivity of the network to crowd features, and the convergence speed of the network becomes very slow, and the network model's ability to fit features is substantially reduced, which affects the accuracy of crowd density estimation. Therefore, the weakly-supervised approach of simply removing the crowd location information saves the cost of labeling the dataset, but limits the performance of the network and does not fundamentally solve the problem.

To solve the above problem, inspired by the optimal iterative learning control methods [47]−[49], reaction–diffusion neural networks [50] and the model latent factor analysis [51]−[54], we reconsider the training approach of the crowd density estimation model and also sample weakly-supervised data labels, i.e., we use only the number of pedestrians in the image as supervised. However, to compensate for the missing crowd location information and to improve the convergence speed of the network and the feature fitting ability, we designed a novel and effective training method, using a parent-child network with the same parameters to learn different features in the crowd, and then using a linear transformation to correct the information location of the features extracted by the parent network using hidden features learned by the child network, to accelerate the network's ability to adapt to the features. Our training method, significantly improves the convergence speed of the network; the network performance as well as the counting accuracy, is not much different from the fully-supervised method, and, since the parent network has the same parameters as the child network, the increment of the number of parameters of the parent-child network model is very small compared to the number of parameters of the parent network, and the increase in the number of parameters is well within the acceptable range compared to the improved performance of the network. To address the above problems, this paper designs a crowd density estimation method based weakly-supervised learning, which trains the network by correlating between global and local image features to improve the performance of the network model. The main contributions of this paper are as follows:

1) We design a weakly-supervised crowd density estimation method, which based on using only the number of crowd as supervised information without using location label supervision. It omits the manual labeling work without losing the crowd density estimation performance and greatly saves the cost of network training compared to existing fully-supervised methods.

2) We design a novel and effective training approach by designing a parent-child network, which uses incremental learning, by the characteristic linear calibration structure to enhance the adaptability of the network to hidden features using transfer factors and offset weights. It improves the performance of weakly-supervised learning methods, and we verify its effectiveness in this task.

3) We design a loss function that adds the error between the parent network features and child network features (L_2) to the ground truth and predicted counting error (L_C), and use gradient descent to optimize the features extracted by the parent-child network to accelerate the convergence speed of network training and improve the accuracy of crowd density estimation.

II. Related work

1) Fully-Supervised/Semi-Supervised Crowd Density Estimation Methods: With the development of big data, machine learning, and convolutional neural networks [55]−[61], a large number of convolutional neural network (CNN)-based crowd density estimation methods have been proposed. Basic CNN is first applied to crowd density estimation, such as CNN-boosting [1], Wang *et al.* [2], these networks use basic CNN layers, including convolutional layer, pooling layer, fully connected layer, no additional feature information is required, which are simple and easy to implement, but the crowd estimation accuracy is low. Multi-column CNN is subsequently widely used, such as MCNN [3], MBTTBF [4], Multi-scale-CNN [5], CP-CNN [6], DADNet [7], these networks usually use different columns to capture multi-scale information. However the information captured by different columns is redundant and wastes many training resources. To solve the problem of redundant feature extraction by multi-column CNN, Single-column CNN is applied to crowd density estimation, such as CSRNet [8], SANet [9], SPN [10], CMSM [11], TEDnet [12], and IA-MFFCN [13]. These networks usually

deploy a single deeper CNN instead of the bloated structure of multi-column network architecture, do not increase the complexity of the network, and have higher training efficiency, so it has received extensive attention. However, with the development of the density map-based method, the background noise in the image seriously affects the display of the detailed information of the crowd distribution, how to filter out the background noise to highlight the crowd location information has become a challenge.

Therefore, attentional mechanisms have been widely introduced into crowd density estimation tasks, and, attentional mechanisms can supplement the features extracted by the backbone network or the head network by providing the capability to encode distant dependencies or heterogeneous interactions to highlight the head position. ADcrowdNetp designs an attention image generation structure [14], attentional neural field (ANF) uses local and global self-attention to capture long-range dependencies [15], attention guided feature pyramid network (AP-FPN) proposes an attention guided feature pyramid network [16], which adaptively combines high-level and low-level features to generate high-quality density maps with accurate spatial location information, and multi-scale feature pyramid network (MFP-Net) designs a feature pyramid fusion module using different depth and scale convolution kernels [17] where the receptive field of CNN is expanded to improve the training speed of the network, PDANet uses a feature pyramid to extract crowd features of different scales to improve counting accuracy [18], and SPN uses the scale pyramid network to effectively capture multi-scale crowd characteristics [10], and obtain more comprehensive crowd characteristic information. Meanwhile, researchers have attempted to transfer Transformer models in the field of natural language processing to the task of crowd density estimation [19]−[23], [62]−[66]. Transformer uses self-attention to capture the global dependency between input and output, where the advantage is that it is not limited by local interactions, can mine long-distance dependencies and can perform parallel calculations, where the most appropriate inductive bias can be learned according to different task objectives, thereby capturing the global context information of the image and modeling the dependencies between global features, which is a good solution to the limited receptive field of CNN, especially in the presence of uneven scales in dense crowds. In 2020, Dosovitskiy *et al.* [19] proposed the vision transformer (ViT) model, an image classification method based entirely on the self-attention mechanism, which is also the first work of Transformer to replace convolution. In 2021, Sun *et al.* [24] demonstrated the importance of global contextual information in the task of crowd density estimation. In 2021, TDCrowd combines ViT and density map to estimate the number of people in the crowd [25], which solves the problem of background noise interference in crowd density estimation, and improves the accuracy of crowd density estimation.

However, the aforementioned CNN or ViT methods require a large number of labels for training, and labeling the crowd density estimation dataset is a laborious task.

2) Weakly-Supervised Crowd Density Estimation Methods: To reduce the cost of labeling the dataset, some weakly-supervised crowd density estimation methods have been developed. In the weakly-supervised methods, there is no need to label any crowd location information, and image-level count labels are used as the weakly-supervised signal for training. In 2016, Borstel *et al.* [37] proposed a weakly-supervised density estimation method based on the Gaussian process, using the number of objects as the label to train the network, but this method partitions the image, so that different partitions will repeat the same target, causing the estimated number of targets to be higher than the actual number. In 2019, Ma *et al.* [38] proposed a weakly-supervised density estimation method using Bayesian loss, which performs expectation calculation from the probability density map estimated from the network, and regresses to estimate the number of people in the crowd, which improves the counting efficiency under the weakly-supervised method. In 2019, Sam *et al.* [36], designed an auto-encoder to train the network in a weakly-supervised way, updating only a small number of parameters during training, in an attempt to achieve a nearly un-supervised method for crowd density estimation. In 2020, Yang *et al*. [39], proposed a network based on soft label ranking, which highlights the supervision of crowd size based on the original crowd density estimation network. In 2020, Sam *et al.* [29], by matching statistics of the distribution of labels, proposed a weakly-supervised training method that does not use image-level location labeling information. To ease the overfitting problem, in 2019, Wang *et al.* [27] explores the generation of synthetic crowd images to reduce the burden of annotation and alleviate overfitting. With the application of ViT in the field of crowd density estimation, in 2021, TransCrowd applied ViT to crowd density estimation for the first time [21], and proposed a weakly-supervised counting method, which greatly improved the accuracy of crowd density estimation in the weakly-supervised mode, but was affected by the simple structure of the model, where extraction of features was limited.

Compared with previous weakly-supervised methods, we proposed a weakly-supervised method based on linear calibration of parent-child network features, which can effectively reduce labeling cost during training, while maintaining state-of-the-art performance, achieving an optimal trade-off between crowd density estimation accuracy and dataset labeling cost.

III. Proposed Method

A. Overview of the Network Architecture

To improve the convergence speed of the network under the weakly-supervised training method, we propose a parent-child network (PC-Net). It exploits the correlation between global and local features in images to enhance the network's ability to fit the features by incrementally learning and continuously linearly correcting the features extracted by the network. The proposed PC-Net structure is shown in Fig. 1. The PC-Net achieves a better balance between accuracy and training costs. Specifically, PC-Net is divided into two parts, the Parent network and Child network, which have the same backbone net-

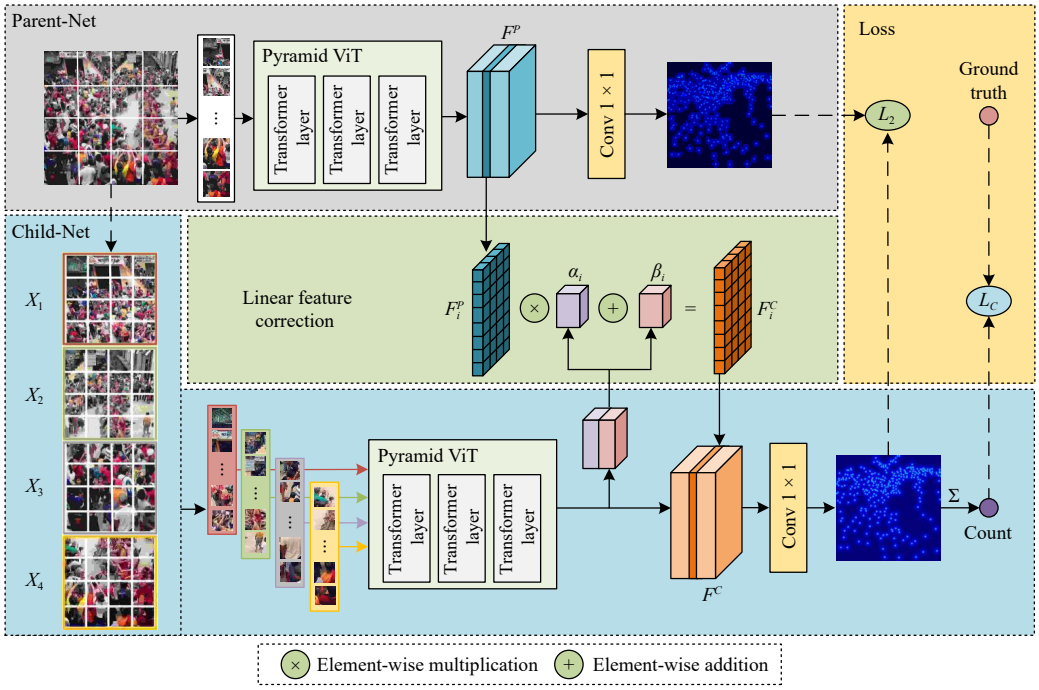


Fig. 1. Overview of the proposed PC-Net architecture. First, the Parent-Net is trained with the global images. Second, learnable parameters α and β are used to corrective features, which are defined according the Parent-Net, namely, for a Parent-Net feature $F_i^P \in F^P$ (i is the index of feature channel), there are a α_i and a β_i that are used to generate a Child-Net feature $F_i^C \in F^C$ by the linear correction. Third, after loading the correction feature F^C to the Child-Net, the partial images are feed into the Child-Net to update the correction parameters.

work. We design a pyramid vision transformer as the feature extraction backbone network to extract crowd features at different levels. In the process of network training, the Parent network learns crowd features through global images, while the Child network learns feature transfer factors and feature bias weights from local images. Then, the crowd features learned by the Parent network are corrected by a linear correction structure to obtain a feature map that contains richer and more accurate global contextual information. Meanwhile, during the training of the network, the Parent and Child networks are updated with the learned weights by gradient descent using different losses to improve the accuracy of the crowd density estimation. Finally, a 1 × 1 convolutional layer is used to output the final density map. In the following sections, we describe our framework in detail.

