Research on RGB-D image recognition technology based on feature fusion and machine learning

Qingbo Sun

Shandong University of Political Science and Law, 250014

20385489@dlvct.edu.cn

Abstract: The three-dimensional RGB-D image contains not only the color and texture information of the two-dimensional image, but also contains the surface geometry information of the target. This article analyzes the RGB-D image recognition methods, including stereo vision technology, structured light technology, etc. By studying the application points of RGB-D image recognition technology under the background of feature fusion and machine learning, the purpose is to improve the richness of image recognition content and provide reference for the smooth development of the follow-up work.

1. Introduction

The currently applied RGB-D image not only contains the color and texture information of the two-dimensional image, but also contains many types of geometric information. Especially with the integration of Kinect, Xtion and other equipment, the level of refinement and clarity of RGB-D image content has also been continuously improved. The optimization of the RGB-D image recognition technology system based on feature fusion and machine learning can not only improve the accuracy of the image recognition results, but also has a positive meaning for improving the value of image utilization.

2. RGB-D image recognition method analysis

2.1 Stereo Vision Technology

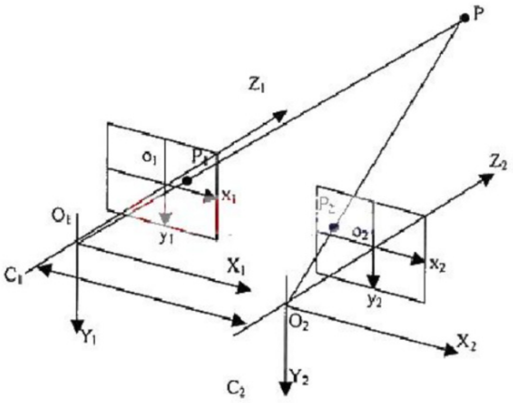


Figure 1 Schematic diagram of binocular vision

Published under licence by IOP Publishing Ltd

The stereo vision method is a traditional way of acquiring depth images. Its principle is to calculate the parallax through the photos of the same scene from multiple perspectives, and obtain the three-dimensional information of the objects in the scene according to the parallax. A general stereo vision system is composed of several ordinary industrial CCD cameras, which require manual camera calibration before acquiring the depth map, which is generally calibrated by the positive checkerboard method. As shown in Figure 1, O_l and O_r are the centers of the two cameras respectively, and the three-dimensional coordinates of point A can be obtained from the similar relationship of triangles. The stereo vision method does not require high-precision sensors, and only a few ordinary cameras are needed to obtain the depth map, but its acquisition of the depth map requires a series of complex algorithms to operate on the original image.

2.2 Structured light technology

Structured light technology is currently a more advanced depth map acquisition method. Because structured light measurement has obvious advantages such as fast speed, high precision, non-contact, easy implementation and automation, it has been widely used in industrial inspection, machine vision, film and television special effects, etc. field. Projectors and cameras are the basic components of structured light systems. Once the projector projects the coded pattern onto the scene to be measured, the camera can capture and shoot the projected pattern, and the acquisition of the three-dimensional information of the scene is the result of matching the code with the projected pattern. Therefore, coding and matching are two important factors for constructing a structured light system. Take Kine ct as an example, its structure is as follows.

Figure 2 Schematic diagram of Kine ct structure

The principle of Kine ct's acquisition of color images is the same as that of ordinary cameras. This article focuses on the principle of Kine ct sensor's acquisition of depth images. Kine ct essentially uses structured light technology to obtain the depth distance. It emits a light source through an infrared emitter. This light source is called laser speckle, which is a diffraction spot formed randomly by laser irradiating the rough surface of an object or penetrating ground glass. These spots will change with the distance of the object from the Kine ct sensor. Microsoft recorded the patterns of these laser speckles in the entire space, and made a one-to-one correspondence with the distance.

3. Application points of RGB-D image recognition technology under the background of feature fusion

3.1 RGB-D image extraction

3.1.1 Gist feature extraction

From the perspective of previous RGB-D image extraction, there are problems such as poor content extraction, high dimensionality of feature data, and low accuracy of algorithm calculation results. In view of this kind of situation, in the process of Gist feature extraction, the extraction algorithm needs to be reasonably selected. Currently, more algorithms are used including HOG algorithm, SIFT algorithm, SURF algorithm, etc. At the same time, in the process of algorithm application, it is also necessary from the perspective of gradient analysis. To start the analysis to improve the accuracy of the calculation results. Moreover, in the calculation process, the overall calculation amount is relatively large, which is 150% higher than the total amount of calculation data in the past, which also requires more reliable algorithms to complete information sorting. In the process of Gist feature extraction, feature data extraction will be completed from naturalness, openness, roughness, expansion and steepness, and then Gabor filters are used to process the content to obtain richer data information. An example of graph extraction is shown in Figure 3.

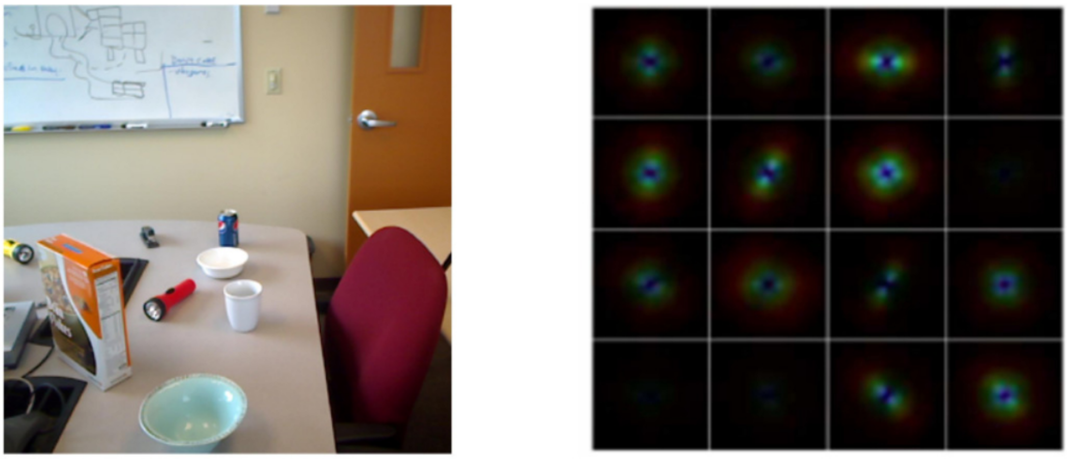


Figure 3 Example diagram of Gist feature extraction

3.1.2 Color-depth map extraction

In the process of data information extraction, it is also necessary to do a good job in the extraction of color-depth maps. In the specific application process, based on the extracted information obtained by the existing Gist algorithm, sub-networks will be used to further improve the image content. Usually, the image is densified to refine the content of the required extraction part, and then the feature points are extracted using the Gist algorithm, and the weighting coefficient is calculated at the same time, so as to obtain the local Gist feature that meets the application requirements. Take the schematic diagram in Figure 3 as an example. In the specific processing process, first, cover the entire picture with a 16×16 grid, and then extract the Gist features in each grid; second, for the existing The divided grid is divided again and organized into 4×4 grids, that is, each small grid is divided into 16 small grids. Calculate the weights according to the correlation between each other to obtain the required analysis results.

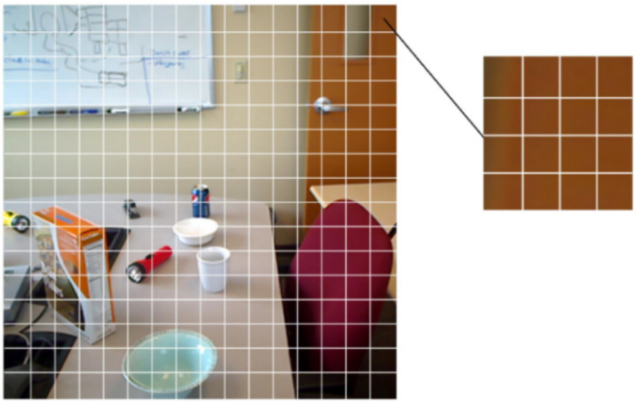


Figure 4 Schematic diagram of Gist algorithm after densification processing

3.2 Feature dimensionality reduction processing

After finishing the above part of the content processing, all the parameter information can be learned based on the Gist feature. At the same time, the basic information in the RGB image will be extracted, including color information, texture information, and geometric information, as shown in Figure 4, after segmentation The picture is converted into 16×16×(4×4) small grid graphics, and the feature dimension that needs to be extracted has been greatly improved. In order to speed up the calculation, the principal component analysis method will also be used to assist the calculation. process. This method belongs to the data dimensionality reduction processing method often used in the process of graph feature extraction. Its principle is shown in Figure 5. The projection point is selected in the coordinate system, and the known high-latitude data is mapped to the low-latitude data by means of the vector projection method. In the latitude space, the projection components are then used for orthogonal processing to obtain the required linear vector to meet specific application requirements. According to the projection results, the amount of projection in different dimensions can also be obtained, and the main direction of the analyzed data information and the criticality of the characteristic content can be understood, so as to complete the dimensionality reduction process. For example, 16×16×(4×4) sub-grid patterns were obtained by the previous division. After dimensionality reduction processing, the total dimensional value can reach 131072, which has rich data information characteristics.

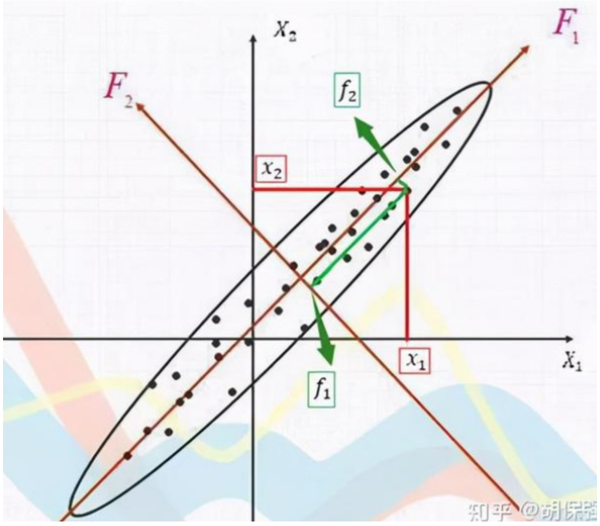


Figure 5 Schematic diagram of principal component analysis

3.3 Characteristic content expression

When performing feature content expression processing, the application process includes the following points: First, perform feature content extraction. According to the bag-of-words model in the initial state, the graphics will be subdivided into several sub-modules in the application, and then the content of these pixels will be subdivided into several sub-modules. Organize so that it can form a representative vector according to a certain rule. In RGB-D images, the Gist algorithm will also be used to optimize the application content in the initial state to obtain more analytically valuable application data. Second, use the K-means clustering algorithm to complete the sample point division processing. In specific applications, it is also necessary to sort out the correlation between the initialization position and the clustering effect and running time, and select 30%-50% of the applications Data, thereby enhancing the use value of the analysis results.

3.4 Simulation experiment analysis

3.4.1 Preparation for simulation experiment

In the construction of this simulation experiment, the RGB-D database is used as a basic reference, and all the pictures in the database are used as simulation experiment data. In terms of image acquisition, the Kinect sensor is used to capture them. The total number of captured RGB images is 5863 sheets, the resolution of each picture is adjusted to 1280×640, in order to facilitate subsequent data analysis work in an orderly manner.

3.4.2 Simulation experiment process

According to the basic content provided by the simulation experiment environment, in the whole experiment process, first, adjust the size of the pictures in the data set, the specific adjustment size is 512×512, and then according to the content in 2.1-2.3, for each picture The information undergoes Gist feature processing to obtain corresponding reference data. Second, use the K-means clustering algorithm to perform in-depth processing, and in the application of the algorithm, set the K value in the algorithm to 500. Third, determine the number of training data in the experiment. In this simulation experiment, 4876 pictures are selected for training, and the data set used for testing is (5863-4876)=987 pictures. The ten-fold cross-validation method is used for content verification, thereby obtaining image recognition accuracy data.

3.4.3 Arrangement of experimental results

Table 1 Analysis of recognition accuracy of different algorithms

Use algorithm	HOG+SVM algorithm	SIFT+SPM algorithm	KD+SVM algorithm	Gist image recognition algorithm
Recognition accuracy	79.3%	86.3%	90.3%	93.2%

From the experimental results in Table 1, it can be understood that the Gist image recognition algorithm used in this article has achieved 93.2% accuracy in image recognition, while the traditional HOG+SVM algorithm has the lowest accuracy, only 79.3%, and the accuracy of the other two types of algorithms The image recognition accuracy of the KD+SVM algorithm reaches 90.3%, and it also has a high image recognition accuracy. However, the algorithm proposed in this article still has a lot of room for improvement. Analyzing the causes of image recognition errors, it is understood that the background color in the confused image is more than 70% similar, which leads to algorithm recognition errors. Therefore, in the subsequent development, it is necessary Focus on the background color distinction to meet the corresponding management needs [1].

4. Application points of RGB-D image recognition technology in the context of machine learning

4.1 Neural network design

When the neural network is designed based on the actual situation, the specific design steps are as follows: First, preprocess the collected images, filter out the required picture information from the RGB-D image, and perform simple preprocessing on its related content. Second, rely on the neural network to establish the corresponding neural network structure, in which the elements in the RGB-D image will be identified, so as to obtain reliable data analysis results. Third, for the extraction of image network features, the specific extraction process can refer to the related algorithms in Chapter 2. Fourth, the neural network recognition system is trained according to the existing database, so that it can continue to learn deeply and meet the recognition needs. Fifth, perform feature fusion processing, and then output the corresponding image recognition results, evaluate the accuracy of the recognition results, and continue to train them to obtain higher accuracy information [2].

4.2 Feature content integration

After the RGB image is obtained, the feature content fusion processing needs to be completed according to the correlation. In the specific fusion processing process, the feature dimensions that need to be extracted need to be improved on the original basis to improve the reliability of the processing results. In order to speed up the calculation, the optimal weight algorithm is also used to assist the calculation process. This method is a data feature fusion processing method often used in the process of graphic feature fusion. Its principle is to select information that meets the optimal calculation requirements from the known data set based on the corresponding basic conditions to meet specific application requirements. According to the calculation results, different types of weights can also be obtained, and the main direction of the analyzed data information and the criticality of the characteristic content can be understood, so as to complete the optimal weight calculation [3].

4.3 Pseudo-colorization of the depth map

After completing the above graphics processing, you need to do a good job of pseudo-color processing. Its main function is to obtain the required color image after processing the existing grayscale image, but the color image is not the true color of the original image, so This process is called pseudo-color processing. In the specific processing process, the corresponding temperature data will be directly drawn on the interface during the infrared imaging design process of the system, and then the acquired temperature data will be processed to obtain the corresponding gray-scale image, and finally obtained by the processing method Pseudo-color image [4].

4.4 Simulation experiment analysis

4.4.1 Preparation for simulation experiment

In the construction of this simulation experiment, the RGB-D database is used as the basic reference, and all the pictures in the database are used as the simulation experiment data. In terms of image acquisition, the Kinect sensor is used to capture them from different angles, and the angle is set to 30. °, 60 ° and 90 °, the total number of captured RGB pictures is 35,263, the resolution of each picture is adjusted to 640 × 640, in order to facilitate the subsequent data analysis work in an orderly manner [5].

4.4.2 Simulation experiment process

According to the basic content provided by the simulation experiment environment, in the whole experiment process, first, adjust the size of the pictures in the data set, the specific adjustment size is 256×256, and the feature fusion processing is performed on each picture information, relying on Caffe Framework to obtain the corresponding reference data. Second, use the C++/CUDA architecture to perform in-depth processing, and in the application of the framework structure, the matrix parameters will also be sorted to obtain a more concise data set. Third, determine the application model of the entire experimental process. In this simulation experiment, the Alex Net model will be used as the main test carrier. The average time of this model on image processing is 1.15ms, with the main frequency 2.90GHZ 16G running content Complete the entire simulation experiment and obtain the accuracy data of image recognition [6].

4.4.3 Arrangement of experimental results

From the experimental results in Table 1, it can be understood that the neural network image recognition algorithm used in this article has achieved 92.3% accuracy in image recognition, while the traditional CNN-RNN algorithm has the lowest accuracy, only 86.3%. The other two types of algorithms are accurate The rate is in the upper middle, and the accuracy of the SN-CNN-SVM algorithm for image recognition has reached 91.1%, and it also has a high image recognition accuracy. However, the algorithm proposed in this article still has a lot of room for improvement. Analyzing the causes of image recognition errors, it is understood that the depth recognition results in the confused images are similar, and the similarity exceeds 60%, which leads to algorithm recognition errors. Therefore, in the follow-up In the development, it is necessary to focus on the research of in-depth identification content to meet the corresponding management needs [7].

Table 2 Analysis of recognition accuracy of different algorithms

Use algorithm	CNN-RNN algorithm	SP+HMP algorithm	SN-CNN-SVM algorithm	Neural network recognition algorithm
Recognition accuracy	86.3%	88.6%	91.31%	92.3%

5. Experimental comparative analysis

According to the above experiment, in the application of Gist image recognition method, the recognition accuracy of the feature image is higher, and in the use process, its overall convenience is relatively high, and it can meet the requirements of image recognition in many situations. The neural network recognition algorithm needs to go through iterative data learning in the application, and the initial state is relatively cumbersome. After a certain number of iterations, its recognition accuracy is relatively stable. In the follow-up research, further discussion of the algorithm application is also needed. Improve the reliability of the analysis results [8].

6. Conclusions

To sum up, in the research process of RGB-D image recognition technology, both feature fusion and machine learning models have good application value. Taking a reasonable way to merge their content can not only speed up image recognition, but also improve The accuracy of image recognition results.

References

[1] Yan Jianwei, Zhang Wenyong, Xie Benliang. Random forest tea fresh leaf classification based on feature fusion [J/OL]. Journal of South China Agricultural University: 1-11 [2021-05-01].

[2] Yuan Mengjiao, Dong Yuning. Network video stream classification based on feature fusion and machine learning [J/OL]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2021(02): 100-108[2021-05-01].

[3] Ma Xiaotian, Wang Chenping, Qiu Yanjun. Prediction of pavement friction performance based on the fusion of macro and micro texture features[J/OL]. Journal of Zhejiang University (Engineering Science Edition): 1-11[2021-05-01].

[4] Feng Kaiyuan, Luo Qingbin, Zheng Minghui, Li Chao. Research on malicious webpage detection method based on multi-feature fusion[J]. Journal of Hubei University for Nationalities (Natural Science Edition), 2021, 39(01): 80-85.

[5] Meng Zhen, Wang Hao, Yu Wei, Deng Sanhong, Zhang Baolong. Research on vocal music classification based on feature fusion [J/OL]. Data analysis and knowledge discovery: 1-22 [2021-05-01].

[6] Ma Jing, Cai Wenjie, Yang Li. Research on Heart Sound Recognition and Classification Based on Machine Learning [J]. Chinese Journal of Medical Physics, 2021, 38(01): 75-79.

[7] Jiang Qianyu, Wang Fengying, Jia Lipeng. Malicious code detection method based on perceptual hash algorithm and feature fusion [J]. Computer Applications, 2021, 41(03): 780-785.

[8] Lu Chujie, Li Sihui. Research on building occupancy rate perception model based on multi-sensor fusion [J]. Computer Applications and Software, 2021, 38(01): 58-62.

Time and Position Aware Graph Neural Networks for Session-based Recommendation

2021 7th International Conference on Computer and Communications (ICCC) | 978-1-6654-0950-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCC54389.2021.9674354

Qingbo Sun
School of Cyberspace Security
Shandong University Of Political Science And Law
Jinan China
e-mail:sun828bo@163.com

Zhijun Zhang*
School of Computer Science and Technology
Shandong Jianzhu University
Jinan China
e-mail:zzjsdcn@163.com

Sheng Sang
School of Computer Science and Technology
Shandong Jianzhu University
Jinan China
e-mail:ss.ssang@qq.com

Fang Dong
School of Computer Science and Technology
Shandong Jianzhu University
Jinan China
dongfang@sdjzu.edu.cn

***Abstract*—Session-based recommendation system is an important part of many e-commerce sites. Its purpose is to recommend according to the interaction behavior of anonymous users in a short time. The latest research is to model the session sequence as a graph and then use the graph neural network to learn the embedding of the item. However, these methods treat the session as a simple sequence and ignore the time interval between user's adjacent interactions. In order to solve this problem, we propose a Time and Position Aware Graph Neural Networks model for the session-based recommendation systems, which can not only learn the embedding of items, but also capture users' interest by using the time interval and sequence information when users browse items. We have conducted sufficient experiments on two e-business datasets, and the experimental results show that our model is superior to baselines.**

***Keywords*—session-based recommendation, graph neural networks, time and position aware**

I. Introduction

As an effective tool to alleviate information overload, recommendation systems are widely used in various e-commerce websites, such as Amazon and eBay. The recommendation systems recommend primarily based on the user's historical behavior or relationship. But in many large-scale e-commerce websites, it will cost a lot of time to obtain the user's historical data, so the user's information is often unknown. At this time, it is very important to recommend based on the user's current session. Therefore, session-based recommendation system comes into being. The difficulty of session-based recommendation system is that it needs to model limited session information to capture users' interest [1].

Collaborative filtering [2] is a classic recommendation algorithm, but the session is anonymous, collaborative filtering cannot be applied in session-based recommendations. Shani et al. [3] and Rendle et al. [4] regarded the session sequence as Markov chain, and then use Markov decision process to make recommendation. However, the Markov chain is only recommended based on the last item of the session, ignoring the historical items clicked by the user, which is difficult to obtain a good recommendation performance. In addition, Markov chain has the problem of dimension explosion, which makes it difficult to apply on large datasets.

Hidasi et al. [5] applied Recurrent Neural Network (RNN) to session-based recommendation for the first time, and they stacked multiple Gate Recurrent Units (GRUs) [6] to learn embedding of items. Li et al. [7] combined the attention mechanism with RNN and used the attention mechanism to filter the noise data in the session. However, the problem of RNN's gradient disappears makes these two models difficult to capture users' global interests , and the session-based recommendation is not just a sequence prediction problem. Liu et al. [8] completely abandoned RNN and proposed the STAMP model, which only uses the attention mechanism to learn users' long-term interest and then makes recommendations based on users' short-term preferences. Wu et al. [9] proposed the SR-GNN model and applied the Graph Neural Network (GNN) in the session-based recommendation of the first time. The SR-GNN uses the advantages of graph neural networks to capture the complex transformation relationships of items in the session, and then generates the embedding of items for recommendation. SR-IEM [10] uses an importance extraction module to assign different importance to different items.

Although previous models have achieved good recommendation performance, we believe that these methods do not take into account the user's time spent on each item in the session, which may result in the information loss. For example, if the user browses a sequence of products: mobile phone, fan, camera, laptop, assuming that the user's browsing time for products is 3s, 1s, 9s, and 2s. It is obvious that the user spends the longest time browsing camera, so we can assume that the user may be more interested in the camera.

Therefore, in order to take advantage of the time information when users browsing items, we propose a novel Time and Position Aware Graph Neural Network model for session-based recommendation systems. As shown in figure 1, first, we construct the session into a graph, second, we use graph neural network to learn the dependencies between items and generate the embedding of the items. Third, we use the time and position aware layer to integrate the time and position information into the embeddings of items. Finally, we use the prediction layer to learn the user's preference and generate a list of candidate items. The main contributions of this work are as follows:

(1) The TPA-GNN model builds the session into a directed weighted graph to capture the transformation relationships between items.

(2) The model uses the Time and Position aware layer to encode items to reflect the importance of each item in the session.

(3) We have conducted extensive experiments on two widely-used datasets (Yoochoose and Diginetica datasets), and the results show that our model performs well.

In this paper, we first introduce the Time and Position Aware Graph Neural Networks, and then, we conduct experiments to verify the effectiveness of the model. Finally, we summarize the full text.

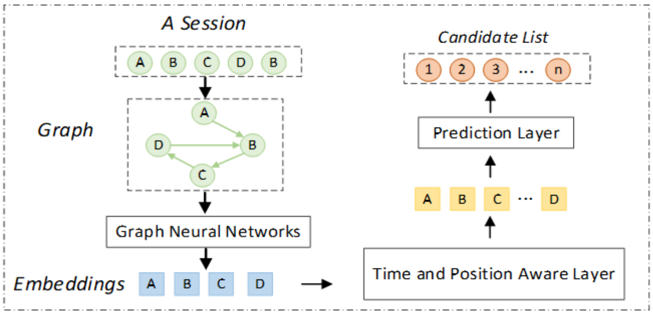


Fig. 1. Process of TPA-GNN layer

II. The Proposed Model

A. Em Description

The set of all items in the session is denoted by $X = \{x_1, x_2, \dots, x_u\}$, where x_i represents item, u represents the number of items in the dataset. We define each session sequence is $s = [x_1, x_2, \dots, x_l]$, where l is the length of the session sequence. Session-based recommendation systems are designed to predict labels x_l based on the first $l - 1$ item of session $[x_1, x_2, \dots, x_{l-1}]$.

We construct the session into a graph. For each session S, we can construct a session graph $G_s = (V_s, E_s)$, where V_s represents the set of items in session S, including items x_{i-1} and items x_i. E_s represents the set of edges in session S, and edges $(x_{i-1}, x_i) \in E_s$.

B. Learning the Embedding of Items

We define $H_{t-1} = \{h_{x_1}, h_{x_2}, \dots, h_{x_n}\}$, where h_{x_i} represents the embedding of the itemx_i in the session, n represents the number of items in the session. In SR-GNN [9], the weight of each edge is affected by the number of occurrences of edges in the graph. However, such weights cannot reflect the dependencies between each item. Inspired by the GAT model [11], [12], in order to adaptively learn the weight of each edge, we use the multi-head attention network to generate the weight.

$$\alpha_{ij} = \frac{\exp\left(LeakyRelu(a^T(Wh_{x_i}||Wh_{x_j})\right)}{\sum_{k\in N_i}\exp\left(LeakyRelu(a^T Wh_{x_i}||Wh_{x_j})\right)} \quad (1)$$

$$\hat{h}_{x_i} = \overset{K}{\underset{k=1}{||}} \sigma\left(\sum\nolimits_{j\in N_i} \alpha_{ij}^k \mathbf{W}^k \mathrm{h}_{x_j}\right) \quad (2)$$

Among them, $h_{x_i}, h_{x_j} \in R^d$ is the embedding of items, which needs to be enhanced by weight $W \in R^{2d\times d}$. a is a weight matrix, $||$ represents the concatenation of matrices.$LeakyRelu$ is an activation function with a negative input slope of 0.2.$k$is the number of heads of the multi head attention mechanism. $\hat{h}_{x_i}$ represents the embedding of items obtained by the transmission of graph neural network.

Then, we use the GRU to update the embeddings of items.

$$H_t = GRU\left(H_{t-1}, \hat{H}_{t-1}\right) \quad (3)$$

C. Time and Position Aware

At present, the latest work mainly uses a attention network to learn users' long-term preferences, and then generates users' interests combined with short-term preferences. But they did not consider the time interval and location information of each item. In our model, we design a time and location aware layer, which considers not only the time interval of user adjacent interaction, but also the location information of items, and then generate time and location aware embedding respectively.

In the dataset, each user interaction corresponds to a timestamp T. For a session $s = [x_1, x_2, \dots, x_l]$, assume that the corresponding timestamp is $T_1, T_2, \dots, T_l$. For the former $(l-1)$ item, we can obtain the corresponding time interval for each item, which is: $T_2 - T_1, T_3 - T_2, \dots, T_l - T_{l-1}$, we take this time as the user's browsing time on the item. In general, the longer a user browses an item, the more interested the user is in the item. In order to facilitate processing, we sort the time intervals from small to large. In this way, each item corresponds to a serial-number of time. Then, for the session S, we can adopt a time embedding layer:

$$T_i = [t_1, t_2, \dots, t_l] \quad (4)$$

where,$t_i \in R^d$ represents the position embedding vector. For example, t_1 indicates the embedding corresponding to the time sequence number of 1. In addition, in the TPA-GNN model, inspired by [11], we use sine and cosine functions to distinguish different positions of items:

$$\begin{aligned} P_i^{d'} &= \sin\left(\frac{1-i}{10000^{d'/d}}\right) \\ P_i^{d'+1} &= \cos\left(\frac{1-i}{10000^{d'/d}}\right) \end{aligned} \quad (5)$$

where, i represents the position of the item in session S, and d represents the dimension of the hidden layer, and d' is a dimension of the vector. For example, when d = 100, $d' = 0,1, \dots 49$.

After the session is trained using the graph neural network, we can get the embedding of each item, which is denoted by $H = [h'_{x_1}, h'_{x_2}, \dots, h'_{x_l}]$. We use the time and position aware layer to encode the item x_i:

$$h_{v_i} = \tanh\left(w_1\left[h'_{v_i}||T_i||P_i\right] + b_1\right) \quad (6)$$

where $w_1 \in R^{d\times 3d}$ is the weight matrix, $b_1 \in R^d$ is the bias term. And $tanh$ is the activation function of the neural network.

In the Time and Position Aware layer, first, we generate a D-dimension time vector for each item with the help of the time embedding layer. Then, we generate a D-dimensional position vector for each position in the session using the sine and cosine formula. Finally, we combined the original embedding vector h'_{v_i} of the item to get the final representation h_{v_i} of the item, including the time and position information of the item.

D. Generating Session Embedding

In a session, the complete session often reflects the user's global interest, while the last item clicked in the session sequence can reflect the user's local preference. we define local preferences $s_c = h_{v_l}$. In addition, we capture users' global interests s_g through a soft attention network:

$$a_i = Z^T \sigma\left(w_2 h_{v_i} + w_3 h_{v_l} + w_4 h_a + b_2\right) \tag{7}$$

$$s_g = \sum_{i=1}^{l} a_i h_{v_i} \tag{8}$$

where, $Z \in R^d$, $w_1, w_2, w_3 \in R^{d \times d}$ is the weight, and $b_2 \in R^d$ is the bias term.h_a is the mean value of the item embedding vector, and it's defined as:

$$h_a = \frac{1}{l} \sum_{i=1}^{l} h_{v_i} \tag{9}$$

Finally, we learn embedding of the session by combining global and local user preferences:

$$h_s = W_5 [s_g || s_c] \tag{10}$$

where $W_5 \in R^{d \times 2d}$ is the weight matrix of the fully connection layer.

E. Prediction Layer

After obtaining the embedding h_s of the session, we dot product it with the embedding of each candidate item, and then apply normalization to get the probability of the user clicking on each item next time:

$$\hat{y}_i = sofmax\left(h_s^T h_{v_i}\right) \tag{11}$$

Finally, we use the cross-entropy loss function to learn the parameters and use the back propagation algorithm to train the model.

$$L(\hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{12}$$

where, y_i is the one-hot encoding of the session label item.

III. Experiments

In order to prove the effectiveness of the TPA-GNN model, we will answer the following two questions:

RQ1: Is the TPA-GNN model better than the latest baselines?

RQ2: Is the Time and Position Aware layer of this model effective?

A. Experimental Configurations

Datasets: We demonstrated the validity of the TPA-GNN model on two widely used datasets, the Yoochoose[1] dataset and the Diginetica[2] dataset. For fair comparison, we use the same treatment as [8][9], first, we filter the session length of 2 and less than 5 items. For Yoochoose, we use the last day's interactive data as the test set and the rest as the training set. For the Diginetica dataset, we use the data of last week as the test set and the rest as the training set. Then, we adopt the data transformation technology of [13], that is, for the session $s = [v_1, v_2, \dots v_l]$, we generate a series of sessions and the corresponding labels: $([v_1, v_2, \dots, v_{l-1}], v_l)$, $([v_1, v_2, \dots, v_{l-2}], v_{l-1})$, … , $([v_1], v_2)$. This can increase the training data and is more conducive to the learning of model parameters.

Evaluation indicators: Following the previous work [8][9], we adopt two frequently used evaluation indicators: P@20 and MRR@20, which respectively reflect the accuracy of the model and the ranking of the label in the candidate projects.

Parameters Setup: We set the dimension of the hidden layer D =100, the number of multiple attention heads K =5, and the minimum batch size of 100. The weight matrix and embedding layer are initialized with a Gaussian distribution with mean value of 0 and variance of 0.01, and the initial values of all bias terms are set to 0. Finally, Adam optimizer was used to optimize the model. The initial value of the learning rate was 0.001, and then it attenuated by 0.1 every three iterations. L2 Penalty set to 10^{-5}.

B. Baseline Method

- POP recommends the most popular items in training set.
- S-POP recommends popular items in the current session.
- Item-KNN [14] recommends using the cosine similarity among the items.
- FMPC [2] uses Markov chain for recommendation.
- GRU4REC [3] utilizes GRU to learn the final representation of the session.
- NARM [7] combines the attention mechanism and RNN to capture users' interest.
- STAMP [8] additionally considers the user's short-term preference.
- SR-GNN [9] uses graph neural network to capture the transformation relationship of items.
- SR-IEM [10] proposes an IEM module to judge the importance of each item in the session.

C. Comparison with Baselines

Answer RQ1, we compare the TPA-GNN model with the common baselines listed in Section 3.2. The experimental results are recorded in Table 1, where the best results in each column are indicated in bold. As can be seen from Table 1, our model achieves the best prediction effect on both P@20 and MRR@20 datasets, which reflects the validity of our model.

[1] http://2015.recsyschallenge.com/challenge.htm

[2] http://cikm2016.cs.iupui.edu/cikm-cup

TABLE I. COMPARISON WITH BASELINE METHOD

Method	Yoochoose 1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
POP	6.71	1.65	0.89	0.20
S-POP	30.44	18.35	21.06	13.68
Item-KNN	51.60	21.81	35.75	11.57
FPMC	45.62	15.01	26.53	6.95
GRU4REC	60.64	22.89	29.45	8.33
NARM	68.32	28.63	49.70	16.17
STAMP	68.74	29.67	45.64	14.32
SR-GNN	70.57	30.94	50.73	17.59
SR-IEM	71.02	31.12	51.31	18.03
TPA-GNN	**71.13**	**31.17**	**52.16**	**18.15**

Among the traditional recommendation algorithms, POP has the worst performance, because it only recommends popular items without considering the difference of interest of each user. S-pop only recommends items that have been viewed, which limits the model's performance. The recommendation performance of item-KNN is further improved, but the sequence information of items is not considered. The FPMC assumes that items are independent of each other, which is difficult to establish.

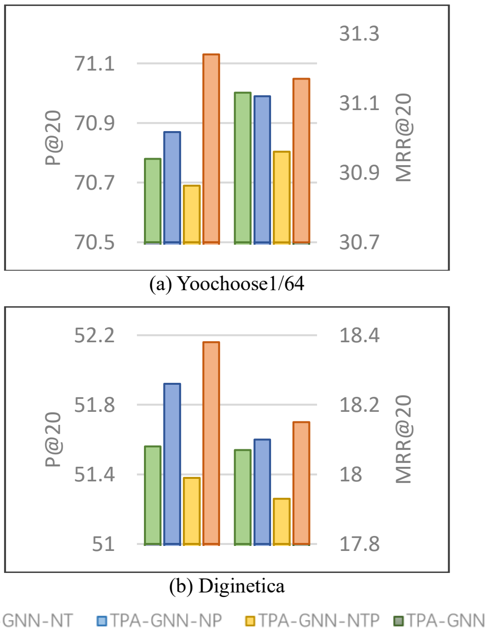
Recommended algorithms based on deep learning are often superior to traditional algorithms, such as GRU4REC, NARM and STAMP. This is because the deep learning algorithm can learn the embedding representation of the item and further get the user's interest. GRU4REC and NARM models model the session as a sequence and use RNN to learn the final representation of the session, but it is difficult to overcome the problem of RNN gradient disappearing. In addition, the session recommendation is not just a sequence prediction problem. STAMP model gives up RNN model completely, only uses the attention mechanism to capture users' long-term interest, and then makes recommendations combined with users' short-term interest.

The recommendation algorithm based on graph neural network further improves the session recommendation performance, because of the graph neural network can capture the transformation relationship between items. The SR-GNN model the session as a graph, then uses the GRU to learn the representation of the item, and finally uses the soft attention mechanism for prediction. The SR-IEM evaluates the importance of considering each project and then generates the final representation of items. These models have achieved better performance, but still inferior to TPA-GNN model, because our model uses graph attention layer to capture the dependencies between items, and then uses the Time and Position Aware layer to generate the final representation of each item in the session, uses soft attention mechanism to learn the user's global preference, and finally combine user's short-term preferences to achieve the best recommendation performance.