B. Backbone Network

In PC-Net, the subject network is divided into two parts, Parent-Net and Child-Net. In order to use incremental learning, linearly correcting the crowd features, Parent-Net and Child-Net have the same network structure. In order to adapt to the problem of scale variability existing in crowd images, a pyramid vision transformer feature extraction backbone network is designed in this paper to extract crowd features at different levels, as shown in Fig. 1, while using a multi-scale window to restrict the calculation of the vision transformer's self-attention mechanism to non-overlapping local regions, which improves computational efficiency. Since the vision transformer can not directly process 2D images, an image preprocessing process is required to convert 2D images into 1D image block sequences before the images are input to pyramid vision transformer. The process of image preprocessing and the structure of the pyramid vision transformer are shown as follows.

1) Image Partition

Before the image is input into the pyramid vision transformer, the 2D image is converted into a 1D image block sequence. To improve the computational efficiency, the input image is divided into $N \times N$ fixed windows, and the image in the window is divided into image blocks of fixed size, and the self-attention calculation is performed in each window, as shown in Fig. 2.

Specifically, input image $I \in \mathbb{R}^{H\times W\times 3}$, divide I into $N\times N$ fixed windows I^n, $n \in [1, N\times N]$, and divide I into $\frac{H}{K}\times\frac{W}{K}$ patches, where the size of each patch is $K\times K\times 3$, and each window I^n contains $\frac{H}{NK}\times\frac{W}{NK}$ patches. Convert them into a 1D images patch sequence $x^n \in \mathbb{R}^{L\times D}$, $n \in [1, N\times N]$, $L=\frac{HW}{N^2K^2}$, $D = K\times K\times 3$. The location mapping of x^n using a learning-based projection $f: x_i^n \to e_i^n$, $i \in [1, D]$ translates the spatial and channel features of the ith image block in the nth window into the embedded features of the nth set of ith vectors, as follows:

$$Z_0^n = \left[e_1^n + p_1^n,\ e_2^n + p_2^n,\ \ldots,\ e_L^n + p_L^n\right],\ n \in [1, N\times N] \quad (1)$$

where Z_0^n denotes the sequence with position information input to the transformer-encoder for the nth window, e_i^n denotes the ith image block in the nth window, p_i^n denotes the position information of the ith image block in the nth window.

2) Pyramid Vision Transformer

When extracting multi-scale crowd features, a multi-layer pyramid vision transformer structure is used. Between layers,

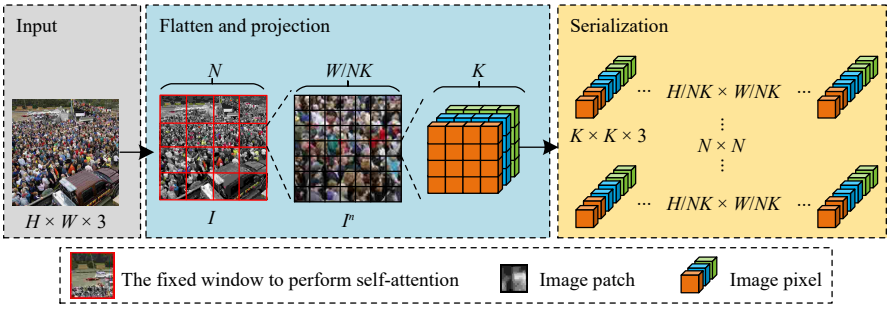


Fig. 2. The process of the image partition.

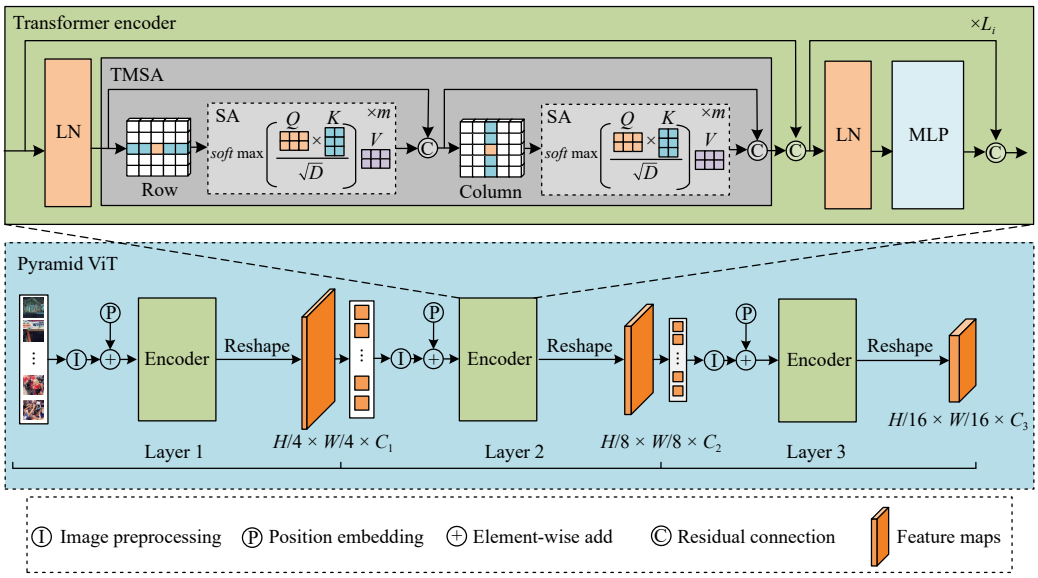


Fig. 3. The structure of pyramid vision transformer. The feature map of each layer needs to be partitioned first to convert the 2D image into a 1D sequence, and then perform feature reshape on the processed 1D sequence to generate 2D features.

the scale of the feature map is controlled by a strategy of progressive shrinkage. Simultaneously, the scheme using multi-scale windows restricts the self-attention calculation process to non-overlapping local windows, and expands the window layer by layer through cross-window connections, which improves the computational efficiency. The method in this paper designs a three-layer transformer-encoder structure, as shown in Fig. 3.

Specifically, the size of the input image is $H \times W \times 3$, the size of the output feature map F_i after Layer i is $H_i \times W_i \times C_i$, and the size of the image patch in Layer i is $K_i \times K_i \times 3$, where $K_1 = 4$, $K_2 = 2$, $K_3 = 2$, the number of Layer i windows is $N_i \times N_i$, where $N_1 = 4$, $N_2 = 2$, $N_3 = 1$, each window of Layer i contains $\frac{H_{i-1}W_{i-1}}{(K_iN_i)^2}$ images patch, and linearly project the image patch into a 1D sequence and embed position information, after the transformer-encoder extracts features, visualize feature sequence rearrangement as feature maps, where C_i is less than C_{i-1}. The transformer-encoder of Layer i includes L_i layers twin multi-head attention mechanism (TMSA) and multi-layer perceptron (MLP), where $L_1 = 2$, $L_2 = 6$, $L_3 = 2$, and each layer is processed by layer normalization (LN) and residual connection. Before TMSA and MLP, LN is used to normalize the feature sequence, which makes the training process more stable and effectively avoids the problem of gradient disappearance or gradient explosion. And residual connection is used after TMSA and MLP, and the features processed by TMSA and MLP are superimposed with the features before processing to avoid the degradation problem of matrix weights in the network. The calculation process is as follows:

$$Z_l^{n'} = TMSA\left(LN\left(Z_{l-1}^n\right)\right) + Z_{l-1}^n \tag{2}$$

$$Z_l^n = MLP\left(LN\left(Z_l^{n'}\right)\right) + Z_l^{n'}. \tag{3}$$

In the formula, Z_{l-1}^n is the output of the nth window in the lth layer, TMSA contains two multi-head attention modules (MSA), as shown in Fig. 3. First, the first MSA performs self-attention calculation on each row, keeps the feature blocks of different rows independent, and aggregates the context information between feature blocks on the horizontal scale. Then, the second MSA performs self-attention calculation for each column, keeps the feature blocks of different columns independent, and aggregates the context information between feature blocks on the vertical scale. Finally, the outputs of the two MSA are concatenated to form a global receptive field, covering the crowd feature information of horizontal and vertical dimensions. The calculation process is as follows:

$$TMSA\left(Z_{l-1}^{n}\right)=\left[MSA_1\left(Z_{l-1}^{n}\right);\ MSA_2\left(Z_{l-1}^{n}\right)\right]W,\ W\in\mathbb{R}^{D\times D}. \tag{4}$$

MSA contains m self-attention (SA) modules. In each independent SA, input sequence Z_{l-1}^{n}, calculate the query (Q), key (K) and value (V) of the sequence, where the process is as follows:

$$[Q,K,V]=Z_{l-1}^{n}W_{Q,K,V},\ W_{Q,K,V}\in\mathbb{R}^{D\times\frac{D}{m}} \tag{5}$$

$$SA\left(Z_{l-1}^{n}\right)=\text{soft}\max\left(\frac{QK^{T}}{\sqrt{D}}\right)V. \tag{6}$$

In the formula, $W_{Q,K,V}$ are learnable matrices, and the outputs of m self-attention modules are connected in series, which can be expressed as

$$MSA\left(Z_{l-1}^{n}\right)=\left[SA_1\left(Z_{l-1}^{n}\right);\ SA_2\left(Z_{l-1}^{n}\right);\ \cdots;\ SA_{12}\left(Z_{l-1}^{n}\right)\right]W$$
$$W\in\mathbb{R}^{D\times D}. \tag{7}$$

MLP contains two linear layers with the Gaussian error linear unit (GELU) activation function. This paper uses the GELU activation function of a standard normal distribution, as shown in (8)

$$GELU(x)=0.5x\left(1+\tanh\left[\sqrt{\frac{2}{\pi}}(x+0.44715x^{3})\right]\right). \tag{8}$$

The first linear layer expands the dimension of the feature sequence from D to 4D, and the second linear layer shrinks the dimension of the feature sequence from 4D to D.

C. Linear Feature Calibration

In order to improve the convergence speed and feature fitting ability of the weakly-supervised crowd counting method during training, we propose a linear feature calibration structure. To achieve feature calibration and transfer between Parent-Net and Child-Net, we consider that the feature parameters of Parent-Net and Child-Net belong to the same linear space V^n (n represents the number of channels of features). Each channel feature in the Child-Net can be transferred from the corresponding channel feature in the Parent-Net by a linear transformation. Fig. 4 shows how the Child-Net feature's parameters are transferred from the Parent-Net by a linear calibration.

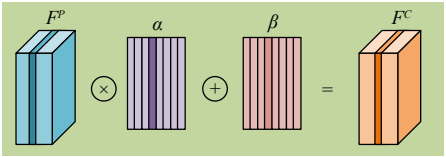


Fig. 4. The process of the linear feature calibration.

In Fig. 4, we define the channel features in Parent-Net as $F^P\in\mathbb{R}^{h\times w\times n}$ (h, w, n represent the length, width, and number of channels of the features, respectively), the feature transfer factors as $\alpha\in\mathbb{R}^{1\times1\times n}$, and the feature bias weights as $\beta\in\mathbb{R}^{1\times1\times n}$, so the process of linear feature correction can be expressed as

$$F^{C}=\left[F_1^{P}\times\alpha_1+\beta_1,\ldots,F_n^{P}\times\alpha_n+\beta_n\right]$$
$$=\left[\begin{bmatrix} f_{11}^{1} & \cdots & f_{1w}^{1}\\ \vdots & \ddots & \vdots\\ f_{h1}^{1} & \cdots & f_{hw}^{1}\end{bmatrix}\times\alpha_1+\beta_1,\ \ldots,\ \begin{bmatrix} f_{11}^{n} & \cdots & f_{1w}^{n}\\ \vdots & \ddots & \vdots\\ f_{h1}^{n} & \cdots & f_{hw}^{n}\end{bmatrix}\times\alpha_n+\beta_n\right]. \tag{9}$$

In the formula, $f_{hw}^{i}\in F^{P}(i\le n)$. The advantage of using linear feature calibration to train the network is that Child-Net inherits the crowd features in Parent-Net well, and retains an extremely strong generalization ability, which continuously improving the calibration of crowd features through transfer factors and bias weights, improving the feature fitting ability of the network. Meanwhile, since Parent-Net and Child-Net have the same backbone network, and the features in Child-Net are transferred through the features in Parent-Net, the parameters learned by the two parts of the network are different. Taking a simple CNN model as an example, assuming that there are L layers of convolution, where the size of each convolutional kernel is $K\times K$, and the number of output channels is fixed to N. Then, the number of Parent-Net parameters is $N\times K\times K\times L$, while the number of Parent-Net parameters is $N\times2\times L$, because Child-Net only needs to learn the transfer factor and bias weights, so the number of Parent-Net parameters is only $2/(K\times K)$ of Parent-Net parameters, which is only 2/9 when the size of convolutional kernel is 3×3. Therefore, the increase of the number of parameters is within an acceptable range.

D. Loss Function

In order to further strengthen the method proposed in this paper, we make full use of the correlation between the local and global crowd feature information to train the network, and improve the accuracy of crowd density estimation. The comprehensive loss function is designed, which consists of L_C loss function and L_2 loss function, as shown in (10)

$$Loss(\theta)=\partial L_{\text{c}}(\theta)+(1-\partial)L_2(\theta). \tag{10}$$

In the formula, L_C is the counting loss of the PC-Net estimated number of people with the ground truth, and L_2 is the MSE loss between PC-Net predicted density map and parent-net predicted density map, where, the L_C counting loss can be expressed as

$$L_C(\theta)=\frac{1}{N}\sum_{i=1}^{N}\left|\frac{F_Y(X_i,\theta)-Y_i}{Y_i}\right|^{2}. \tag{11}$$

In the formula, N denotes the number of images in the training set, $F_Y(X_i,\theta)$ denotes the estimated number of people obtained from the X_i ($i=1,\ldots,N$) images, and θ denotes a set of parameters that can be learned; Y_i denotes the true number of people in the X_i ($i=1,\ldots,N$) images. L_2 loss can be expressed as

$$L_2(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\|Z(X_i,\theta) - Z_P(X_i,\theta)\|_2^2. \quad (12)$$

In the formula, N denotes the number of images in the training set, X_i represents the ith image of the input, θ denotes a set of parameters that can be learned, and $Z(X_i,\theta)$ denotes the prediction result of PC-Net and $Z_P(X_i,\theta)$ denotes the prediction result of Parent-Net.

E. Crowd Density Map Generation

The crowd features extracted by PC-Net contains the location information of each pedestrian. We use a focal inverse distance transform (FIDT) to process the features to generate a visualized crowd density map [67]. The specific process can be expressed as follows: if there are Z pedestrian feature points in an image, the following processing is performed on the feature images:

$$P(x,y) = \min_{(x',y')\in Z}\sqrt{(x-x')^2+(y-y')^2} \quad (13)$$

$$I = \frac{1}{P(x,y)^{(AP(x,y)+B)}+C}. \quad (14)$$

In (13), Z denotes the set of all crowd feature points, and for any feature point (x, y), the Euclidean distance $P(x, y)$ is calculated with its nearest feature point (x', y'). Since the distance between feature points varies greatly, it is difficult to perform distance regression directly, so the inverse function is used for regression, as shown in (14), where I is the processing result of FIDT, C is an additional constant, usually set to 1, to avoid division by 0 in the calculation process, and $P(x, y)$ is exponentially processed to slow down the decay of the crowd head information, and I is visually displayed to generate a visual crowd density map. Finally, the predicted crowd density values are obtained by 2D integrating and summing the generated density maps. In the experiments, $A = 0.02$, $B = 0.75$ were set.

IV. Experiments

A. Training Process

In the training phase, one iteration updates parameters for two models. As shown in Fig. 1, first, the data are fed into Parent-Net for training, and the global feature F^P is optimized using the gradient descent method, as follows:

$$F^P = F^{P'} - \varepsilon \times \nabla L_C(\theta). \quad (15)$$

In the formula, εdenotes the learning rate of Parent-Net, and L_C is the count loss of the Parent-Net estimated number of crowd with Ground Truth. Second, we use the Linear Feature Calibration structure to transfer F^P channel-by-channel into Child-Net to obtain F^C, the process of transfer, as shown in (9). Since the transfer factor α and the bias weight β used in linear feature calibration need to be learned by Child-Net, we need to feed the local image data into Child-Net and optimize F^C with the gradient descent method, as follows:

$$F^C = F^{C'} - \mu \times \nabla \mathrm{Loss}(\theta). \quad (16)$$

In the formula, μdenotes the learning rate of Child-Net. Loos is the value of the integrated loss function designed in this paper. In the testing phase, we use the best-performing model on the test set to make an inference.

B. Training Hyper-Parameter Settings

During training we use the Adam optimizer, Batch_size is set to 16, the learning rate εin the Parent-Net and μ in the Child-Net are initialized as 0.0001, reduced by 0.5 times after every 50 epochs, where the GELU function is used as an activation function to improve the training speed and effectively avoid the disappearance and explosion of the gradient. We use l2 regularization of 0.0001 to avoid over-fitting. Since the images in the dataset have different resolutions, the resolution of all images is adjusted to 768 × 768. The experimental environment is shown in Table I.

TABLE I
EXPERIMENTAL ENVIRONMENT (TABLE I INTRODUCES THE EXPERIMENTAL ENVIRONMENT PARAMETERS FROM THE ASPECTS OF SYSTEM, FRAME, LANGUAGE, CPU, GPU AND RAM)

Name	Parameter
System	Windows 11
Frame	Pytorch
Language	Python
CPU	Intel (R) Core (TM) i7-10870H CPU @ 2.50GHz
GPU	NVIDIA GEFORCE GTX 3060
RAM	16.00 GB

C. Datasets

In this work, extensive experiments are conducted on five crowd datasets of ShanghaiTech Part A, ShanghaiTech Part B, UCF_CC_50, UCF_QNRF and JHU-CROWD++. Unlike fully-supervised methods, only count-level labels are used as supervision information in the training process. Choose a representative crowd image on each dataset, as shown in Fig. 5. The crowd images in each dataset have different degrees of uneven crowd scale variation.

1) ShanghaiTech [3]: It has 1198 crowd images with a total of 330165 people. The dataset contains two parts, A and B. Part A includes 482 highly crowded crowd images, of which 300 form the training dataset and the remaining 182 form the testing dataset; Part B includes 716 relatively sparse crowd images, of which 400 images form the training dataset, and the remaining 316 images form the testing dataset.

2) UCF_CC_50 [68]: It has 50 crowd images, these images have different resolutions and different viewing angles. The number of pedestrians per crowd image varies from 94 to 4543, with an average of 1280 pedestrians per image. Due to the limited number of images in this dataset and the large span of the number of people in the image, five-fold cross-validation is used in this dataset.

3) UCF_QNRF [69]: It has 1535 crowd images with a total of 12 500 people, of which 1201 form the training sample set and the remaining 334 form the test sample set. The number of pedestrians per crowd image varies from 49 to 12 865, with an average of 815 pedestrians per image.

4) JHU-CROWD++ [70]: It is an unconstrained dataset

Fig. 5. Crowd images from five crowd datasets. (a) From the ShanghaiTech Part A dataset; (b) From the ShanghaiTech Part B dataset; (c) From the UCF_CC_50 dataset; (d) From the UCF_QNRF dataset; (e) JHU-CROWD++ dataset.

with 4372 images that are collected under various weather-based conditions such as rain, snow, etc. and contains 2722 training images, 500 validation images, and 1600 testing images. This dataset contains 1.5 million annotations at both image level and head-level. The total number of people in each image ranges from 0 to 25 791.