D. *The Influence of Time and Position Aware Layer*

The Time and Position Aware layer can learn the contribution of each project more accurately. In order to answer RQ 2 and prove the effectiveness of the time and Position Aware layer, we design the following comparison experiments:

- TPA-GNN-NT: It does not include the time embedding vector.
- TPA-GNN-NP: It does not include position vectors.
- TPA-GNN-NTP: It does not include time embeddings, nor contains position vectors.



(a) Yoochoose1/64

(b) Diginetica

Fig. 2. Performance of Time and Position Aware layer

We conducted experiments with these comparison models and TPA-GNN on two datasets, and the results are shown in Figure 2. Obviously, the TPA-GNN model achieves better results with the help of the Time and Position layer. The TPA-GNN-NTP model has the worst results because it takes into account neither the time interval nor the position information of items of the session. The performance of TPA-GNN-NT model and TPA-GNN-NP model is better than that of TPA-GNN-NTP model, but lower than that of TPA-GNN model, which proves that both time encoder and position encoder can reflect the importance of items. It is worth noting that the TPA-GNN-NP model performs slightly better than the TPA-GNN-NT model in the Diginetica dataset, possibly because the location information of items in the session is more important than the time information.

IV. CONCLUSION

We propose a Time and Position aware graph neural network model for session-based recommendation systems, which can learn user's interest more accurately. Experimental results show that the TPA-GNN model is better than the nine baselines on two different datasets. As for future work, we will use more auxiliary information to learn the embedding of the item, such as neighbor sessions.

ACKNOWLEDGMENT

This paper is made possible thanks to the generous support from the Natural Science Foundation of Shandong Province (ZR202103030440), PhD Funding Project of Shandong

Jianzhu University (No.X19044Z)，Science and Technology Program Project of Shandong Colleges and Universities (No.J17KA070), Undergraduate Education Reform Project of Shandong Province (2021).

REFERENCES

[1] Wang, Shoujin, et al. "A survey on session-based recommender systems." ACM Computing Surveys (CSUR) 54.7 (2021): 1-38.

[2] Su X, Khoshgoftaar T M. A Survey of Collaborative Filtering Techniques[J]. Advances in Artificial Intelligence, 2009, 2009(12).

[3] Shani G, Heckerman D, Brafman R I, et al. An MDP-Based Recommender System[J]. Journal of Machine Learning Research, 2005, 6(1):1265-1295.

[4] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation[C]// Proceedings of the 19th International Conference on World Wide Web,WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010. ACM, 2010.

[5] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based Recommendations In Recurrent Neural Networks[J]. ICLR '16, 2015.

[6] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[7] Jing Li, Ren P, Chen Z, et al. Neural Attentive Session-based Recommendation[J]. In:Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, Singapore Singapore, 2017.

[8] Qiao Liu, Zeng Y, Mokhosi R, et al. STAMP:Short-Term Attention/Memory Priority Model for Session-based Recommendation[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018.

[9] Wu S, Tang Y, Zhu Y, et al. Session-Based Recommendation with Graph Neural Networks[C]// Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence. 2019.

[10] Pan Z, Cai F, Y Ling, et al. Rethinking Item Importance in Session-based Recommendation[J]. In:Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Virtual Event China, 2020.

[11] Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances In Neural information processing Systems. Pp 5998 -- 6008.

[12] Veličković P, Cucurull G, Casanova A, et al (2018) Graph Attention Networks. In ICLR '16,2018.

[13] Tan YK, Xu X, Liu Y (2016) Improved Recurrent Neural Networks for Session-based Recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, Boston MA USA:17-22.

[14] Sarwar B, Karypis G, Konstan J, Reidl J (2001) Item-based collaborative filtering recommendation algorithms. In:Proceedings of the Tenth International Conference on World Wide Web-www '01. ACM Press, Hong Kong, Hong Kong: 285 -- 295.

A Forensic Method for DeepFake Image based on Face Recognition

Jian Wu
School of Cyberspace Security, Shandong University of Political Science and Law, Jinan 250014
+86 13808932046
jinanwujian@163.com

Kai Feng
School of Cyberspace Security, Shandong University of Political Science and Law, Jinan 250014
+86 18813003272
fkdhy@163.com

Xu Chang
School of Cyberspace Security, Shandong University of Political Science and Law, Jinan 250014
+86 13791036385
changxumail@163.com

Tongfeng Yang
School of Cyberspace Security, Shandong University of Political Science and Law, Jinan 250014
+86 15053130726
yangtf2014@126.com

Abstract

DeepFake digital images have serious negative impacts on news integrity, legal forensics, and social security. In order to detect the DeepFake digital images more accurately, a method based on face recognition is proposed. Face image feature vectors are extracted by Facenet, and the Euclidean distances among the vectors of different face images are calculated as classification principle. Then, the machine learning algorithms is trained to perform binary classification of real and fake face images. The experimental results on the Celeb-DF data set show that the proposed method has better detection effect than the existing detection methods.

CCS CONCEPT

Computing methodologies → Artificial intelligence → Computer vision → Computer vision representations → Image representations

Keywords

DeepFake, Face Recognition, Image Forensic

1 Introduction

In the era of highly developed digital technology, as the functions of various image editing and processing software become more and more complete and popular, people can easily forge and tamper with digital images. Especially in the past two years, applications such as FakeApp and ZAO have appeared. The representative series of DeepFake technology can use deep learning technologies such as generative adversarial networks (GAN) to easily forge high-precision information such as faces in images or videos. Not only is it difficult for the human eye to recognize, but also existing images/Video forensics analysis technology is also completely ineffective [1]. When such images are used in news reports, legal forensics, insurance claims, etc., it will cause irreversible negative effects, cause violations of civil rights, and even undermine social stability, national security and international order.

With the use of DeepFake technology in the case of fraud, some experts and scholars have worried that it will become an information warfare weapon that misleads public opinion. Therefore, How to conduct forensic analysis on DeepFake image or video is attracting more and more attention from governments and technology enterprises all over the world. In August 2019, the U.S. Defense Advanced Research Projects Agency (DARPA) held a theme day event to "prevent the widespread spread of maliciously forged images, audio, and video generated by artificial intelligence", marking the "Media Assurance" project has shifted to the detection of DeepFake technology. In September 2019, Facebook said that it would jointly launch a "DeepFake detection challenge" with Microsoft and MIT universities, and invested $10 million to promote the development of DeepFake detection technology [3]. Since 2019, media including CCTV have extensively reported on the application and security threats of DeepFake technology, which has aroused the attention and concerns of domestic scholars and the general public. With the open source of some DeepFake techniques, DeepFake image has a great potential to spread, but the research on DeepFake image forensics has just started.

DeepFake images use the discriminator and generator's adversarial mode to continuously generate images. The feedback from the discriminator enables the generation model to produce highly realistic fake images. In theory, existing forensic analysis methods can be used as discriminators to promote the generator. Bypassing known detection methods produces unrecognizable fake images. Therefore, forensics on DeepFake images is more challenging than manual fake images.

2 Previous Works

At the end of 2017, a user named DeepFake created the term DeepFake by using machine learning to "change" a porn star's

HPCCT & BDAI 2020, July 3–6, 2020, Qingdao, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7560-3/20/07…$15.00
https://doi.org/10.1145/3409501.3409544

face in real time. The term DeepFake was also created, which represents the use of deep learning technology to realize digital images of people, audio and video generation or modification, to achieve the purpose of deceiving the audience with the information content. The original DeepFake uses a deep autoencoder to replace the face areas of the two images.

The late open source tool faceswap-GAN adds the adversarial loss and perceptual loss into the encoding and decoding structure, and realizes the GAN-based DeepFake, which can automatically adjust the eye movement, the head posture, the facial expression and even the light, to create a highly realistic image or video. This kind of deep forgery technology is still developing innovatively [5]. As a result, a large number of existing blind forensics methods[6,7] fail due to high error rate when detecting deep forgery information [8].

The blind forensics of DeepFake images can be summarized as a binary classification problem. The key is to discover the use of various features that can reveal the traces of specific forgeries. The existing research results are mainly divided into three categories: one based on the difference between the human image in video and the normal human image, including detecting whether people blink in the video [9], whether the head posture is continuous and natural [10], analyzing whether the color of the eyes in the contrast image is consistent, and whether the tooth gap is visible [11], etc. This type of method can only perform forensic analysis on specific scenes, and can't deal with new DeepFake technologies. The second type is detection methods based on the statistical characteristics of natural images. For example, using the traditional method for analyzing the inconsistency of image illumination response, Koopman divided the image region into eight groups and calculated the average PRNU mode noise of each group. According to the difference of the standardized cross-correlation scores between the real image and the forged image, the differences are distinguished [12]; Zhang et al. [13] used the Bag-of-words method to extract 64-dimensional SURF (Speeded Up Robust Features) features of human faces and tested them with Support vector machine, random forest and multi-layer perceptron to distinguish the exchanged facial images from the real ones. However, feature design and extraction of this kind of methods mainly rely on manual extraction of human experience, because feature extraction and classifier training are carried out alone, can not guarantee the synchronous optimization of both. The third type is a detection method based on deep learning technology. For example, [14] uses a four-layer CNN network and combines it with Inception to train on its self-built DeepFake data set. The detection accuracy rate is up to 98.4%. In reference [15], a set of GAN based false face images is tested by a CNN with three convolution layers and two fully connected layers, which is based on the high-pass filter used in image steganalysis, the detection accuracy of 256×256 resolution face image is over 99%. Li et al. used VGG16 and three ResNet networks for training and detection based on the resolution inconsistency between the fake distorted facial region and the surrounding images, combined with UADFV and DeepFakeTIMIT data sets, the results show that the detection accuracy of ResNet is higher than that of Literature [10] and [14]. [17] proposed one type of twin network containing Dense modules detects DeepFake images, and the accuracy of testing in five sets of DeepFake images based on GANs is more than 90%. However, the above methods only detect the known DeepFake techniques, without considering the detection problem of the unknown forgery methods. In particular, a considerable proportion of the images in the existing DeepFake data set have obvious tamper characteristics, even if the human being can easily identify whether they are forgeries or not, and the neural network is trained by using this kind of data set with poor quality of forgery, it will lead to weak detection and generalization ability of the network, which can not be used to obtain the higher quality of the fake video or image in the real network.

3 Celeb-DF data set

In order to better evaluate existing DeepFake detection methods and promote the development of detection methods, in 2019, Li et al. constructed Celeb-DF data set containing DeepFake videos with better visual quality [18]. Specifically, the Celeb-DF data set includes 408 real videos and 795 synthetic videos generated with DeepFake. The average video length in the Celeb-DF data set is 13 seconds, and they all have a standard 30FPS frame rate. Contrast to the images in UADDV [19], DeepFake-TIMIT [20], and the DeepFake subset in FaceForenscics++ [21], the Celeb-DF data set solves the problem of low resolution of synthesized faces, color inconsistency, visible facial parts of the original face, temporal flickering, etc., eliminating forgery traces that can be directly discerned by the naked eye, some sample frames of Celeb-DF are shown in Fig.1. The authors of [18] tested several existing detection methods, such as Two-stream DNN, MesoNet, HeadPose, Fwa, VA-MLP, Multi-task, Xception, on Celeb-DF data set, the greatest decline was seen in the AUC(Area Under the Receiver Operating Characteristics), which dropped from the highest level of 99.7% to 38.7%. The reason for the decrease in detection performance is as the author points out, these methods can not detect the available spurious features in the Celeb-DF data set, such as low resolution, color mismatch and visible boundaries. Thus, using Celeb-DF machine learning data set expected to improve detection performance of DeepFake.

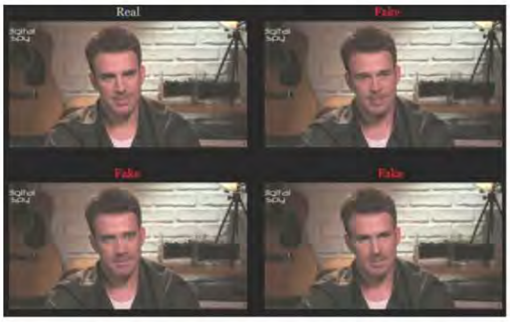


Figure 1. Sample frames of Celeb-DF

4 Our Method based on Face Recognition

By comparing the fake image and the original image in the Celeb-DF data set, it can be seen that the "face-changing video" is a fake image formed by covering the face of a star with the face of the original video character through DeepFake technology. In order to ensure that the fake traces are not found, the facial expression is more natural, and the original face image needs to be appropriately transformed. This results in a certain difference between the fake image and the face of the original image. Therefore, if the original person can be obtained, the face recognition technology can be used to determine whether the face image has been tampered with using face-changing technology. In most cases, this assumption can be satisfied.

In order to extract a face image suitable for network training, ffmpeg is used to extract frames from the video in Celeb-DF. Because the appearance of adjacent frames in the video changes little, an image is extracted every 10 frames. In order to reflect the diversity of faces in the data set as much as possible, and reduce the computation, only a small number of images are extracted from each image video. Then, MTCNN is used to locate and capture the face in the extracted image. Because the face regions are of different sizes, all face images are scaled to 128x128. Considering the balanced data set, we use the face image extracted from the real video in Celeb-DF as a positive example, each video extracts 40 human face images, and each fake video extracts 10 human face images and get the same number of negative examples as the positive examples. Specifically, there are 5886 fake face images and 5886 original face images.

5 Network Structure Design

In this paper, the network structure of face recognition mainly includes two parts: a feature extraction module and a classification module. The network structure design is mainly shown in Figure 2. Among them, the feature extraction uses the facenet network proposed by Google, and its main function is to convert facial images Map to a multi-dimensional space to get the vector representation of the image in space, and then calculate the spatial Euclidean distance between the feature vectors to represent the degree of acquaintance between the two images [22]. The spatial Euclidean distance between the same faces is small, and the spatial Euclidean distance between different faces is far, so face recognition can be realized through the spatial mapping of face images. Facenet uses a deep neural network-based image mapping method and a Triple Loss-based loss function to train the neural network, the loss function is shown as formula (1).

$$Loss = \sum_{i=1}^{N} \left[\| f(A^{(i)} - f(P^{(i)}) \|_2^2 - \| f(A^{(i)} - f(N^{(i)}) \|_2^2 + \alpha \right]_+ \quad (1)$$

Where A is a picture of a person, P is a picture of same person as A, N is a picture of another person. [z]+ denote max(z,0).

In network training, a batch of face images is first input into a convolutional neural network structure (such as vggnet, googlenet, ResNet) for feature extraction, and then the features are L2 normalized so that all image features are mapped to a hypersphere, the difference caused by the imaging environment of the sample is avoided, and the image embedding space is finally obtained after L2 normalization. The feature vector (embedding) uses Triplet Loss as the loss function for model optimization. The guiding principle of its optimization is to make the distance between the samples of the same face smaller than the distance of samples of different faces. By using a large number of face images to update and optimize the network parameters, the final facenet network can output multi-dimensional feature vector according to the face image directly, and can compare the difference of different images. Facenet was originally applied in traditional face recognition and achieved good recognition results. In our work, the trained network structure 20170512-110547 is directly used for feature extraction, the main model adopts a deep network Inception-ResNet, the output face feature dimension is 128.

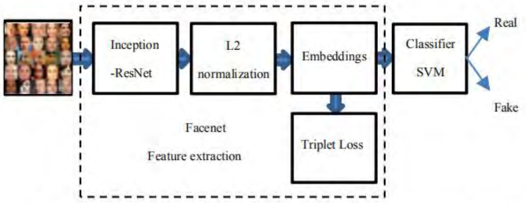


Figure 2. The structure of face recognition and classifier

In traditional facenet-based face recognition applications, when the Euclidean distance between the feature vectors of two face images is greater than 1.1, the faces are considered to belong to different people. But in the DeepFake image, the forged image often has the high similarity with the original image, therefore, can not copy the threshold as the classification standard. Therefore, in order to determine a reasonable Euclidean distance as a classification criterion, in the classification module, the facial features extracted by the feature are embedded into the spatial feature vector (embedding) as an input, using a machine learning algorithm (such as SVM, etc.) to learn the difference between the Euclidean distance of the features of real and fake face images, and select reasonable classification criteria through machine learning to achieve the two classification of real and fake face images.

6 Experimental Results and Analysis

The experimental platform is windows10/Opencv/ Python3.7/Tensorflow1.14. In order to prove the difference between real face images and fake face images, and to verify the feasibility of this solution, we first selected real human faces and fake human faces from the same person (Angela Julie) for comparison, part of the face image is shown in Figure 3.

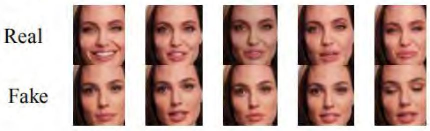


Figure 3. The face images of Angelina Julie

By using facenet to map the test images into vector space of 128 dimensions, and calculate the Euclidean distance between real and real, and the Euclidean distance between real and fake, it is found that the test images are distinguishable, as shown in Table 1.

Although the Euclidean distance were all less than 1.1, it can be seen that the detection results of the two sets have significant differences. It is verified that the Euclidean distance difference between the feature vectors of the real image and the fake image can be used to classify the DeepFake images. So, we select 585 real images and forged image from the face image set, which extract from Celeb-DF as described in section 4.1. The overall data is divided into two parts: training and testing. The training data accounts for 80%; the test data accounts for 20%. First, the training and test data are all generated using the facenet network structure to generate embedding vectors, and then the vectors are used for training and classification. In the algorithm, the classification accuracy is tested. The classification algorithms used are GBDT(Gradient Boosting Decision Tree), logistic, and SVM. The results are as follows:

Table 1. Euclidean distance between different images

Test No.	Real-Real	Real-Fake
0	0.658	1.033
1	0.531	0.921
2	0.67	0.812
3	0.579	0.97
4	0.655	1.091
5	0.619	0.91
6	0.772	0.868
7	0.534	0.775
8	0.873	0.872
9	0.937	0.894
10	0.77	0.916
11	0.963	0.89
Average	0.713	0.913

Table 2. Accuracy results of three classifier

Test No.	Logistic	GBDT	SVM
1	0.8	0.75	0.8
2	0.84	0.84	0.83
3	0.76	0.84	0.75
4	0.79	0.79	0.88
5	0.78	0.88	0.84
Average	0.794	0.82	0.82

It can be seen from the test data in the table above that the three machine learning algorithms have a certain classification effect, and the overall recognition accuracy is about 80%, but the recognition accuracy of the GBDT and SVM algorithms is more stable.

In order to verify the advantages and disadvantages of this method and other traditional DeepFake detection methods, we choose the classification method of GBDT Algorithm and 7 algorithms listed in the literature [18] to carry out the contrast test in Celeb-DF data set. The performance of each methods is measured by using *AUC* as shown in formula (2).

$$AUC = \frac{TP + TN}{P + N} \tag{2}$$

Where *TP* denotes the number of samples correctly classified as positive samples, and *TN* represents the number of samples correctly classified as negative samples, *P* is the number of positive samples, *N* is the number of negative samples. The value range of AUC is in [0,1], and higher value denotes better detection performance. Many of these methods use a deep neural network but we haven't the code for training the network, so the *AUC* of traditional method is copy from reference [18]. The ROC curve of the method based on face recognition (FR) relative to the *AUC* index is shown in Figure 4.

Table 3. *AUC* performance of each methods on Celeb-DF

Methods	*AUC*(%)
Two-stream	55.7
Meso4	53.6
HeadPose	54.8
FWA	53.8
VA-MLP	48.8
Multi-task	36.5
Xception	38.7
FR	83.6

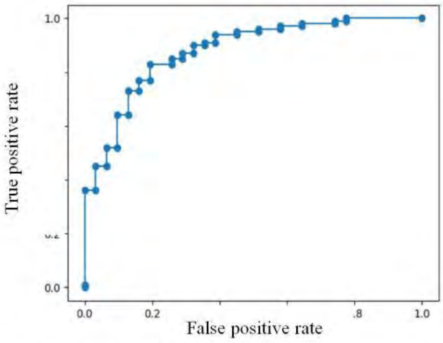


Figure 4. The ROC of the method based on face recognition

As can be seen from the Table 3, the method based on face recognition obtains better detection performance than traditional DeepFake detection methods. The reason may be the traditional detection methods are based on low resolution, color mismatch and visible boundary artifacts to detect DeepFake image. When such visual artwork is removed from the video in the Celeb-DF data set, the performance of these detection methods is reduced. This can also be attributed to the fact that many of these methods are not trained on Celeb-DF data set and therefore do not apply to detect the synthetic video.

7 Conclusion

In this paper, a new method of DeepFake image forensic analysis based on face recognition is proposed. Combining with face recognition technology such as facenet, we test Celeb-DF, which is a high quality data set, experimental results show the effectiveness of the proposed method. In addition, the amount of data used in the experiment is small, which can reduce the threshold for identification and analysis, and has obvious advantages over some deep learning methods. However, the method in this paper needs to obtain the original image of the identified person, and there are some visual differences between the forged image and the original image, so it can not be used for forensic analysis of such DeepFake methods as Face2Face, which only modify the expression of the character, the extension of the method is affected to a certain extent. In the next step, more artificial intelligence techniques(such as GAN) should be used to improve this method.

8 ACKNOWLEDGMENTS

This work is supported by 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 2019RKB01369; Open Fund of the Key Lab of Forensic Science KF202015, Ministry of Justice, China (Academy of Forensic Science).

9 REFERENCES

[1] Pavel Korshunov, Sebastien Marcel. DeepFakes: a New Threat to Face Recognition? Assessment and Detection. arXiv:1812.08685,2018.

[2] Lin X, Li J H, Wang S L, Liew A W C , Cheng F , Huang X S. Recent Advances in Passive Digital Image Security Forensics: A Brief Review. Engineering, 2018, 4:29-39.

[3] https://finance.sina.com.cn/stock/relnews/us/2019-09-06/doc-iicezzrq3832029.shtml.

[4] Korshunova I, Shi W, Dambre J, and Theis L. Fast face-swap using convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. Italy:2017,3677-368.

[5] JY Zhu, T Park, P Isola, AA Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In: Proceedings of the IEEE International Conference on Computer Vision. Italy:2017,2223-2232.

[6] Galball J, and Marcel S. Face anti-spoofing based on general image quality assessment. In:22nd International Conference on Pattern Recognition . Sweden: 2014 ,1173-1178.

[7] Wen D, Han H, Jain A K. Face spoof detection with image distortion analysis. IEEE Transactions on Information Forensics and Security,2015,10(4):746-761.

[8] Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Saeid Nahavandi. Deep Learning for DeepFakes Creation and Detection. arXiv:1909.11573,2019.

[9] Yuezun Li, Ming-Ching Chang, Siwei Lyu. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In:International Workshop on Information Forensics and Security, Hong Kong, 2018.

[10] Yang X, Li Y, Lyu S. Exposing deep fakes using inconsistent head poses. In:2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Brighton:2019, 8261-8265.

[11] Matern F, Riess C, Stamminger M. Exploiting visual artifacts to expose DeepFakes and face manipulations. In:2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). USA: 2019,83-92 .

[12] Koopman M, Rodriguez A M, Geradts Z. Detection of DeepFake video manipulation. In: The 20th Irish Machine Vision and Image Processing Conference (IMVIP) . Ireland: 2018, 133-136.

[13] Zhang Y, Zheng L, Thing V L. Automated face swapping and its detection. In:2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP). Singapore: 2017, 15-19.

[14] Darius Afchar, Vincent Nozick, Junichi Yamagishi, Isao Echizen. Mesonet: a compact facial video forgery detection network. In: IEEE International Workshop on Information Forensics and Security (WIFS). Hong Kong, 2018.

[15] Mo H, Chen B, Luo W. Fake faces identification via convolutional neural network. In :Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security . Innsbruck, Austria, 2018, 43-47.

[16] Li, Y., and Lyu, S. Exposing DeepFake videos by detecting face warping artifacts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, 46-52.

[17] Hsu, C. C., Zhuang, Y. X., and Lee, C. Y. . Deep fake image detection based on pairwise learning. Preprints, 2019050013.

[18] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi and Siwei Lyu. Celeb-DF: A New Dataset for DeepFake Forensics. arXiv preprint arXiv:1909.12962v2.2019.

[19] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2019.

[20] Pavel Korshunov and Sebastien Marcel. DeepFakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685, 2018.

[21] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In ICCV, 2019.

[22] Florian Schroff, Dmitry Kalenichenko, James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. CVPR, 2015

2019 IEEE 5th International Conference on Computer and Communications

Blind Forensic Method Based on Convolutional Neural Networks for Image Splicing Detection

Jian Wu, Xu Chang, Tongfeng Yang, Kai Feng
School of Cyberspace Security
Shandong University of Political Science and Law
Jinan, China
e-mail: jinanwujian@163.com

Abstract—**In order to effectively detect whether digital images are spliced, a blind forensics method of digital image splicing based on deep learning is proposed. The method uses high-pass filter to preprocess the image, weakens the negative influence of image content on tampering forensic analysis, implements feature selection and classification based on convolutional neural networks(CNNs) to realize the classification of real images and spliced images. Experiments on the Columbia image splicing detection evaluation dataset and comparison with traditional forensic methods show that the proposed method can achieve better detection accuracy.**

Keywords-blind forensic; image splicing detection; convolutional neural networks

I. Introduction

Nowadays digital technology has highly developed, with the increasing popularity of image editing software, digital images can be easily forged and falsified, making it difficult or even impossible for human eyes to recognize. Especially when such images are used for scientific research, legal forensics, news reports, insurance claims, etc., it will have irreversible negative effects. Therefore, image forensics technology for identifying the accuracy, integrity and originality of digital images has emerged as the times require, and has become a research hotspot in the field of computer forensics [1].

In the existing digital image forensics technology, image blind forensics belongs to a passive authentication technology, which does not need to embed the authentication information in the image in advance, and has higher application value than the active authentication technology such as digital watermark. Due to the variety of methods for image tampering, the existing digital image blind forensics methods are also different. Among them, the copy-splicing tampering between different images is to splicing different parts of two or more images to achieve the forgery effect, as a frequently used image tampering method, and its forensic analysis method has received extensive attention from scholars. The blind forensic analysis methods mainly include the following three categories: First, the detection method based on the consistency of imaging equipment. This method is similar to the image source forensics method. It is based on the imaging consistency feature of digital cameras to identify whether an image comes from different photographic equipment. The current research methods include CFA interpolation detection, CCD mode noise detection, color difference detection, and camera response function detection, etc. [2]. Second, the detection method for the post-splicing processing operation. For the tampering, after the image is spliced, jpeg compression, de-noising, blurring, etc. are often used to erase the traces left by the tampering to the image. The method determines the tampering area by analyzing the characteristics of a tampering operation. Existing forensic analysis methods include jpeg recompression detection, fuzzy retouch detection, resampling detection, and the like. Third, a kind of detection method based on the statistical characteristics of natural images. The method will identify the tamper image and the natural image as a two-class problem, extract the high-order statistical characteristics of the image as a feature, and train the svm and other classifiers to complete the detection.

In recent years, deep neural networks have been widely used in the field of images, such as image segmentation, image fusion, image classification, etc., and some scholars have applied it to image tampering for evidence. For example, Chen et al. [3] proposed the use of deep learning to solve the problem of median filtering forensics. This algorithm can significantly improve the detection performance of median filter tampering images. Yang et al [4] proposed a Laplacian convolutional neural network algorithm to detect reacquired images, and the detection accuracy of different size image libraries exceeds 95%.The literature [5] proposed the use of convolutional neural network to solve the problem of camera source forensics, and the classification accuracy rate of 27 kinds of cameras exceeded 94%. It can be seen that deep learning technology has broad application prospects in image tampering and evidence collection.

Therefore, this paper proposes a method based on convolutional neural networks for image tampering blind forensics. The method adopts a network including a pre-processing layer, the convolution layer, the BN layer, the ReLU layer and the pooling layer to realize automatic extraction of image features; a fully connected network of one layer is used to implement categorization of tampering images and natural images. We trained and tested a CNN using the Columbia image splicing detection evaluation dataset. The experimental results show that the proposed method has higher detection accuracy than the traditional detection method for the spliced image detection.

978-1-7281-4742-0/19/$31.00 ©2019 IEEE

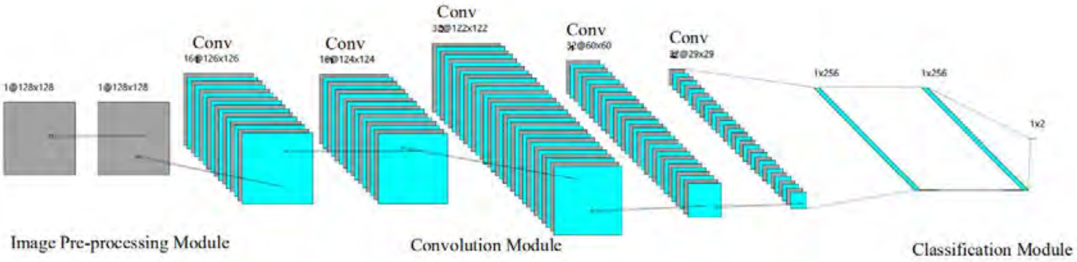


Figure 1. The network structure in our experiments. This structure consists of one image processing layer, five convolutional layers, and three fully connected layers. The form "$a@b\times b$" means the number of feature maps a and resolution $b\times b$ of the corresponding layer.

II. Network Structure

In this paper, a CNN model as shown in Fig. 1 is constructed for image splicing detection. The whole framework is divided into image pre-processing module, convolution module and output module. The original image enters the convolution module after high-pass filtering in the pre-processing module, and performs convolution operation to extract features in the convolution module. Finally, the output module outputs the probability of the category to which the image belongs, and the whole classification is completed.

A. Brief Review of CNNs

As one of the most representative deep learning approaches, CNNs have attracted a significant amount of attention given their ability to automatically learn classification features directly from data, such as images, text and speech. Especially in objection recognition and image classification tasks, the CNNs used the trainable filters and pooling operations alternatingly to the raw input images, and have achieved superior performance.

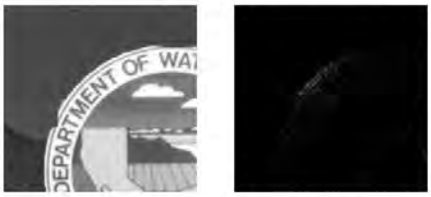
CNNs combine three architectural ideas: local receptive fields, shared weights, and pooling. Compared to standard feed forward neural networks with similar size layers and images, CNNs have much fewer connection. Hence, they are much easier to train the parameters. A typical CNN is composed of the convolutional layer and the classification layer. The convolutional layer usually follows a pooling operation, which also called pooling layer.

Since the introduction of AlexNet [6] in 2012 by Hinton, the CNNs have become the dominating approach for image classification and image segmentation. Various new architectures, including NiN[7], VGG[8], Inception[9], ResNet [10], DenseNet[11], and NASNet[12], have been proposed since then. They vary in connection style between the layers and how convolution and pooling operations are realized. From the references, we can see a steady trend of model accuracy improvement. For example, the top-1 validation accuracy on ImageNet in NASNet-A has been raised to 82.7%.

B. Image Pre-processing Module

When traditional CNNs are applied to image segmentation and image classification research, it mainly focuses on identifying the specific content of image representation. The extractive characteristics of machine learning are often related to image content. The difference is that the main object of feature extraction in image tampering analysis is the tampering leftover processing trace, and the content of the image should be suppressed as noise affecting learning. For this reason, Chen et al. performed a median filtering residual calculation on the image before the convolution operation, which can significantly improve the detection ratio of the median filtering operation; Peng Zhao et al. used three SRM filtering kernels to preprocess the image, which input to the faster R-CNN network. Similar to reference[13], we use the high-pass filter used in the literature [14] to preprocess the image. The filter kernel is shown in the formula (1) below. The comparison of the filtering effect for a certain picture is shown in Figure 2. Comparing the two pictures, it can be seen that the high-pass filtering pre-processing can eliminate the influence of image content on CNNs training and learning, highlight the splicing traces, and help to improve the tampering classification accuracy.

$$F=\frac{1}{12}\begin{bmatrix}-1 & 2 & -2 & 2 & -1\\ 2 & -6 & 8 & -6 & 2\\ -2 & 8 & -12 & 8 & -2\\ 2 & -6 & 8 & -6 & 2\\ -1 & 2 & -2 & 2 & -1\end{bmatrix} \qquad (1)$$



Original image Filtering image

Figure 2. The filtering effect on a certain picture.

C. Convolution Module

As shown in Figure 1, the convolution module is composed of the convolution layer, the batch normalization layer (BN layer), the activation function layer and the pooling layer. The convolution layer convolves the different convolution kernels with the image channels provided by the pre-processing layer by sliding the window, thereby extracting different features of the original image. The convolutional layer is followed by the BN layer. Because the input range of each layer of data is constantly changing during the network training process, and this change will be amplified in the network, resulting in the network training process is more difficulty. With the BN layer, the data is normalized to effectively train the network while helping to suppress network overfitting. In order to improve the nonlinear mapping ability of the network, a nonlinear activation layer is connected behind the BN layer. Our algorithm uses a modified linear unit (ReLU) as the activation function. Compared to the saturated nonlinear function such as TanH, the ReLU function can solve the gradient explosion or gradient disappearance problem, and also help to speed up the convergence. It is worth noting that in the traditional CNNs network structure for image classification and image recognition, the maximum pooling layer is connected after the activation function layer to achieve the extraction of the main features of the image, and at the same time, it reduces the network scale and improve the convergence speed. However, in the image tampering analysis, it is easy to eliminate the hidden tampering traces by using the maximum pooling operation. Therefore, the network does not use the pooling layer after the first two convolutional layers, and only use the average pooling layer after the last three convolutional layers for considering more pixel effects. In the process of extraction of tampering features in this algorithm, in order to achieve the classification effect, five conventional convolutional layers are set, namely "Conv1", 16 filters, the size is $3\times3\times1$; "Conv2", 16 filters, size $3\times3\times16$; and "Conv3" with 32 filters of size $3\times3\times16$; and "Conv4" with 32 filters of size $3\times3\times32$; and "Conv5" with 32 filters of size $3\times3\times32$. Each convolution layer's step size is equal to 1, they all followed by BN layer and the activation layer, Conv3, Conv4, and Conv5, is followed by an average pooling layer, where each convolution layer uses *k* filters to convolute with the input to generate *k* new feature channels for subsequent processing.

D. Classification Module

Following the convolutional module is two fully connected layers each containing 256 neurons. Each neuron in the fully connected layer is connected to all neurons in its previous layer. The fully connected layer can integrate the localized information of the class in the pooling layer. In order to improve the non-liner mapping performance of CNN, the activation function connected to each neuron in the fully connected layer also uses the ReLu function. The loss function layer connected after the activation function layer is classified by the softmax logistic regression function, and the cross entropy is used as the loss function. In order to avoid training over-fitting, the network uses the dropout regularization method to invalidate some implicit nodes, that is, these nodes do not participate in the forward and backward propagation of CNN.

III. Experimental Verification

A. Image Sets and Image Expansion

The experiment used a standard test library established by Columbia University's DVMM lab for the analysis splicing forgeries in images. The database includes 1845 bmp gray images of 128×128 pixels, of which 933 real images and 912 splicing images, and no post-processing such as landscaping after splicing. The database includes animals, plants, landscapes, and buildings, divided into five sub-categories based on the combination of smoothing and texture: uniform texture, uniform smooth, texture texture, smooth smooth, and texture smooth.

The amount of data in the Columbia image library is small. When used for deep learning, it tends to lead to poor generalization of the model. In order to increase the diversity of data, the original image can usually be randomly cropped, translated, scaled, mirrored, rotated, added noise and other data expansion processing. But for the image tamper analysis, because the purpose is to determine whether an image has been modified, and the data expansion method of changing the image pixel value is likely to cause the real image to be mistaken for tampering. Therefore, the data expansion in this paper only performs horizontal mirroring and rotation of 3 specific angles (90°,180°,270°) for all images of the Columbia image set, and does not change the image itself and its corresponding category label. In this way, a total of 8 samples can be obtained for each image, thereby expanding the number of original image sets by eight times. The specific example is shown in Figure 3. To verify the validity of the algorithm, in each experiment, 80% of the same proportion of real and stitched images were randomly selected from the extended image library for training the CNN network, 10% of the images were used for verification, and the remaining 10% were used for test.

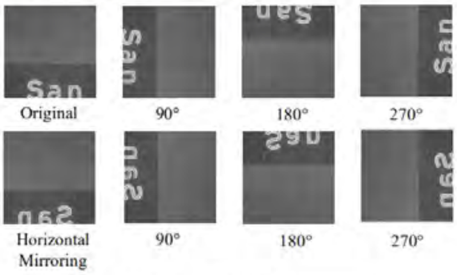


Figure 3. Example of a image expansion.

B. Experimental Parameter Settings

The CNN network in this paper is designed and implemented by Matlab2019a. The hardware platform is Intel(R) Xeon(R) E5-1603 2.8GHz, 32GB RAM and an NVIDIA GeForce TITAN GPU. The learning parameters of the network are set to stochastic gradient descent with momentum SGDM, the momentum is 0.9, the initial learning rate is 0.001, the L2 regularization coefficient is 0.004, the learning rate is decreased by 0.1, and the changing period of learning rate is 8. The minibatch is 16, the maximum number of iterations is 25, the verification period is once every 100 sheets, and the Dropout probability is 50%.

C. Comparison with Different Number of layers

Three experiments were carried out to study the effect of convolution layer number on the performance of CNN. In the experiment, the optimal number of convolution layers used in CNN is identified by changing the number of layers from 4 to 7. The classification accuracy of the test set and the training time implemented by CNN in each scene is evaluated. The experimental results show that when the number of convolution layers is 5, and the accuracy of its implementation is 95.51%, which is better than that of other layers. When the number of convolution layers is bigger than 5, we find that the classification accuracy is reduced but the training time is increased. Therefore, we chose to use five-layer convolution operation. The details are shown in Table I.

TABLE I. RESULTS OF DIFFERENT NUMBER OF LAYERS

Number of layers	Accuracy	Training time(s)
4	93.56%	2782
5	95.51%	2859
6	92.97%	3074
7	94.05%	3299

D. Comparison with Different Dataset

We used the accuracy index to test the convolutional neural network in the Columbia image splicing detection evaluation dataset(DVMM), with different data volumes and pre-processing conditions. A total of four experimental environments were tested. The accuracy of the experimental results is as Table II shows. It can be seen from the comparison experiments that a large number of training samples are very necessary for deep learning, which can greatly improve the performance of the classification, and the high-pass filter can significantly improve the classification effect.

TABLE II. RESULTS OF DIFFERENT EXPERIMENTAL ENVIRONMENT

Experimental Environment	Dataset	Accuracy
1	DVMM	62.20%
2	Expansion of DVMM	75.65%
3	DVMM with high-pass filter	77.02%
4	Expansion of DVMM with high-pass filter	95.51%

The iterative curve in the fourth experimental environment is shown in Figure 4. It can be seen that as the number of iterations increases, the detection accuracy increases gradually, but when the number of iterations reaches the second, the classification effect tends to be stable.

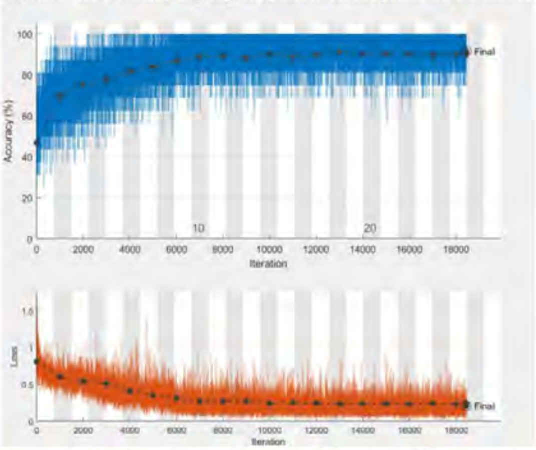


Figure 4. The Iterative Curve in the Fourth Experimental Environment.

E. Comparison with Different Algorithms

The accuracy of this algorithm is compared with the accuracy of several traditional image splicing detection algorithms. These methods use the same Colombia splicing detection evaluation database. The comparison results are shown in Table III. Comparing the data in the table III, it can be found that the performance of the CNN-based image splicing detection algorithm used in this paper is better than the existing algorithms listed. For DVMM dataset with gray images, our scheme achieves a performance gain of at least 1.96% over the methods in reference [17] and [18]. In this sense, our method is more preferable to gray images. It shows that deep learning technology combines feature extraction and classifier training, which can significantly improve classification accuracy and learning efficiency compared with traditional shallow learning methods (such as SVM).

TABLE III. RESULTS OF REFERENCES ON DVMM

Methods	Accuracy
[15]	78.80%
[16]	86.36%
[17]	93.55%
[18]	93.36%
Proposed	95.51%

IV. CONCLUSION

This paper proposes a method of image splicing blind forensics based on convolutional neural networks. This method adds a high-pass filter to the image pre-processing in the traditional CNN network, and builds a CNN network in the deep learning tool provided by Matlab 2019a to evaluate it on the Columbia image splicing detection evaluation dataset. The experiment results show that the image splicing detection method based on deep neural network can automatically learn how to detect the image splicing and tampering traces, without considering feature extraction and classification design, which has obvious advantages over traditional detection methods. The next step is to design the effective network architecture for localizing the tampered regions in the splicing image.

ACKNOWLEDGMENT

This material is based upon work supported by Program for Young Innovative Research Team in Shandong University of Political Science and Law; Key Laboratory of Evidence-Identifying in Universities of Shandong (SDUPSL); Projects of Shandong Province Higher Educational Science and Technology Program under Grant No.J16LN19, J18KA357, J18KA383; Projects of Shandong University of Political Science and Law under Grant No. 2016Z03B, 2015Z03B.

REFERENCES

[1] Markos Zampoglou, Symeon Papadopoulos, Yiannis Kompatsiaris. Large-scale evaluation of splicing localization algorithms for web images.Multimedia Tools and Applications, 2017,pp.4801-4834.

[2] Khurshid Asghar, Zulfiqar Habiba and Muhammad Hussain. Copy-move and splicing image forgery detection and localization techniques: a review. Australian Journal of Forensic Sciences, 2017, pp.1-27.

[3] Chen J, Kang X, Liu Y, Z. Jane Wang. Median Filtering Forensics Based on Convolutional Neural Networks. Signal Processing Letters IEEE, 2015.

[4] Peng P Y, Rong R N, Yao Z. Recapture Image Forensics Based On Laplacian Convolutional Neural Networks// International Workshop on Digital-forensics and Watermaking, 2016.

[5] Baroffio L, Bondi L, Bestagini P, Tubaro S.. Camera identification with deep convolutional networks. arXiv:1603.01068v1,2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp.1097-1105, 2012.

[7] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv preprint arXiv:1312.4400, 2013.

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

[9] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Re□thinking atrous convolution for semantic image segmentation. CoRR, abs/1706.05587, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Wein□berger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261-2269. July, 2017.

[12] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. CoRR, abs/1707.07012, 2017.

[13] Peng Zhou, Xintong Han, Vlad I. Morariu, Larry S. Davis. Learning Rich Features for Image Manipulation Detection. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1053-1061.

[14] Yinlong Qiana, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. Proc. SPIE 9409, Media Watermarking, Security, and Forensics 2015.

[15] YUAN Quanqiao, SU Bo1, ZHAO Xudong, LI Shenghong. Image splicing detection based on high frequency wavelet Markov features. Journal of Computer Applications, 2014, pp.1477-1481.

[16] LIU Xiaoxia, LI Feng, XIONG Bing. Image splicing detection using Weber local descriptors. Computer Engineering and Applications, 2013, pp.140-143.

[17] Z. He, W. Lu, W. Sun, and J. Huang, Digital image splicing detection based on Markov features in DCT and DWT domain, Pattern Recognit., 2012, vol. 45, no. 12, pp. 4292-4299, .

[18] X. Zhao, S. Wang, S. Li and J. Li, Passive Image-Splicing Detection by a 2-D Noncausal Markov Model, IEEE Transactions on Circuits and Systems for Video Technology, 2015, vol. 25, no. 2, pp. 185-199.

Research Progress in Blind Forensics of Digital Image Smooth Filtering

Jian Wu(✉), XinHua Tang, Kai Feng, and TongFeng Yang

School of Cyberspace Security, Shandong University of Political Science and Law,
Jinan 250014, China
jinanwujian@163.com

Abstract. Digital image blind forensics technology is a hot research direction in the field of information security. It can realize the authenticity and integrity verification of digital images without embedding the authentication information in the image in advance. In recent years, it has been widely researched and quickly developed. In this paper, a post-processing method-smoothing filter commonly used in digital image processing is summarized. The traditional forensic methods and representative work proposed in the early stage are summarized and discussed. Combined with deep learning technology, the main deep learning based smooth filtering is introduced in detail. Combined with deep learning technology, the main detection method based on smooth filtering is introduced in detail. Finally, we discussed the problems that still needed to be solved by blind learning technology based on deep learning and future research trends.

Keywords: Digital image forensics · Deep learning · Smooth filtering

1 Introduction

In today's highly developed digital technology, with the increasing popularity of image editing software, digital images can be easily forged and falsified, making it difficult or even impossible for human eyes to recognize. Especially when such images are used in scientific research, legal forensics, news reports, insurance claims, etc. will have irreversible negative impacts. Therefore, image forensics techniques for identifying the accuracy, integrity and originality of digital images have emerged as the times require, and have become a concern in the field of computer forensics [1]. In the existing digital image forensics technology, image blind forensics belongs to a passive authentication technology, which does not need to embed authentication information in the image in advance, so it has higher authentication and higher application value than active authentication technologies such as digital watermarking.

The blind forensic analysis methods currently involved mainly include the following three categories: First, the detection method based on imaging device consistency [2]. This method is similar to the image source forensics method and is based on the imaging consistency feature of digital cameras to identify whether the image is from different imaging devices, the current research methods include CFA interpolation detection, CCD

X. Sun et al. (Eds.): ICAIS 2020, CCIS 1252, pp. 359–370, 2020.
https://doi.org/10.1007/978-981-15-8083-3_32

mode noise detection, color difference detection, camera response function detection, etc. Second, based on the natural image statistical characteristics detection method [3]. The tamper image and the natural image are regarded as two classification problems, and the high-order statistical characteristics of the image are extracted as features, and the training classifier completes the detection. Third, the detection method for the post-splicing processing operation is used for the tampering after the image splicing, such as JPEG compression [4], resampling [5], filtering [9] and other operations to erase the traces left by tampering to the image, this method determines the tampering area by analyzing the characteristics of a tampering operation.

Since the smooth filtering operation is a very common post-processing method, it can be used to hide the tampering target edge. Therefore, many scholars have realized the forensic analysis of tampering images by detecting the smooth filtering operation traces, which are proposed to include SPAM [9], MFF [12] and MFR [14] and other series of results. In recent years, the rise of deep learning has injected new vitality into the forensic method. Many scholars have improved the deep learning network structure according to the smoothing filter tampering operation characteristics. Combining deep learning technology with image tamper detection has achieved a lot of creative results [25, 31–40]. This paper will introduce the development of traditional detection methods and deep learning techniques in the field of image smoothing blind forensics in recent years. The situation is summarized, and its technical characteristics, existing problems, and future research trends are discussed.

2 Traditional Detection Method

The smooth filtering operation is mainly divided into two categories: one is linear filtering, such as average filtering, Gaussian filtering, etc. The second type is nonlinear filtering, such as median filtering. Smooth filtering is a common post-processing technique for image denoising and smoothing, and is often used by counterfeiters to destroy the statistical properties and tampering artifacts that occur during tampering. Median filtering can even reduce the reliability of forensic tools, such as eliminating linear correlation between adjacent pixels to hide evidence of resampling [6], or to remove blockiness statistics introduced by JPEG compression to achieve the anti-forensics for JPEG compression detection [7], therefore, the existing research work mainly focuses on the detection and analysis of median filtering, including the detection method based on streaking characteristics, the detection method based on adjacent pixel correlation, and the residual-based on median filtering detection method and detection method based on frequency domain residual.

2.1 Detection Method Based on Streaking Characteristics

As early as 1987, Bovik and other scholars [8] found that the median filtered image not only has good edge retention characteristics, but also usually contains many constant image blocks. This feature is called the streaking artifact characteristic of median filtered image. In 2010, Kirchner et al. proposed the median filtering detection method based on the first-order difference image histogram ratio based on the characteristics of streaking artifact [9], which performed well on uncompressed images. For JPEG compressed images, the scholars used the subtractive pixel adjacency model (SPAM) [10] in the field of image steganalysis to detect the median filtering operation. The SPAM feature is to model the first-order difference values of the horizontal, vertical and diagonal directions of the image into *n*-order Markov chains, and combine their transition probability matrices into a set of feature vectors, when the JPEG quality factor is greater than 70, the detection algorithm performs better, but it decreases rapidly as the quality factor decreases. And because the SPAM feature dimension is very high, when the number of pixels of the image to be tested decreases, the detection performance will also be significantly deteriorated. Also based on the streaking property, Cao et al. [11] calculated the first-order differential zero-value probability *f* of the image texture region as the median-filtered statistical fingerprint feature, and designed another median filter detection method, which can also preliminarily distinguish median filter from other operations such as image scaling bilinear scaling (BS), Gaussian filter (GF) and mean filter (AF). However, the detection performance of JPEG compressed images has not changed significantly.

2.2 Detection Method Based on Adjacent Pixel Correlation

The median filtering detection method based on streaking characteristics does not perform well for image post-processing such as JPEG compression. The main reason is that SPAM and other features after various filtering operations are often masked by JPEG compression effects, so other filtering traces need to be considered. From 2011 to 2012, a series of detection methods based on adjacent pixel correlations were produced. In 2011, Yuan [12] observed that there are some common pixels in the overlapping filter window, and the median filtering of these overlapping windows introduces special features (local dependent effects) into the image. The author defines 44 features based on sequential statistics and gray values contained in the filter window, named median filtering forensics features (MFF) and based on MFF combined with SVM to achieve median filtering detection (recorded as MFF method). The test results show that the algorithm can effectively distinguish other smoothing filters such as median filtering and mean filtering. In terms of JPEG compression and low resolution images, although the detection performance is weakened with the reduction of JPEG compression factor and image resolution, however, the proposed algorithm is superior to the median filtering detection method based on streaking characteristics, and the localization of the local median filtering region in uncompressed images is realized for the first time. In 2012, Chen et al. [13] found that the median filtering and other filtering operations differed in the first-order differential cumulative distribution function of the image. At the same time, the correlation between local image differences was also studied, and the global directions were extracted for these two directions. The global probability feature (GPF) and the local correlation feature (LCF) form a 56-dimensional global and local feature GLF. Experimental results show that the feature vector can be well distinguished from the median value after training with SVM. The filtered image and the original image are less affected by JPEG compression and image resolution.

2.3 Detection Method Based on Median Filtering Residual

In 2013, Kang et al. [14] defined the difference image obtained by filtering the median value and calculating the difference from the original image as the median filter residual (MFR). Based on the characteristics that the MFR is reduced after the image is again median filtered, the median value of the image to be detected at the input is calculated by MFR, which is modeled as a linear autoregressive (AR) model, using 10-dimensional autoregressive coefficients as a detection feature (denoted as AR method). Compared with SPAM, MFF and GLF methods, AR method is not easily affected by image content and JPEG compression, and the detection result of median filtering is better, but the method extracts too few features. Robustness needs to be improved. In 2016, the research team [15] further proposed a median filtering residual difference (MFRD) based median filtering forensics technique (denoted as MFRD method). First, the multi-directional MFRD is grouped according to the directivity and symmetry, then the autoregressive model is established and its model parameters and histogram features are extracted respectively. Finally, all the grouped features are combined into 48-dimensional median filter detection features, which are trained and tested in the hybrid image database composed of five image databases of UCID [27], BR [28], DID [29], NRCS [30] and BOWS2 [43] by SVM. Compared with the above SPAM, MFF, GLF and AR detection results, MFRD method reduces the interference from image content and JPEG compression block effect trace, greatly improves the detection accuracy for JPEG compressed image and low-resolution image, and can better distinguish median filter and image scaling, low-pass Gaussian filter and mean filter and other operations.

2.4 Detection Method Based on Frequency Domain Residual

Xu et al. [16] observed that after the second filtering, the frequency domain residuals of both the original image and the filtered image will show as band-pass signals, but the bandwidth and other parameters are different, and different filtering operations correspond to different bandpass filters of different bandwidth. Therefore, the image is filtered again by a frequency domain low-pass filter at the detection end to obtain the frequency domain residual and convert it to the normalized Radon domain. Finally, the Radon transform curve is fitted to an 8-order Fourier series. The 18 parameters of the Fourier series are used as the classification feature VF for filtering detection, and the filtering is performed by SVM. The experimental results show that the proposed algorithm has a good detection effect on Gaussian filtering, mean filtering and median filtering, and can judge the size of the filter, and has good robustness for JPEG compression, but the method does not test the low-resolution image.

2.5 Summary of Traditional Detection Methods

Table 1 lists representative research results of traditional detection methods, not only lists feature dimensions, classification algorithms, and image sets, but also specifically points out the detection performance and detection image resolution when jpeg compression. The reason is that JPEG is the most widely used image format in image transmission and storage applications. Image compression will cover up the traces of smooth filtering.

Therefore, the detection algorithm needs to consider the robustness of JPEG compression with a small quality factor. The performance of low-resolution image detection is based on small images. The detection period of the block is beneficial to the classification algorithm for the localization analysis of the tamper region. Therefore, some studies compare the low resolution image and JPEG compression to show the advanced nature of the algorithm. In general, the traditional smooth filtering detection the feature design and extraction of the method mainly rely on manual experience manual extraction, and combined with traditional machine learning techniques such as SVM for training. Because feature extraction and classifier training are performed separately, they cannot be optimized simultaneously. With the rise of deep learning technology, the traditional smooth filtering detection method is gradually transitioning to the detection method based on deep learning.

Table 1. Traditional detection methods.

Paper, Year	Type of forgery	Features (dimension)	Classifier	Dataset	Performance and limitations
[9] 2010	MF	SPAM(686)	SVM	6500 images	Performance is acceptable while JPEG qualities >70, but deteriorates with decreasing image resolution
[11] 2010	MF, GF AF, BS	*f*	–	UCID	Performance degrades significantly in JPEG compressed images
[12] 2011	MF, GF, AF	MFF(44)	SVM	BOWS2, BR, NRCS, UCID	Robust to the low image quality(i.e., low resolution and JPEG); Local median filtering area can be localized
[13] 2012	MF	GLF(56)	SVM	9000 images from BOWS2, NRCS, DID	Significant performance improvement in the case of low resolution and strong JPEG post-compression
[14] 2013	MF, GF, AF, Scaling	MFR(10)	SVM	6690 images From UCID, BR,BOWS2, DID, NRCS	Performance is acceptable while JPEG qualities as low as 30; The method can identify median filtering in small image block
[15] 2016	MF, GF, AF, Scaling	MFRD(48)	SVM	6690 images From UCID, BR, BOWS2, DID, NRCS	The proposed detector performs better than SPAM, MFF, GLF, AR methods, especially in the detection of median filtering under heavy JPEG compression

(*continued*)

Table 1. (*continued*)

Paper, Year	Type of forgery	Features (dimension)	Classifier	Dataset	Performance and limitations
[16] 2013	MF, GF, AF	VF(18)	SVM	BOWS2	The method can predict parameters of MF, GF and AF filters

3 Smooth Filtering Detection Method Based on Deep Learning

3.1 Introduction to Convolutional Neural Networks

Deep learning is an emerging direction in the field of machine learning. By simulating the human brain to automatically learn the abstract features of each level of data, it is more convenient and realistic to extract the essential features of data [17]. Since 2006, Hinton [18] proposed a multi-layer Restricted Boltzmann Machine (RBM) based on probability map model. Deep learning has become a leading tool in image processing and computer vision. In particular, Convolutional Neural Network (CNN) [19], Deep Belief Network (DBN) [20], Stacked Auto-Encoder (SAE) [21], long and short time memory deep models such as Long-Short Term Memory (LSTM) [22] and Generative Adversarial Network (GAN) [23] have produced a large number of breakthroughs in various fields [24]. Among them, CNN directly processes 2D images through weight sharing and convolution operations, which avoids the complex feature extraction and data reconstruction process in traditional pattern recognition algorithms, which is deeply concerned by researchers. Especially with the generation of large-scale image data and the rapid development of computer hardware performance, convolutional neural network and its improved method have achieved breakthrough results in image segmentation, image recognition and image classification. In the field of digital image smoothing and filtering forensics, CNN has gradually replaced SVM and become the mainstream feature extraction and classification tools.

3.2 Smooth Filtering Detection Based on Deep Learning

Chen et al. [25] proposed the CNN-based median filtering forensics method for the first time, and designed a network structure consisting of one preprocessing layer, five convolutional layers and three fully connected layers. The pre-processing layer filters the input image median value and calculates the difference (MFR) from the original image as the input of the subsequent convolution layer, the principle is that MFR can suppress the original content pair classification such as image edge and texture. The effect, and the MFR of the image after filtering, is significantly reduced relative to the MFR of the untampered image. The network selects the ReLU activation function and the maximum pooling operation after the convolutional layer, and sets the dropout mechanism at the full connection layer to prevent the network from overfitting. Training and testing by image set extracted in BOSSbase 1.01 [26], UCID [27], BOSS RAW [28], DID [29] and NRCS [30], the detection rate of this method (defined as MFR method) is significantly improved compared with AR, MFF and GLF algorithms, but it is only suitable for detection of MF tampering.

In order to realize the preprocessing of training and test images, and extract the weak residual information left after image tampering, Bayar et al. [31] designed a new convolution structure with constraints on the first layer of traditional CNN, adding constraints on the convolution kernel property, as shown in

$$\begin{cases} w_k^{(1)}(0,0) = -1 \\ \sum_{l,m \neq 0} w_k^{(1)}(l,m) = 1 \end{cases} \tag{1}$$

Where w is the new convolution kernel and w(0,0) is the value of the center of the convolution kernel. The convolutional layer constrained by this constraint can learn the pixel and the relationship features around the pixel, not just the image content itself. By using the network structure established by the constrained convolution layer and the following two convolution layers, two maximum pooling layers and three full connection layers, we can realize the classification and forensics of four tampering modes, such as median filter, Gaussian blurring, additive white Gaussian noise and resampling. This shows that the constrained convolutional network structure can suppress the influence of image content and capture the operating characteristics, and realize the automatic learning of image pre-processing convolution kernel. In 2017, the team transformed the image tampering manipulation parameter estimation problem into a classification problem [32], and divided the parameter spaces of the four tampering methods, such as resizing, JPEG compression, median filtering, and Gaussian blurring, into disjoint subsets. Each subset of parameters is assigned to a different class. The scaling factor, the JPEG compression factor, the median filter kernel size, the Gaussian blur kernel size, and the fuzzy parameters are separately estimated based on a constrained convolution network similar to the literature [31].

Some scholars have not preprocessed the image, but use a more complex network structure and training method than the literature [31, 32] for detection. In 2018, Boroumand et al. [40] considered four basic image processing-low-pass filtering, high-pass filtering (sharpening), denoising and tone adjustment (including contrast and gamma adjustment), designed to accommodate resolution changes and JPEG compressed image manipulation detection method. This method increases the network depth, width (increased number of convolution kernels), and the amount of training data than the above studies. The network structure uses 8 convolutional layers and 3 fully connected layers. Considering that the average pooling operation will eliminate valuable noise, no pooling operation is used after the first two layers, and the average pooling operation is used after each of the last six convolution layers. The network training adopts the method of training the small image and the image of any size in stages and parts to ensure that the network can adapt to the detection of the size change of the detected image. One of the highlights is that the detection accuracy of the MF operation without the training set can also be more than 94%.

In order to achieve more tamper-type detection including smooth filtering, in 2019, Wu et al. [33] defined compression, blurring, morphology, contrast manipulation, additive noise, resampling and quantization as level0, and then these types of tampering are subdivided, such as subdividing blurring into Gaussian blur, box blurring, median blur, etc., increasing in turn, up to a total of 385 tamper types at level 5, including even in painting based on deep learning. The network structure of the design is divided into two sub-networks. The first sub-network mainly adopts the VGG network structure, which is responsible for tampering the trace feature extraction. In the first layer of image preprocessing, not only the tape proposed in the literature [2] is utilized, but also the SRM (spatial rich models) [34], which performs well in the field of steganalysis, is introduced. A total of 16 convolution kernels were designed for image preprocessing. The detection accuracy of the sub-network for 25 types of tampering operations in the level 1 can reach more than 85%, but even with a deeper and wider network structure, the recognition rate of 385 tamper types can only reach 51.8%. The second sub-network is responsible for locating the tampered area. The innovation is to use the tampering location problem as a local anomaly detection problem. The tamper-resistant regional features are taken as the main features, and the local features that differ from the main features are mapped to a specific tampering type. It can be seen from the research that in the image detection based on image space domain, by adding more types of filtering kernels to preprocess the image, it is possible to learn richer tampering features, and can realize the detection and analysis of more tampering modes including median filtering and Gaussian filtering.

Since the filtering operation is equivalent to removing certain frequency components in the frequency domain, the variation of the image in the frequency domain can also reflect the corresponding filtering operation. For example, in [35], a conversion layer is added before CNN, and the original image is converted into a frequency domain image by discrete Fourier transform and logarithmic transformation, and then input to the convolutional layer of CNN, and the image frequency is filtered by CNN. The possibility of domain feature learning realizes the detection of mean filtering, Gaussian filtering and median filtering under different parameter settings. The detection accuracy of low resolution image and JPEG compressed image is higher than that of MFF, AR, MFR and AAP mentioned in [35]. In [36], the CNN-based filtering image frequency domain feature learning method is also designed for the above three common filtering operations. The main difference is that the conversion layer adopts the filtering residual in frequency (FRF) of the input image block, and improves the recognition of the network model for the primary and secondary filtering operations based on the frequency domain features of the image.

In addition to traditional CNN, some scholars have customized the CNN, or introduced a new deep learning model (GAN) into the smoothing filter detection. In view of the weak nonlinear learning ability of traditional CNN networks, Tang et al. [37] introduced the multi-layer perceptron layer (mlpconv) contained in the NIN network [41] into the convolutional layer of the CNN network in 2018, and used MLP's nonlinear learning ability to learn the nonlinear characteristics of median filtering operation. Compared with the literature [25], the method solves the problem of median filtering detection of low resolution images and low quality compression factor JPEG images better. Higher detection accuracy can also be obtained without increasing the filter residual calculation. The study also performs the nearest neighbor interpolation and amplification on the original image, which expands the difference between the traced traces of the median filter between the falsified image and its original version, which is beneficial to improve the detection performance. In 2019, Shan et al. [38] further analyzed the influence of the block effect generated by JPEG compression on the median filtering feature, deblocking the JPEG image by the maximum a posteriori (MAP) framework to eliminate the block effect, and using the image deblocking method effectively suppressing the interference of JPEG compression, and then deblocking the image into the fused filtered residual (FFR) layer composed of MFR, average filtered residual (AFR), and Gaussian filtered residual (GFR) for the second step of preprocessing. The difference is blended to highlight the fingerprint left by the MF. Finally, the output of the FFR layer becomes the subsequent network input to further classify multiple features using a tailored parallel two-way CNN. Its detection performance is superior to the MFF, AR and MFR methods. In 2018, Jin et al. [39] applied the GAN to the median filter detection of low-resolution image and JPEG image for the first time. It does not analyze grayscale images like the above research methods, but directly to RGB images. The dark channel residual (DCR) set 3 × 3, 5 × 5, 7 × 7 three kinds of filter kernel analysis and detection, not only make full use of the rich information contained in the color image, but also improve the indicators of AUC, recall and F1 detection. Moreover, it is possible to directly perform detection and analysis on color images widely used in a real environment.

3.3 Summary of Deep Learning Smoothing Filter Detection Methods

Digital image smoothing detection is a category of pattern recognition. The successful application of deep learning in the field of pattern recognition has also led to the development of smooth filtering detection technology. Since 2015, a series of achievements including CNN, GAN, NIN, LSTM and other deep learning technologies have emerged. The main innovations of these studies are: 1) design or introduce MFR, SRM and other filters to achieve effective calculation of residual signals; 2) comprehensively use a variety of filters to extract features from the detected images, and use new network structure guarantees the diversity of the residual signal; 3) reduces the residual signal loss by reducing the pooling operation (such as canceling the pooling layer at the network front) or using the average pooling instead of the maximum pooling in the network training process. Compared with the traditional detection method, the training data is expanded in various ways, the feature learning is more autonomous, and the various detection performances are significantly improved, which also shows the great potential of deep learning technology in the field of digital image blind forensics.

4 Conclusion and Outlook

Based on the various traces left in the digital image by the tampering operation, the forensic personnel can discover digital image tampering without prior knowledge. Among various image blind evidence methods, image smoothing detection technology based on machine learning plays an important role. From the technical point of view, the image smoothing filter detection can be summarized as a two-category problem. The key is to extract the features of extracting specific tamper traces through machine learning. The difficulty and challenge lies in how to extract all kinds of features from the weak traces left by tamper operation, which take into account the distinguishing ability and generalization ability. At present, there is no perfect solution.

With the advent of the artificial intelligence era, deep learning technology has brought effective solutions for a series of digital image forensics problems such as smooth filtering detection. However, to date, deep learning based methods have not exhibited superior performance in the image recognition field. The main reasons are as follows: 1) Most of the current network structures adopt the network structure framework of image recognition. The detection method is susceptible to interference from image content, which is not conducive to extracting weak tampering operation features such as smooth filtering, and cannot guarantee end-to-end learning. Feature learning still requires some manual intervention; 2) Compared with image recognition mature datasets such as ImageNet, image tampering deep learning training sample sets are not enough to train more complex network structures, and the existing training data are mostly generated based on the simplified tampering mode, lacking the data set close to the real tampering scene. With the advent of image forgery based on deep learning [44], the detection of image tampering becomes more and more difficult. Although deep learning technology has broad prospects in the field of forensics, it still needs to carry out in-depth research in combination with new technologies and new theories, in order to solve the problems of end-to-end learning and feature learning as soon as possible.

Acknowledgment. This material is based upon work supported by 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; Key Laboratory of Evidence-Identifying in Universities of Shandong (SDUPSL); Projects of Shandong Province Higher Educational Science and Technology Program under Grant No. J16LN19, J18KA357, J18KA383.

References

1. Zhang, J., Li, Y., Niu, S., Cao, Z., Wang, X.: Improved fully convolutional network for digital image region forgery detection. Comput. Mater. Continua **60**(1), 287–303 (2019)
2. Gao, S., Xu, G., Hu, R.-M.: Camera model identification based on the characteristic of CFA and interpolation. In: Shi, Y.Q., Kim, H.-J., Perez-Gonzalez, F. (eds.) IWDW 2011. LNCS, vol. 7128, pp. 268–280. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32205-1_22
3. Ye, S., Sun, Q., Chang, E.C.: Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: Proceedings of 2007 IEEE International Conference on Multimedia and Expo, Beijing, China (2007)
4. Bianchi, T., Piva, A.: Detection of nonaligned double JPEG compression based on integer periodicity maps. IEEE Trans. Inf. Forensics Secur. **7**(2), 842–848 (2012)
5. Bappy, J.H., Mohammed, T.M., et al.: Detection and localization of image forgeries using resampling features and deep learning. In: CVPRW, pp. 1181–1189 (2017)
6. Kirchner, M., Bohme, R.: Hiding traces of resampling in digital images. IEEE Trans. Inf. Forensics Secur. **3**(4), 582–592 (2008)
7. Stamm, M.C., Liu, K.J.R.: Anti-forensics of digital image compression. IEEE Trans. Inf. Forensics Secur. **6**(3), 1050–1065 (2011)
8. Bovik, A.C.: Streaking in median filtered images. IEEE Trans. Acoust. Speech Signal Process. **35**(4), 493–503 (1987)

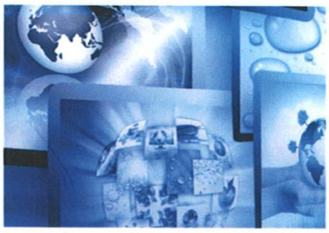
9. Kirchner, M., Fridrich, J.: On detection of median filtering in digital images. In: Proceedings of SPIE-Electronic Imaging 2010: Media Forensics and Security II, San Jose, CA, USA, 17–21 January 2010. International Society for Optics and Photonics, Bellingham, 7541101–7541112 (2010)
10. Pevny, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Trans. Inf. Forensics Secur. **5**(2), 215–224 (2010)
11. Cao, G., Zhao, Y., Ni, R., Yu, L., Tian, H.: Forensic detection of median filtering in digital images. In: Proceedings of 2010 IEEE International Conference on Multimedia and Expo, Singapore, Singapore, 19–23 July 2010, pp. 89–94. IEEE, Piscataway (2010)
12. Yuan, H.: Blind forensics of median filtering in digital images. IEEE Trans. Inf. Forensics Secur. **6**(4), 1335–1345 (2011)
13. Chen, C., Ni, J., Huang, R., Huang, J.: Blind median filtering detection using statistics in difference domain. In: Proceedings of Information Hiding, Berkeley, CA, USA, May 2012
14. Kang, X., Stamm, M.C., Peng, A., Ray Liu, K.J.: Robust median filtering forensics using an autoregressive model. IEEE Trans. Inf. Forensics Secur. **8**(9), 1456–1468 (2013)
15. Peng, A.J., Kang, X.G.: Median filtering forensics based on multi-directional difference of filtering residuals. Chin. J. Comput. **39**(3), 503–515 (2016)
16. Xu, F.-Y., Su, Y.-T.: Smoothing filtering detection for digital image forensics. J. Electron. Inf. Technol. **35**(10), 2287–2293 (2013)
17. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
18. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine learning, pp. 791–798. ACM (2007)
19. Sahiner, B., Chan, H.P., Petrick, N., et al.: Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. IEEE Trans. Med. Imaging **15**(5), 598–610 (1996)
20. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
21. Poultney, C., Chopra, S., Cun, Y.L.: Efficient learning of sparse representations with an energy-based model. In: Advances in Neural Information Processing Systems, pp. 1137–1144 (2007)
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
23. Fang, W., Zhang, F., Sheng, V.S., Ding, Y.: A method for improving CNN-based image recognition using DCGAN. Comput. Mater. Continua **57**(1), 167–178 (2018)
24. Oquab, M., Bottou, L., Laptev, I., et al.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1717–1724 (2014)
25. Chen, J., Kang, X., Liu, Y., Wang, Z.J.: Median filtering forensics based on convolutional neural networks. IEEE Signal Process. Lett. **22**(11), 1849–1853 (2015)
26. Bas, P., Filler, T., Pevny, T.: Break our steganographic system: the ins and outs of organizing BOSS. In: International Conference on Information Hiding, pp. 59–70 (2011)
27. Schaefer, G., Stich, M.: UCID - an uncompressed colour image database. In: Proceedings of SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose (2004)
28. http://boss.gipsa-lab.grenoble-inp.fr/BOSSRank/index.php?mode=VIEW&tmpl=materials
29. Gloe, T., Böhme, R.: Dresden image database (DID), the ‘Dresden image database’ for benchmarking digital image forensics. In: Proceedings of ACM Symposium Applied Computing, March 2010, vol. 2, pp. 1584–1590 (2010)
30. NRCS United States Department of Agriculture, Natural Resources Conservation Service Photo Gallery (2002). http://photogallery.nrcs.usda.gov

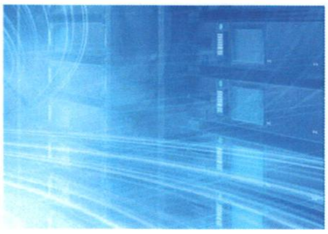
31. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10 (2016)
32. Bayar, B., Stamm, M.C.: A generic approach towards image manipulation parameter estimation using convolutional neural networks. In: The 5th ACM Workshop, pp. 147–157. ACM (2017)
33. Wu, Y., AbdAlmageed, W., Natarajan, P.: ManTra-Net: manipulation tracing network for detection and localization of image forgeries with anomalous features. In: CVPR, pp. 9543–9552 (2019)
34. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **7**(3), 868–882 (2012)
35. Liu, A., Zhao, Z., Zhang, C., Su, Y.: Smooth filtering identification based on convolutional neural networks. Multimedia Tools Appl. **78**(19), 26851–26865 (2016). https://doi.org/10.1007/s11042-016-4251-z
36. Yang, B., Zhang, T., Chen, X.-Y.: Local blur detection of digital images based on deep learning. J. Appl. Sci. Electron. Inf. Eng. **36**(2), 321–330 (2018)
37. Tang, H., Ni, R., Zhao, Y., Li, X.: Median filtering detection of small size image based on CNN. J. Vis. Commun. Image Represent. **51**, 162–168 (2018)
38. Shan, W., Yi, Y., Qiu, J., Yin, A., et al.: Robust median filtering forensics using image deblocking and filtered residual fusion. IEEE Access **7**, 17174–17183 (2019)
39. Jin, X., Jing, P., Su, Y.: AMFNet: an adversarial network for median filtering detection. IEEE Access **6**, 50459–50567 (2018)
40. Boroumand, M., Fridrich, J.: Deep learning for detecting processing history of images media. Watermarking Secur. Forensics. **9**, 213-1–213-9 (2018)
41. Lin, M., Chen, Q., Yan, S.: Network in network. In: Proceedings of ICLR (2014)
42. IEEE's signal processing society-camera model identification (2018). https://www.kaggle.com/c/sp-society-camera-model-identification
43. Bas, P., Furon, T.: Break our watermarking system. http://bows2.ec-lille.fr/2nd
44. Li, C., Jiang, Y., Cheslyar, M.: Embedding image through generated intermediate medium using deep convolutional generative adversarial network. Comput. Mater. Continua **56**(2), 313–324 (2018)

深度学习与深度合成

SHENDUXUEXIYUSHENDUHECHENG

吴　剑◎著





中国纺织出版社有限公司 | 国家一级出版社
全国百佳图书出版单位

图书在版编目（CIP）数据

深度学习与深度合成 / 吴剑著. --北京：中国纺织出版社有限公司，2020.10
ISBN 978-7-5180-8072-4

Ⅰ.①深… Ⅱ.①吴… Ⅲ.①机器学习 Ⅳ.①TP181

中国版本图书馆CIP数据核字（2020）第209826号

策划编辑：顾文卓　　特约编辑：王　霞
责任校对：　　　　　责任印制：储志伟

中国纺织出版社有限公司出版发行
地址：北京市朝阳区百子湾东里A407号楼　邮政编码：100124
销售电话：010—67004422　传真：010—87155801
http: //www.c-textilep. com
中国纺织出版社天猫旗舰店
官方微博http://weibo.com/2119887771
山东道克图文快印有限公司印刷　各地新华书店经销
2020年10月第1版第1次印刷
开本：787×1092　1/16　印张：11.75
字数：300千字　定价：68.00元

目　　录

A Deep Learning Approach to Detection of Warping Forgery in Images[⋆]

Tongfeng Yang, Jian Wu, Guorui Feng, Xu Chang and Lihua Liu

Shandong University of Political Science and Law, Jinan Shandong China
yangtf2014@126.com

Abstract. In recent years, image forensics has received full attention from researchers. A large number of algorithms for image smoothing, JPEG compression, copy-move, and shear tampering were published. However, there are still many image tampering algorithms that are not involved. In this paper, we publish a dataset of image warping, which contains more than 10000 images, and propose a novel convolutional neural network called DWF-CNN to identify warped images. In experiments, we compared the performance with 4 alternative networks. The proposed network with the preprocessing layer of the SRM layer and Bayar convolutional layer got the best result, which reached to the accuracy of 99.36%. The experiments also showed that the network with the regular convolutional layer performed even worse than a random guess. It illustrates the importance of the well-designed preprocessing layer in this research area again.