D. Evaluation Metric

In this paper, we use mean absolute error (MAE), mean squared error (MSE), and mean absolute percentage error (MAPE) as evaluation metrics for PC-Net performance. MAE is the average absolute value of the difference between the target and estimated densities, and it is the average L_1 loss between the target and estimated densities. It can highlight outliers in the data, and its value is not affected by the influence of outliers, making it more robust in evaluating algorithm performance. MSE is the average squared value of the difference between the target density and the estimated density, and it is the average L_2 loss between the target density and the estimated density, which can penalize larger error values. MSE usually magnifies the effect of squared error to make it easier to distinguish between models with larger error values. MAPE is a measure of the relative error between the estimated and actual values, which makes it easier to compare the variability of algorithms on different datasets, and it uses the percentage error to measure the prediction error, which is more convenient in practice, more intuitive, easy to explain. MAPE can avoid the problem of "mean squared error inflation" that tends to occur in MSE, i.e., when there are outlier values in the dataset, as the impact on MAPE is smaller. In summary, the three metrics MAE, MSE, and MAPE are chosen to evaluate the algorithm in this paper, which can well demonstrate the robustness as well as the accuracy of PC-Net. The calculation is shown as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{C}_i - C_i\right| \tag{17}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{C}_i - C_i\right)^2} \tag{18}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\hat{C}_i - C_i}{C_i}\right|. \tag{19}$$

In the formula, N represents the number of test images, C_i represents the actual number of people in the ith image, and $\hat{C}_i$ represents the estimated number of people in the ith image. When the values of MAE, MSE and MAPE are smaller, the error between the estimated number of people and the actual number of people is smaller, indicating that the effect of the experiment is better.

E. Experiment 1: Comparisons With State-of-the-Art Methods

The ShanghaiTech dataset is a crowded and multi-scale dataset, to verify the counting performance of PC-Net. Experiments are performed on this dataset and compared with state-of-the-art methods, and the results of MAE, MSE and MAPE are given in Table II. The UCF_CC_50 dataset includes 50 grayscale images, where the images have different resolutions and viewing angles, which is a very challenging dataset with various crowd scenes and a limited total number of images; Therefore, five-fold cross-validation is performed to maximize the use of samples, and the dataset is randomly divided into 5 equal parts. Each part contains 10 images, four of which are used as the training dataset, and the remaining one is used as the testing dataset, where a total of five trainings and testings are performed. Finally the average value of the error index is taken as the final experimental result, and compared with state-of-the-art methods. The results of MAE, MSE and MAPE are given in Table II. UCF_QNRF dataset is also a crowded and multi-scale dataset, which is collected from three different datasets and includes various scenes around the world. The total number of images and the total number of people far exceed the first three datasets, and compared with state-of-the-art methods, the results of MAE, MSE and MAPE are given in Table II. JUU-CROWD++ is a super large dataset, which contains crowd images under various complex weather conditions. Compared with state-of-the-art methods, the results of MAE, MSE and MAPE are given in Table II.

1) Performance on the ShanghaiTech Dataset: In this paper, PC-Net is compared with state-of-the-art methods, and the results are show shown in Table II**,** where we divide these methods into two groups. The first group is the fully-supervised methods, which uses location information and population number information as supervised information. The second group is the weakly-supervised methods, which uses only population number information as supervised information.

TABLE II
COMPARISON OF PC-NET AND THE STATE-OF-THE-ART METHODS ON THE SHANGHAITECH, UCF_CC_50, UCF-QNRF AND JHU_CROWD++ DATASETS. L DENOTES THE TRAINING LABEL CONTAINS LOCATION INFORMATION, AND C DENOTES THE TRAINING LABEL CONTAINS POPULATION NUMBER INFORMATION. RED AND BLUE INDICATE THE FIRST AND THE SECOND-BEST PERFORMANCES, RESPECTIVELY

Method	Venue	Label	Part A	Part B	UCF_CC_50	UCF_QNRF	JHU_CROWD++
		L/C	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE
MCNN [3]	CVPR16	+/+	110.2/173.2/28.2%	26.4/41.3/27.1%	377.6/509.1/35.9%	277.0/426.0/32.5%	160.6/377.7/24.4%
CSRNet [8]	CVPR18	+/+	68.2/115.0/15.8%	10.6/16.0/9.3%	266.1/397.5/26.3%	–/–/–	72.2/249.9/26.4%
CFF [71]	ICCV19	+/+	65.2/109.4/14.9%	7.2/12.2/6.1%	–/–/–	93.8 /146.5/13.1%	–/–/–
TEDnet [12]	CVPR19	+/+	64.2/109.1/14.7%	8.2/12.8/7.1%	249.4/354.5/24.2%	113.0/188.0/16.1%	–/–/–
PCC-Net [72]	TCSVT19	+/+	73.5/124.0/17.2%	11.0/19.0/9.7%	240.0/315.5/23.1%	148.7/247.3/22.3%	–/–/–
RPNet [73]	CVPR20	+/+	61.2/96.9/13.9%	8.1/11.6/7.0%	–/–/–	–/–/–	–/–/–
ASNet [74]	CVPR20	+/+	57.8/90.1/13.0%	–/–/–	174.8/251.6/15.8%	91.5/159.7/12.6%	–/–/–
AMRNet [75]	ECCV20	+/+	61.5/98.3/13.9%	7.0/11.0/6.0%	184.0/256.8/16.8%	86.6/152.2/11.9%	–/–/–
DM-Count [76]	NeurIPS20	+/+	59.7/95.7/13.5%	7.4/11.8/6.3%	211.0/291.5/19.6%	85.6/148.3/11.7%	–/–/–
GL [77]	CVPR21	+/+	61.3/95.4/13.9%	7.3/11.7/6.3%	–/–/–	84.3/147.5/11.5%	59.9/259.5/20.1%
SFCN [78]	IJCV21	+/+	64.8/107.5/14.9%	7.6/13.0/6.5%	214.2/318.2/20.1%	102.0/171.4/14.3%	62.9/247.5/22.2%
LW-Count [79]	TCSVT22	+/+	69.7/100.5/16.2%	10.1/12.4/8.9%	239.3/307.6/23.1%	149.7/238.4/22.5%	90.2/311.8/35.2%
SSR-HEF [80]	TII22	+/+	55.0/88.3/12.3%	**6.1/9.5/5.2%**	173.3/260.4/15.7%	**70.2/128.6/9.4%**	**51.3/101.6/17.4%**
ST-Net [81]	TMM22	+/+	**52.9/83.6/11.2%**	**6.3/10.3/5.3%**	**162.0/230.4/14.5%**	87.9/166.4/12.9%	–/–/–
MFFNet [82]	TIM23	+/+	107.3/188.5/27.3%	12.7/35.2/11.4%	323.5/482.8/33.9%	142.1/271.1/21.1%	–/–/–
CTASNet [83]	TCSVT23	+/+	**54.3/87.8/12.2%**	6.5/10.7/5.5%	**158.1/221.9/14.1%**	**80.9/139.2/11.0%**	–/–/–
Scale-Aware [84]	CEE23	+/+	58.6/98.5/13.2%	7.5/8.5/6.5%	210.2/260.8/19.7%	–/–/–	–/–/–
L2R [26]	TPMAI19	–/+	73.6/112.0/17.2%	13.7/21.4/12.4%	279.6/408.1/27.9%	124.0/196.0/17.9%	–/–/–
Yang *et al.* [39]	ECCV20	–/+	104.6/145.2/26.4%	12.3/21.2/11.0%	–/–/–	–/–/–	–/–/–
MATT [33]	PR21	–/+	80.1/129.4/19.0%	11.7/17.5/10.4%	355.0/550.2/38.2%	–/–/–	–/–/–
SUA [34]	ICCV21	–/+	68.5/125.6/16.4%	12.3/17.9/11.1%	–/–/–	119.2/213.3/17.1%	–/–/–
TransCrowd [21]	SCIS22	–/+	66.1/105.1/15.2%	9.3/16.1/8.1%	–/–/–	97.2/168.5/13.4%	56.8/193.6/19.6%
PC-Net (Ours)	–	–/+	**58.7/89.5/13.3%**	**7.3/10.4/6.3%**	**217.3/309.7/20.5%**	**84.8/148.9/11.6%**	**52.2/103.9/17.8%**

According to Table II, PC-Net is very competitive with the first group. Although MAE, MSE, and MAPE do not achieve the optimal results, they are more advantageous than most of the fully-supervised methods such as GL, LW-Count, etc., PC-Net largely closes the gap in counting performance between weakly-supervised methods and fully-supervised methods, and its labeling cost is much lower than that of fully-supervised methods. The advantage of PC-Net over the second group is more obvious, as MAE, MSE and MAPE are better than the existing weakly-supervised methods. On Part A, MAE, MSE and MAPE are improved by 11.2%, 14.8% and 12.5%, respectively, and on Part B, MAE, MSE and MAPE are improved by 21.5%, 35.4% and 22.2%, respectively. Thus, it is demonstrated that PC-Net can achieve the best density estimation performance with a weakly-supervised training mode by training with feature linear correction. Figs. 6(a) and 6(b) shows some visualization results of PC-Net on Part A and Part B datasets.

It can be seen that PC-Net performs well on two datasets, generating accurately distributed density maps with high resolution, and the prediction results are close to the true values. Comparing Figs. 6(a) and 6(b), the ShanghaiTech Part A dataset is extremely crowded and has little change in crowd scale, while the ShanghaiTech Part B dataset is relatively sparse but has large change in crowd scale, which indicates that PC-Net can be a good fit for different degrees of crowd scale changes. The third column of Fig. 6, gives the heat map of the Parent-Net output, and we use the red box to mark out the obvious misidentification or omission of identification. It can be seen that extracting the crowd features using only Parent-Net can easily produce misidentification of crowd features. The process of crowd feature correction and transfer, on the other hand, corrects the location information of the crowd well, which further compensates for the lack of crowd location information under the weakly-supervised crowd counting method and further improves the accuracy of the crowd counting.