Keywords: Image forensics · Convolutional Neural Networks · Image Warping.

1 Introduction

With the popularity of digital cameras and mobile phones, more and more digital images and videos have been captured and published, and the number of pictures on the Internet has increased dramatically. Users can easily prettify, modify, and tamper the content of images with common image processing software, e.g. Photoshop. These software are designed to be easy to use and allows image tampering without expertise. Moreover, some jobs specializing in image processing, such as advertising design and graphic designers, have been derived. Related courses are offered in most computer-related majors too.

[⋆] Supported by Program for Young Innovative Research Team in Shandong University of Political Science and Law(Network Information Security and Forensics, Intelligent Information Processing);Big data and Artificial Intelligence Legal Research Collaborative Innovation Center of Shandong University of Political Science and Law; Key Laboratory of Evidence-Identifying in Universities of Shandong (SDUPSL); Projects of Shandong Province Higher Educational Science and Technology Program under Grant No.J16LN19, J18KA357, J18KA383; Projects of Shandong University of Political Science and Law under Grant No. 2016Z03B, 2015Z03B,2019KQ13Z.

Consequently, image forgery is becoming a rampant problem. Some approaches were proposed to protect the authenticity of image content by adding watermarks[4, 8], hashes and etc, which called active image forensic. Images need to be preprocessed before they are published. In contract, passive techniques that need no image preprocessing are more useful but more challenging.

In recent years, a variety of image tampering detection techniques have been proposed, each algorithm targets one or several tampering methods. Image smoothing[16, 16, 23], splicing[6, 21, 11] , JPEG compression[14, 3, 12, 18, 1], and copy-move tampering[5, 19, 15, 20, 22] are the most concerned tampering methods.

However, many tampering methods are not involved, such as image warping, image overlay, and content recognition in recent years. These algorithms are also used frequently in image processing.

Most proposed methods based on deep learning use a preprocessing layer to reduce the interference from image contents on image forensics. It is because, with an image processing software, the forgery images tend to close to the authentic ones not only visually but also statistically.

Fridrich[9] proposed 15 kernels for steganalysis. Zhou[25] selected 3 of those well-designed kernels as the first layer of their network for image tamping detection and achieved state-of-the-art performance compared to alternative methods. This layer often called SRM or SRMConv2d and is proved to be very effective in several follow-up works[15, 24].

Bayar[2] proposed a novel constrained convolutional layer with kernels having fixed weight -1 in their center. The network can handle multiple types of image forensics tasks.

Image warping is a prevalent method of image tampering. In photoshop, it called "liquefy". With this method, the face and shape of a person can be adjusted. This challenges the authenticity of images on the Internet.

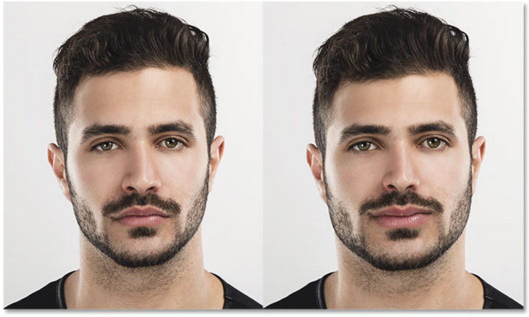


Fig. 1. Sample images of image warping. The left is authentic and right is warped.

In this paper, we propose a novel CNN network for image warping forgery. The network consists of two blocks: preprocessing block and regular CNN. We test the first block of 5 forms, and compared their performances and analyzed the results.

Section 2 describes the method of building the public image warping dataset. Section 3 gives the architecture of our proposed CNN network and section 4 demonstrates the result of our experiments. Section 5 gives the conclusion.

2 Image Warping Dataset

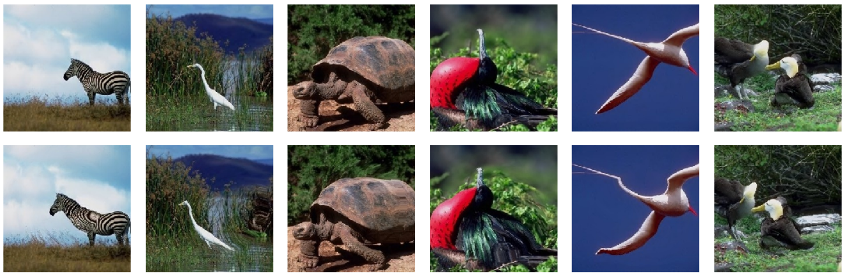


Fig. 2. Sample images of the dataset. Images in the top line are authentic and the bottom ones are warped.

As far as we know, there is no image warping dataset yet. Manually building a dataset of sufficient size is very labor-intensive. Fortunately, it is possible to construct the dataset using algorithms. The algorithm proposed by [10] is employed in this paper as it has many advantages: only pixels in the circular selection will be distorted, the father to the center of the circle, the smaller the distortion of pixels, and no changes on the edge of the circle, the image changes are uniform and natural.

We cut the images in the authentic set(AU) of CASIA v2.0[7] to 256×256 as negative samples and the warped images as positive ones.

The images bigger than 256×256 were cut to 256×256, and smaller ones were discarded. We randomly selected parameters of image warping and warped each image.

Equation 1 gives the warping algorithm with is called liquefy in Photoshop, while $\vec{x}$ is coordinates of the pixels in the warped image, and $\vec{u}$ is coordinates of the pixels in the source image. $\vec{c}$ and r donate the center coordinate and radius of the circle, and $\vec{m}$ donates the coordinate of user's mouse.

$$\vec{u} = \vec{x} - \left(\frac{r^2 - |\vec{x} - \vec{c}|^2}{(r^2 - |\vec{x} - \vec{c}|^2) + |\vec{m} - \vec{c}|^2} \right) (\vec{m} - \vec{c}) \tag{1}$$

The warped image on the Internet is visually different from the original image. We set r to 100. Even so, it can be seen from the figure2 that the visual difference between the distorted image and the original image is still tiny. If there is no original image for comparison, it is almost difficult to judge whether it has been warped.

We then randomly select the center of circle $\vec{c}$ and make sure the whole circle is in the image. We build vector $\vec{d}$ with the length of $r/2$ and random selected direction θ (Equation 2), then let $\vec{m} = \vec{d} + \vec{c}$.

$$\vec{d} = \{\frac{r\cos(\theta)}{2}, \frac{r\sin(\theta)}{2}\}, \theta \in [0, 2\pi] \tag{2}$$

Finally, we got 10066 images, while 5033 images are warped, and the others are original. All images have a size of 256x256. The dataset now is available on Github[1].

3 Proposed Convolutional Neural Network

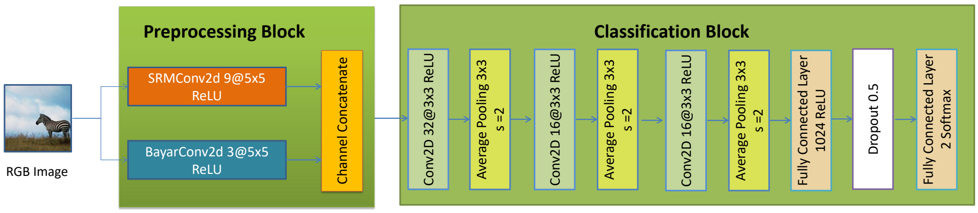


Fig. 3. The architecture of CNN network proposed in this paper. The network contains two block. The preprocessing block contains preprocessing layer to improve the classification performance and classification block is a well-designed CNN network for classification.

3.1 The Structure of Typical CNN Network

A typical CNN network consists of several convolutional layers and fully connected layers. The input of the first convolutional layer is the image from the dataset, while the input of other layer is the output of the previous layer which called feature map. Let us donate the feature map (output) of the layer n by F^n, the kernel and bias of convolution by W^n and B^n, convolution operator by $*$. We have:

$$F^n = \varphi(F^{n-1} * W^n + B^n) \tag{3}$$

where F^0 is the input image and F^{n-1} donate the output of the above layer where $n > 0$.

[1] https://github.com/taiji1985/ImageWarpingDataset

In the pooling layer, image(or feature map) is downsampled to reduce the number of parameters and the computation load. Max-pooling and average-pooling are the most commonly used.

The fully connected layer is a linear layer with an activation function. The number of its parameters is product of the size of the input vector and output vector. As it has so many parameters, it is very easy to over-fitting. A technique called dropout is adopted. It randomly drops out some dimensions of the input vector in each training step, which makes the rest dimensions more discriminate.

3.2 The Overview of Proposed CNN Network

Proposed DWF-CNN network is shown in Fig. 3 which consists of two blocks: the preprocessing block and classification block. We use SRMConv2d[25] layers with 9 filters and BayarConv2d[2] layers with 3 filters and concatenate their output in channel dimension as the preprocessing block.

In classification block, we use 3 convolutional layers with a kernel size of 3×3 and ReLU activation function. Each of those layers is followed by one average pooling layer with a size of 3×3 and a step of 2. Two fully connected layers are employed at the end of the network with 1024 and 2 neurons respectively. The first fully connected layer use ReLU activation function and the last one uses softmax for output.

3.3 The Preprocessing Block

We build the SRMConv2d layer with 9 filters by the 3 kernels given in Equation 4. The input of this layer is an image with 3 color channels: red, green and blue, so the shape of the filter is $(3, 5, 5)$.

$$K_1 = \frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, K_2 = \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}, K_3 = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Let $\vec{0}$ donate the 5×5 zero matrix, the 9 filters can be represented by Equation 5. In other words, we let the kernel K_i given above as the weights for one input channel, and the weights for the other two channels are set to zero matrix. It can separately process the three channels of the image to extract features with higher discrimination.

$$\begin{bmatrix} K_1 & \vec{0} & \vec{0} \\ \vec{0} & K_1 & \vec{0} \\ \vec{0} & \vec{0} & K_1 \\ K_2 & \vec{0} & \vec{0} \\ \vec{0} & K_2 & \vec{0} \\ \vec{0} & \vec{0} & K_2 \\ K_3 & \vec{0} & \vec{0} \\ \vec{0} & K_3 & \vec{0} \\ \vec{0} & \vec{0} & K_3 \end{bmatrix} \tag{5}$$

BayarConv2d is a constrained convolutional layer while the center value of the filters fixes to -1, and the sum of other values in one kernel fix to 1. The amount of all weights normalize to zero. We use 3 such filters in our network.

Let w_{ij} donates the weight of 1 filter of BayarConv2d layer, we have:

$$\begin{cases} \sum_i \sum_j w_{ij} = 0 \\ w_{0,0} = -1 \end{cases} \tag{6}$$

We merge the output of the SRMConv2d layer and BayarConv2d layer in the channel dimension and get 12 channels at all. We also tested other forms of preprocessing block(Fig. 4) in our experiments. The above form got better accuracy.

The Preprocessing Block (b) combines SRMConv2d, BayarConv2d, and regular convolutional layers and gets 32 channels for output. This structure tried to get a better result but failed. Block (c-e) contains a single type of preprocessing layer to test the performance of each one. Block (c) and (d) got results lower than the proposed effect.

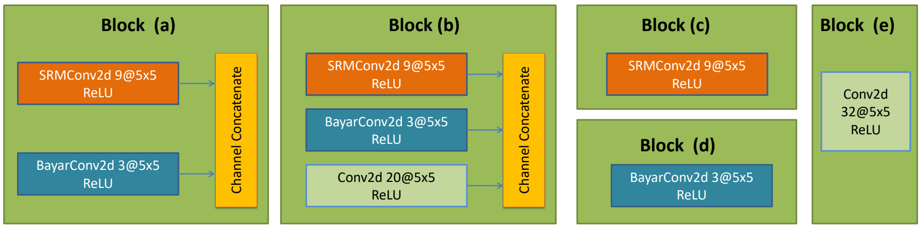


Fig. 4. Five forms of preprocessing block. (a) Block with the SRMConv2d and BayarConv2d layer. (b) Block with the SRMConv2d, BayarConv2d, and regular Conv2d layer. (c) Block with the SRMConv2d layer. (d) Block with BayarConv2d layer. (e) Block with the regular Conv2d layer.

3.4 The Classification Block

The first convolutional layer in classification block has 32 filters of 3×3, and the second and third convolutional layer has 16 filters with the same size. They all use the ReLU activation function. We use symmetric padding in each of those layers.

In VGG[17] network, the number of filters grows twice after each two-layer, while the number of our network decreases. We had tested the architecture like VGG, but it performed pool as it had too many parameters to converge.

Those convolutional layers are followed by three 3×3 average pooling layer instead of 2×2 pooling used in classic CNN network like Alexnet[13] or VGG[17]. The bigger size increases the receptive field.

We employ two fully connected layers and use a dropout layer between them with a drop rate of 0.5.

4 Experiments

We split our image warping dataset randomly into two parts by 9:1, the first part for training, and the second part for evaluation.

We tested five forms of preprocessing block demonstrated by Fig. 4 : (a) the proposed block with SRMConv and BayarConv2d,(b) block with SRMConv2d, BayarConv2d and normal Conv2d which has 32 channels at all. (c) block with only the SRMConv2d layer with 9 filters. (d) block with only BayarConv2d layer with 3 filters. (e) block with a normal Conv2d layer.

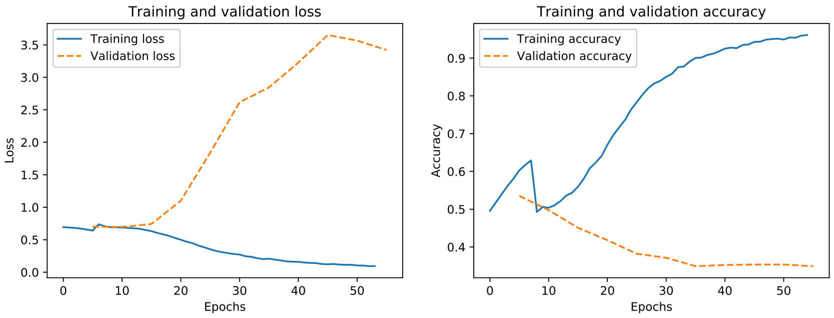
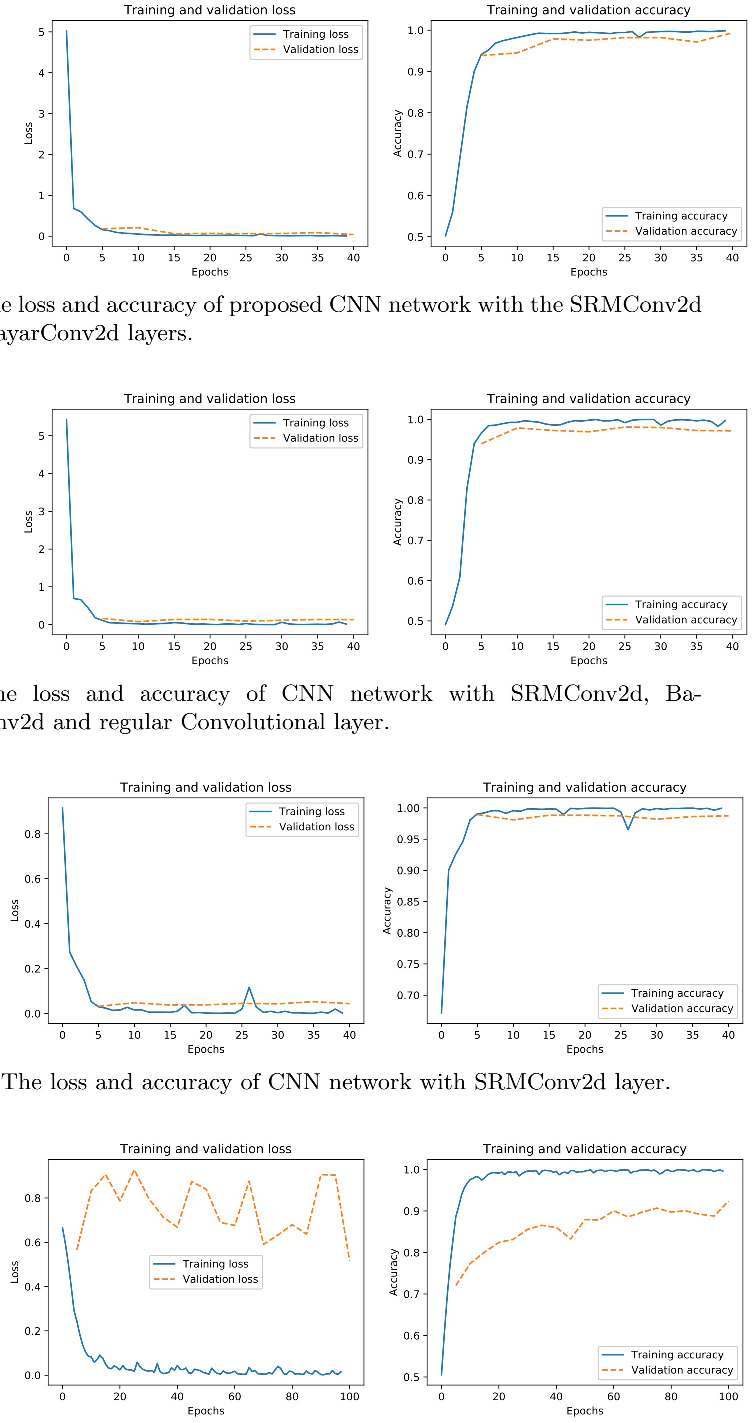


Fig. 5. The loss and accuracy of CNN network with regular convolutional layer in preprocessing block.

We used Google Colaboratory[2] in our experiments, which is free. It has 24G GDDR5 memory and a speed of 2.91 tflops for each runtime. The optimizer

[2] https://colab.research.google.com



(a) The loss and accuracy of proposed CNN network with the SRMConv2d and BayarConv2d layers.

(b) The loss and accuracy of CNN network with SRMConv2d, BayarConv2d and regular Convolutional layer.

(c) The loss and accuracy of CNN network with SRMConv2d layer.

(d) The loss and accuracy of CNN network with BayarConv2d layer.

Fig. 6. The loss and accuracies of CNN network with 4 forms of the preprocessing block.

Adam with learning rate 0.001 and loss function of sparse categorical cross-entropy were employed in our networks.

The result of our experiments is impressive. It shows that the CNN network with regular convolutional layer performed worse than random guess during continuous training, but our model achieved an accuracy of up to 99.36%, the best results of each model shown in Table 1.

Table 1. Classification accuracies of several CNN networks with different preprocessing block.

Form of Processing Block	Accuracy
(a) SRMConv2d and BayarConv2d	**99.36%**
(b) SRMConv2d,BayarConv2d and Conv2d	98.20%
(c) SRMConv2d	98.94%
(d) BayarConv2d	92.38%
(e) Conv2d	53.54%

Firstly, we tested the block with the regular convolutional layer(preprocessing block(e)) and showed the result in Fig.5 . The experiment showed that accuracy continuously declined to 34.92% while we trained 100 epochs. The validation accuracy shows that this model misclassified the images more than the correct classification and worse than the random guess dues to that the CNN network always ignores the deformation of images. CNNs are better at identifying the content of images, including deformed content. It is an advantage in content recognition, but a disadvantage in distortion detection.

We also tested the CNN network with the BayarConv2d layer with 3 filters, which got an accuracy of 92.38% shown in Fig.6 (d). The training accuracy reaches 97% after 13 epochs, and increase slowly after that, while the validation accuracy grows slowly and unhesitatingly.

The CNN network with the SRMConv2d layer got an accuracy of 98.94% quickly, after 10 epochs. The network trained faster than the others while the SRMConv2d layer is not trainable, and the total number of parameters in the network was smaller.

The results of the proposed network with preprocessing block(a) show in Fig.6 (a), which got an accuracy of 99.36%. Note that the preprocessing block of this network contains SRMConv2d layer and BayarConv2d layer but no regular convolutional layer.

In contrast, we used all of the layers included SRMConv2d, BayarConv2d, and regular convolutional layer in preprocessing block but got worse accuracy of 98.20% shown in Fig.6 (b). Combined with the above experiments, it can be concluded that the regular convolutional layer has a negative impact on accuracy.

5 Conclusion

Image warping is a common method of beautifying and forging in image processing. We proposed a dataset with more than 10000 images and proposed an effective identification method. We intensely studied the influence of the preprocessing layer on the classification results through experiments. The effect of traditional CNN with regular convolutional layers is deplorable because of the visual similarity between the distorted image and the original image, even lower than randomly guess. The well-designed preprocessing layer proposed can reduce the impact of content on the image and focus on image tampering forensics.

References

1. Barni, M., Bondi, L., Bonettini, N., Bestagini, P., Costanzo, A., Maggini, M., Tondi, B., Tubaro, S.: Aligned and non-aligned double JPEG detection using convolutional neural networks. CoRR **abs/1708.00930** (2017), http://arxiv.org/abs/1708.00930
2. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. IEEE Transactions on Information Forensics and Security **13**(11), 2691–2706 (Nov 2018). https://doi.org/10.1109/TIFS.2018.2825953
3. Bin Li, Shi, Y.Q., Jiwu Huang: Detecting doubly compressed jpeg images by using mode based first digit features. In: 2008 IEEE 10th Workshop on Multimedia Signal Processing. pp. 730–735 (Oct 2008). https://doi.org/10.1109/MMSP.2008.4665171
4. Chan, K.C., Moon, Y.S., Cheng, P.S.: Fast fingerprint verification using subregions of fingerprint images. IEEE Transactions on Circuits and Systems for Video Technology **14**(1), 95–101 (Jan 2004). https://doi.org/10.1109/TCSVT.2003.818358
5. Cozzolino, D., Poggi, G., Verdoliva, L.: Efficient dense-field copy–move forgery detection. IEEE Transactions on Information Forensics and Security **10**(11), 2284–2297 (Nov 2015). https://doi.org/10.1109/TIFS.2015.2455334
6. Cozzolino, D., Poggi, G., Verdoliva, L.: Splicebuster: A new blind image splicing detector. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 (Nov 2015). https://doi.org/10.1109/WIFS.2015.7368565
7. Dong, J., Wang, W., Tan, T.: Casia image tampering detection evaluation database. pp. 422–426 (07 2013). https://doi.org/10.1109/ChinaSIP.2013.6625374
8. Farid, H., Lyu, S.: Higher-order wavelet statistics and their application to digital forensics. In: 2003 Conference on Computer Vision and Pattern Recognition Workshop. vol. 8, pp. 94–94 (June 2003). https://doi.org/10.1109/CVPRW.2003.10093
9. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security **7**(3), 868–882 (June 2012). https://doi.org/10.1109/TIFS.2012.2190402
10. Gustafsson, A.: Interactive image warping. https://www.gson.org/thesis/warping-thesis.pdf (1993), [Online; accessed 19-July-2008]
11. Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting fake news: Image splice detection via learned self-consistency. CoRR **abs/1805.04096** (2018), http://arxiv.org/abs/1805.04096
12. Korus, P., Huang, J.: Multi-scale fusion for improved localization of malicious tampering in digital images. IEEE Transactions on Image Processing **25**(3), 1312–1326 (March 2016). https://doi.org/10.1109/TIP.2016.2518870

13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (May 2017). https://doi.org/10.1145/3065386, http://doi.acm.org/10.1145/3065386
14. Pevny, T., Fridrich, J.: Detection of double-compression in jpeg images for applications in steganography. IEEE Transactions on Information Forensics and Security **3**(2), 247–258 (June 2008). https://doi.org/10.1109/TIFS.2008.922456
15. Rao, Y., Ni, J.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 (Dec 2016). https://doi.org/10.1109/WIFS.2016.7823911
16. Rhee, K.H.: Median filtering detection using variation of neighboring line pairs for image forensics. Journal of Electronic Imaging **25**(5), 1 – 13 (2016). https://doi.org/10.1117/1.JEI.25.5.053039, https://doi.org/10.1117/1.JEI.25.5.053039
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556 pp. 1–7 (09 2014)
18. Taimori, A., Razzazi, F., Behrad, A., Ahmadi, A., Babaie-Zadeh, M.: Quantization-unaware double jpeg compression detection. Journal of Mathematical Imaging and Vision **54**(3), 269–286 (Mar 2016). https://doi.org/10.1007/s10851-015-0602-z, https://doi.org/10.1007/s10851-015-0602-z
19. Wen, B., Zhu, Y., Subramanian, R., Ng, T., Shen, X., Winkler, S.: Coverage — a novel database for copy-move forgery detection. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 161–165 (Sep 2016). https://doi.org/10.1109/ICIP.2016.7532339
20. Wu, Y., Abd-Almageed, W., Natarajan, P.: Image copy-move forgery detection via an end-to-end deep neural network. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1907–1915 (March 2018). https://doi.org/10.1109/WACV.2018.00211
21. Wu, Y., Abd-Almageed, W., Natarajan, P.: Deep matching and validation network - an end-to-end solution to constrained image splicing localization and detection. CoRR **abs/1705.09765** (2017), http://arxiv.org/abs/1705.09765
22. Wu, Y., Abd-Almageed, W., Natarajan, P.: BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VI, pp. 170–186 (09 2018)
23. Yang, J., Ren, H., Zhu, G., Huang, J., Shi, Y.Q.: Detecting median filtering via two-dimensional ar models of multiple filtered residuals. Multimedia Tools Appl. **77**(7), 7931–7953 (Apr 2018). https://doi.org/10.1007/s11042-017-4691-0, https://doi.org/10.1007/s11042-017-4691-0
24. Yue Wu, W.A., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgerieswith anomalous features (2019)
25. Zhou, P., Han, X., Morariu, V., Davis, L.: Learning rich features for image manipulation detection. pp. 1053–1061 (06 2018). https://doi.org/10.1109/CVPR.2018.00116

Image Tampering Detection for Splicing based on Rich Feature and Convolution Neural Network

Tongfeng Yang
Shandong University of Political Science and Law
No.63 East Jiefang Road
Jinan, China
addressyangtf2014@126.com

Jian Wu
Shandong University of Political Science and Law
No.63 East Jiefang Road
Jinan, China
jinanwujian@163.com

Zhifeng Fang
Shandong University of Political Science and Law
No.63 East Jiefang Road
Jinan, China
fangzf@sdu.edu.cn

ABSTRACT

Image splicing is a widely used image tampering method. The detection of these methods has also been widely concerned by researchers. We propose a detection method based on rich feature and convolution neural network. In order to avoid the interference of image content on classification, we use a high pass filter to preprocess the image, then a well-designed novel convolutional neural network is used to classify the images. In the experiment, the network performed better than traditional methods on the Columbia image splicing detection evaluation dataset.

CCS Concepts

• **Computing methodologies→Neural networks; Matching; Image processing;**

Keywords

Image forensics; Convolutional Neural Networks; Image.

1. INTRODUCTION

With the rapid increase of image data on the Internet and the wide application of image editing software, the authenticity of the images people see on the Internet has been greatly challenged. For a variety of purposes, people will pre-processing their images before release. After being transmitted through the Internet, tampered digital image sometimes bring serious mass incidents, especially in the academic, political and military fields.

Image splicing is a simple process that crops and pastes regions from the same or separate sources. It can make objects that do not belong to the image appear in the image. The forged image often leaves traces at the splicing place. Therefore, forgers often use image blur, median filtering, double JPEG compression and other means to mask the splicing traces. This makes it difficult for people to distinguish the forged images with the naked eye.

The existing work focuses on the following three classes: the methods based on camera noise pattern, the methods based on color filter array (CFA) and the methods based on post-processing detection.

The first class of splicing localization algorithms is based on camera noise model. Because of the different electrical characteristics of cameras, there will be noise in the photos taken. With the help of the detection of noise model, we can judge the difference between the original image and the patch of the spliced image[13, 5, 3, 4, 11].

The second class of splicing localization algorithms is based on CFA interpolation patterns. CFA is used to convert the output of the camera's sensor to the pixel value in RGB format, and CFAs for different cameras are also different. The CFA of spliced image is different because of the different source of image blocks, so image splicing can be detected by CFA[6,7].

The third class of splicing localization algorithms is based on detection of post-processing operations, including image smoothing detection[16,18] and double JPEG compression detection[14, 2, 10, 17, 1].

At the same time, convolution neural network has made a landmark breakthrough in the field of image recognition. The traditional convolution neural network is designed for recognition. The lower layers are used to extract low-level features such as edge, angle, line, etc., the upper layers are used to extract object representation features, and the full connection layers are used for regression or classification.

The detection target of image tampering detection is to determine whether there is tampering. The tampered image is similar to the original image in content, so the general convolution neural network perform poor, sometimes the accuracy is lower than the random guess (< 0.5).

Fridrich[8] gives 15 convolution kernels for hidden information detection. Zhou[20] selected three cores for image preprocessing, and achieved good results in forgery image forensics. In this paper, we only use the most effective kernel of those kernels.

In this paper, we carefully design a deep network based on rich features and convolution neural network. Firstly, a selected rich feature kernel is used to suppress image content and enhance the forgery trace of the image. Then, we use the neural network including 5 convolution layers and 2 full connection layers to classify the image. Batch normalization, Dropout technology and ReLU activation function are used to enhance the generalization ability of the network.

HPCCT & BDAI 2020, July 3 – 6, 2020, Qingdao, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7560-3/20/07 $15.00
https://doi.org/10.1145/3409501.3409530

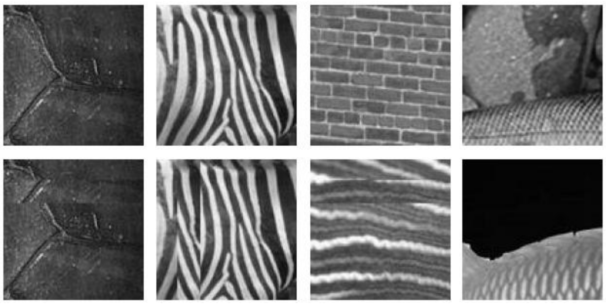


Figure 1. Sample images of the columbia image splicing detection evaluation dataset. The images in first line are authentic, while the second line are spliced.

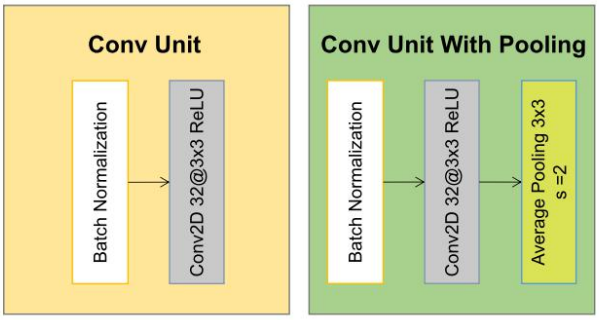


Figure 2. Convolutional unit with and without pooling layer.

2. PROPOSED NEURAL NETWORK

2.1 Overview

The whole network demonstrated by Figure 3 is composed of preprocessing block and classification block. SRM layer is used in preprocessing block to reduce the impact of image content on classification. Five convolution units are used in the classification block. The first two units do not contain the pooling layer, and the last two units contain the average pooling layer.

The last part of the classification block contains two full connection layers, the first contains 1024 neurons, and the second contains one neuron as the output layer. Dropout technology is used between the two connection layers with a parameter of 0.5.

Sigmoid activation function is used in the output layer, and ReLU activation function is used in the other full connection layer and convolution layers.

2.2 Dataset Enhancements

Deep learning algorithm has a high demand for the size of data set. When the model is trained with a small data set, the optimization function is used to fall into the local minimum, resulting in poor generalization ability.

So, we expanded the dataset. Each image is horizontally mirrored, and then rotated by 90 , 180 and 270 degrees respectively. The original image and the processed image are all put into the enhanced dataset.

2.3 The Preprocessing Block

Convolutional neural network is designed to solve the problem of content recognition. The low-level convolution layer can induce the bottom image features such as edge, angle and edge, and the high-level features depict a part of the object, and the top level is used for classification. However, in the image forensics, the forgery image and the original image can contain the same object, so using the general convolution neural network can not detect the forgery problem very well.

Therefore, when tampering forensics, it is necessary to preprocess the image to reduce the impact of image content on classification as much as possible.

SRM[20] has proved to be an effective preprocessing layer. SRM core is a well-designed matrix as shown in Equation 1, whose center is a negative number, and the signs of adjacent weights are opposite to highlight the high-frequency characteristics of images which can enhance the splicing trace in the image. Figure 1 shows sample images from columbia image splicing detection evaluation dataset, and Figure 4 shows an simple of the original image and the filtered image from the dataset.

$$K = \frac{1}{12}\begin{bmatrix} -1 & +2 & -2 & +2 & -1 \\ +2 & -6 & +8 & -6 & +2 \\ -2 & +8 & -12 & +8 & -2 \\ +2 & -6 & +8 & -6 & +2 \\ -1 & +2 & -2 & +2 & -1 \end{bmatrix} \qquad (1)$$

2.4 Classification Block

We design two convolution units, the first unit is composed of batch normalization and convolution layer, while the second is composed of batch normalization, convolution layer and upper sampling layer.

The whole classification network consists of five convolution units and two fully connected layers. Since the size of the image in the dataset is 128×128 , in order to avoid losing too much information in the convolution process, the first two convolution blocks in the classification block do not contain the upper sampling layer, and the last three are followed by the upper sampling layer. The technical details are further described below.

2.5 Convolution Units

The convolution unit consists of a batch standardization layer, a convolution layer and an optional upper sampling layer demonstrated by Figure 2.

In the conv2d layer, we use the "valid" padding function instead of the "same" function. Compared with the "same" method, "valid" does not bring padding regions. The characteristics of these regions are similar to those brought by splicing, which will interfere with classification.

The ReLU function is used in the convolution layer, which is widely used in many convolution neural networks. It simulates the characteristics of biological neurons, and has fast operation speed and can solve the gradient vanishing problem well.

In the deep learning network, with the increase of network depth, the training becomes more and more difficult and the convergence becomes slower. The main reason is that the deep learning is based on the back propagation algorithm, which depends on the gradient calculation. The ReLU activation function and batch normalization try to solve this problem from different angles.

In addition to using the ReLU activation function, the network we give also uses the batch normanlization technology which is performed before each convolution layer.

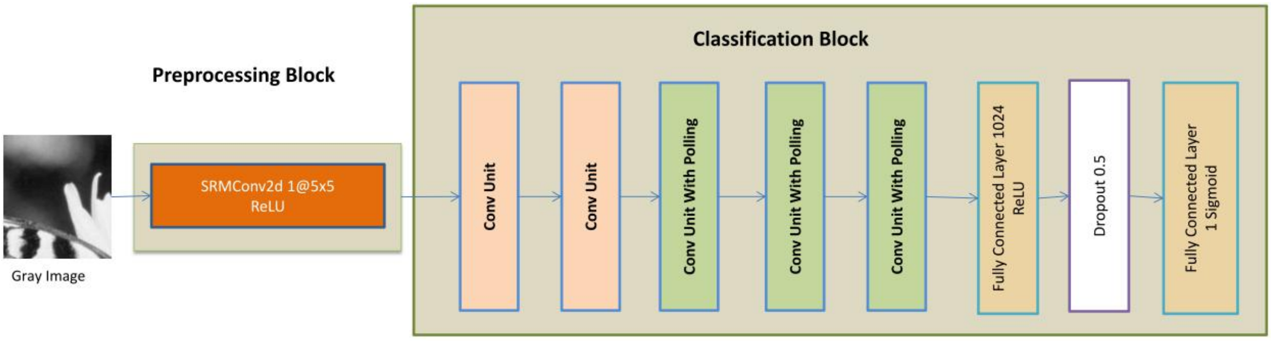


Figure 3. The overall network architecture proposed.

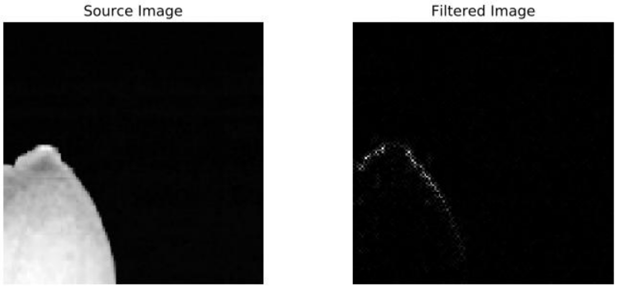


Figure 4: Simple of source and filtered image from Columbia dataset.

Batch normalization is mainly used to solve the internal covariate shift problem. After the image is preprocessed, the pixel value tends to be very small and the image looks darker visually. After batch normalization, the mean of pixel values is converted to 0, and the variance is converted to 1. This kind of data entering the convolution layer will significantly reduce the value of the second-order normal form of convolution kernel weights.

The size of the upper sampling layer is 3×3, which has a better receptive field than that of 2×2. Step size is 2, and the receptive fields of the neighboring pixels after sampling are overlapped. Compared with the non overlapping receptive field, the loss of information is less. According to our previous research results, the upper sampling type uses mean pooling instead of maximum pooling.

2.6 Fully Connected Layers

Two fully connected layers are used in this paper. The first layer follows the last convolution unit and contains 1024 neurons and the second layer contains only one neuron as output.

There is a dropout layer with a parameter of 0.5 between the two full connection layers to avoid overfitting and poor generalization due to too many parameters. The first full connection layer uses the ReLU activation function, while the second uses the sigmoid function to adapt to the classification labels.

The size of the upper sampling layer is 3×3, which has a better receptive field than that of 2×2. Step size is 2, and the receptive fields of the neighboring pixels after sampling are overlapped. Compared with the non overlapping receptive field, the loss of information is less. According to our previous research results, the upper sampling type uses mean pooling instead of maximum pooling.

3. EXPERIMENTS

3.1 Dataset

The experiment used the image splicing evaluation dataset from Columbia's DVMM laboratory. The image in this dataset is an uncompressed grayscale image of 128×128 pixels , with a total of 1845 images. The numbers of the authentic and spliced images are approximately the same while 933 are authentic and 912 are spliced. The spliced image is obtained by splicing the authentic image along the edge of the object or transversely (longitudinal), and specific can be divided into five categories: homogenic smooth, homogenic textured, textured-smooth, textured-textured and smooth-smooth.

The deep neural network has a large number of parameters, thus small dataset is not sufficient to fully optimize the parameters. So the enhancement of the dataset is a regular operation, which generally includes: mirroring, rotation, adding noise, adjusting tone, etc. But for image tampering forensics, adding noise will interfere with classification, so only mirroring and rotation operation is used. We select an image from the dataset and present the mirrored and rotated images in Figure 5.

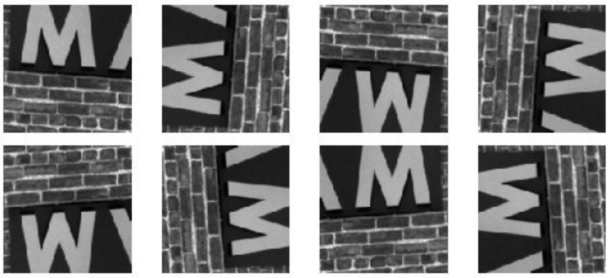


Figure 5. Original and expanded images.

3.2 Experiment Setting

The experiments were completed on the Google Colaboratory platform. Each execution unit contains two 2.30 GHZ Intelli (R) xeon CPUs, 12.72 GB memory and 8 remote TPUs. Python 3.0 ,Google Tensorflow 1.15.0 and the 2.2.4-tf version of Keras embedded in Tensorflow were used.

The premise of using TPU is that the number of samples in the training set is an integer multiple of 8, so that the data can be divided into 8 TPUs equally. Therefore, we divided the expanded dataset into approximately 9:1 training set and test set.

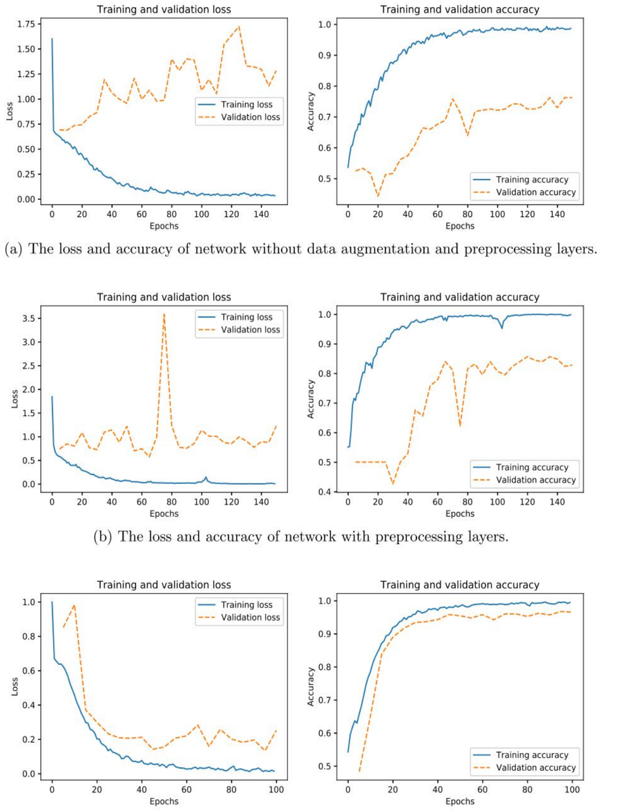
Adam with learning rate of 0.001 was used as the optimization function ,while binary cross entropy as loss function. We set the maximum number of iterations to 150 and take the best evaluation results obtained during the iteration as the reported accuracy.

We validated the accuracy on test set every **5** iterations, and each iteration took about 10 seconds in experiments.

3.3 Network Structure Evaluation

In the experiment, we evaluated the effect of pre-processing layer and data expansion on the results. We first evaluated the network without the data augmentation and pre-processing layers (abbreviated as **NN** below), then evaluated the network that only included the pre-processing layer(abbreviated as **NP**), and finally evaluated the structure that included both. The accuracy and loss function curves of the proposed networks are shown in Figure 6.

Experiments show that both the pre-processing layer and the data expansion play a positive role in improving the classification accuracy. The results are shown in Table 1.



(a) The loss and accuracy of network without data augmentation and preprocessing layers.

(b) The loss and accuracy of network with preprocessing layers.

(c) The loss and accuracy of proposed network with data augmentation and preprocessing layers.

Figure 6. Effect evaluation of pre-processing layer and data enhancement.

3.4 Comparison with Different Algorithms

The algorithm presented in this paper is compared with several published algorithms in accuracy. It can be seen that compared with the existing algorithm, the proposed algorithm has a significant improvement, as shown in Table 2.

It is worth noting that in the process of network structure optimization and parameter tuning, we find that the results of multiple execution of the same network structure are not the same. This may be caused by the following reasons: firstly, as the total number of samples is small, when the training set obtained by random segmentation has better representativeness, the test results will be better. Secondly, because of the large number of parameters, the neural network often falls into local minimum value while the data is not enough, thus the random initial value of the parameters having a certain impact on the results. The best results of multiple execution are given in this paper and even the worst(accuracy: 0.9428) is still higher than the comparison algorithm.

Table 1. Accuracy of networks

Network	Accuracy
NN	0.7622
NP	0.8565
Proposed Network	**0.9673**

Table 2. Results of related work on DVMM

Network	**Accuracy**
Wavelet Markov[15]	0.7880
Weber[12]	0.8636
Markov in DCT and DWT[9]	0.9355
2-D Noncausal Markov[19]	0.9336
Proposed Network	**0.9673**

4. CONCLUSIONS

In this paper, a convolutional neural network is designed to detect the image mosaic. The interference of image content to the detection process is reduced by using the carefully selected filter core, and the performance of classification is further improved by expanding the data set. Experiments show that the preprocessing layer and data set expansion do have a positive effect. Compared with the related work, the results also prove that the network presented in this paper has better classification performance.

5. ACKNOWLEDGMENTS

Supported by Program for Young Innovative Research Team in Shandong University of Political Science and Law(Network Information Security and Forensics, Intelligent Information Processing);Big data and Artificial Intelligence Legal Re search Collaborative Innovation Center of Shandong Uni versity of Political Science and Law; Key Laboratory of Evidence-Identifying in Universities of Shandong (SDUPSL); Projects of Shandong Province Higher Educational Science and Technology Program under Grant No.J16LN19, J18KA357, J18KA383; Projects of Shandong University of Political Sci ence and Law under Grant No. 2016Z03B, 2015Z03B,2019KQ13Z.

6. REFERENCES

[1] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro. Aligned and non-aligned double JPEG detection using convolutional neural networks. CoRR, abs/1708.00930, 2017.

[2] Bin Li, Y. Q. Shi, and Jiwu Huang. Detecting doubly compressed jpeg images by using mode based first digit features.

In 2008 IEEE 10th Workshop on Multimedia Signal Processing, pages 730﹣735, Oct 2008.

[3] M. Chen, J. Fridrich, M. Goljan, and J. Luk´ as. Determining image origin and integrity using sensor noise. IEEE Transactions on information forensics and security, 3(1):74﹣90, 2008.

[4] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva. A bayesian-mrf approach for prnu-based image forgery detection. IEEE Transactions on Information Forensics and Security, 9(4):554﹣567, 2014.

[5] D. Cozzolino, G. Poggi, and L. Verdoliva. Splicebuster: A new blind image splicing detector. In 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pages 1﹣6, Nov 2015.

[6] A. E. Dirik and N. Memon. Image tamper detection based on demosaicing artifacts. In 2009 16th IEEE International Conference on Image Processing (ICIP), pages 1497﹣1500. IEEE, 2009.

[7] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. Image forgery localization via fine-grained analysis of cfa artifacts. IEEE Transactions on Information Forensics and Security, 7(5):1566﹣1577, 2012.

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

[9] Z. He, W. Lu, W. Sun, and J. Huang. Digital image splicing detection based on markov features in dct and dwt domain. Pattern Recognition, 45(12):4292—4299.

[10] P. Korus and J. Huang. Multi-scale fusion for improved localization of malicious tampering in digital images. IEEE Transactions on Image Processing, 25(3):1312﹣1326, March 2016.

[11] C.-T. Li and Y. Li. Color-decoupled photo response non-uniformity for digital image forensics. IEEE Transactions on Circuits and Systems for Video Technology, 22(2):260﹣271, 2011.

[12] X. B. LIU Xiaoxia, LI Feng. Image splicing detection using weber local descriptors. Computer Engineering and Applications, 49(12):140, 2013.

[13] B. Mahdian and S. Saic. Using noise inconsistencies for blind image forensics. Image and Vision Computing, 27(10):1497﹣1503, 2009.

[14] T. Pevny and J. Fridrich. Detection of double-compression in jpeg images for applications in steganography. IEEE Transactions on Information Forensics and Security, 3(2):247﹣258, June 2008.

[15] X. Z. Quanqiao Y, Bo S U. Image splicing detection based on high frequency wavelet markov features. Journal of Computer Applications, 34(5):1477, 2014.

[16] K. H. Rhee. Median filtering detection using variation of neighboring line pairs for image forensics. Journal of Electronic Imaging, 25(5):1﹣13, 2016.

[17] A. Taimori, F. Razzazi, A. Behrad, A. Ahmadi, and M. Babaie-Zadeh. Quantization-unaware double jpeg compression detection. Journal of Mathematical Imaging and Vision, 54(3):269﹣286, Mar 2016.

[18] J. Yang, H. Ren, G. Zhu, J. Huang, and Y.-Q. Shi. Detecting median filtering via two-dimensional ar models of multiple filtered residuals. Multimedia Tools Appl., 77(7):7931﹣7953, Apr. 2018.

[19] X. Zhao, S. Wang, S. Li, and J. Li. Passive image-splicing detection by a 2-d noncausal markov model. IEEE Transactions on Circuits and Systems for Video Technology, 25(2):185﹣199, Feb 2015.

[20] P. Zhou, X. Han, V. Morariu, and L. Davis. Learning rich features for image manipulation detection. Pages 1053﹣1061, 06 2018.

Metadata of the chapter that will be visualized in SpringerLink

Book Title	Artificial Intelligence and Security	
Series Title		
Chapter Title	A Deep Learning Approach to Detection of Warping Forgery in Images	
Copyright Year	2020	
Copyright HolderName	Springer Nature Switzerland AG	

Corresponding Author	Family Name	**Yang**
	Particle	
	Given Name	**Tongfeng**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Shandong University of Political Science and Law
	Address	Jinan, Shandong, China
	Email	yangtf2014@126.com
Author	Family Name	**Wu**
	Particle	
	Given Name	**Jian**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Shandong University of Political Science and Law
	Address	Jinan, Shandong, China
	Email	
Author	Family Name	**Feng**
	Particle	
	Given Name	**Guorui**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Shandong University of Political Science and Law
	Address	Jinan, Shandong, China
	Email	
Author	Family Name	**Chang**
	Particle	
	Given Name	**Xu**
	Prefix	
	Suffix	

	Role	
	Division	
	Organization	Shandong University of Political Science and Law
	Address	Jinan, Shandong, China
	Email	
Author	Family Name	**Liu**
	Particle	
	Given Name	**Lihua**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Shandong University of Political Science and Law
	Address	Jinan, Shandong, China
	Email	

Abstract

In recent years, image forensics has received full attention from researchers. A large number of algorithms for image smoothing, JPEG compression, copy-move, and shear tampering were published. However, there are still many image tampering algorithms that are not involved. In this paper, we publish a dataset of image warping, which contains more than 10000 images, and propose a novel convolutional neural network called **DWF-CNN** to identify warped images. In experiments, we compared the performance with 4 alternative networks. The proposed network with the preprocessing layer of the SRM layer and Bayar convolutional layer got the best result, which reached to the accuracy of 99.36%. The experiments also showed that the network with the regular convolutional layer performed even worse than a random guess. It illustrates the importance of the well-designed preprocessing layer in this research area again.

Keywords

Image forensics - Convolutional neural networks - Image warping

A Deep Learning Approach to Detection of Warping Forgery in Images

Tongfeng Yang(✉), Jian Wu, Guorui Feng, Xu Chang, and Lihua Liu

Shandong University of Political Science and Law, Jinan, Shandong, China
yangtf2014@126.com

AQ1

Abstract. In recent years, image forensics has received full attention from researchers. A large number of algorithms for image smoothing, JPEG compression, copy-move, and shear tampering were published. However, there are still many image tampering algorithms that are not involved. In this paper, we publish a dataset of image warping, which contains more than 10000 images, and propose a novel convolutional neural network called **DWF-CNN** to identify warped images. In experiments, we compared the performance with 4 alternative networks. The proposed network with the preprocessing layer of the SRM layer and Bayar convolutional layer got the best result, which reached to the accuracy of 99.36%. The experiments also showed that the network with the regular convolutional layer performed even worse than a random guess. It illustrates the importance of the well-designed preprocessing layer in this research area again.

Keywords: Image forensics · Convolutional neural networks · Image warping

1 Introduction

With the popularity of digital cameras and mobile phones, more and more digital images and videos have been captured and published, and the number of pictures on the Internet has increased dramatically. Users can easily prettify, modify, and tamper the content of images with common image processing software, e.g. Photoshop. These software are designed to be easy to use and allows image tampering without expertise. Moreover, some jobs specializing in image processing, such as advertising design and graphic designers, have been derived. Related courses are offered in most computer-related majors too.

Consequently, image forgery is becoming a rampant problem. Some approaches were proposed to protect the authenticity of image content by adding watermarks [4, 8], hashes and etc., which called active image forensic. Images need to be preprocessed before they

Supported by Program for Young Innovative Research Team in Shandong University of Political Science and Law(Network Information Security and Forensics, Intelligent Information Processing);Big data and Artificial Intelligence Legal Research Collaborative Innovation Center of Shandong University of Political Science and Law; Key Laboratory of Evidence-Identifying in Universities of Shandong (SDUPSL); Projects of Shandong Province Higher Educational Science and Technology Program under Grant No.J16LN19, J18KA357, J18KA383; Projects of Shandong University of Political Science and Law under Grant No. 2016Z03B, 2015Z03B,2019KQ13Z.

X. Sun et al. (Eds.): ICAIS 2020, LNCS 12240, pp. 1–10, 2020.
https://doi.org/10.1007/978-3-030-57881-7_10

are published. In contract, passive techniques that need no image preprocessing are more useful but more challenging.

In recent years, a variety of image tampering detection techniques have been proposed, each algorithm targets one or several tampering methods. Image smoothing [16, 23], splicing [6, 11, 21], JPEG compression [1, 3, 12, 14, 18], and copy-move tampering [5, 15, 19, 20, 22, 27] are the most concerned tampering methods.

However, many tampering methods are not involved, such as image warping, image overlay, and content recognition in recent years. These algorithms are also used frequently in image processing.

Most proposed methods based on deep learning use a preprocessing layer to reduce the interference from image contents on image forensics. It is because, with an image processing software, the forgery images tend to close to the authentic ones not only visually but also statistically.

Fridrich [9] proposed 15 kernels for steganalysis. Zhou [25] selected 3 of those well-designed kernels as the first layer of their network for image tamping detection and achieved state-of-the-art performance compared to alternative methods. This layer often called SRM or SRMConv2d and is proved to be very effective in several follow-up works [15, 24].

Bayar [2] proposed a novel constrained convolutional layer with kernels having fixed weight -1 in their center. The network can handle multiple types of image forensics tasks.

Image warping is a prevalent method of image tampering. In photoshop, it called "liquefy". With this method, the face and shape of a person can be adjusted. This challenges the authenticity of images on the Internet (Fig. 1). AQ2

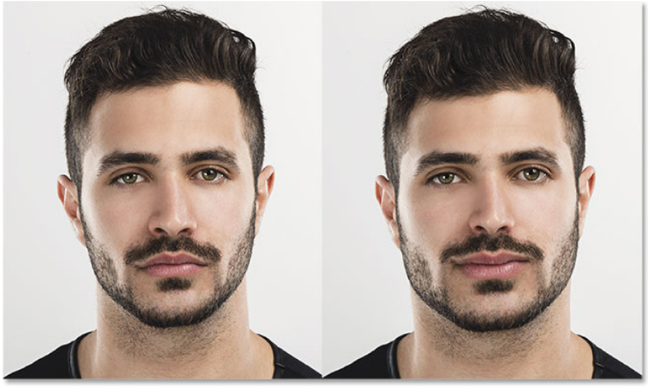


Fig. 1. Sample images of image warping. The left is authentic and right is warped.

In this paper, we propose a novel CNN network for image warping forgery. The network consists of two blocks: preprocessing block and regular CNN. We test the first block of 5 forms, and compared their performances and analyzed the results.

Section 2 describes the method of building the public image warping dataset. Section 3 gives the architecture of our proposed CNN network and Sect. 4 demonstrates the result of our experiments. Section 5 gives the conclusion.

2 Image Warping Dataset

As far as we know, there is no image warping dataset yet. Manually building a dataset of sufficient size is very labor-intensive. Fortunately, it is possible to construct the dataset using algorithms. The algorithm proposed by [10] is employed in this paper as it has many advantages: only pixels in the circular selection will be distorted, the father to the center of the circle, the smaller the distortion of pixels, and no changes on the edge of the circle, the image changes are uniform and natural.

We cut the images in the authentic set(AU) of CASIA v2.0 [7] to 256 × 256 as negative samples and the warped images as positive ones.

The images bigger than 256 × 256 were cut to 256 × 256, and smaller ones were discarded. We randomly selected parameters of image warping and warped each image.

Equation 1 gives the warping algorithm with is called liquefy in Photoshop, while x is coordinates of the pixels in the warped image, and u is coordinates of the pixels in the source image. c and r donate the center coordinate and radius of the circle, and m donates the coordinate of user's mouse.

$$u = x - \left(\frac{r^2 - |x-c|^2}{r^2 - |x-c|^2 + |m-c|^2}\right)(m-c) \tag{1}$$

The warped image on the Internet is visually different from the original image. We set r to 100. Even so, it can be seen from the Fig. 2 that the visual difference between the distorted image and the original image is still tiny. If there is no original image for comparison, it is almost difficult to judge whether it has been warped.

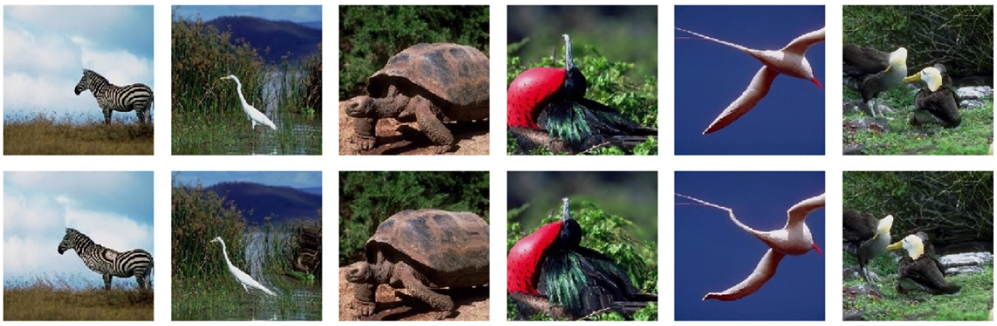


Fig. 2. Sample images of the dataset. Images in the top line are authentic and the bottom ones are warped.

We then randomly select the center of circle c and make sure the whole circle is in the image. We build vector d with the length of r/2 and random selected direction θ (Eq. 2), then let m $= d + c$.

$$d = \left\{\frac{r\cos(\theta)}{2}, \frac{r\sin(\theta)}{2}\right\}, \theta \in [0, 2\pi] \tag{2}$$

Finally, we got 10066 images, while 5033 images are warped, and the others are original. All images have a size of 256 x 256. The dataset now is available on Github[1].

[1] https://github.com/taiji1985/ImageWarpingDataset.

3 Proposed Convolutional Neural Network

3.1 The Structure of Typical CNN Network

A typical CNN [26, 28, 29] network consists of several convolutional layers and fully connected layers. The input of the first convolutional layer is the image from the dataset, while the input of other layer is the output of the previous layer which called feature map. Let us donate the feature map (output) of the layer n by F^{n}, the kernel and bias of convolution by W^n and B^n, convolution operator by $*$, activation function by ϕ. We have:

$$F^n = \phi(F^{n-1} * W^n + B^n) \tag{3}$$

where F^0 is the input image and $F^{\mathrm{n\ -\ 1}}$ donate the output of the above layer while $n > 0$.

In the pooling layer, image (or feature map) is downsampled to reduce the number of parameters and the computation load. Max-pooling and average-pooling are the most commonly used.

The fully connected layer is a linear layer with an activation function. The number of its parameters is product of the size of the input vector and output vector. As it has so many parameters, it is very easy to over-fitting. A technique called dropout is adopted. It randomly drops out some dimensions of the input vector in each training step, which makes the rest dimensions more discriminate.

3.2 The Overview of Proposed CNN Network

Proposed **DWF-CNN** network is shown in Fig. 3 which consists of two blocks: the preprocessing block and classification block. We use SRMConv2d [25] layers with 9 filters and BayarConv2d [2] layers with 3 filters and concatenate their output in channel dimension as the preprocessing block.

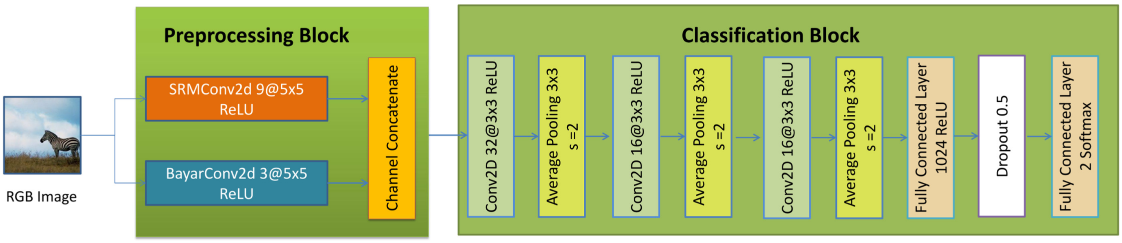


Fig. 3. The architecture of CNN network proposed in this paper. The network contains two block. The preprocessing block contains preprocessing layer to improve the classification performance and classification block is a well-designed CNN network for classification.

In classification block, we use 3 convolutional layers with a kernel size of 3 × 3 and ReLU activation function. Each of those layers is followed by one average pooling layer with a size of 3 × 3 and a step of 2. Two fully connected layers are employed at the end of the network with 1024 and 2 neurons respectively. The first fully connected layer use ReLU activation function and the last one uses softmax for output.

3.3 The Preprocessing Block

We build the SRMConv2d layer with 9 filters by the 3 kernels given in Eq. 4. The input of this layer is an image with 3 color channels: red, green and blue, so the shape of the filter is (3, 5, 5).

$$K_1 = \frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, K_2 = \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}, K_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Let **0** donate the 5 × 5 zero matrix, the 9 filters can be represented by Eq. 5. In other words, we let the kernel K_i given above as the weights for one input channel, and the weights for the other two channels are set to zero matrix. It can separately process the three channels of the image to extract features with higher discrimination.

$$\begin{bmatrix} K_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & K_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & K_1 \\ K_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & K_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & K_2 \\ K_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & K_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & K_3 \end{bmatrix} \tag{5}$$

BayarConv2d is a constrained convolutional layer while the center value of the filters fixes to −1, and the sum of other values in one kernel fix to 1. The amount of all weights normalize to zero. We use 3 such filters in our network.

Let w_{ij} donates the weight of 1 filter of BayarConv2d layer, we have:

$$\begin{cases} \sum\limits_{\mathrm{i}} \sum\limits_{j} w_{ij} = 0 \\ w_{0,0} = -1 \end{cases} \tag{6}$$

We merge the output of the SRMConv2d layer and BayarConv2d layer in the channel dimension and get 12 channels at all. We also tested other forms of preprocessing block (Fig. 4) in our experiments. The above form got better accuracy.

The Preprocessing Block (b) combines SRMConv2d, BayarConv2d, and regular convolutional layers and gets 32 channels for output. This structure tried to get a better result but failed. Block (c–e) contains a single type of preprocessing layer to test the performance of each one. Block (c) and (d) got results lower than the proposed effect.

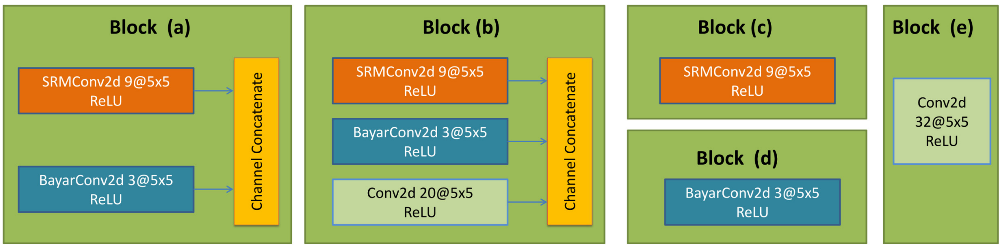


Fig. 4. The Preprocessing Block (b) combines SRMConv2d, BayarConv2d, and regular convolutional layers and gets 32 channels for output. This structure tried to get a better result but failed. Block (c-e) contains a single type of preprocessing layer to test the performance of each one. Block (c) and (d) got results lower than the proposed effect.

3.4 The Classification Block

The first convolutional layer in classification block has 32 filters of 3×3, and the second and third convolutional layer has 16 filters with the same size. They all use the ReLU activation function. We use symmetric padding in each of those layers.

In VGG [17] network, the number of filters grows twice after each two-layer, while the number of our network decreases. We had tested the architecture like VGG, but it performed pool as it had too many parameters to converge.

Those convolutional layers are followed by three 3×3 average pooling layer instead of 2×2 pooling used in classic CNN network like Alexnet [13] or VGG [17]. The bigger size increases the receptive field.

We employ two fully connected layers and use a dropout layer between them with a drop rate of 0.5.

4 Experiments

We split our image warping dataset randomly into two parts by 9:1, the first part for training, and the second part for evaluation.

We tested five forms of preprocessing block demonstrated by Fig. 4: (a) the proposed block with SRMConv and BayarConv2d, (b) block with SRMConv2d, BayarConv2d and normal Conv2d which has 32 channels at all. (c) block with only the SRMConv2d layer with 9 filters. (d) block with only BayarConv2d layer with 3 filters. (e) block with a normal Conv2d layer.

We used Google Colaboratory[2] in our experiments, which is free. It has 24G GDDR5 memory and a speed of 2.91 tflops for each runtime. The optimizer Adam with learning rate 0.001 and loss function of sparse categorical cross-entropy were employed in our networks.

The result of our experiments is impressive. It shows that the CNN network with regular convolutional layer performed worse than random guess during continuous training, but our model achieved an accuracy of up to 99.36%, the best results of each model shown in Table 1.

[2] https://colab.research.google.com.

Table 1. Classification accuracies of several CNN networks with different pre-processing block.

Form of Processing Block	Accuracy
(a) SRMConv2d and BayarConv2d	**99.36%**
(b) SRMConv2d,BayarConv2d and Conv2d	98.20%
(c) SRMConv2d	98.94%
(d) BayarConv2d	92.38%
(e) Conv2d	53.54%

Firstly, we tested the block with the regular convolutional layer (preprocessing block(e)) and showed the result in Fig. 5. The experiment showed that accuracy continuously declined to 34.92% while we trained 100 epochs. The validation accuracy shows that this model misclassified the images more than the correct classification and worse than the random guess dues to that the CNN network always ignores the deformation of images. CNNs are better at identifying the content of images, including deformed content. It is an advantage in content recognition, but a disadvantage in distortion detection.

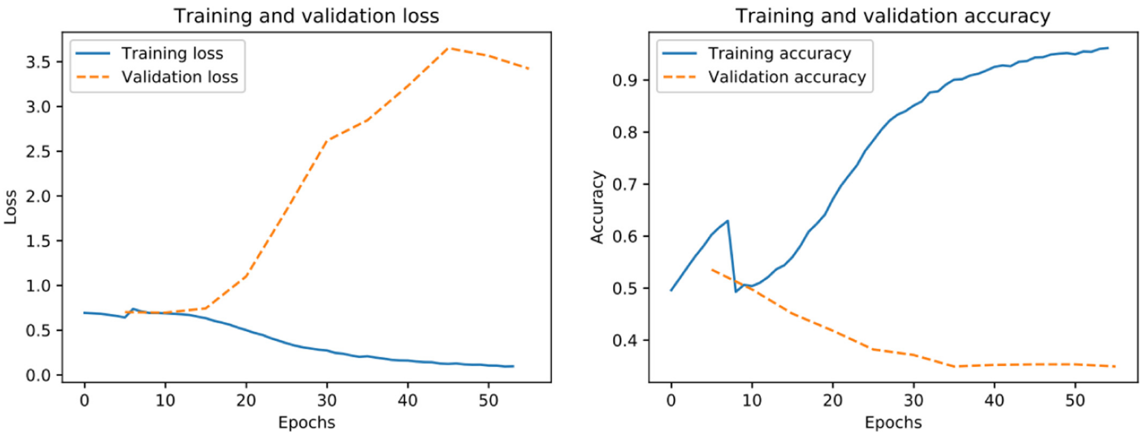
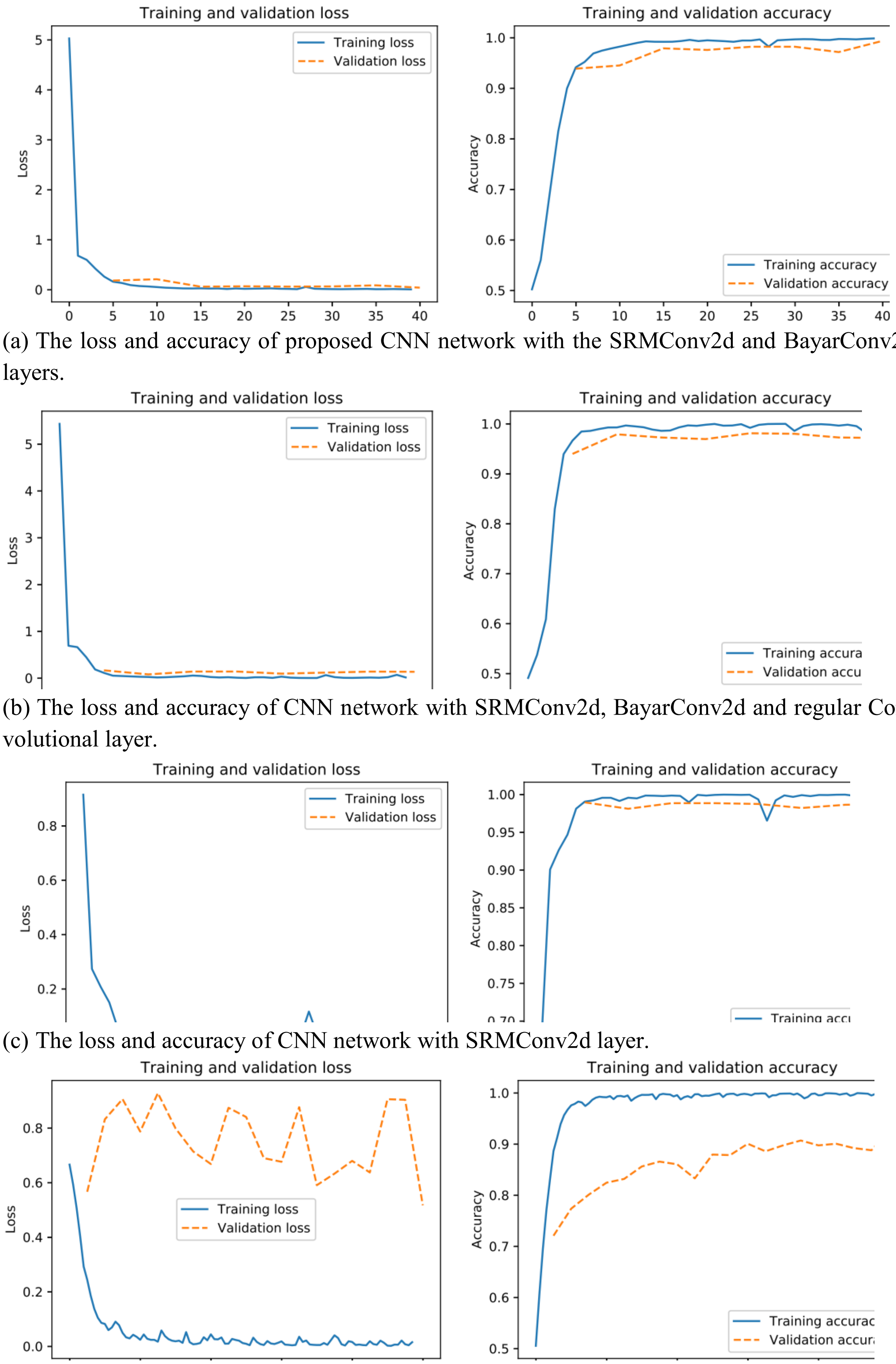


Fig. 5. The loss and accuracy of CNN network with regular convolutional layer in preprocessing block.