2) Performance on the UCF_CC_50 Dataset: According to Table II, under the weakly-supervised training, compared to the second group, PC-Net outperforms other weakly-supervised methods on the UCF_CC_50 dataset, with MAE, MSE and MAPE improving by 38.8%, 43.7% and 46.3%, respectively, which proves the superiority of PC-Net. However, compared with the first group, PC-Net has obvious shortcomings, probably because the data in this dataset is limited and the number of people included in the images spans a relatively large range. The prediction results are not stable enough, and there are a small number of images with large

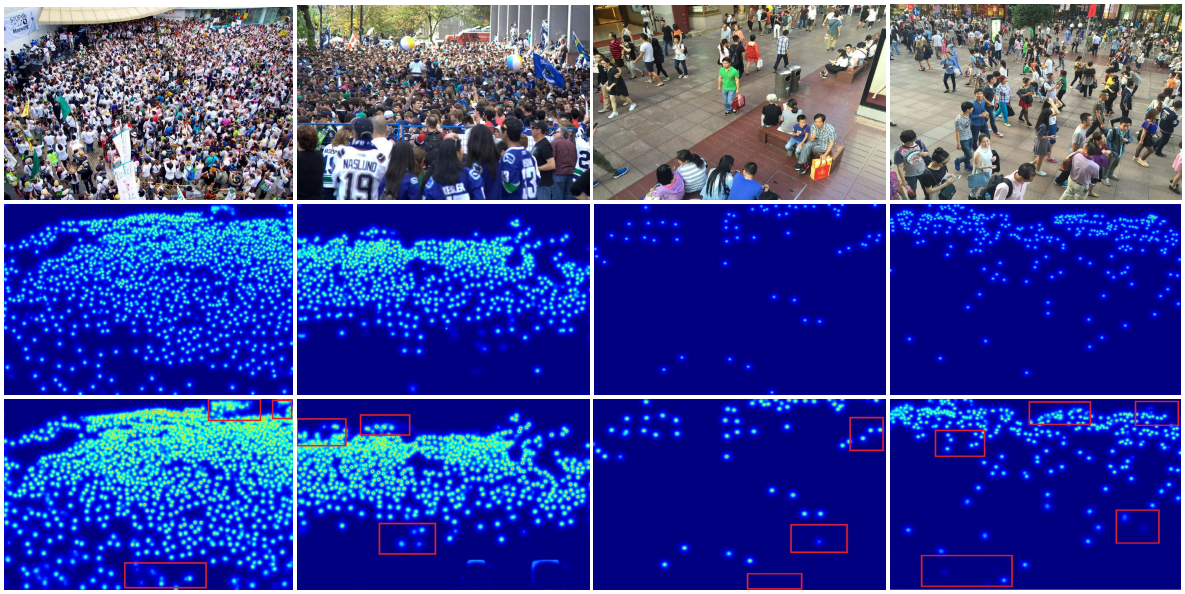


Fig. 6. Visualization results of the density maps on (a) ShanghaiTech Part A and (b) ShanghaiTech Part B, respectively.

errors, which leads to a decrease in the performance of the method. Fig. 7 shows some visualization results of PC-Net on UCF_CC_50 dataset.

In the second column of Fig. 7, the crowd density map generated by PC-Net is given, it can be seen that PC-Net can make good predictions and generate accurate density maps in crowded scenarios with variable scales, and the generated density maps have different sparsity for different scales of crowds, but the estimated values have some errors relative to the real values, such as the first set of images, which are a small number of images with large errors in the test of this paper. It is possible that the low brightness of the image is affecting the counting performance of the network. To further evaluate the visualized crowd density images, we manually label several samples containing crowd locations and perform a visual display of crowd locations, as shown in the third column of Fig. 7. A new set of evaluation metrics, structural similarity (SSIM) and peak signal-to-noise ratio (PSNR), were also used to evaluate the generated crowd density maps with labeled density maps, which compensate for the shortcomings of the one-dimensional evaluation metrics such as MAE and MSE. The experimental results show that PC-Net can fit the location information of the crowd well, and although there are some location errors, they are within the acceptable range. To summarize, PC-Net's counting performance is slightly insufficient in the face of extremely crowded crowds, so more data is needed for training to improve the accuracy of the model on extremely crowded datasets.

3) Performance on the UCF_QNRF Dataset: According to Table II, compared with the second group of methods, in the weakly-supervised mode, the MAE, MSE and MAPE of PC-Net improved by 12.8%, 11.6% and 13.4%, respectively, which indicates a significant improvement in the prediction

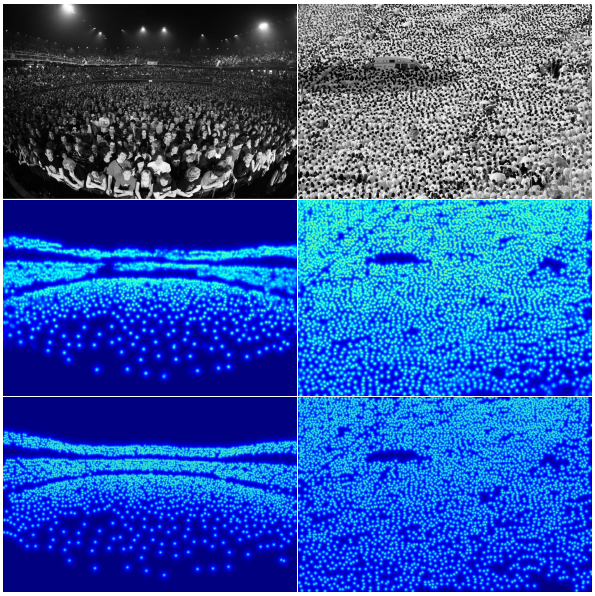


Fig. 7. Visualization results of the density maps on UCF_CC_50.

effect. PC-Net achieved optimal counting accuracy on this dataset and showed excellent robustness. Compared with the first group of methods, PC-Net also outperforms some of the fully-supervised training methods, such as L2R and TEDnet, etc., further narrowing the gap in counting performance between weakly-supervised training methods and fully-supervised training methods, and comparing some of the most advanced crowd density estimation methods, PC-Net greatly

reduces the injection cost of the dataset label, although its performance is slightly worse. Fig. 8 shows some visualization results of PC-Net on the UCF_QNRF dataset.

Count: 254
Estimate: 249

Count: 415
Estimate: 407

Fig. 8. Visualization results of the density maps on UCF_QNRF.

It can be seen that PC-Net has a good ability to fit the crowd of different scales in the first image of Fig. 8, and generates an accurate and high resolution density map, which reflects that PC-Net has a good ability to solve the problem of drastic changes in the scale of the crowd. PC-Net also generates an accurate density map for the denser crowd in the second image, but there is a certain error in the estimated value relative to the real value, which is a small number of images with large errors in the test of this paper's method, probably because the difference in lighting interferes with the counting accuracy, and more training is needed in the next step to improve the robustness of the model and exclude large errors.

4) Performance on the JHU-CROWD++ Dataset: According to Table II, PC-Net has a great advantage over both the first and second group of methods, and it is superior to the weakly-supervised methods, such as the advanced method TransCrowd. In addition, compared with fully-supervised methods, such as MCNN and CSRNet, the counting accuracy of PC-Net has been significantly improved on this dataset, and MAE, MSE and MAPE all achieved the second best performance, which proves the effectiveness of our method. Fig. 9 shows some visualization results of PC-Net on JHU-CROWD++ dataset, including the plots of crowd density in rainy and snowy days. It can be seen that PC-Net can better process the crowd images under the deteriorating weather conditions.

F. Experiment 2: Actual Experiment

In order to test the performance of PC-Net in practical applications, we conducted experiments in several real scenarios. To ensure the applicability and universality of the experiments, images taken by cameras on campuses, subway stations and city roads were randomly selected as test set, The test set contains a total of 400 images with more than 10 scenes, each containing a number of people ranging from 0 to 2000, all with a resolution of 768 × 768, and these data generally have uneven scales, background noise and other common

Count: 7
Estimate: 6

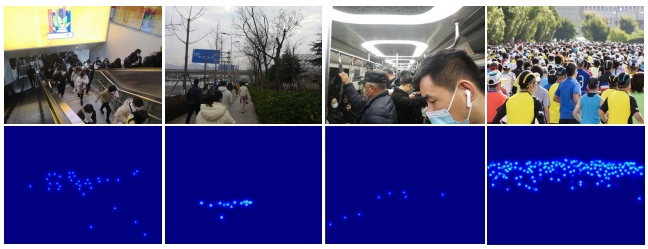
Count: 10
Estimate: 9

Fig. 9. Visualization results of the density maps on JHU-CROWD++.

factors that affect the accuracy of crowd density estimation. We conducted multiple groups of experiments and took the average value as the result of the test, and the experimental results are shown in Table III. Fig. 10 shows some visualization results of actual experiment.

TABLE III
COMPARISON OF PC-NET AND THE OTHER METHODS ON THE RANDOM DATASET

Method	Venue	Label L/C	MAE	MSE	MAPE
MCNN [3]	CVPR16	+/+	102.3	157.4	24.5%
CSRNet [8]	CVPR18	+/+	59.8	90.21	12.3%
LW-Count [79]	TCSVT22	+/+	61.3	88.7	13.4%
TransCrowd [21]	SCIS22	−/+	57.4	89.4	12.8%
PC-Net (Ours)	−	−/+	**49.7**	**75.3**	**10.6%**



Count :39
Estimate: 34

Count: 14
Estimate: 13

Count: 14
Estimate: 8

Count: 438
Estimate: 387

Fig. 10. Visualization results of the density maps of the actual experiment.

It can be seen that the test results of PC-Net on the unfamiliar dataset still outperformed the compared algorithms, and the MAE, MSE, and MAPE all obtained the optimal results. Here we randomly selected the visualization results of four scenes, and we can see that PC-Net has some adaptability to scenes that have never been seen before, and can also generate accurate and high-resolution crowd density maps, and the predicted crowd density is within an acceptable error range compared with the real crowd density. However, the results of the multi-scene test reveal the problem where the migration of

PC-Net for multiple scenes is slightly insufficient, such as the third and fourth group of images. The error of the crowd density in this scene is obviously slightly larger, and the main problem is the poor adaptability of PC-Net for this scene. Therefore, PC-Net needs to increase the training sample and test in multiple scenes to adjust its model parameters and also increase its adaptability to multiple scenes.

V. Discussion

A. Study of Training Hyper-Parameter Settings

In the training process of the network, the selection of the initial training hyper-parameter is crucial to the success of the network training. Setting good parameters can help avoid gradient disappearance or gradient explosion during the network training process, and make the neural network learn the features of the data more quickly and accurately, and improve the training effect and generalization ability of the model. In order to determine the optimal initialization parameters, we discussed the effects of different Batch_size, learning rate, activation function and optimizer on the performance of PC-Net in ShanghaiTech Part A. The experimental results are shown in Fig. 11.