We also tested the CNN network with the BayarConv2d layer with 3 filters, which got an accuracy of 92.38% shown in Fig. 5 (d). The training accuracy reaches 97% after 13 epochs, and increase slowly after that, while the validation accuracy grows slowly and unhesitatingly.

The CNN network with the SRMConv2d layer got an accuracy of 98.94% quickly, after 10 epochs. The network trained faster than the others while the SRMConv2d layer is not trainable, and the total number of parameters in the network was smaller.

The results of the proposed network with preprocessing block(a) show in Fig. 6 (a), which got an accuracy of 99.36%. Note that the preprocessing block of this network contains SRMConv2d layer and BayarConv2d layer but no regular convolutional layer.



(a) The loss and accuracy of proposed CNN network with the SRMConv2d and BayarConv2d layers.

(b) The loss and accuracy of CNN network with SRMConv2d, BayarConv2d and regular Convolutional layer.

(c) The loss and accuracy of CNN network with SRMConv2d layer.

(d) The loss and accuracy of CNN network with BayarConv2d layer.

Fig. 6. The loss and accuracies of CNN network with 4 forms of the preprocessing block.

In contrast, we used all of the layers included SRMConv2d, BayarConv2d, and regular convolutional layer in preprocessing block but got worse accuracy of 98.20% shown in Fig. 6 (b). Combined with the above experiments, it can be concluded that the regular convolutional layer has a negative impact on accuracy.

5 Conclusion

Image warping is a common method of beautifying and forging in image processing. We proposed a dataset with more than 10000 images and proposed an effective identification method. We intensely studied the influence of the preprocessing layer on the classification results through experiments. The effect of traditional CNN with regular convolutional layers is deplorable because of the visual similarity between the distorted image and the original image, even lower than randomly guess. The well-designed preprocessing layer proposed can reduce the impact of content on the image and focus on image tampering forensics.

References

1. Barni, M., et al.: Aligned and non-aligned double JPEG detection using convolutional neural networks. CoRR abs/1708.00930 (2017) http://arxiv.org/abs/1708.00930
2. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. IEEE Trans. Inf. Forensics. Secur. **13**(11), 2691–2706 (2018)
3. Bin, L., Shi, Y.Q., Jiwu, H.: Detecting doubly compressed JPEG images by using mode based first digit features. In: 2008 IEEE 10th Workshop on Multimedia Signal Processing. pp. 730–735 (2008). https://doi.org/10.1109/MMSP.2008.4665171
4. Chan, K.C., Moon, Y.S., Cheng, P.S.: Fast fingerprint verification using subregions of fingerprint images. IEEE Trans. Circ. Syst. Video Technol. **14**(1), 95–101 (2004). https://doi.org/10.1109/TCSVT.2003.818358
5. Cozzolino, D., Poggi, G., Verdoliva, L.: Efficient dense-field copy–move forgery detection. IEEE Trans. Inf. Forensics Secur. **10**(11), 2284–2297 (2015). https://doi.org/10.1109/TIFS.2015.2455334
6. Cozzolino, D., Poggi, G., Verdoliva, L.: Splicebuster: a new blind image splicing detector. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6, November 2015 https://doi.org/10.1109/WIFS.2015.7368565
7. Dong, J., Wang, W., Tan, T.: Casia image tampering detection evaluation database, pp. 422–426, July 2013 https://doi.org/10.1109/ChinaSIP.2013.6625374
8. Farid, H., Lyu, S.: Higher-order wavelet statistics and their application to digital forensics. In: 2003 Conference on Computer Vision and Pattern Recognition Workshop. vol. 8, pp. 94–94 (June 2003). https://doi.org/10.1109/CVPRW.2003.10093
9. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Trans Inf. Forensics Secur. **7**(3), 868–882 (2012). https://doi.org/10.1109/TIFS.2012.2190402
10. Gustafsson, A.: Interactive image warping. https://www.gson.org/thesis/warping- thesis.pdf (1993). Accessed 19 July 2008
11. Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting fake news: Image splice detection via learned self-consistency. CoRR abs/1805.04096 (2018). http://arxiv.org/abs/1805.04096
12. Korus, P., Huang, J.: Multi-scale fusion for improved localization of malicious tampering in digital images. IEEE Trans. Image Process. **25**(3), 1312–1326 (2016)

13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classfication with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
14. Pevny, T., Fridrich, J.: Detection of double-compression in jpeg images for applications in steganography. IEEE Trans. Inf. Forensics Secur. **3**(2), 247–258 (2008). https://doi.org/10.1109/TIFS.2008.922456
15. Rao, Y., Ni, J.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (Dec 2016). https://doi.org/10.1109/WIFS.2016.7823911
16. Rhee, K.H.: Median filtering detection using variation of neighboring line pairs for image forensics. J. Electron. Imag. **25**(5), 1–13 (2016)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556 pp. 1–7 (2014)
18. Taimori, A., Razzazi, F., Behrad, A., Ahmadi, A., Babaie-Zadeh, M.: Quantization-unaware double JPEG compression detection. J. Math. Imaging Vis. **54**(3), 269–286 (2016). https://doi.org/10.1007/s10851-015-0602-z
19. Wen, B., Zhu, Y., Subramanian, R., Ng, T., Shen, X., Winkler, S.: Coverage—a novel database for copy-move forgery detection. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 161–165 Sep 2016
20. Wu, Y., Abd-Almageed, W., Natarajan, P.: Image copy-move forgery detection via an end-to-end deep neural network. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1907–1915 March 2018
21. Wu, Y., Abd-Almageed, W., Natarajan, P.: Deep matching and validation network an end-to-end solution to constrained image splicing localization and detection. CoRR abs/1705.09765 (2017), http://arxiv.org/abs/1705.09765
22. Wu, Y., Abd-Almageed, W., Natarajan, P.: BusterNet: detecting copy-move image forgery with source/target localization: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VI, pp. 170–186 (2018)
23. Yang, J., Ren, H., Zhu, G., Huang, J., Shi, Y.-Q.: Detecting median filtering via two-dimensional AR models of multiple filtered residuals. Multimed. Tools. Appl. **77**(7), 7931–7953 (2017). https://doi.org/10.1007/s11042-017-4691-0
24. Yue Wu, W.A., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgerieswith anomalous features (2019)
25. Zhou, P., Han, X., Morariu, V., Davis, L.: Learning rich features for image manipulation detection, pp. 1053–1061 (2018). https://doi.org/10.1109/CVPR.2018.00116
26. Zhang, J., Li, Y., Niu, S., Cao, Z., Wang, X.: Improved fully convolutional network for digital image region forgery detection. Comput. Mater. Con. **60**(1), 287–303 (2019)
27. Cui, Q., McIntosh, S., Sun, H.: Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs. Comput. Mater. Con. **055**(2), 229–241 (2018)
28. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
29. Fang, W., Zhang, F., Sheng, V.S., Ding, Y.: A method for improving CNN-based image recognition using DCGAN. Comput. Mater. Con. **57**(1), 167–178 (2018)

Author Queries

Chapter 10

Query Refs.	Details Required	Author's response
AQ1	This is to inform you that corresponding author has been identified as per the information available in the Copyright form.	
AQ2	Please check and confirm if the inserted citation of fig 1 is correct. If not, please suggest an alternate citation. Please note that figure should be cited sequentially in the text.	

VTD-Net: Depth Face Forgery Oriented Video Tampering Detection based on Convolutional Neural Network

Tongfeng Yang,Jian Wu,Lihua Liu,Xu Chang,Guorui Feng
Shandong University of Political Science and Law, Jinan 250014, P. R. China
E-mail: yangtf2014@126.com

Abstract: Face is the basis of identity authentication in many software, and the rise of generative adversary network makes the forgery of face easier than ever, which brings great challenges to information security. We propose a novel deep convolution neural network called VTD-Net to recognize faces generated by adversarial learning. The network is full-pipeline which composed of face location, interception, scaling and detection. In the experiment, we use the latest challenging face forgery dataset Celeb-DF, evaluated the forgery detection performance at frame level and video level, and achieved state-of-the-art results.

Key Words: Image Forensics, Generative Adversary Network, Face Swap

1 Introduction

The development of image processing technology and the popularization of image editing tools make image tampering easy. However, the manual image tampering technology has a huge workload in the processing of video, and it is difficult to match the mouth shape and expression of people in video. However, the emergence of generative adversary network(GAN) makes the image forgery become automatic. A large number of face swap software has appeared on the Internet for ordinary people to use, and there are also face swap projects with open source code on community such as GitHub. This makes video face swap extremely simple.

At present, a large number of methods for manual image forgery have been proposed. Most methods aim at one or several specific forgery methods, image smoothing[1, 2], splicing[3–5] , JPEG compression[6–9], and copy-move tampering[10–14] are the most concerned tampering methods.

Aiming at face swap algorithm, researchers have proposed some detection algorithms[15–22]. Detecting whether a video or image is forged is essentially a binary classification problem. These algorithms usually evaluate the accuracy of the original video and the forged video. The evaluation results published by these algorithms have reached a very high level, some of which have reached almost perfect results. Does this mean that the problem has been completely solved?

From the observation of the datasets used by these algorithms, it can be seen that the datasets they use are often distinguishable by the naked eye. The brightness and hue of the fake face are obviously different from those around the face, and there is flickering phenomenon in the video. This kind of video on the network can be seen through at a glance, causing little damage.

In addition, the evolution of deep forgery technology is also a reason. To solve this problem, researchers construct a new Celeb-DF[23] dataset. They generate high-resolution images (256×256). By randomly changing the brightness, contrast, color distortion and sharpness of the input image, the face is enhanced, which improves the diversity of training data and eliminates the obvious difference between the fake face and the surrounding image. It is almost impossible to distinguish through human eyes. The performance of existing algorithms on this dataset is very poor.

In this paper, a whole pipeline is designed to detect the videos with fake faces. Both of the forgery detection at frame level and video level have achieved good results.

Our contributions are as follows: firstly, we provide an effective preprocessing layer to improve the recognition ability of the network to the forged image. Secondly, an efficient classification network with fewer parameters is designed. Thirdly, 82.56% accuracy and 90.2% AUC(Area Under Curve) are achieved on the challenging Celeb-DF dataset, which is much higher than the currently reported AUC value of 55.7%.

2 Related Work

Fridrich[24] designed 15 high pass convolution kernels for steganalysis. Zhou[25] chose 3 of them as the front layer of its image tamper detection neural network, which achieved good results. This is in line with the theoretical analysis: when the image is spliced, high-frequency traces will remain on the edge of the splicing.

Bayar[26] designs a special convolution layer, which limits the center of the convolution kernel to - 1 and the sum of the whole convolution kernel to 0. This design weakens the interference of image content on tamper detection.

Zhou[27] combines the two streams of GoogLeNet and expression of local noise residue and camera features to realize the detection of tampered face.

Afchar[15] proposed Mesonet: by constructing CNN network to detect face forgery, and improved it with inception technology. The network has achieved good results on the relatively simple Face2Face and deepfake datasets.

Yang[19] extracted face pose as a clue to solve the problem of incongruity between face and head pose in face swap video, and classified it using SVM.

Supported by Program for Young Innovative Research Team in Shandong University of Political Science and Law(Network Information Security and Forensics, Intelligent Information Processing);Big data and Artificial Intelligence Legal Research Collaborative Innovation Center of Shandong University of Political Science and Law; Key Laboratory of Evidence-Identifying in Universities of Shandong (SDUPSL); Projects of Shandong Province Higher Educational Science and Technology Program under Grant No.J16LN19, J18KA357, J18KA383; Projects of Shandong University of Political Science and Law under Grant No. 2016Z03B, 2015Z03B,2019KQ13Z.

Li[20] analyzes whether the face in the video is distorted by extracting the landmark on the face for forgery detection. This algorithm has a good effect on the early forgery algorithm which uses distorted face to insert face.

By observing the two early face forgery datasets, Face2Face and faceswap, Matern[18] found many forgery traces, such as the inconsistency of the brightness of the two eyes, the shadow of the nose and so on, and extracted the visual features for classification.

Nguyen[22] proposes a multi task learning algorithm to detect tampered images and further locate tampered regions.

Rössler[28] released a dataset called Faceforensics++, and implemented face swap video detection by modifying XceptionNet.

The above algorithm performs well in traditional datasets but pool in Celeb-DF dataset with high challenges, and the AUC measure is no more than 56% in the experiment.

3 Proposed Neural Network

3.1 Overview

Our model uses a video as input, uses FFmpeg to decode the video, positions and segments the head image in the decoded frame, then puts the head image into the preprocessing layer after zooming, and then puts the preprocessed image into a CNN to get the result of whether each frame image has been tampered, and then infers whether the video is tampered according to the detection result of the image. As shown in the Fig.1.

3.2 Video Frame Extraction

The video consists of several consecutive frames. We use FFmpeg to decode the video, and decode the video into many pictures. Because the adjacent frames have high similarity in the image, we take out one image every 10 frames(5 frames if video is short). In order to increase the diversity of datasets, we do not take too many images from a video, but only a few images from each image. The number of original video and forged video provided in the dataset is unbalanced. In order to get the positive and negative examples of equivalence, we extract 40 images from each original video as the counter example and 10 from each forged video as the positive example.

3.3 Face Detection and Extraction

MTCNN[29] is employed to detect the face in the frame. MTCNN uses a cascaded CNN structure, which completes face detection and face alignment simultaneously through multi task learning. The network outputs the position of the face in the image and the key points of the face, including the positions of eyes, nose and mouth.

We only use the face detection function of MTCNN, but not the human face alignment function. Face alignment is to put the face in the center of the image for face recognition, but it will cause the distortion of the edge pixels. The core of face swap detection is to judge whether the image is changed or not. The edge image generated by alignment will bring interference to face swap detection, so we do not use face alignment.

Not only that, in order to avoid the network to detect the identity of the face, we also rotate the face image in training set. First, the face is cut from the image, and then is rotated 90 degrees, 180 degrees and 270 degrees respectively.

These face extracted from the video and the rotated face form the training set.

In order to facilitate further image classification, all face images are zoomed to size of 128×128.

3.4 High-pass Block

We build the SRMConv2d[25] layer with 9 filters by the 3 kernels given in Equation 1. The input of this layer is an image with 3 color channels: red, green and blue, so the shape of the filter is $(3, 5, 5)$.

$$\begin{aligned}
K_1 &= \frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
K_2 &= \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix} \\
K_3 &= \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{1}$$

Let O donate the 5×5 zero matrix, the 9 filters can be represented by Equation 2. In other words, we let the kernel K_i given above as the weights for one input channel, and the weights for the other two channels are set to zero matrix. It can separately process the three channels of the image to extract features with higher discrimination.

$$\begin{bmatrix} K_1 & O & O \\ O & K_1 & O \\ O & O & K_1 \\ K_2 & O & O \\ O & K_2 & O \\ O & O & K_2 \\ K_3 & O & O \\ O & K_3 & O \\ O & O & K_3 \end{bmatrix} \tag{2}$$

BayarConv2d is a constrained convolutional layer while the center value of the filters fixes to -1, and the sum of other values in one kernel fix to 1. The amount of all weights normalize to zero. We use 3 such filters in our network.

Let w_{ij} donates the weight of 1 filter of BayarConv2d layer, we have:

$$\begin{cases} \sum_i \sum_j w_{ij} = 0 \\ w_{0,0} = -1 \end{cases} \tag{3}$$

We merge the output of the SRMConv2d layer and BayarConv2d layer in the channel dimension and get 12 channels at all.

3.5 Classification Block

The neural network block for classification consists of three convolution layers and two full connection layers.

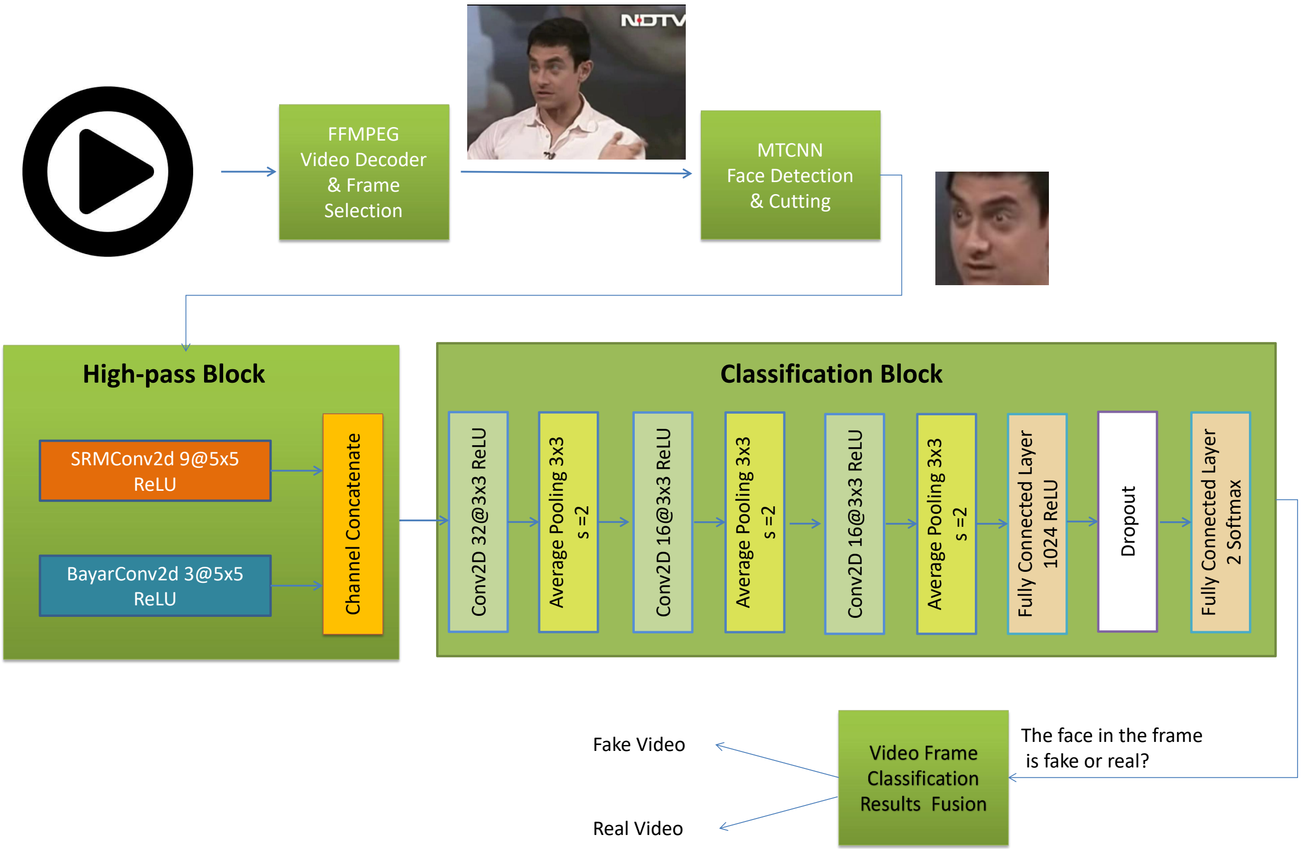


Fig. 1: Overall Network Architecture

Each convolution layer is followed by a pooling layer, which is then activated using the ReLU activation function.

The three convolution layers are 3 × 3 in size. The first convolution layer uses 32 channels, the second uses 16 channels, and the third uses 16 channels. The purpose of this design is to reduce the number of parameters as much as possible, speed up convergence, and delay the occurrence of over fitting phenomenon.

The role of pooling layer is to reduce the size of the image and the number of parameters. According to our previous work, we chose average pooling instead of max pooling. The step is 2 and the pooling size is 3 × 3.

In the last full connection layer of this block network, the first layer contains 1024 neurons, and the second layer contains 2 neurons as output, using softmax activation function.

The loss function of the network is set to sparse categorical crossentropy, and the evaluation methods are accuracy and Area Under Curve(AUC).

3.6 Video Frame Classification Results Fusion

The second full connection layer of the network will provide two 0-1 values for each image, and the second value can be regarded as the probability of whether the image is tampered with. We get the probability of the same video from the mean of the probability of all the extracted images belonging to the same video. If it is greater than 0.5, it is considered as tampered video.

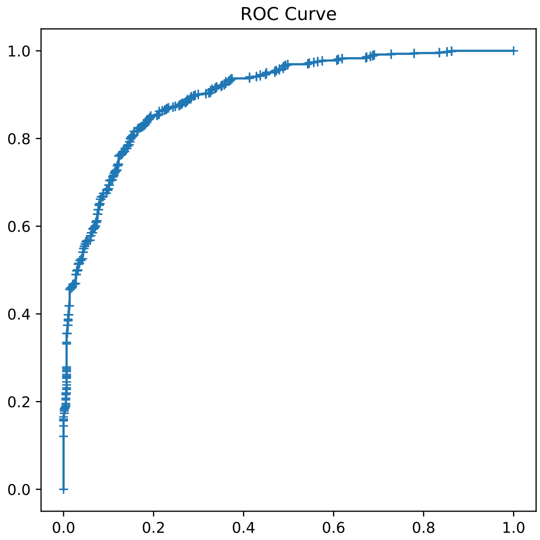


Fig. 2: ROC Curve of Frame Level Tampering Detection

4 Experiments

4.1 Dataset

In the experiment, we use a relatively novel celeb DF dataset, which contains very realistic video, which can hardly be distinguished by human eyes. Its performance is much higher than other existing datasets. In the process of video generation, a variety of optimization strategies are adopted to improve its performance, including: directly generating 256 × 256 head image, rather than relying on zoom, randomly adjusting brightness, contrast, color deviation and sharpness to expand the dataset, improving the color consistency of the generated results, better face insertion technology and solving the time flicker problem.

The dataset contains 158 original videos, 795 fake videos and 250 original videos from YouTube. Because of the imbalance of data, each original video extracts about 40 images, and each forged video extracts about 10 images. After that, the redundant images extracted from the forgery video are discarded to get the same amount of training set and test set images. We use MTCNN[29] to extract the face of the extracted image, and then reduce the face to 128x128. Finally, 5886 original heads and the same number of forged heads were obtained.

4.2 Division of training and test set

We divide the dataset by video (video level split), that is, all the head images generated by the same video are either divided into training sets or test sets. Nine out of ten images from video are treated as training sets and the rest are test sets.

We used to divide the dataset randomly based on the pictures, and the classification accuracy is more than 99%. The reason is that if a video generates 10 pictures, nine of them belong to the training set and one belongs to the test set according to the probability, then the nine pictures belong to the training set have the label of whether they are forged images or not. This image in the testing set can be judged by the similarity (i.e. face recognition) only through the comparison with the training set, without judging whether they are forged. This method is disadvantageous to train the network for detecting the forged image, although the accuracy is so high.

4.3 Data Augmentation

In order to avoid training the network to recognize the face as much as possible, we expand the face in the training set, mirror the face horizontally, rotate 90 degrees, 180 degrees and 270 degrees. Face from different angles avoids the recognition of face similarity by network.

4.4 Frame Level Training and Test

We detect whether the extracted head image is tampered, then put the processed data into the network for training, and use the test set to evaluate the accuracy and AUC.

We get the accuracy of 0.8256. The ROC curve is shown in Figure 2, and the corresponding AUC (area under curve) value is 0.9016. The results of this experiment are compared with those published in Celeb-DF, as shown in Table 1.

The curve of loss and accuracy during training is shown in Figure 3.

4.5 Video Level Tampering Detection

The 1176 pictures in the test set used in the frame level classification belong to 74 videos. In this step of experiment, we successfully identified 68 videos, only 6 videos were misjudged, with an accuracy of 91.89%. The corresponding ROC curve is shown in Figure 4, and the AUC value is 0.9548.

Table 1: AUC(%) of Performance of Each Method

Methods	AUC(%)
Two-stream[27]	55.7
Meso4[15]	53.6
MesoInception4[15]	49.6
HeadPose[19]	54.8
FWA[20]	53.8
VA-MLP[18]	48.4
VA-LogReg[18]	46.9
Multi-task	36.5
Xception	38.7
Proposed Method	**90.2**

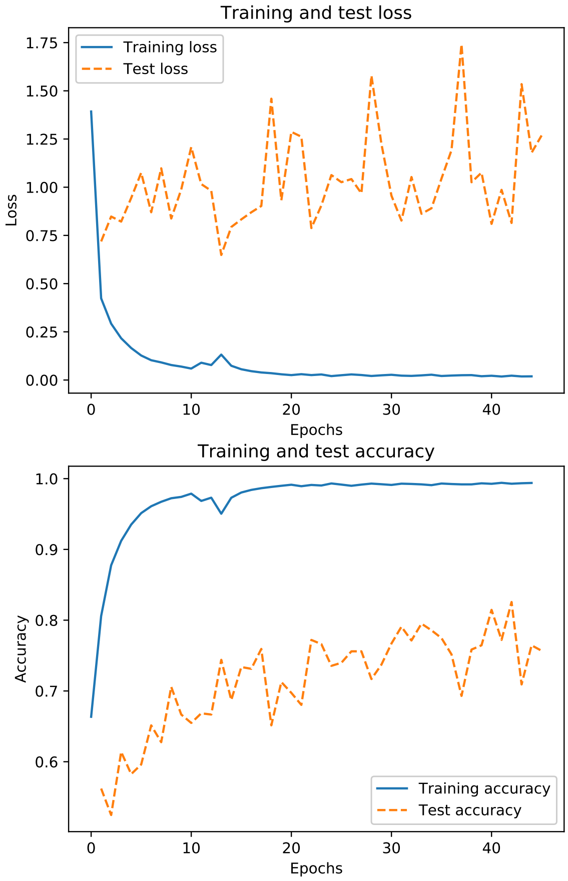


Fig. 3: The Loss and Accuracy Curve of Proposed Method

5 Conclusion

In this paper, we present a detection algorithm for deep face forgery, which has achieved good results in the novel celeb DF dataset with high difficulty. In the experiment, we also found that the initial weight of the randomly generated network will have a certain impact on the whole training results, which is expected to be improved in the follow-up work.

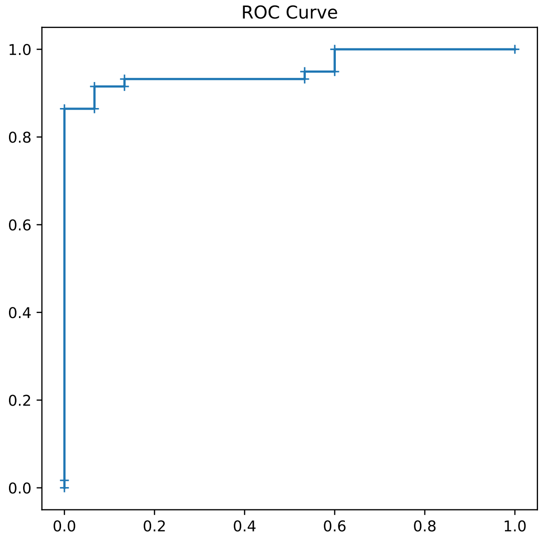


Fig. 4: ROC Curve of Video Level Tampering Detection

References

[1] Kang Hyeon Rhee. Median filtering detection using varia-tion of neighboring line pairs for image forensics.Journal ofElec-tronic Imaging, 25(5):1 – 13, 2016.

[2] Jianquan Yang, Honglei Ren, Guopu Zhu, Jiwu Huang,and Yun-Qing Shi.Detecting median filtering via two-dimensional ar models of multiple filtered residuals.Multi-media Tools Appl., 77(7):7931–7953, 2018.

[3] D. Cozzolino, G. Poggi, and L. Verdoliva. Splicebuster:A new blind image splicing detector. In 2015 IEEE International Workshop on Information Forensics and Security(WIFS), 2015:1–6.

[4] Yue Wu, Wael Abd-Almageed, and et. Deepmatching and validation network - an end-to-end solution toconstrained image splicing localization and detection.CoRR,abs/1705.09765, 2017.

[5] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A.Efros. Fighting fake news: Image splice detection via learnedself-consistency.CoRR, abs/1805.04096, 2018.

[6] T. Pevny and J. Fridrich. Detection of double-compression injpeg images for applications in steganography.IEEE Trans-actions on Information Forensics and Security, 3(2):247–258,2008.

[7] Bin Li, Y. Q. Shi, and Jiwu Huang. Detecting doubly compressed jpeg images by using mode based first digit features.In 2008 IEEE 10th Workshop on Multimedia Signal Processing, 2008:730–735.

[8] P. Korus and J. Huang. Multi-scale fusion for improved localization of malicious tampering in digital images.IEEE Transactions on Image Processing, 25(3):1312–1326, 2016.

[9] Ali Taimori, Farbod Razzazi, and et. Quantization-unaware doublejpeg compression detection.Journal of Mathematical Imag-ing and Vision, 54(3):269–286, 2016.

[10] D. Cozzolino, G. Poggi, and L. Verdoliva. Efficient densefield copy–move forgery detection.IEEE Transactions on Information Forensics and Security, 10(11):2284–2297, 2015.

[11] B. Wen, Y. Zhu, R. Subramanian, T. Ng, and et. Coverage — a novel database for copy-move forgerydetection. IEEE International Conference on Image Processing (ICIP), 2016:161–165.

[12] Y. Rao and J. Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In 2016 IEEE International Workshop on Information Forensics and Security (WIFS), 2016:1–6.

[13] Y. Wu, W. Abd-Almageed, and P. Natarajan. Image copymove forgery detection via an end-to-end deep neural network. In2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018:1907–1915.

[14] Yue Wu, Wael Abd-Almageed, and Prem Natarajan.Buster-Net: Detecting Copy-Move Image Forgery with Source Target Localization: 15th European Conference, Munich, Germany,2018:8–14.

[15] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and IsaoEchizen. Mesonet: a compact facial video forgery detectionnetwork.CoRR, abs/1809.00888, 2018.

[16] D. Güera. E. J.Delp. Deepfake video detection using recurrent neural networks. 15th IEEE InternationalConference on Advanced Video and Signal Based Surveil-lance (AVSS), 2018:1–6.

[17] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi:Exposing AI generated fake face videos by detecting eyeblinking.CoRR, abs/1806.02877, 2018.

[18] F. Matern, C. Riess, and M. Stamminger. Exploiting vi- sual artifacts to expose deepfakes and face manipulations. In 2019 IEEE Winter Applications of Computer Vision Work- shops (WACVW), 2019:83–92.

[19] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses.CoRR, abs/1811.00661, 2018.[19] F. Matern, C. Riess, and M. Stamminger.Exploiting vi-sual artifacts to expose deepfakes and face manipulations. In2019 IEEE Winter Applications of Computer Vision Work-shops (WACVW), 2019:83–92.

[20] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts.CoRR, abs/1811.00656, 2018.

[21] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convo-lutional strategies for face manipulation detection in videos.CoRR, abs/1905.00582, 2019.

[22] Huy H. Nguyen, Fuming Fang, Junichi Yamagishi, andIsao Echizen.Multi-task learning for detecting and seg-menting manipulated facial images and videos.CoRR,abs/1906.06876, 2019.

[23] Pu Sun Honggang Qi Yuezun Li, Xin Yang and Siwei Lyu.Celeb-df: A new dataset for deepfake forensics.arXivpreprint arXiv:1909.12962, 2019.

[24] J. Fridrich and J. Kodovsky. Rich models for steganalysis ofdigital images.IEEE Transactions on Information Forensicsand Security, 7(3):868–882, 2012.

[25] Peng Zhou, Xintong Han, Vlad Morariu, and Larry Davis.Learning rich features for image manipulation detection.2018:1053–1061.

[26] B. Bayar and M. C. Stamm. Constrained convolutional neuralnetworks: A new approach towards general purpose imagemanipulation detection.IEEE Transactions on Information-Forensics and Security, 13(11):2691–2706, 2018.

[27] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S.Davis. Two-stream neural networks for tampered face detection.CoRR, abs/1803.11276, 2018.

[28] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images.CoRR,abs/1901.08971, 2019.

[29] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao.Joint face detection and alignment using multi-task cascaded convolutional networks.CoRR, abs/1604.02878, 2016.

A Self-Supervised Learning Network for Student Engagement Recognition From Facial Expressions

Wen-Long Zhang, Rui-Sheng Jia, Hu Wang, Cheng-Yue Che, and Hong-Mei Sun

Abstract— **Student engagement in online learning is an important indicator for measuring learning effectiveness. Due to the fact that facial video data of students during online learning contains a wider range of information such as time, current research has begun to focus on obtaining student engagement from video data. These studies primarily rely on supervised learning methods and have achieved certain success. However, the longstanding lack of large-scale and high-quality labeled data, as well as the time-consuming and laborious sample labeling work, have to some extent hindered their further improvement. To solve this problem, this paper proposes a self-supervised learning method, Facial Masked Autoencoder (FMAE), which is used to construct a student engagement recognition model. This method uses a masked autoencoder to process a large number of unlabeled facial videos, and performs self-supervised pre-training by learning masked facial features from the reconstruction process. In order to promote the encoder to better mask learning for the face, a new facial mask strategy and reconstruction module have been proposed. With this method, the model can not only focus on important facial regions, but also obtain more accurate appearance features and spatio-temporal details. Experiments have demonstrated that the proposed method achieves excellent results on DAiSEE and EmotiW datasets, showing its potential in the task of student engagement recognition.**

Index Terms— **Online learning, student engagement, self-supervised learning, masked autoencoder.**

I. Introduction

NOWADAYS, online education has attracted wide attention. With its flexibility and portability, it can provide rich educational resources, and allow students to experience national or even global quality teaching resources without going out [1]. However, there are some limitations to this form of education, one of which is the lack of feedback and interaction. In traditional classroom education, teachers usually adopt some corresponding teaching methods, such as observing students' facial expressions and other behaviors to grasp student engagement in real time. However, in online education, face-to-face communication between students and teachers is often lacking [2]. Therefore, the ability to assess student engagement and provide timely feedback or intervention in online education is an important factor that affects students' learning outcomes and ensures their learning effects.

The identification methods of student engagement can be divided into two categories according to the type of input data: the identification methods of student engagement based on static data and the identification methods of student engagement based on dynamic data. The former uses static images as input, while the latter is designed to identify student engagement in dynamic image sequences or videos. Since methods based on static data ignore the critical temporal information of the face, this paper will focus on methods based on dynamic data. This method has higher accuracy and can better reflect student engagement.

The identification of student engagement based on dynamic data is mainly relies on supervised learning methods. Currently, researchers have exploited diversified deep neural networks for this assignment, including 2D/3D convolutional neural networks (CNN) [15], [16], [17], ensemble neural networks [18], [19], [20], [21] and a more advanced Transformer based architecture [22], [23]. Despite the remarkable success of supervised learning methods in the task of student engagement identification, there are still several obstacles that limit the further development of this field. Firstly, the currently available datasets for student engagement recognition are rather limited, and the training samples within these datasets are relatively single (only contain videos of a few subjects and only cover a few scenes). Secondly, supervised learning methods are prone to overfitting and therefore have poor generalization ability when applied to other datasets or practical applications. Finally, the collection of large-scale and high-quality annotation data is a time-consuming and laborious task [3]. Considering that there are a large number of unlabeled facial videos on the Internet, the task of student engagement recognition can be accomplished by self-supervised learning methods (Fig. 1. illustrates the process of our self-supervised learning method in contrast with the supervised learning method).

Self-supervised learning has achieved remarkable achievement in multiple deep learning researches. Among them, MAE, as one of the important methods of generative self-supervised learning, has recently took out hitherto unknown results in numerous deep learning researches [4]. Initially, this method was applied to masked language modeling in the field

Manuscript received 6 May 2024; revised 4 July 2024 and 22 July 2024; accepted 26 July 2024. Date of publication 31 July 2024; date of current version 23 December 2024. This work was supported by the Humanities and Social Science Fund of the Ministry of Education of the People's Republic of China under Grant 22YJAZH036. This article was recommended by Associate Editor Z. Tang. *(Corresponding authors: Rui-Sheng Jia; Hong-Mei Sun.)*

The authors are with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China (e-mail: zhangwenlong@sdust.edu.cn; jrs@sdust.edu.cn; wanghu@sdust.edu.cn; chechengyue@sdust.edu.cn; skd991915@sdust.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2024.3436029.

Digital Object Identifier 10.1109/TCSVT.2024.3436029

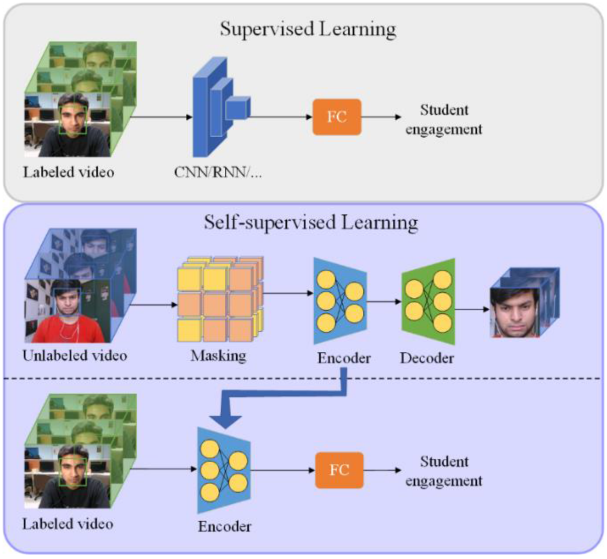


Fig. 1. Supervised and self-supervised learning methods.

of natural language processing, where models are pre-trained by a strategy of first masking and then reconstruction. With the success of BERT based mask methods [5], mask autoencoders have also been continuously explored in the visual field. In the realm of point cloud learning, McP-BERT has shown tremendous potential [6]. It has successfully optimized the process of self-supervised pre-training for point clouds by ingeniously introducing multi-choice token and utilizing the high-level semantics information learned from transformers. Additionally, VideoMAE extends MAE to the video fields and has achieved deep impression results on many universal video datasets [7]. In order to better perform the masking task, some studies have adopted a variety of different design options, for example pixel level masking [8], [9], token level masking [10], [11], depth feature based masking [12], and the use of ViT [13]. In addition, in order to better model the spatio-temporal patterns of input data, recent studies have also explored strategies such as masked motion modeling [14] and tube masking [7]. Inspired by these research strategies, this paper proposes a Facial Masked Autoencoder (FMAE) based on self-supervised learning, which learns masked facial features during the reconstruction process to solve the dilemma that supervised learning methods face in video-based student engagement recognition. The main contributions of this paper are summarized as follows:

1) A new self-supervised learning network is proposed for student engagement recognition. In this method, mask autoencoder is applied to process a large number of unlabeled facial video data, which effectively reduces the cost of sample annotation.
2) An encoder with a new masking strategy is designed, which forces the encoder to pay more attention to the spatio-temporal details of the specific facial regions through the mask operation of these regions in the time series, so as to improve the learning ability of facial features.
3) An effective reconstruction module is constructed, which helps the model accurately reconstruct the masked facial region through joint reconstruction loss and adversarial loss, thereby promoting the model to learn more detailed and rich facial representations during the reconstruction process.
4) The experimental results indicate that FMAE has achieved remarkable achievements on DAiSEE and EmotiW datasets, which proves its effectiveness in the task of student engagement recognition.

The structure of the remaining parts of this paper is as follows. The second section discusses the relevant research in the field of student engagement recognition. The third section describes the details of the proposed method in this paper. The fourth section presents the experimental results and analysis on DAiSEE and EmotiW datasets. The fifth section provides a summary of this paper and looks forward to future work directions and potential improvement space.

II. Related Works

In the research of using video data to obtain student engagement, researchers mainly adopt supervised learning methods and strive to develop more advanced deep learning architectures to extract valuable spatial-temporal information from original facial videos, so as to obtain student engagement. In general, three trends can be summed up.

Firstly, one trend is to straight utilize 3D CNNS to obtain combined spatio-temporal features from original face videos. Geng et al. [15] used C3D model in their study to identify student engagement by modeling appearance (facial expression) and motion information in videos. Zhang et al. [16] innovatively introduced I3D, an excellent network in the field of behavior recognition, into the field of student engagement recognition. The structure of the model is improved and optimized according to the characteristics of the student engagement recognition dataset, so that it could accurately analyze and evaluate student engagement from facial video data, thus significantly improving the accuracy of student engagement recognition. In addition, Mehta et al. [17] based on 3D DenseNet and self-attention mechanisms designed a neural network model to identify and assess student engagement and emotional state in online education. The self-attention block in the model helps to extract enhanced facial features (spatial features, temporal features, spatial-temporal features), which enables the model to effectively detect the student engagement status in the video sequence, ultimately achieving satisfactory accuracy results.

Secondly, the second trend is to use an ensemble model, first using a 2D CNN to obtain facial features from each static frame, and then using RNN to synthesize the dynamic temporal information of all frames. Wu et al. [18] designed a feature-based method that they obtain facial features and the upper part of the body features from videos through 2D convolutional neural networks, and combined Long Short-Term memory (LSTM) and Gated Recurrent Unit (GRU) to classify the extracted features to identify the degree of student engagement. In addition, Zhu et al. [19] designed a GRU

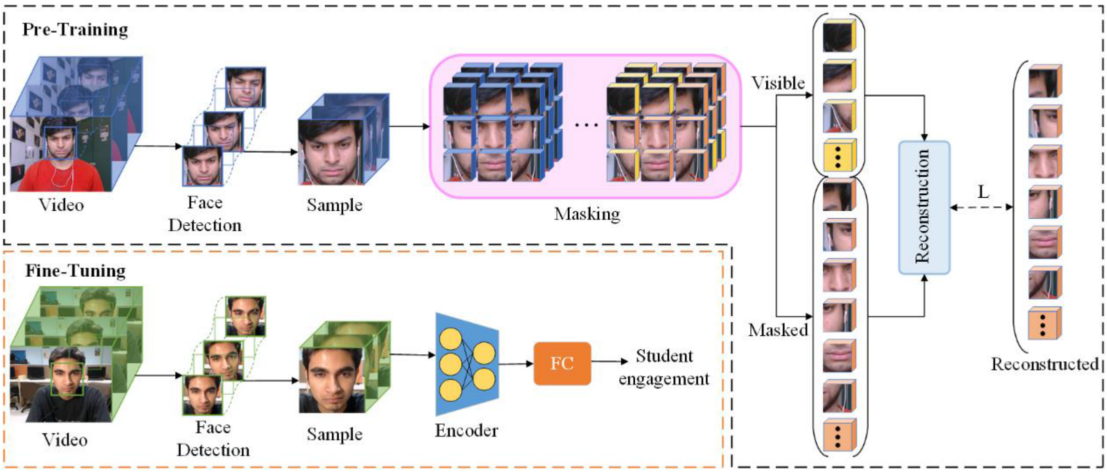


Fig. 2. Overview of the FMAE architecture. It consists of two phases: self-supervised pre-training and fine-tuning. Pre-training architecture: FMAE learns facial representation from unlabeled network video data in a self-supervised learning manner. Fine-tuning architecture: The pre-trained model is fine-tuned to make it suitable for the student engagement recognition task.

model for attention mechanism, which extracts facial features, posture features and CNN features from videos. The final student engagement level is predicted by combining these three different types of features. Additionally, Liao et al. [20] proposed a deep face spatio-temporal network (DFSTN) for online learning, which uses a pre-trained SE-ResNet-50 to extract the spatial features of the face and uses an LSTM with global attention to generate attention hidden states, so as to detect student engagement. In addition, Abedi and Khan [21] treated detecting student engagement as a classification mission related to spatio-temporal information and proposed a new end-to-end ResNet + TCN hybrid neural network. Among them, ResNet is used to extract spatial features of faces from successive video frames, while TCN obtains the degree of engagement by processing temporal changes of faces in video frames.

Recently, with the rise of Transformer, several studies have begun to exploit its global dependency modeling capabilities for better performance, which forms a third trend. Xusheng et al. [22] designed a new class attention method based on ViT [13], called CavT. In this method, the video sequence is divided into blocks along temporal and spatial dimensions, each block contains the same part of the adjacent image, and the blocks are converted into patch embeddings using linear projection. Finally, ViT with a class attention module is used to process these patch embeddings. Through unified training of long videos of variable length and short videos of fixed length, this method can carry out student engagement learning and achieve good accuracy. Chen et al. [67] proposed an innovative Multi-relation Perception Network (MRAN), which deeply explores and learns multi-level relationships among local regions, between global-local features, and among different samples by comprehensively focusing on the significant features of the whole face and local areas. With this multi-dimensional feature capture, MRAN is able to extract richer facial features from different angles. Additionally, Qin et al. [23] proposed a multi-task network known as SwinFace, which is based on the Swin Transformer and capable of performing a variety of tasks including face recognition, expression recognition, age estimation, and facial attribute estimation. To address potential conflicts between tasks and to meet their unique requirements, they have integrated a Multi-Level Channel Attention (MLCA) module into each subnet. This module can adaptively select the most appropriate feature levels and channels to accurately execute tasks. With this design, SwinFace not only demonstrates a deep understanding of facial features, ensuring exceptional accuracy in the recognition task of student engagement, but also excels across all tasks.

However, existing student engagement recognition methods mainly use supervised learning methods to analyze manually annotated data, so they are limited by existing student engagement datasets. Different from them, this paper designs a self-supervised learning method that could learn effective facial features from a quantity of unlabeled face video data and apply it to the field of student engagement recognition.

III. Proposed Method

A. FMAE

Considering the task of student engagement recognition as a whole, it can be analyzed from two different aspects. The first is the area associated with facial features, which includes various regions of the face (eyes, nose, mouth, etc.), mainly studying the facial shape and texture of these area; Secondly, temporal information in facial movements needs to be taken into account, so spatio-temporal modeling is very necessary. To achieve this goal, a new self-supervised learning framework FMAE is proposed, as shown in Fig. 2. The framework applies facial masking strategy to mask the unlabeled video data. Then, the masked image is reconstructed from the visible input using an encoder decoder architecture, and a reconstruction module is used to train the encoder between the reconstructed image and the input image, which is applied to the recognition task of student engagement.

FMAE mainly consists of two parts: facial mask strategy and reconstruction module. For a given training dataset $S =$

$\{V_i\}_{i=1}^{N}$, where N is the number of videos in the dataset and V is a single video in the input dataset. From the original input video V, the facial region is obtained by tracking and cropping, and then random time sampling is performed to obtain $v \in \mathbb{R}^{C\times T\times H\times W}$ $(C, T, H, W$ respectively represent the number of channels of the processed video, the length of the video, and the height and width of a single frame of the video). Through the facial masking strategy, additionally map v into n visible tokens and $(p - n)$ masked tokens using a predefined masking ratio $r = (p - n)/p$. Visible tokens are represented as $K_v \in \mathbb{R}^{n\times d}$, masked tokens are represented as $K_m \in \mathbb{R}^{(p-n)\times d}$, where d is the embedded dimension, and p is the total number of tokens mapped from v, for a given token $p = T/t \times H/h \times W/w$ with $t \times h \times w$ dimension. Therefore, FMAE passes specific areas of the face through the above tokens for mask learning, which can better focus on important features of the face. The visible tokens are mapped through the encoder to the latent feature space z, which captures the key information of the facial representation. Then, the decoder will utilize the information in the latent feature space z to reconstruct it into $(p - n)$ masked tokens. The reconstruction process aims to recover the masked information in order to enhance understanding of facial features. It is quite a challenging task to reconstruct a spatio-temporal face from original pixels, so a reconstruction module is designed for better training.

The FMAE model learns rich and detailed face representations from face videos through self-supervised learning, and then fine-tuning [24] is used to make it suitable for the task of student engagement recognition. For a given dataset $S_e = \{v_j, y_j\}_{j=1}^{N}$ of student engagement, a linear fully connected layer with embedding parameters w is introduced to align the latent feature space obtained through the encoder with the label space of the video data of student engagement. In the fine-tuning, the backbone network is frozen and only the w parameters of the full connection layer are updated. This can effectively utilize the learned facial representations and associate them with labels of student engagement, thereby improving the performance of student engagement recognition.

B. Facial Masking Strategy

Since video can be regarded as the temporal extension of static appearance, and there is correspondence between video frames, such temporal correlation may lead to information leakage, where masked content in one frame may be visible in another frame [25]. Therefore, in order to make better use of spatio-temporal information of video data, a facial mask strategy is designed inspired by strategies such as tube mask [7]. The architecture of this strategy is shown in Fig. 3. Specifically, the strategy processes each spatio-temporal cube by dynamic tracking and masking, which means that the same facial region is masked at the spatial location of different time frames of the video, thus maintaining consistency while shielding the influence of correlation information. In this way, when performing later reconstruction tasks, the model needs to overcome the challenge introduced by the mask, which is to recover the complete facial representation from the visible part of the facial information. Therefore, through this masking strategy, the model is encouraged to learn the local and global features of the face, so as to better utilize the spatio-temporal information in the video data and improve the learning effect of the face representation.

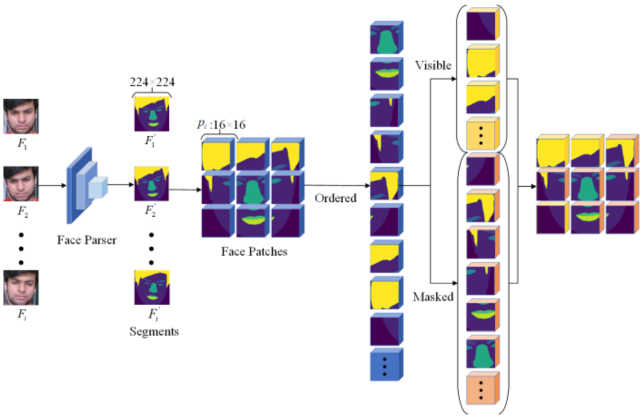


Fig. 3. Facial mask strategy.

Before the mask operation, face parsing is performed first [26]. Through face parsing, the facial image can be divided into different regions, such as left eye, right eye, nose, mouth, hair, skin and background, as shown in Equation (1). Based on the characteristics of these facial regions, mask strategies that are more suitable for faces can be designed.

$$s = \left\{f_{le}, f_{re}, f_{no}, f_{mo}, f_{ha}, f_{sk}, f_{bg}\right\} \tag{1}$$

Firstly, different priorities are set for different facial regions according to their importance and attention, as shown in Equations (2) and (3). In general, the left eye, right eye, nose, mouth and other regions of the face provide more abundant and important facial feature information, so these facial regions are set higher priority. Next, using a predefined masking ratio, the mask operation is first applied to the facial regions with higher priority. This means that the information of these regions will be hidden during the encoding and reconstruction process, thus ensuring the accurate acquisition and reconstruction of important facial features. Subsequently, mask operations are performed on areas such as hair, skin, and background. With this selective mask operation, it is possible to reduce the reconstruction of irrelevant or secondary facial information, thereby improving the quality and accuracy of facial reconstruction.

$$S_{high} = \{f_{le}, f_{re}, f_{no}, f_{mo}\} \tag{2}$$

$$S_{low} = \left\{f_{ha}, f_{sk}, f_{bg}\right\} \tag{3}$$

By adopting this masking strategy, map the input v into n visible tokens and $(p - n)$ masked tokens. Next, these visible tokens are used to encode and reconstruct the spatio-temporal variations of the face. Facial mask provides a more efficient masking strategy for facial reconstruction tasks, which enables more accurate acquisition of facial appearance features and spatio-temporal details. Algorithm 1 shows the processing of the facial mask operation.

Algorithm 1 Facial Masking Procedure

```
Require: $v \in \mathbb{R}^{C \times T \times H \times W}, r$
$regionMap \leftarrow FaceParser\ (v)$   ▷ $Face - Parsing$
    $regionMap \in \{f_{le}, f_{re}, f_{no}, f_{mo}, f_{ha}, f_{sk}, f_{bg}\}$
$s = \{f_{le}, f_{re}, f_{no}, f_{mo}, f_{ha}, f_{sk}, f_{bg}\}$ ▷$Prioritize\ Regions$
$p = T/t \times H/h \times W/w$   ▷ $tokens\ for\ each\ v$
    ▷ $(3D\ cube\ tokens\ have\ \dim ension\ of\ t \times h \times w\ each)$
$(p - n) \leftarrow r \times p$   ▷ $Number\ of\ masked\ tokens$
$K_m \leftarrow \{\}$   ▷$Initializemaskedtokens$
$S_{high} = \{f_{le}, f_{re}, f_{no}, f_{mo}\}$
    $S_{low} = \{f_{ha}, f_{sk}, f_{bg}\}$   ▷ $Set\ priority$
$for\ s'\ in\ s\ do$
    $K_m \leftarrow \{s'\}$
    $if\ len\,(K_m) == (p - n)\ then$
        $break$
    $end\ if$
$end\ for$
$K_v \leftarrow K - K_m$   ▷ $K\ is\ all\ tokens\ from\ v$
```

Algorithm 2 Training procedure for FMAE

```
Require: $Ft, En\,(Encoder), De\,(Decoder), D, S, r, w, S_e$
$while\ not\ converged\ do$ ▷ $FT - MAE\ pre - training$
    $v \leftarrow S$   ▷ $sample\ batch$
    $\{K_m, K_v\} \leftarrow Ft\,(v, r)$
        ▷ $Facial\ Trajectory\ Masking\,(See\ Algorithm1)$
    $K'_m \leftarrow DeEn\,(K_v)$
    $\{D\} \leftarrow \nabla_{\{D\}}\mathcal{L}^d\left(K_m, K'_m\right)$
    $K'_m \leftarrow DeEn\,(K_v)$
    $\{En, De\} \leftarrow \nabla_{\{En,De\}}\mathcal{L}^g\left(K_m, K'_m\right)$
$end\ while$
$while\ not\ converged\ do$
    $\{v, y\} \leftarrow S_e$   ▷ $sample\ batch$
    $K \leftarrow v$   ▷$K\ is\ all\ tokens\ from\ v$
    $y' \leftarrow wEn\,(K_v)$
    $\{w\} \leftarrow \nabla_{\{w\}}\mathcal{L}_e\left(y, y'\right)$   ▷ $Linear\,Probing$
$end\ while$
```

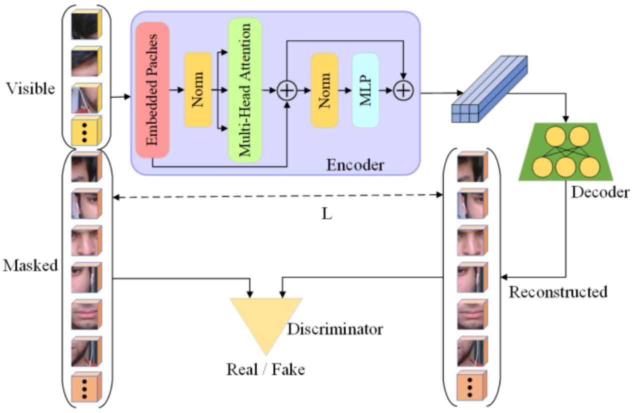


Fig. 4. Reconstruction module.

C. Reconstruction

After the facial masking, the facial reconstruction is realized through the cooperative work of the encoder and decoder. The architecture of the reconstruction module is shown in Fig. 4. The n visible tokens obtained after the mask operation are entered to the encoder, and the encoder maps these visible tokens to the latent feature space z. The visible tokens are used as a guideline to manufacture the masked region of the face, which enables the decoder to reconstruct the masked tokens from the latent feature space. In the reconstruction module, the encoder and decoder adopt the identical architecture as the ViT [13]. In order to guide the training of the reconstruction process, the reconstruction loss [27] and adversarial loss [28] are introduced to help the model learn how to accurately reconstruct the masked facial region by combining these two loss functions between the reconstructed cube and the masked cube. Algorithm 2 outlines the training process of the FMAE method. FMAE mainly facilitates model training by introducing reconstruction loss and adversarial loss.

Reconstruction loss. The reconstruction loss aims to minimize the difference between each pixel of the generated image and the original image, helping the model generate results that are more accurate and closer to the target. For the input visible tokens K_v, its visible facial information is utilized by the mask autoencoder in the reconstruction module to reconstruct the masked facial region K'_m. In order to better complete the process, the weight in the mask autoencoder is updated by minimizing the mean square error loss in spatio-temporal facial patterns. Reconstruction loss is defined as follows:

$$\mathcal{L}_r = \frac{1}{N} \sum_{i=1}^{N} \left\| K_m^{(i)} - {K'}_m^{(i)} \right\|_2 \tag{4}$$

where N is the total number of videos in the input dataset S, $K_m^{(i)}$ and ${K'}_m^{(i)}$ are the masked tokens and the reconstructed tokens of the i-th video data in S.

It should be noted that since the reconstruction loss is calculated based on the difference between image pixels, its use may lead to the reduction of the average pixel error of the whole image, which may cause image blurring and information loss. To better perform the reconstruction task, the model is trained by joint adversarial loss.

Adversarial loss. In order to better reconstruct the masked spatio-temporal facial region and learn richer and more effective feature representation, adversarial loss is introduced in the reconstruction module. Adversarial loss is based on the idea of generative adversarial network, by training a discriminator to evaluate the authenticity of the reconstructed image, while training the reconstruction module to deceive the discriminator to the maximum extent. This process of adversarial training can help the reconstruction module learn to generate more realistic and clearer reconstruction results, so as to compensate for the blurring and information loss problems that may be caused by the reconstruction loss. Adversarial loss is defined as follows:

$$\mathcal{L}_a^d = \frac{1}{N} \sum_{i=1}^{N} \left[\log D\left(K_m^{(i)}\right) + \log\left(1 - D\left({K'}_m^{(i)}\right)\right) \right] \tag{5}$$

$$\mathcal{L}_a^g = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 - D\left({K'}_m^{(i)}\right)\right) \tag{6}$$

where $\log D\left(K_m^{(i)}\right)$ is the probability that the discriminator determines the real data (i.e., masked tokens) as real data, and $\log\left(1 - D\left({K'}_m^{(i)}\right)\right)$ is the probability that the discriminator determines the false data (i.e., reconstructed tokens) as false

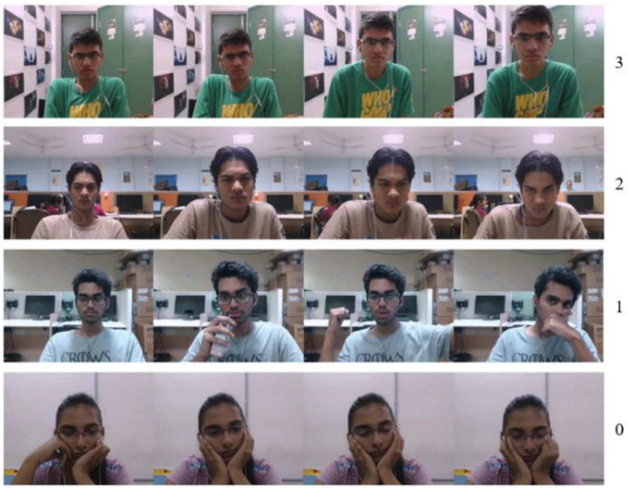


Fig. 5. Examples from the DAiSEE dataset: the first row represents high engagement, the second row represents engagement, the third row represents low engagement, and the fourth row represents disengagement.

data. The overall loss formula in this framework is as follows:

$$\mathcal{L}^g = \mathcal{L}_r + \mathcal{L}_a^g \tag{7}$$

$$\mathcal{L}^d = \mathcal{L}_a^d \tag{8}$$

Thus, better results can be achieved in the reconstruction task by using reconstruction loss and adversarial loss in combination. The reconstruction loss ensures the accuracy and consistency of the reconstructed image, while the adversarial loss provides an additional optimization mechanism to make the reconstructed results clearer and more realistic.

IV. Experiments

In this section, experiments are performed on the DAiSEE and EmotiW datasets to evaluate the proposed model. Before presenting the results, the experimental setup is first described, including the dataset, evaluation metrics, and experimental details. Afterwards, the proposed method is compared with the current SOTA method. Finally, a series of method analysis and ablation studies are carried out on the proposed method.

A. Experimental Settings

1) Dataset:

a) DAiSEE: This dataset consists of videos from 112 online learners, with a total of 9068 video samples. [29]. The videos were labeled according to the four states of the learners when watching the online course, including boredom, confusion, frustration, and engagement. Each state is divided into four levels: level 0(very low), level 1(low), level 2(high), and level 3(very high). The focus of this paper is to classify the degree of student engagement in online learning. The length of each video is 10 seconds, the frame rate is 30 frames per second, and the resolution is 640 × 480 pixels. Fig. 5. shows sample examples of different categories in the dataset. In our experiments, these datasets are used to fairly compare the previous methods with the proposed method. The final results report the performance on 1784 test videos.

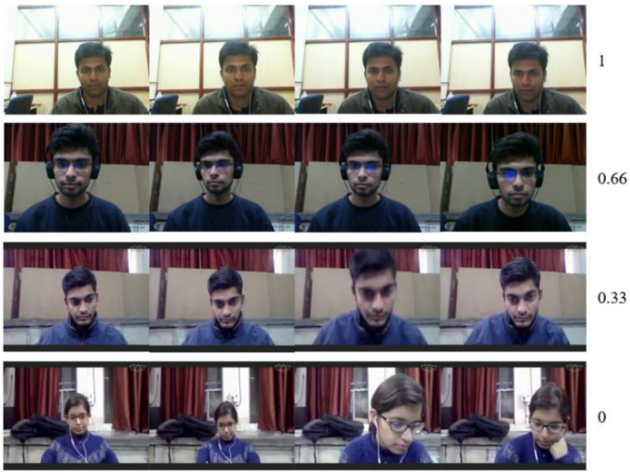


Fig. 6. Examples from the EmotiW dataset: the first row represents high engagement, the second row represents engagement, the third row represents low engagement, and the fourth row represents disengagement.

b) EmotiW: This dataset is provided for measuring student engagement in the sub challenge of EmotiW [30]. The dataset contains videos of 78 people (25 females and 53 males, aged from 19 to 27 years) during online learning. There are 262 videos in total, which include 148 training videos, 48 validation videos and 67 test videos. The videos have a resolution of 640 × 480 pixels, 30 frames per second, and the length of each video is approximately 5 minutes. The level of engagement for every video is divided into four values corresponding to the lowest to highest engagement levels, where 0 indicates disengagement at all, 0.33 indicates low engagement, 0.66 indicates engagement, and 1 indicates high engagement. Fig. 6. presents some examples of samples from different categories in the dataset. In this subchallenge, only the training set and the validation set are publicly available, and we use the validation set to validate the proposed method.

2) Evaluation Metrics:

a) Accuracy: In this work, accuracy [31] is expressed as the number of correctly classified samples divided by the sum of the number of positive (correctly classified) and negative (misclassified) samples of all test samples, expressed by the formula as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

where TP, TN, FP and FN are true positive, true negative, false positive and false negative respectively.

b) MSE: Mean Squared Error (MSE) is a commonly used measure of the difference between the predicted value of the model and the actual observed value, which measures the squared average distance between the true value of the data and the predicted value of the model. MSE is defined as follows:

$$MSE = \frac{1}{B}\sum_{i=1}^{B}\left(y_i - \widehat{y_i'}\right)^2 \tag{10}$$

where B is the number of samples contained in a batch, y_i is the true value of the i-th sample, and y_i' is the predicted value of the i-th sample.

c) Precision: Precision measures the proportion of samples predicted to be positive in a task that are actually positive. The precision is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

where TP and FP are respectively predicted as positive samples and actually are positive samples, while predicted as positive samples are actually negative samples.

d) Recall: The recall rate measures the proportion of samples predicted to be positive out of the actual positive samples in the task. Recall is defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

e) F1-Score: F1-Score take into account both precision and recall, and is used to measure the overall performance of the task. F1-Score are defined as follows:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{13}$$

f) Implementation details: Firstly, for a given facial video, there is a high degree of redundancy between consecutive frames. Therefore, in order to consider meaningful frames in the temporal direction, a minimum temporal stride of 2 is used for frame sampling. Given an input video of size 3×16×224×224, generate 8×14×14 spatio-temporal patterns of size 2 × 16 × 16. When applying a facial mask strategy, FMAE performs mask operations on these spatio-temporal cubes using a predefined masking ratio. Relevant studies have shown that 90% mask is more suitable for model training [7], [42]. The goal of FMAE is to produce masked tokens from fewer visible tokens. After the masking operation, each token is mapped via the encoder to a latent feature space of dimension 768. Based on this latent feature space, the masked facial spatio-temporal cube is reconstructed. ViT-B is used as the encoder backbone. Some of the hyperparameters in the pre-training are as follows: $lr = base\ learningrate \times batch\ size/256$, relative to the overall batch size, the basic learning rate is linearly scaled [32]. AdamW optimizer [33], basic learning rate $1.5e-4$, momentum $\beta_1 = 0.9, \beta_2 = 0.95$ and learning rate scheduler with cosine decay are used for self-supervised pre-training [34]. In regard to fine-tuning process, the linear probing method is used, using Adam optimizer [35], $\beta_1 = 0.5$, $\beta_2 = 0.9$, the base learning rate is $1e-4$ and the weight decays to 0.

B. Performance Comparison

Firstly, FMAE is compared with the most advanced supervised learning methods on DAiSEE and EmotiW datasets in Table I and Table II respectively. On the DAiSEE dataset, FMAE outperforms the previous best methods (i.e., ResNet + TCN and Optimized ShuffleNet v2) with significant accuracy, achieving a noteworthy improvement of 0.84%. As for similar observations on the EmotiW dataset, in this experiment, the last fully connected layer is reshaped to adapt the model to the regression task. The FMAE method also achieves a significant MSE, outperforming most supervised learning methods and achieving comparable performance (0.0112 difference) to the current best method (i.e., MAGRU). The experiments in Table I and Table II show that FMAE can learn powerful facial representation and is well applied to the task of student engagement recognition.

TABLE I
PERFORMANCE COMPARISION ON DAiSEE DATASET

Dataset	Model	Accuracy (%)
DAiSEE	C3D [15]	48.10
	I3D [16]	52.40
	LRCN [36]	57.90
	DFSTN [20]	58.84
	C3D + TCN [21]	59.97
	DERN [37]	60.00
	ResNet + LSTM [21]	61.50
	3D DenseAttNet [17]	63.59
	ResNet + TCN [21]	63.90
	Optimized ShuffleNet v2[38]	63.90
	ours	64.74

TABLE II
PERFORMANCE COMPARISION ON EMOTIW DATASET

Dataset	Model	MSE
EmotiW	Dhall et al. (Baseline) [39]	0.1
	ResNet + TCN [21]	0.096
	C3D [15]	0.0904
	DenseAttNet [17]	0.0877
	Swin-L [40]	0.0813
	I3D [16]	0.0741
	DFSTN [20]	0.0736
	CavT [22]	0.0667
	MAGRU [19]	0.0517
	ours	0.0629

In addition, a comparison of the confusion matrix of FMAE on the DAiSEE dataset and state-of-the-art supervised learning methods is given in Fig. 7. It is worth noting that due to the highly unbalanced distribution of samples in the DAiSEE dataset, the method adopted in Fig. 7. (a)-(d) fails to perform a good classification of disengagement and low engagement samples. Fig. 7. (e)ResNet + TCN method, due to the adoption of customized sampling strategy and weighted loss, some samples of disengagement and low engagement can be correctly classified, but the classification of engagement and high engagement is affected, which makes the overall recognition effect worse. Fig. 7. (f) FMAE method, compared with other methods, the number of samples in the main diagonal of the confusion matrix is significantly increased. This result indicates that FMAE has learned rich and detailed facial representation in the pre-training process, which makes the recognition effect of different student engagement significantly improved.

C. Ablation Study

Extensive ablation studies are conducted on the DAiSEE and EmotiW datasets to show the effectiveness of each component.

1) Masking Ratio: In order to thoroughly investigate the specific impact of different masking ratios on model performance, a series of experiments are carried out on two datasets DAiSEE and EmotiW, which select multiple different values of

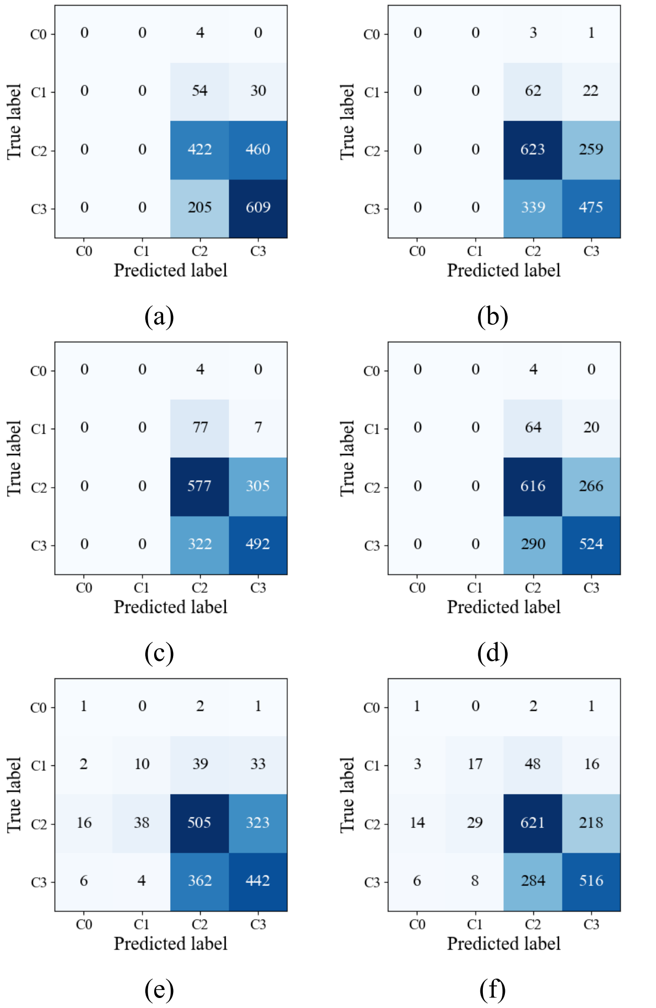


Fig. 7. Confusion matrices of different methods applied on the DAiSEE dataset (a) C3D fine tuning [15], (b) ResNet + LSTM [21], (c) C3D + TCN [21], (d) ResNet + TCN [21], (e) ResNet + TCN with weighted sampling and weighted loss [21], (f)FMAE (ours).

masking ratio in the range of [0.25,0.95]. The purpose of this experiment is to provide a clear understanding of the effect of masking ratio on experimental performance to enhance feature understanding while maintaining the efficiency of the reconstruction task. As can be seen from Fig. 8, a masking ratio of 90% is the optimal masking ratio for FMAE. When the masking ratio is smaller, the reconstruction task is able to obtain more information, which reduces the ability of feature understanding. If the masking ratio is set too large, especially above 90%, the reconstruction task becomes extremely challenging, resulting in a lack of sufficient information to complete accurate reconstruction and learn detailed features, which makes the overall performance degrade. Therefore, after experimental verification, 90% is consistently chosen as the optimal masking ratio in all experiments to ensure that sufficient information could be obtained while maintaining high feature comprehension in the reconstruction task.

2) *Minimum Temporal Stride:* The exploration is conducted on the DAiSEE dataset to evaluate the performance of student engagement recognition tasks under five different settings of minimum temporal sampling strides. This experiment delved into the delicate balance between recognition accuracy and computational complexity, aiming to determine the optimal minimum temporal sampling stride. According to the trade-off relationship between recognition accuracy and computational complexity under various stride size settings in Fig. 9, it can be found that when the minimum sampling stride size is set to 2, the accuracy of student engagement recognition only shows a slight decrease compared with the more refined stride size, but the computational complexity is significantly reduced at this time. This indicates that the minimum sampling stride size of 2 is the most reasonable choice, which not only ensures the accuracy of identification, but also has a more efficient performance.

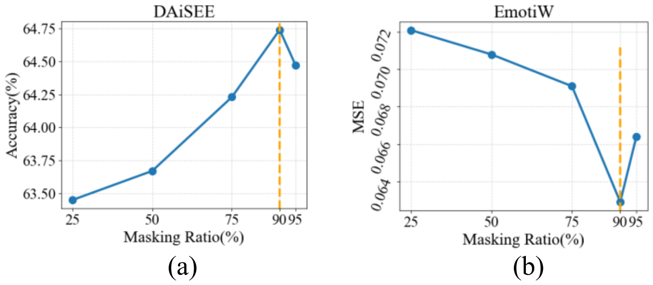


Fig. 8. Impact of Masking Ratio. Comparing the impact of different masking ratios on engagement recognition in the DAiSEE and EmotiW datasets.

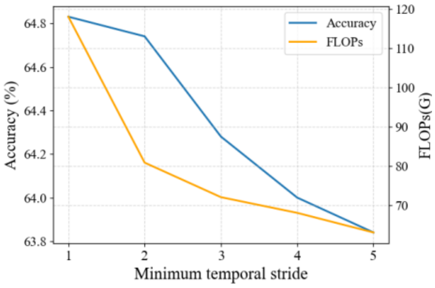


Fig. 9. Effect of the minimum time sampling stride.

3) *Masking Strategies:* This ablation experiment aims to compare the differences between the proposed facial mask strategy and the existing mask strategies, namely frame mask, random mask and tube mask strategies. The frame masking strategy [41] is to partially mask the frame data, and the random masking strategy [42] is to randomly select some image regions for masking. The experiments in Table III show that these two methods are relatively less effective. This is due to the inherent relationship between adjacent frames, which is not processed by the two masking methods, resulting in information leakage in the mask and reconstruction process, which ultimately affects the accuracy of experimental results. Although the tube masking strategy [7] masks the same region of different frames, the mask region of this strategy is randomly selected. In contrast to the proposed face masking strategy, it does not select the regions in the face that contain important information for masking. Therefore, the experimental results of the tube mask are also slightly inferior.