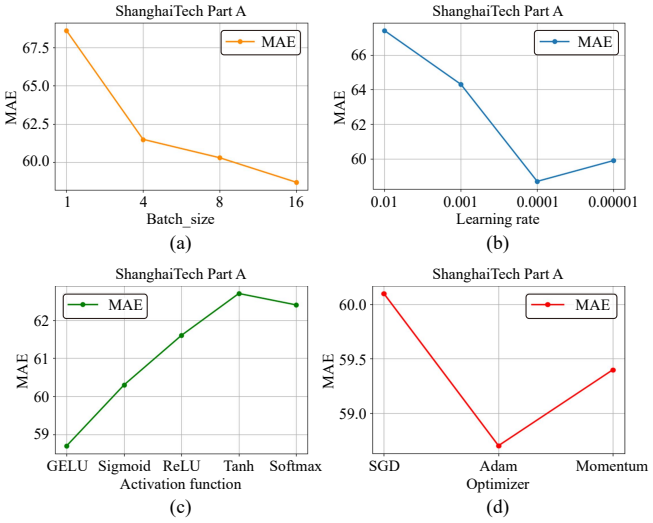


Fig. 11. Visualization results of the study of different initialization hyper-parameter settings. (a) Denotes the MAE values for different Batch_size; (b) Denotes the MAE values for different Learning rate; (c) Denotes the MAE values for different Activation function; (d) Denotes the MAE values for different Optimizer.

It can be seen that PC-Net is more sensitive to Batch_size and learning rate during the training process. As the Batch_size increases, the parallel performance of GPU is fully utilized, thus speeding up the training of the model. A larger Batch_size requires more memory storage, and a larger Batch_size may lead to overfitting because the model is more likely to memorize a larger Batch_size and thus fail to learn the overall features of the input data; therefore, on balance, we set the Batch_size to 16. Due to the complexity of the crowd density estimation task and the depth of PC-Net layers, we consider setting a smaller initial learning rate in order to avoid unstable or scattered training. The experimental results show that optimal model performance is achieved when the initial learning rate is set to 0.0001. For the activation function and the optimizer, the experimental results show that PC-Net is less sensitive to them. We compared five activation functions (GELU, Sigmoid, ReLU, Tanh, Softmax) and three optimizers (SGD, Adam, Momentum). The experimental results show that PC-Net achieves optimal results when GELU is used as the activation function and Adam is used as the optimizer. In summary, we set the Batch_size to 16, set the initial learning rate to 0.0001, and use GELU as the activation function and Adam optimizer at the beginning of the training.

B. Study of Backbone Network

With CNN-based deep learning, because CNNs have a small receptive field, this limits the upper limit of the global feature extraction range of the network. Therefore, CNN-based methods are more capable of extracting local crowd information in small intervals, but are not enough for global crowd information extraction of the whole image, which makes it difficult for CNN-based methods to establish global context features. However, ViT has the advantage of capturing long context dependencies and a global receptive domain, which is a good remedy for this deficiency of CNN. We calculated effective receptive fields for both VGG network and ViT. Specifically, we measure the effective receptive field of different layers as the absolute value of the gradient of the center location of the feature map with respect to the input. Results are averaged across all channels in each map for 16 randomly selected images, with results in Fig. 12.

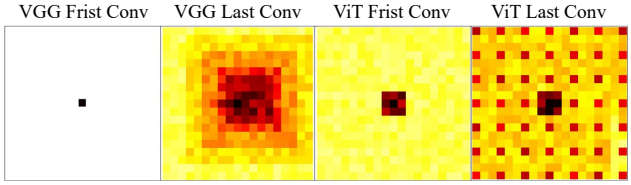


Fig. 12. Visualization results of the effective receptive fields for VGG and ViT.

We observe that lower layer effective receptive fields for ViT are indeed larger than in VGG, and while VGG effective receptive fields grow gradually, ViT receptive fields become much more global midway through the network. ViT receptive fields also show strong dependence on their center patch due to their strong residual connections. Overall, VGG effective receptive fields are highly local and grow gradually, ViT effective receptive fields shift from local to global. To further verify the superiority of the performance of pyramid vision transformer, we conducted an ablation study using the first 10 layers of VGG16 replacing pyramid vision transformer as the backbone network of PC-Net, keeping the other structures the same; the results are shown in Table IV.

As can be seen, the performance of the pyramid vision transformer is significantly better than that of VGG. On the Part A dataset, MAE, MSE and MAPE are improved by 5.6%, 2.2% and 6.3%, respectively. On the Part B dataset, MAE, MSE and MAPE are improved by 23.2%, 27.3% and 24.1%,

TABLE IV
RESULTS OF BACKBONE NETWORK ABLATION STUDY

Method	Part A	Part B	UCF_CC_50	UCF_QNRF	JHU_CROWD++
	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE
PC-Net (VGG)	62.2/91.5./14.2%	9.5/14.3/8.3%	243.2/343.2/24.1%	98.4/167.3/13.7%	68.4/141.2/24.6%
PC-Net (Ours)	**58.7/89.5/13.3%**	**7.3/10.4/6.3%**	**217.3/309.7/20.5%**	**84.8/148.9/11.6%**	**52.2/103.9/17.8%**

TABLE V
RESULTS OF PYRAMID VISION TRANSFORMER ABLATION STUDY

Method	Part A	Part B	UCF_CC_50	UCF_QNRF	JHU_CROWD++
	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE	MAE/MSE/MAPE
L1	105.3/187.4/26.6%	13.5/39.4/12.2%	331.3/491.2/35.0%	141.5/261.1/21.0%	107.9/369.8/45.3%
L1+L2	65.1/114.2/14.9%	8.9/14.3/7.7%	257.3/363.3/25.2%	104.1/171.2/14.6%	71.2/175.9/25.9%
L1+L2+L3 (Ours)	**58.7/89.5/13.3%**	**7.3/10.4/6.3%**	**217.3/309.7/20.5%**	**84.8/148.9/11.6%**	**52.2/103.9/17.8%**
L1+L2+L3+L4	58.6/89.9/13.3%	7.2/10.5/6.2%	218.2/308.3/20.6%	85.1/148.5/11.7%	51.3/102.3/17.9%
L1+L2+L3+L4+L5	60.1/98.3/13.6%	8.3/14.5/7.2%	245.4/312.5/23.7%	102.3/141.4/14.4%	61.3/165.4/21.5%

respectively. On UCF_CC_50 dataset, MAE, MSE and MAPE are improved by 10.6%, 9.8% and 14.9%, respectively. On UCF_QNRF dataset, MAE, MSE and MAPE are improved by 13.8%, 11.0% and 15.3%, respectively. On the JHU-CROWD++ dataset MAE, MSE and MAPE are improved by 23.7%, 26.4% and 27.6%, respectively. This is further proof of the superiority of PC-Net's performance.

C. Study of Pyramid Vision Transformer

The pyramid vision transformer structure proposed in this paper consists of three layers of ViT; to verify its rationality, ablation experiments were conducted on five datasets, keeping the other structures the same in the experiments to test the performance of the pyramid vision transformer under different configurations. The results are shown in Table V, where L* represents the number of layers of ViT in pyramid vision transformer.

As can be seen, the performance of PC-Net improves as the first three layers of ViT are stacked in the pyramid vision transformer, but when the ViT is stacked to the 4th layer, the performance of the model is almost the same as the 3-layer ViT, and even some metrics appear to decrease; when the number of layers of ViT continues to increase to the 5th layer, the performance of the model starts to decrease rapidly. We believe that as the depth of the network increases, the gradients in the backpropagation may become very small, leading to the gradient vanishing problem, or the gradients become very large, leading to the gradient exploding problem. These problems can make the training process difficult and make convergence impossible. Moreover, as the depth of the network increases, the number of parameters in the network increases exponentially, which can over-fit the network and make it unable to generalize to new datasets, reducing the generalization ability of the network. Therefore, we take the above factors into consideration and set the number of layers of ViT in pyramid vision transformer as 3.

D. Study of Linear Feature Calibration

In this paper, we propose a new training method using a linear feature calibration to train the network through incremental learning, which utilizes the correlation between global and local image features. To verify its effectiveness, we tested the convergence speed of the network under different supervision methods on the ShanghaiTech dataset, and the results are shown in Fig. 13.

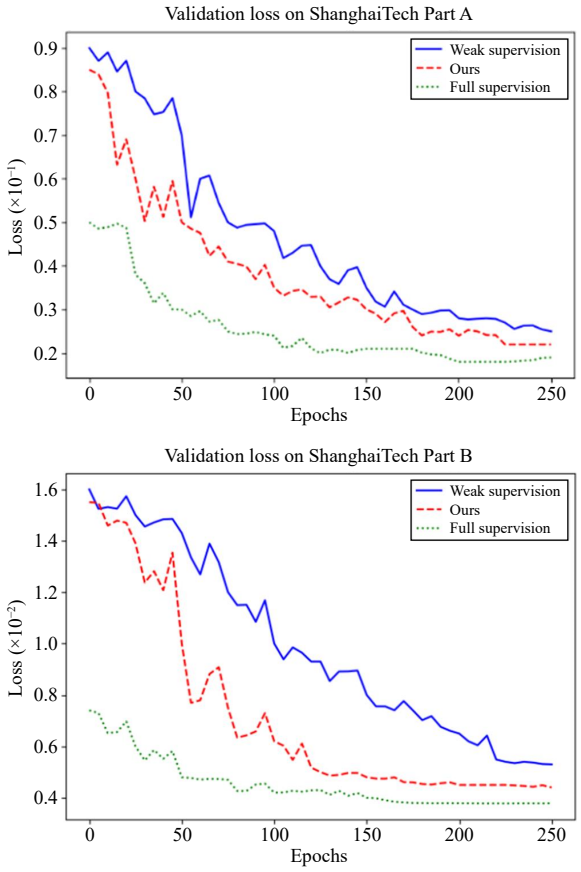


Fig. 13. Convergence speed of networks under different supervision methods. The abscissa is the training Epochs, and the ordinate is the loss value during training. The three training methods sample the same backbone network, which is the backbone network proposed in this paper.

Here, the "weakly-supervised" training method means that instead of using the linear feature calibration structure proposed in this paper, a channel attention fusion approach is used, and the features extracted from the Parent-Net and Child-Net are weighted and fused. It can be seen that the convergence speed and the fitting ability of our proposed the training method are clearly better than those of the "weakly-supervised" training method. However, it can be seen that, compared with the fully-supervised method, the convergence stability of PC-Net is poor during the training process. The reason is that during the training process, there is uncertainty in the sample labels, which increases the learning difficulty of the model, and the model may be affected by noise and learn the wrong features, resulting in overfitting or under fitting, causing the model to converge unstably. We believe that we can try to use a nonlinear feature correction process to increase the stability of the training process.

E. Study of Loss function

The loss function is very important in the training process of the network, and using different loss functions will have a great impact on the performance of the model regression, therefore, a comprehensive loss function is designed and the weight of L_2 and L_C is adjusted by loss weights. To obtain the optimal loss function, experiments are conducted on the ShanghaiTech and UCF_QNRF datasets, and the values of ∂ is discussed. The results are shown in Fig. 14.

As can be seen, different weight loss functions have an impact on the performance of the network model, and MAE and MSE change with ∂, showing a trend of decreasing and then increasing, which proves the rationality of the two-part loss function. The optimal values of MAE and MSE were obtained at $\partial = 0.6$, which proved the improvement of the comprehensive loss function on the network performance.

F. Study of Network Parameters

To analyze the parameter complexity and time complexity of PC-Net, we compared MAE, Params, and inference time on the ShanghaiTech dataset, and the experimental results are shown in Table VI.

As can be seen, the advantage of PC-Net is that it uses a weakly-supervised training method, which reduces the training cost; the MAE as well as the density estimation performance is good, however, the number of parameters is slightly larger and the required inference time is longer. Therefore, the performance of PC-Net suffers and the density estimation accuracy decreases when applied to devices with limited computational resources, such as embedded devices. Therefore, in future work, we consider studying a lightweight method based on PC-Net to analyze the parameter bottleneck layer in PC-Net, find the part of the network that consumes the most time and computational resources, and compress it to improve the training and application of the network.

VI. Conclusion and Future Work

In this paper, an effective weakly-supervised crowd density estimation method is proposed and a novel training method is used to achieve an optimal balance between training costs and counting performance. The network mainly consists of a pair of parent-child networks and a linear feature calibration structure. Specifically, the parent network is used to extract the crowd features, the child network is used to extract the feature correction parameters and bias weights, and the features are calibrated using the linear feature calibration structure to improve the convergence speed as well as the fitting ability of the network. In addition, a pyramid vision transformer is used as the backbone network of the PC-Net to solve the problem of uneven scale in the crowd, while the spatial correlation and crowd sensitivity of density map are enhanced by global-local feature loss and counting loss.

In future work, we will study a crowd counting and positioning method based on PC-Net, which can not only achieve

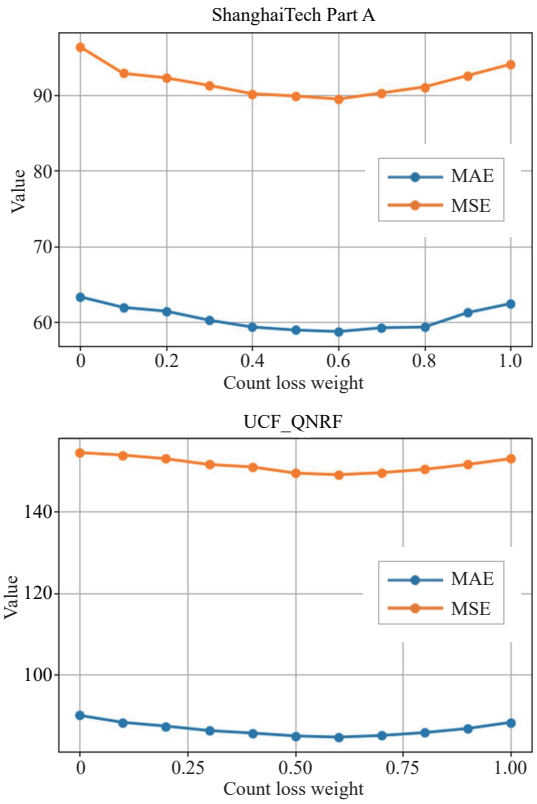


Fig. 14. MAE and MSE in ShanghaiTech Part A and UCF_QNRF datasets under different counting loss weights.

TABLE VI
VI Comparison of the Params, MAE and Running Time of PC-Net and Other Methods on the ShanghaiTech Dataset

Method	Lable	Params (M)	MAE	GPU (MS)	CPU (S)
CSRNet [8]	+/+	16.26	68.2	85.5	24. 3
BL [38]	+/+	21.50	61.5	63.7	25.8
DUBNet [85]	+/+	18.05	64.4	414.8	51.3
SFCN [78]	+/+	38.60	64.8	81.9	24.1
DKPNet [86]	+/+	30.63	55.6	89.4	23.7
ST-Net [81]	+/+	15.56	52.9	61.3	21.4
PC-Net (Ours)	−/+	**36.8**	**58.7**	**75.4**	**22.6**

a better personal positioning function and counting accuracy, but also the number of parameters is smaller and more stable during training.

REFERENCES

[1] E. Walach and L. Wolf, "Learning to count with CNN boosting," in *Proc. 14th European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 660–676.

[2] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proc. 23rd ACM Int. Conf. Multimedia*, Brisbane, Australia, 2015, pp. 1299–130.

[3] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016, pp. 589–597.

[4] V. Sindagi and V. Patel, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 1002–1012.

[5] Y. Wang, S. Hu, G. Wang, C. Chen, and Z. Pan, "Multi-scale dilated convolution of convolutional neural network for crowd counting," *Multimed. Tools Appl.*, vol. 79, no. 1–2, pp. 1057–1073, Jan. 2020.

[6] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid CNNs," in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 1879–1888.

[7] D. Guo, K. Li, Z.-J. Zha, and M. Wang, "DADNet: Dilated-attention-deformable ConvNet for crowd counting," in *Proc. 27th ACM Int. Conf. Multimedia*, Nice, France, 2019, pp. 1823–1832.

[8] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018, pp. 1091–1100.

[9] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 757–773.

[10] X. Chen, Y. Bin, N. Sang, and C. Gao, "Scale pyramid network for crowd counting," in *Proc. IEEE Winter Conf. Applications Computer Vision*, Waikoloa, USA, 2019, pp. 1941–1950.

[11] L. Huang, S. Shen, L. Zhu, Q. Shi, and J. Zhang, "Context-aware multi-scale aggregation network for congested crowd counting," *Sensors*, vol. 22, no. 9, p. 3233, Apr. 2022.

[12] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder networks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 6126–6135.

[13] Y.-C. Li, R.-S. Jia, Y.-X. Hu, D.-N. Han, and H.-M. Sun, "Crowd density estimation based on multi scale features fusion network with reverse attention mechanism," *Appl. Intell.*, vol. 52, no. 11, pp. 13097–13113, Sept. 2022.

[14] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "ADCrowdNet: An attention-injective deformable convolutional network for crowd understanding," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 3220–3229.

[15] A. Zhang, L. Yue, J. Shen, F. Zhu, X. Zhen, X. Cao, and L. Shao, "Attentional neural fields for crowd counting," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 5713–5722.

[16] H. Chu, J. Tang, and H. Hu, "Attention guided feature pyramid network for crowd counting," *J. Vis. Commun. Image Represent.*, vol. 80, p. 103319, Oct. 2021.

[17] T. Lei, D. Zhang, R. Wang, S. Li, W. Zhang, and A. K. Nandi, "MFP-Net: Multi-scale feature pyramid network for crowd counting," *IET Image Process.*, vol. 15, no. 14, pp. 3522–3533, Dec. 2021.

[18] S. Amirgholipour, W. Jia, L. Liu, X. Fan, D. Wang, and X. He, "PDANet: Pyramid density-aware attention based network for accurate crowd counting," *Neurocomputing*, vol. 451, pp. 215–230, Sept. 2021.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learning Representations*, 2021.

[20] S. Yang, W. Guo, and Y. Ren, "CrowdFormer: An overlap patching vision transformer for top-down crowd counting," in *Proc. 31st Int. Joint Conf. Artificial Intelligence*, Vienna, Austria, 2022, pp. 1545–1551.

[21] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, "Transcrowd: Weakly-supervised crowd counting with transformers," *Sci. China Inf. Sci.*, vol. 65, no. 6, p. 160104, Apr. 2022.

[22] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Montreal, Canada, 2021, pp. 548–558.

[23] U. Sajid, X. Chen, H. Sajid, T. Kim, and G. Wang, "Audio-visual transformer based crowd counting," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Montreal, Canada, 2021, pp. 2249–2259.

[24] G. Sun, Y. Liu, T. Probst, D. P. Paudel, N. Popovic, and L. Van Gool, "Boosting crowd counting with transformers," arXiv preprint arXiv: 2105.10926, 2021.

[25] P. T. Do, "Attention in crowd counting using the transformer and density map to improve counting result," in *Proc. 8th NAFOSTED Conf. Information and Computer Science*, Hanoi, Vietnam, 2021, pp. 65–70.

[26] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in CNNs by self-supervised learning to rank," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1862–1878, Aug. 2019.

[27] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 8190–8199.

[28] G. Olmschenk, J. Chen, H. Tang, and Z. Zhu, "Dense crowd counting convolutional neural networks with minimal data using semi-supervised dual-goal generative adversarial networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition: Learning with Imperfect Data Workshop*, 2019, pp. 21–28.

[29] D. B. Sam, A. Agarwalla, J. Joseph, V. A. Sindagi, R. V. Babu, and V. M. Patel, "Completely self-supervised crowd counting via distribution matching," in *Proc. 17th European Conf. Computer Vision*, Tel Aviv, Israel, 2022, pp. 186–204.

[30] Z. Zhao, M. Shi, X. Zhao, and L. Li, "Active crowd counting with limited supervision," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 565–581.

[31] Y. Liu, L. Liu, P. Wang, P. Zhang, and Y. Lei, "Semi-supervised crowd counting via self-training on surrogate tasks," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 242–259.

[32] V. A. Sindagi, R. Yasarla, D. S. Babu, R. V. Babu, and V. M. Patel, "Learning to count in the crowd from limited labeled data," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 212–229.

[33] Y. Lei, Y. Liu, P. Zhang, and L. Liu, "Towards using count-level weak supervision for crowd counting," *Pattern Recognit.*, vol. 109, p. 107616, Jan. 2021.

[34] Y. Meng, H. Zhang, Y. Zhao, X. Yang, X. Qian, X. Huang, and Y. Zheng, "Spatial uncertainty-aware semi-supervised crowd counting," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Montreal, Canada, 2021, pp. 15529–15539.

[35] S. Khaki, H. Pham, Y. Han, A. Kuhl, W. Kent, and L. Wang, "DeepCorn: A semi-supervised deep learning method for high-throughput image-based corn kernel counting and yield estimation," *Knowl. Based Syst.*, vol. 218, p. 106874, Apr. 2021.

[36] D. B. Sam, N. N. Sajjan, H. Maurya, and R. V. Babu, "Almost unsupervised learning for dense crowd counting," in *Proc. 33rd AAAI Conf. Artificial Intelligence*, Honolulu, USA, 2019, pp. 8868–8875.

[37] M. von Borstel, M. Kandemir, P. Schmidt, M. K. Rao, K. Rajamani, and F. A. Hamprecht, "Gaussian process density counting from weak supervision," in *Proc. 14th European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 365–380.

[38] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 6141–6150.

[39] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe, "Weakly-

supervised crowd counting learns from sorting rather than locations," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 1–17.

[40] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, USA, 2008, 1–7.

[41] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surv.*, vol. 48, no. 1, p. 7, Aug. 2015.

[42] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Leveraging GPS-less sensing scheduling for green mobile crowd sensing," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 328–336, Aug. 2014.

[43] C. Liu, X. Wen, and Y. Mu, "Recurrent attentive zooming for joint crowd counting and precise localization," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 1217–1226.

[44] Z.-Q. Cheng, J.-X. Li, Q. Dai, X. Wu, and A. Hauptmann, "Learning spatial awareness to improve crowd counting," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 6151–6160.

[45] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. 23rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2010, pp. 1324–1332.

[46] L. Wen, D. Du, P. Zhu, Q. Hu, Q. Wang, L. Bo, and S. Lyu, "Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network," arXiv preprint arXiv: 1912.01811, 2019.

[47] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, and W. Paszke, "An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 53, no. 6, pp. 3461–3473, Jun. 2023.

[48] X. Xin, Y. Tu, V. Stojanovic, H. Wang, K. Shi, S. He, and T. Pan, "Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems," *Appl. Math. Comput.*, vol. 412, p. 126537, Jan. 2022.

[49] C. Zhou, H. Tao, Y. Chen, V. Stojanovic, and W. Paszke, "Robust point-to-point iterative learning control for constrained systems: A minimum energy approach," *Int. J. Robust Nonlinear Control*, vol. 32, no. 18, pp. 10139–10161, Dec. 2022.

[50] X. Song, N. Wu, S. Song, and V. Stojanovic, "Switching-like event-triggered state estimation for reaction–Diffusion neural networks against DoS attacks," *Neural Process. Lett.*, vol. 55, no. 7, pp. 8997–9018, Dec. 2023.

[51] W. Li, X. Luo, H. Yuan, and M. C. Zhou, "A momentum-accelerated Hessian-vector-based latent factor analysis model," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 830–844, Mar.–Apr. 2023.

[52] D. Wu, X. Luo, Y. He, and M. C. Zhou, "A prediction-sampling-based multilayer-structured latent factor model for accurate representation to high-dimensional and sparse data," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022. DOI: 10.1109/TNNLS.2022.3200009

[53] X. Luo, Y. Zhou, Z. Liu, L. Hu, and M. C. Zhou, "Generalized Nesterov's acceleration-incorporated, non-negative and adaptive latent factor analysis," *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2809–2823, Sep.–Oct. 2022.

[54] D. Wu and X. Luo, "Robust latent factor analysis for precise representation of high-dimensional and sparse data," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 4, pp. 796–805, Apr. 2021.

[55] W. Zhao, M. Wang, Y. Liu, H. Lu, C. Xu, and L. Yao, "Generalizable crowd counting via diverse context style learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5399–5410, Aug. 2022.

[56] F. Zhu, H. Yan, X. Chen, and T. Li, "Real-time crowd counting via lightweight scale-aware network," *Neurocomputing*, vol. 472, pp. 54–67, Feb. 2022.

[57] J. T. Zhou, L. Zhang, J. Du, X. Peng, Z. Fang, Z. Xiao, and H. Zhu, "Locality-aware crowd counting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3602–3613, Jul. 2022.

[58] C. Xu, D. Liang, Y. Xu, S. Bai, W. Zhan, X. Bai, and M. Tomizuka, "Autoscale: Learning to scale for crowd counting," *Int. J. Comput. Vis.*, vol. 130, no. 2, pp. 405–434, Feb. 2022.

[59] J. Zhang, "Knowledge learning with crowdsourcing: A brief review and systematic perspective," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 5, pp. 749–762, May 2022.

[60] Z. Liu, N. Wu, Y. Qiao, and Z. Li, "Performance evaluation of public bus transportation by using DEA models and Shannon's entropy: An example from a company in a large city of China," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 4, pp. 779–795, Apr. 2021.

[61] Y. Zheng, Q. Li, C. Wang, X. Wang, and L. Hu, "Multi-source adaptive selection and fusion for pedestrian dead reckoning," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 12, pp. 2174–2185, Dec. 2022.

[62] X. Deng, S. Chen, Y. Chen, and J.-F. Xu, "Multi-level convolutional transformer with adaptive ranking for semi-supervised crowd counting," in *Proc. 4th Int. Conf. Algorithms, Computing and Artificial Intelligence, Sanya*, China, 2021, p. 2.

[63] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-constrained spatial transformer network for video crowd counting," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Shanghai, China, 2019, pp. 814–819.

[64] Y. Fang, S. Gao, J. Li, W. Luo, L. He, and B. Hu, "Multi-level feature fusion based locality-constrained spatial transformer network for video crowd counting," *Neurocomputing*, vol. 392, pp. 98–107, Jun. 2020.

[65] Z. Wu, L. Liu, Y. Zhang, M. Mao, L. Lin, and G. Li, "Multimodal crowd counting with mutual attention transformers," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Taipei, China, 2022, pp. 1–6.

[66] Q. Wang, T. Han, J. Gao, and Y. Yuan, "Neuron linear transformation: Modeling the domain shift for crowd counting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3238–3250, Aug. 2022.

[67] D. Liang, W. Xu, Y. Zhu, and Zhou, Y, "Focal inverse distance transform maps for crowd localization," *IEEE Trans. Multimedia*, vol. 25, pp. 6040–6052, 2023.

[68] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Portland, USA, 2013, pp. 2547–2554.

[69] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 544–559.

[70] V. A. Sindagi, R. Yasarla, and V. M. Patel, "JHU-CROWD++: Large-scale crowd counting dataset and a benchmark method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2594–2609, May 2022.

[71] Z. Shi, P. Mettes, and C. Snoek, "Counting with focus for free," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 4199–4208.

[72] J. Gao, Q. Wang, and X. Li, "PCC Net: Perspective crowd counting via spatial convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3486–3498, Oct. 2020.

[73] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe, "Reverse perspective network for perspective-aware object counting," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 4373–4382.

[74] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang, and Y. Pang, "Attention scaling for crowd counting," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 4705–4714.

[75] X. Liu, J. Yang, W. Ding, T. Wang, Z. Wang, and J. Xiong, "Adaptive mixture regression network with local counting map for crowd counting," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 241–257.

[76] B. Wang, H. Liu, D. Samaras, and M. Hoai, "Distribution matching for crowd counting," in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, p. 135.

[77] J. Wan, Z. Liu, and A. B. Chan, "A generalized loss function for crowd counting and localization," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, USA, 2021, pp. 1974–1983.

[78] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Pixel-wise crowd understanding via synthetic data," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 225–245, Jan. 2021.

[79] Y. Liu, G. Cao, H. Shi, and Y. Hu, "LW-count: An effective lightweight encoding-decoding crowd counting network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6821–6834, Oct. 2022.

[80] J. Chen, K. Wang, W. Su, and Z. Wang, "SSR-HEF: Crowd counting

with multiscale semantic refining and hard example focusing," *IEEE Trans. Industr. Inform.*, vol. 18, no. 10, pp. 6547–6557, Oct. 2022.

[81] M. Wang, H. Cai, X.-F. Han, J. Zhou, and M. Gong, "STNet: Scale tree network with multi-level auxiliator for crowd counting," *IEEE Trans. Multimedia*, vol. 25, pp. 2074–2084, 2023.

[82] X. Zhang, L. Han, W. Shan, X. Wang, S. Chen, C. Zhu, and B. Li, "A multi-scale feature fusion network with cascaded supervision for cross-scene crowd counting," *IEEE Trans. Instrum. Meas.*, vol. 72, p. 5007515, Feb. 2023.

[83] Y. Chen, J. Yang, B. Chen, and S. Du, "Counting varying density crowds through density guided adaptive selection CNN and transformer estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 3, pp. 1055–1068, Mar. 2023.

[84] V. K. Sharma, R. N. Mir, and C. Singh, "Scale-aware CNN for crowd density estimation and crowd behavior analysis," *Comput. Electr. Eng.*, vol. 106, p. 108569, Mar. 2023.

[85] M.-H. Oh, P. Olsen, and K. N. Ramamurthy, "Crowd counting with decomposed uncertainty," in *Proc. 34th AAAI Conf. Artificial Intelligence*, New York, USA, 2020, pp. 11799–11806.

[86] B. Chen, Z. Yan, K. Li, P. Li, B. Wang, W. Zuo, and L. Zhang, "Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Montreal, Canada, 2021, pp. 16045–16055.

Yong-Chao Li received the M.S. degrees in computer science and technology from Shandong University of Science and Technology in 2023. During this time, his research interests included image processing and deep learning. He is currently a Ph.D. candidate in intelligent information and communication systems at Ocean University of China. His current research interests include image processing, computational imaging and deep learning.

Rui-Sheng Jia received the M.S. and Ph.D. degrees in computer science and technology from Shandong University of Science and Technology in 2002 and 2010, respectively. He is currently a Full Professor at the College of Computer Science and Engineering, Shandong University of Science and Technology, and is the Leader of a Natural Science Foundation of Shandong Province. His research interest includes artificial intelligence, computer vision, information fusion, microseismic monitoring and inversion. He has more than 30 first-author publications and has more than 50 co-author publications.

Ying-Xiang Hu received the M.S. degrees in computer science and technology from Shandong University of Science and Technology in 2023. During this time, her research interests included image processing and deep learning. She is currently a Ph.D. candidate in computer science and technology at Nanjing University of Science and Technology. Her current research interests include image processing, crowd counting and deep learning.

Hong-Mei Sun received the B.S. and M.S. degree in computer science from Shandong University of Science and Technology candidate in 1995 and 2005, respectively. She is currently an Associate Professor at the College of Computer Science and Engineering, Shandong University of Science and Technology, and is the Leader of a Key Research and Development Projects of Shandong Province. Her research interests include computer vision, deep learning and software engineering. She has more than 20 first-author publications and has more than 50 co-author publications.