TABLE III

ABLATION STUDIES OF DIFFERENT MODULES. FS: FACIAL STRATEGY, RL: RECONSTRUCTION LOSS, AL: ADVERSARIAL LOSS

Masking Strategy	DAiSEE				EmotiW
	Accuracy	Precision	Recall	F1-Score	MSE
Random	59.17	0.57	0.58	0.58	0.0748
Frame	57.51	0.56	0.57	0.56	0.0783
Tube	62.98	0.60	0.62	0.61	0.0671
Facial	64.74	0.63	0.65	0.64	0.0629

TABLE IV

ABLATION STUDIES OF MASKING STRATEGIES

Modules	DAiSEE				EmotiW
	Accuracy	Precision	Recall	F1-Score	MSE
FMAE	64.74	0.63	0.65	0.64	0.0629
w/o FS	62.98	0.60	0.62	0.61	0.0671
w/o RL	63.64	0.60	0.63	0.61	0.0668
w/o AL	63.79	0.61	0.64	0.62	0.0664
w/o FS&RL	60.83	0.56	0.59	0.58	0.0769
w/o FS&AL	61.69	0.58	0.61	0.59	0.0758

TABLE V

ABLATION STUDIES OF THE INTERPLAY EFFECTS BETWEEN MASKING STRATEGIES AND VARIOUS MODULES. ACC.: ACCURACY, PRE.: PRECISION, REC.: RECALL, F1: F1-SCORE

Masking Strategy	Modules		DAiSEE				EmotiW
	RL	AL	Acc.	Pre.	Rec.	F1	MSE
Random	✓		58.26	0.56	0.58	0.57	0.0792
		✓	57.72	0.56	0.56	0.56	0.0802
	✓	✓	59.17	0.57	0.58	0.58	0.0748
Frame	✓		57.42	0.54	0.56	0.55	0.0825
		✓	57.36	0.53	0.56	0.55	0.0833
	✓	✓	57.51	0.56	0.57	0.56	0.0783
Tube	✓		61.69	0.58	0.61	0.59	0.0758
		✓	60.83	0.56	0.59	0.58	0.0769
	✓	✓	62.98	0.60	0.62	0.61	0.0671
Facial	✓		63.79	0.61	0.64	0.62	0.0664
		✓	63.64	0.60	0.63	0.61	0.0668
	✓	✓	64.74	0.63	0.65	0.64	0.0629

4) Different Modules: Keep other components fixed and remove some modules from the framework to verify the validity of the corresponding modules. Firstly, the validity of the facial masking strategy is verified. From Table IV, it can be observed that the accuracy of the model decreases significantly when the facial masking strategy is not used. This is mainly attributed to the fact that this strategy masks the regions containing important information in the face, forcing the model to learn these important facial features, so it makes a significant contribution to improving the accuracy of the model. Secondly, the reconstruction loss and adversarial loss in the model are removed respectively to verify the effectiveness of their separate effects. According to the experimental results in Table IV, when the reconstruction loss is not used, the accuracy of the model decreases by 1.1%. This may be because in the absence of fine-grained control of reconstruction loss, only the use of adversarial loss cannot guide the reconstruction process of the model at a more detailed level, thus affecting the performance of the model. Similarly, the accuracy of the model is also reduced when adversarial loss is not used. This indicates that adversarial loss plays an important role in model training, which enables the model to learn richer and more detailed facial representations through adversarial optimization mechanism. When the reconstruction loss and adversarial loss are combined in FMAE and applied to the training process of the model, the model shows the best performance. This indicates that the combination strategy of the two loss functions can better guide the model to accurately reconstruct the masked facial region, so as to improve the performance of the model. Finally, the facial masking strategy and different loss modules in the model are removed at the same time, so as to verify the effectiveness of the two modules acting simultaneously. According to the accuracy results shown in Table IV, the accuracy is greatly reduced and it can be concluded that the best results can only be obtained when all modules work together.

5) Interplay Effects: In this part, more in-depth ablation experiments are carried out to further verify the interaction between the masking strategy and the combination of different modules. In this ablation study, four masking strategies are combined with different loss functions in the reconstruction module through exhaustive experiments. From the experimental data in Table V, the following observations can be obtained: 1) When the same masking strategy is used, the experimental data show that the combined reconstruction loss and adversarial loss methods are generally superior in performance to those relying on a single loss function. Specifically, the performance improvement achieved by this combination strategy is up to 2.15% in accuracy, which further confirms that the combined design of loss functions in the reconstruction module has a key effect on the performance. 2) When using the same loss function, the combined approach with the facial masking strategy showed a significant performance improvement. This achievement is mainly attributed to the unique design of the facial masking strategy, which enables the model to focus more on the information of key facial regions. This result is consistent with the ablation results of masking strategy, which further confirms the important role of facial masking strategy in the optimization of model performance. 3) The combination scheme using the facial masking strategy and the designed reconstruction module (including reconstruction loss and adversarial loss) achieves the best results in the experiment. This result demonstrates that the design of FMAE is superior in the acquisition of facial key information and efficient reconstruction, and is the best combination scheme.

D. Visualization Analysis

In order to further verify the effectiveness of FMAE method, the facial parsing process and the learned facial feature map are visualized. As shown in Fig. 10., the level of engagement decreases from left to right. From the visualization results of facial parsing in the second line, it can be noticed that the method we adopted can effectively divide the facial region and distinguish areas such as glasses and hand occlusion, which provides strong support for the subsequent mask operation. In the third line of Fig. 10., the results of a visualization experiment with Grad-CAM [43] are shown, from which it

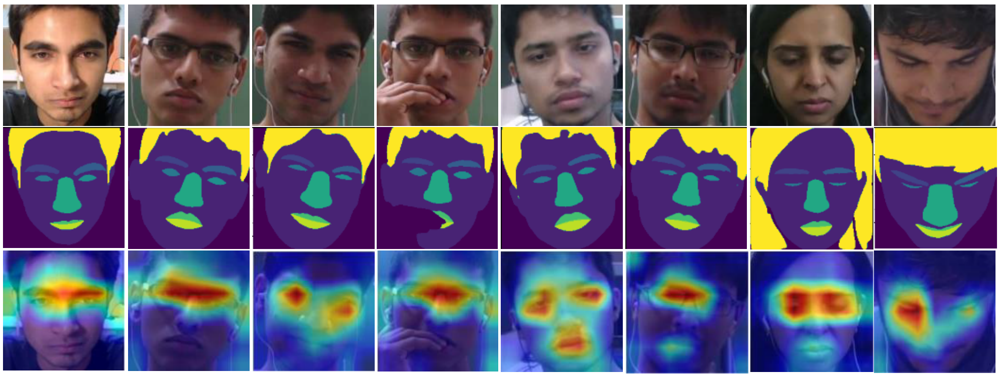


Fig. 10. Facial parsing and Grad - CAM visualization. The facial frame images obtained by cropping and other processing in the original video in the first row. The second row of face parsing results is visualized. The third row is visualized by Grad-CAM.

can be observed that the FMAE method is able to learn meaningful face representations and pay attention to key face areas. In addition, since the non-frontal poses also contain rich facial information related to student engagement, it can be noted in the third row that even in the presence of occlusion and non-frontal poses, the method is still able to focus on key areas such as eyes, thus obtaining a good facial representation.

E. Extended Experiments

FMAE learns rich and detailed facial representation from facial videos through self-supervised learning. In order to verify its ability to understand facial features, it is applied to facial expression recognition (FER) [44] task for further verification. Facial expression recognition refers to the classification of facial expressions by analyzing people's facial features. In order to evaluate the performance of FMAE in facial expression recognition task, experiments were conducted on AFEW [39], CREMA-D [45], and RAVDESS [46] datasets. The AFEW dataset is a collection of audio-video short clips gathered from movies and television series, consisting of 773 training samples and 383 validation samples. The dataset contains seven basic expressions (anger, disgust, fear, happiness, sadness, surprise, and neutral), and each video clip is labeled with a single expression category. CREMA-D is a high-quality audio-visual dataset consisting of 7442 video clips from 91 actors. The dataset covers six expressions, which are happy, sad, angry, fear, disgust, and neutral. RAVDESS is an audio-visual dataset of emotional speech and song. It consists of 2880 video clips from 24 professional actors with eight expressions (i.e., 7 basic expressions and calmness).

In the Table VI, the performance of FMAE in facial expression recognition task is compared in terms of accuracy. The results show that FMAE exhibits considerable competitiveness compared to SOTA methods. Specifically, on the AFEW dataset, FMAE achieved an accuracy of 62.71%, representing a performance improvement of 3.29% compared to the previous optimal model. This result indicates that FMAE is also competent for facial expression recognition tasks. Similarly, FMAE also shows considerable accuracy on both the CREMA-D and RAVDESS datasets. Although it is slightly weaker than the method of MSAF [60], CFN-SR [61] and Sinha et al. [54], this may be due to the fact that these methods leverage multimodal information from the data and provide more comprehensive and abundant decision information for facial expression recognition through other modalities, which makes these methods achieve more superior performance. Compared with these methods, FMAE still demonstrates good performance without using audio information. Through these results, it can be shown that FMAE learns robust, comprehensive and accurate facial features through the self-supervised way of mask autoencoder, which makes it competent for other face-related tasks.

TABLE VI
EXTENDED RESEARCH ON FACIAL EXPRESSION RECOGNITION TASKS

Dataset	Model	Accuracy (%)
AFEW	LBP-TOP (baseline) [47]	38.90
	Meng et al. [48]	51.18
	Kumar et al. [49]	55.17
	Li et al. [50]	54.30
	VGG13+VGG16+ResNet [51]	59.42
	ours	62.71
CREMA-D	Vougioukas et al. [52]	55.26
	Eskimez et al. [53]	65.67
	Sinha et al. [54]	75.02
	GRU [55]	55.01
	GAN [56]	58.71
	ViT [57]	67.81
	SepTr [58]	70.47
	ours	71.58
RAVDESS	AV-LSTM [59]	65.80
	MCBP [60]	71.32
	MSAF [60]	74.86
	CFN-SR [61]	75.76
	MMTM [62]	73.12
	ERANNs [63]	74.80
	ours	74.83

V. CONCLUSION

FMAE is an efficient self-supervised learning model that uses a large number of unlabeled face videos to cope with the dilemma of supervised learning methods in the task of student engagement recognition and promote its development. The model introduces two key designs, the facial masking strategy and the reconstruction module, to make the video reconstruction task more challenging, thus encouraging the model to learn more representative features. However, because the model adopts ViT architecture as the encoder, the number of parameters of the model is large, which is not conducive to deployment in lightweight environment. The future research direction can focus on optimizing the parameter number of the model in order to provide the inference speed of the model and save computing resources. This can be achieved by compressing the model [64], model pruning [65], or designing more efficient architectures. In the process of promoting the development of student engagement recognition, privacy and ethical issues should also be taken seriously [66]. Future research should focus on how to identify student engagement in online learning while protecting student privacy.

References

[1] Y. Cui et al., "A survey on big data-enabled innovative online education systems during the COVID-19 pandemic," *J. Innov. Knowl.*, vol. 8, no. 1, Jan. 2023, Art. no. 100295.

[2] S. K. Banihashem, O. Noroozi, P. den Brok, H. J. A. Biemans, and N. T. Kerman, "Modeling teachers' and students' attitudes, emotions, and perceptions in blended education: Towards post-pandemic education," *Int. J. Manage. Educ.*, vol. 21, no. 2, Jul. 2023, Art. no. 100803.

[3] Y. Wang et al., "FERV39k: A large-scale multi-scene dataset for facial expression recognition in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20890–20899.

[4] K. He et al., "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2022, pp. 16000–16009.

[5] L. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, vol. 1, 2019, p. 2.

[6] K. Fu, M. Yuan, S. Liu, and M. Wang, "Boosting point-BERT by multichoice tokens," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 1, pp. 438–447, Aug. 2023.

[7] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 10078–10093.

[8] M. Chen et al., "Generative pretraining from pixels," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1691–1703.

[9] L. Zhang, X. Zhang, Q. Wang, W. Wu, X. Chang, and J. Liu, "RPMG-FSS: Robust prior mask guided few-shot semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6609–6621, Nov. 2023.

[10] S. Goley, R. Pradhan, and A. Welch, "Towards masked autoencoding pre-training for wide area motion imagery," in *Proc. Geospatial Informat. XIII*, Jun. 2023, pp. 51–64.

[11] P. Gao et al., "Mimic before reconstruct: Enhancing masked autoencoders with feature mimicking," *Int. J. Comput. Vis.*, vol. 132, no. 5, pp. 1546–1556, May 2024.

[12] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14648–14658.

[13] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[14] X. Sun et al., "Masked motion encoding for self-supervised video representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2235–2245.

[15] L. Geng, M. Xu, Z. Wei, and X. Zhou, "Learning deep spatiotemporal feature for engagement recognition of online courses," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 442–447.

[16] H. Zhang, X. Xiao, T. Huang, S. Liu, Y. Xia, and J. Li, "An novel end-to-end network for automatic student engagement recognition," in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2019, pp. 342–345.

[17] N. K. Mehta, S. S. Prasad, S. Saurav, R. Saini, and S. Singh, "Three-dimensional DenseNet self-attention neural network for automatic detection of student's engagement," *Appl. Intell.*, vol. 52, no. 12, pp. 13803–13823, 2022.

[18] J. Wu, B. Yang, Y. Wang, and G. Hattori, "Advanced multi-instance learning method with multi-features engineering and conservative optimization for engagement intensity prediction," in *Proc. Int. Conf. Multimodal Interact.*, Oct. 2020, pp. 777–783.

[19] B. Zhu, X. Lan, X. Guo, K. E. Barner, and C. Boncelet, "Multi-rate attention based GRU model for engagement prediction," in *Proc. Int. Conf. Multimodal Interact.*, Oct. 2020, pp. 841–848.

[20] J. Liao, Y. Liang, and J. Pan, "Deep facial spatiotemporal network for engagement prediction in online learning," *Appl. Intell.*, vol. 51, pp. 6609–6621, Oct. 2021.

[21] A. Abedi and S. S. Khan, "Improving state-of-the-art in detecting student engagement with resnet and TCN hybrid network," in *Proc. 18th Conf. Robots Vis. (CRV)*, May 2021, pp. 151–157.

[22] X. Ai, V. S. Sheng, C. Li, and Z. Cui, "Class-attention video transformer for engagement intensity prediction," 2022, *arXiv:2208.07216*.

[23] L. Qin et al., "SwinFace: A multi-task transformer for face recognition, expression recognition, age estimation and attribute estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2223–2234, Apr. 2024.

[24] M. Assran et al., "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15619–15629.

[25] H. Zhu, Y. Chen, G. Hu, and S. Yu, "Information-density masking strategy for masked image modeling," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 1619–1624.

[26] C. Chen, X. Li, L. Yang, X. Lin, L. Zhang, and K. K. Wong, "Progressive semantic-aware style transformation for blind face restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11891–11900.

[27] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by in painting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2544, pp. 2536–2544.

[28] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–12.

[29] A. Gupta, A. D'Cunha, K. Awasthi, and V. Balasubramanian, "DAiSEE: Towards user engagement recognition in the wild," 2016, *arXiv:1609.0188*.

[30] A. Kaur, A. Mustafa, L. Mehta, and A. Dhall, "Prediction and localization of student engagement in the wild," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2018, pp. 1–8.

[31] S. Batra et al., "DMCNet: Diversified model combination network for understanding engagement from video screengrabs," *Syst. Soft Comput.*, vol. 4, Dec. 2022, Art. no. 200039.

[32] P. Goyal et al., "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[34] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[36] J. Donahue et al., "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[37] T. Huang, Y. Mei, H. Zhang, S. Liu, and H. Yang, "Fine-grained engagement recognition in online learning environment," in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2019, pp. 338–341.

[38] Y. Hu, Z. Jiang, and K. Zhu, "An optimized CNN model for engagement recognition in an E-Learning environment," *Appl. Sci.*, vol. 12, no. 16, p. 8007, Aug. 2022.

[39] A. Dhall, "EmotiW 2019: Automatic emotion, engagement and cohesion prediction tasks," in *Proc. Int. Conf. Multimodal Interact.*, Oct. 2019, pp. 546–550.

[40] Z. Liu et al., "Video Swin transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3192–3201.

[41] Y. Lee, H. Seong, and E. Kim, "Iteratively selecting an easy reference frame makes unsupervised video object segmentation easier," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 2, pp. 1245–1253.

[42] C. Feichtenhofer, Y. Li, and K. He, "Masked autoencoders as spatiotemporal learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 35946–35958.

[43] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[44] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1195–1215, Jul. 2022.

[45] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, “CREMA-D: crowd-sourced emotional multimodal actors dataset,” *IEEE Trans. Affect. Comput.*, vol. 5, no. 4, pp. 377–390, Oct. 2014.

[46] S. R. Livingstone and F. A. Russo, “The ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in north American English,” *PLoS ONE*, vol. 13, no. 5, May 2018, Art. no. e0196391.

[47] T. S. Ashwin, and R. M. R. Guddet, “Affective database for e-learning and classroom environments using Indian students’ faces, hand gestures and body postures,” *Future Gener. Comput. Syst.*, vol. 108, pp. 334–348, Jul. 2020.

[48] D. Meng, X. Peng, K. Wang, and Y. Qiao, “Frame attention networks for facial expression recognition in videos,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3866–3870.

[49] V. Kumar, S. Rao, and L. Yu, “Noisy student training using body language dataset improves facial expression recognition,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 756–773.

[50] S. Li et al., “Bi-modality fusion for emotion recognition in the wild,” in *Proc. Int. Conf. Multimodal Interact.*, Oct. 2019, pp. 589–594.

[51] S. A. Bargal, E. Barsoum, C. C. Ferrer, and C. Zhang, “Emotion recognition in the wild from videos using images,” in *Proc. 18th ACM Int. Conf. Multimodal Interact.*, Oct. 2016, pp. 433–436.

[52] K. Vougioukas, S. Petridis, and M. Pantic, “Realistic speech-driven facial animation with GANs,” *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1398–1413, May 2020.

[53] S. E. Eskimez, Y. Zhang, and Z. Duan, “Speech driven talking face generation from a single image and an emotion condition,” *IEEE Trans. Multimedia*, vol. 24, pp. 3480–3490, 2022.

[54] S. Sinha, S. Biswas, R. Yadav, and B. Bhowmick, “Emotion-controllable generalized talking face generation,” 2022, *arXiv:2205.01155*.

[55] A. Shukla, K. Vougioukas, P. Ma, S. Petridis, and M. Pantic, “Visually guided self supervised learning of speech representations,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6299–6303.

[56] G. He, X. Liu, F. Fan, and J. You, “Image2Audio: Facilitating semi-supervised audio emotion recognition with facial expression image,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3978–3983.

[57] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio spectrogram transformer,” 2021, *arXiv:2104.01778*.

[58] N.-C. Ristea, R. T. Ionescu, and F. S. Khan, “SepTr: Separable transformer for audio spectrogram processing,” 2022, *arXiv:2203.09581*.

[59] E. Ghaleb, M. Popa, and S. Asteriadis, “Multimodal and temporal perception of audio-visual cues for emotion recognition,” in *Proc. 8th Int. Conf. Affect. Comput. Intell. Interact. (ACII)*, Sep. 2019, pp. 552–558.

[60] L. Su, C. Hu, G. Li, and D. Cao, “MSAF: Multimodal split attention fusion,” 2020, *arXiv:2012.07175*.

[61] Z. Fu et al., “A cross-modal fusion network based on self-attention and residual structure for multimodal emotion recognition,” 2021, *arXiv:2111.02172*.

[62] H. R. V. Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida, “MMTM: Multimodal transfer module for CNN fusion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13286–13296.

[63] S. Verbitskiy, V. Berikov, and V. Vyshegorodtsev, “ERANNs: Efficient residual audio neural networks for audio pattern recognition,” *Pattern Recognit. Lett.*, vol. 161, pp. 38–44, Sep. 2022.

[64] T. Choudhary, V. Mishra, A. Goswami, and J. Sarangapani, “A comprehensive survey on model compression and acceleration,” *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5113–5155, Oct. 2020.

[65] A. Onan, S. Korukoğlu, and H. Bulut, “A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification,” *Inf. Process. Manage.*, vol. 53, no. 4, pp. 814–833, Jul. 2017.

[66] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. H. Deng, “Privacy-preserving federated deep learning with irregular users,” *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1364–1381, Mar. 2022.

[67] D. Chen, G. Wen, H. Li, R. Chen, and C. Li, “Multi-relations aware network for in-the-wild facial expression recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3848–3859, Sep. 2023.

Wen-Long Zhang was born in Shandong, China, in 1999. He received the B.S. degree from Qingdao University of Science and Technology, China, in 2022. He is currently pursuing the M.S. degree with Shandong University of Science and Technology. His research interests include image processing and deep learning.

Rui-Sheng Jia is currently a Full Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, China, and the Leader of the Natural Science Foundation of Shandong Province, China. He has more than 30 first-author publications and has more than 50 co-author publications. His research interests include artificial intelligence, computer vision, information fusion, microseismic monitoring, and inversion.

Hu Wang was born in Anhui, China, in 1999. He received the B.S. degree from Qingdao University of Technology, China, in 2022. He is currently pursuing the M.S. degree with Shandong University of Science and Technology. His research interests include image processing and deep learning.

Cheng-Yue Che was born in Shandong, China, in 2001. She received the B.S. degree from Shandong University of Science and Technology, China, in 2023, where she is currently pursuing the M.S. degree. Her research interests include image processing and deep learning.

Hong-Mei Sun received the B.S. and M.S. degrees in computer science from Shandong University of Science and Technology, China, in 1995 and 2005, respectively. She is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, and the Leader of the Key Research and Development Projects of Shandong Province, China. She has more than 20 first-author publications and has more than 50 co-author publications. Her research interests include computer vision, deep learning, and software engineering.

NT-Net: A Semantic Segmentation Network for Extracting Lake Water Bodies From Optical Remote Sensing Images Based on Transformer

Hai-Feng Zhong, Qing Sun, Hong-Mei Sun, and Rui-Sheng Jia

Abstract—**The automatic extraction of lake water is one of the research hotspots in the field of remote sensing image processing. Due to the small interclass variance between lakes and other ground objects, and the complex texture characteristics of lake boundaries, existing methods often have problems such as over-segmentation and inaccurate boundary segmentation when segmenting lake water bodies. To alleviate these problems, this article designs an end-to-end semantic segmentation network [noise-canceling transformer network (NT-Net)] for the automatic extraction of lake water bodies from remote sensing images. Aiming at the problem of over-segmentation caused by nonlake objects, an interference attenuation module is designed in the network. This module can model the key features that are distinguishable and suitable for segmenting lake water by analyzing the difference in feature representation between lakes and other ground objects, thus suppressing the feature representation of nonlake objects. To more accurately segment the lake boundary, a multilevel transformer module is designed. This module can capture the context association of boundary information and enhance the feature representation of boundary information by using the self-attention mechanism. The comparative experimental results show that, compared with the current mainstream semantic segmentation networks, the method in this article has advantages in extracting lake water bodies comprehensively and coherently.**

Index Terms—**Convolutional neural network (CNN), lake segmentation, optical remote sensing images, semantic segmentation, transformer.**

I. Introduction

THE rapid development of remote sensing technology has provided finer and higher resolution optical remote sensing images [1]. Segmenting lake water bodies from optical remote sensing images can be widely used in water area change detection, water resource protection, global climate change research, and flood disaster emergency response [2], [3], [4], [5]. Therefore, how accurately extracting lake water bodies in remote sensing images is the primary task in exploring its application value. The current automatic extraction of lake water faces the following challenges.

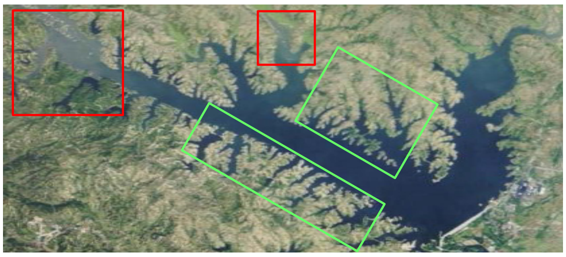


Fig. 1. Remote sensing image of lake water bodies. The red boxes represent disturbed areas similar to lake water bodies. Green boxes represent tiny tributaries at the boundary of the lake water bodies.

1) There are many surface objects similar to lake water bodies in remote sensing images, such as grass, jungle, and silt beside the lake. As shown by the red box in Fig. 1, the lake water body at the boundary exhibits similar spectral characteristics to the surrounding soil due to its shallow depth. This similarity between nonuniform objects results in a small interclass variance between lake water bodies and other ground objects.
2) In addition to the over-segmentation problem, how to accurately segment the boundaries of lake water bodies is also a difficult problem to solve. For example, as shown in the green box in Fig. 1, under the impact of lake water on the soil on the shore, a large number of capillary lake water bodies will appear at the boundary. Since these lake water bodies are too small in scale and do not have obvious texture features, they are difficult to identify.

In response to the above problems, this article designs a noise-canceling transformer network (NT-Net) to use the image's local and global information to achieve accurate extraction of lakes. NT-Net adopts an asymmetric "encode–decode" structure as a shared backend. Specifically, to address the over-segmentation problem caused by nonlake water objects, an interference attenuation module (IAM) is designed. The IAM can not only use the positional relationship of features to model the differential features between lake water and other objects, but also adaptively give more feature weights

Manuscript received 6 June 2022; revised 12 July 2022; accepted 27 July 2022. Date of publication 8 August 2022; date of current version 19 August 2022. This work was supported by the Humanities and Social Science Fund of the Ministry of Education of the People's Republic of China under Grant 21YJAZH077. *(Corresponding authors: Hong-Mei Sun; Rui-Sheng Jia.)*

Hai-Feng Zhong, Hong-Mei Sun, and Rui-Sheng Jia are with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China (e-mail: shm0221@163.com; jrs716@163.com).

Qing Sun is with the Dongfang College, Shandong University of Finance and Economics, Taian 271000, China.

Digital Object Identifier 10.1109/TGRS.2022.3197402

to the lake water by using the dependencies between feature channels. To improve the segmentation accuracy (Acc) of the lake boundary degree, the multilevel transformer (MT) module is designed. On the one hand, the MT can integrate multiscale contextual information about boundary features to preserve the integrity and continuity of boundary contours by modeling long-range semantic dependencies of boundary information. On the other hand, this module can use high-level semantic features to assist the modeling process of low-level texture features, thus mapping out more accurate boundary details. Unlike existing transformers, the MT can model the feature output of each downsampling stage, thus learning multiscale features that preserve spatial localization and optimize discontinuous lake boundaries. The main contributions of this article are as follows.

1) We designed a new semantic segmentation network NT-Net based on convolutional neural network (CNN) and transformer. NT-Net makes full use of CNN's ability to capture local detail information and transformer's ability to model long-range information, effectively improving the segmentation Acc of lake water bodies.
2) An IAM is designed in NT-Net, which effectively alleviates the problem of over-segmentation caused by nonlake objects by deeply mining the feature differences between lake water bodies and other ground objects.
3) An MT module that can focus on building global dependencies is designed. This module can perform multilevel global self-attention modeling on feature information to accurately express the continuous positional relationship between boundary information.

The remainder of this article is organized as follows. Section II discusses the previous work on lake segmentation. Section III introduces the detailed structure of the NT-Net. In Section IV, we illustrate the implementation steps of the experiment and analyze the experimental results in detail. Section V discusses the rationality of some parameters in NT-Net. Finally, Section VI summarizes our conclusions.

II. Related Work

Traditional lake segmentation methods generally use threshold segmentation [6], clustering [7], and support vector machines [8] to extract lake water bodies in remote sensing images. For example, McFeeters [9] proposed a lake water segmentation method based on the normalized difference water index (NDWI) by using the band reflection of surface objects and achieved relatively ideal segmentation results. Obida *et al.* [10] proposed a segmentation method of river water using K-means clustering technique. This method divides the river area and the background area based on the shallow features of the lake water bodies. However, the above methods cannot solve the problem of vegetation and soil noise because it is difficult to choose an appropriate threshold when the lake and background are mixed in pixels. To solve noise interference in the segmentation process, Wu *et al.* [11] proposed a lake segmentation method based on support vector machines. This method builds a mathematical model for noise reduction by analyzing the characteristics of noise, thus reducing the negative impact of noise information. Although the Acc of traditional methods in extracting lake water bodies continues to improve, most of them use low-level features such as the color, shape, and texture of lake water bodies to complete the segmentation process. Since the deep features of lake water cannot be used to refine the segmentation results, the segmentation Acc is low.

CNNs have been shown to be effective in capturing both shallow detailed features and deep abstract features in images [12]. To make full use of the deep abstract features, researchers try to apply CNN to the segmentation task of lake water bodies to accurately segment lakes. For example, Chen [13] used CNN to segment water bodies in cases such as narrow lakes and wide lakes, and demonstrated that deep learning methods were more accurate than traditional methods when segmenting lakes in images. However, this method cannot perceive global contextual information, so the segmentation results have a low degree of continuity. To improve the perception of global information to achieve dense prediction of lake pixels, Hu *et al.* [14] designed a multibranch aggregation module using dilated convolutions. The module uses dilated convolutions with different dilation rates to sample the given input, so that the network can establish the correspondence between local pixels and global pixels, thus protecting the integrity of lake water. Network models that use similar structures include MSLWE-Net [15] and PA-U-Net [16]. Although the above methods can better perceive the context distribution of image information, they cannot accurately locate the boundary information of lake water. To segment the lake boundary more accurately, some methods try to use the attention mechanism [17], [18] in the network model to give more attention to the boundary information adaptively. For example, MEC-Net [19], DAU-Net [20], FWE-Net [21], and MIE-Net [22] all use compression and excitation operations to redistribute feature weights for ground information, thus focusing more on feature channels containing boundary information. In addition, some methods [23], [24] also try to use loss functions that can refine the boundary modeling process. For example, Miao *et al.* [25] used the edge-weighted loss to assign more weights to the pixels near the boundary of the lake water body, so that the network could pay more attention to the boundary information during the decoding process, thus improving the segmentation Acc of the boundary.

The above methods only capture the detailed features of the boundary from the local scope and do not use the contextual information of the boundary to improve the coherence of the contour. Inspired by the great success of Visio transformer (ViT) [26] in the field of image classification, some researchers have tried to exploit the self-attention mechanism [27], [28], [29] to obtain contextual associations of boundary features. For example, BA-Net [30] utilizes a transformer similar to ViT to capture informative long-range relationships and obtain more coherent boundary results. However, transformer involves complex matrix operations, so the computational complexity of BA-Net is high. In order to improve the operating efficiency of the algorithm, Gao *et al.* [31] designed STransFuse using Swin transformer [32]. STransFuse shifts

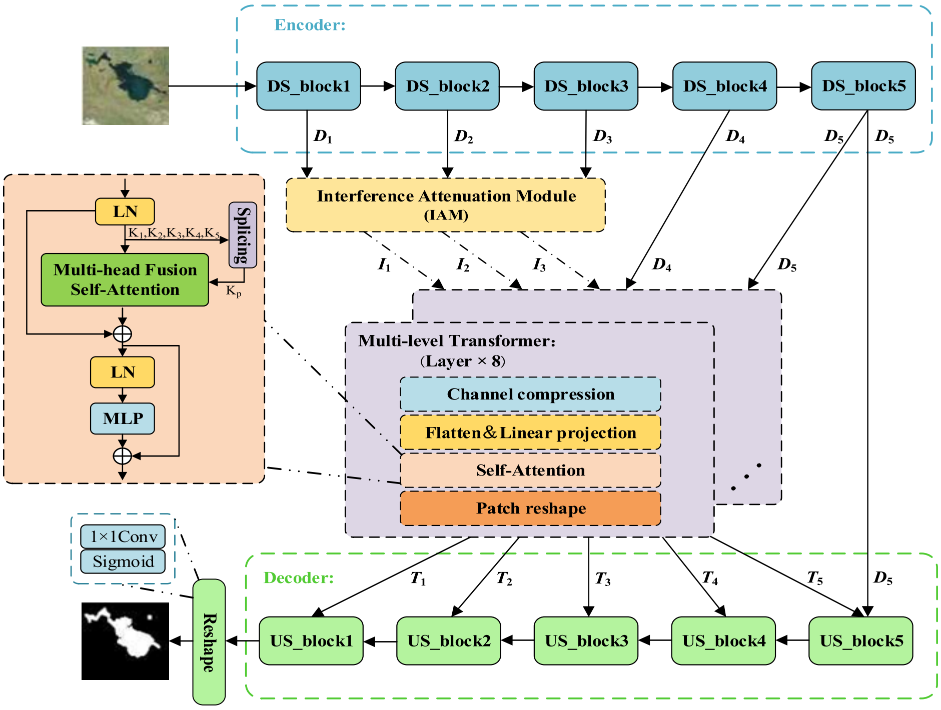


Fig. 2. Overview of NT-Net. The workflow of NT-Net is divided into two parts: encoding and self-attention process, and decoding process.

the focus of self-attention from global information to local information within each patch, thus reducing the computational complexity of the model. Xu *et al.* [33] proposed a network model for remote sensing image segmentation by transforming the backbone structure of the Swin transformer. The network uses depthwise separable convolutions [34] and a more flexible image window, which further reduces the computational complexity of the model. Although the above methods can reproduce the boundary contours of lakes well, these methods cannot identify the detailed texture of lake boundaries due to the weak inductive bias ability of the transformer. Inspired by the above work, we use CNN and transformer as the network's backbone to design a network model that can accurately extract lake water.

III. NT-Net

A. NT-Net Semantic Segmentation Network

To solve the problem of over-segmentation of lake water and accurately segment the boundary of lakes, this article designs a deep neural network NT-Net based on semantic segmentation. The network is mainly composed of encoder, decoder, IAM, and MT module, and its structure is shown in Fig. 2.

1) Encoding and Self-Attention Process: The encoder processes the input remote sensing images $I \in \mathbb{R}^{H\times W\times C}$ with successive downsampling blocks (DS-blocks) to generate a multiscale feature stream $D_i(i = 1, 2, 3, 4, 5)$. Before D_i is input into the MT for self-attention processing, D_1, D_2, and D_3 need to be sent to IAM to reduce the negative impact of nonlake water objects on the segmentation process and generate feature streams I_1, I_2, and I_3. Considering that most of the interference generated by nonlake water objects exists in the low-level feature maps, only D_1, D_2, and D_3 are processed by IAM here. After that, the MT performs feature compression, flattening, and serialization on the input I_1, I_2, I_3, D_4, and D_5 to generate feature sequences for the self-attention process. Then, the multihead fusion attention mechanism (MF-SA) processes these feature sequences, thus completing the global self-attention process for features at different levels.

The reshaped feature stream $T_i(i = 1, 2, 3, 4, 5)$ can be sent to the decoder. The decoder uses the upsampling block (US_block) to upsample T_i to restore the detailed features of the lake water gradually. Finally, the binary segmentation image of the lake water bodies is output after sigmoid processing.

2) Decoding Process: First, it is necessary to reshape the feature sequence generated by the self-attention process. This is because the output of the self-attention process is a 1-D feature sequence, which cannot be directly upsampled. The patch reshape operation is required to restore the dimension of the feature sequence to a 2-D size.

B. Encoder

In NT-Net, the function of the encoder is to use the DS-block to extract the feature information of multiple scales from the remote sensing image for the self-attention process of the MT. The structure of the DS-block is shown in Fig. 3.

DS-block consists of a convolutional layer and a pooling layer, whose input feature stream is $\text{FE}_i \in \mathbb{R}^{H\times W\times C_n}$ and the output feature stream is $\text{FE}_{i+1} \in \mathbb{R}^{(H/2)\times(W/2)\times C_m}$, where $n < m$. The convolution layer draws on the idea of residual structure and improves the information diversity in the process

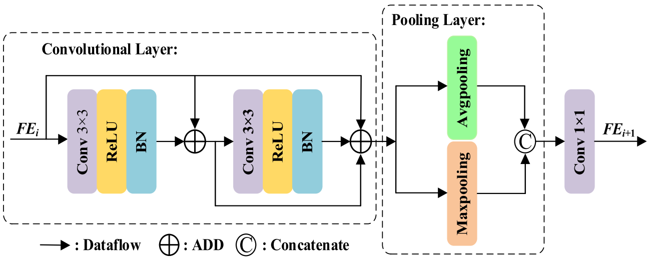


Fig. 3. Structure of the proposed DS-block.

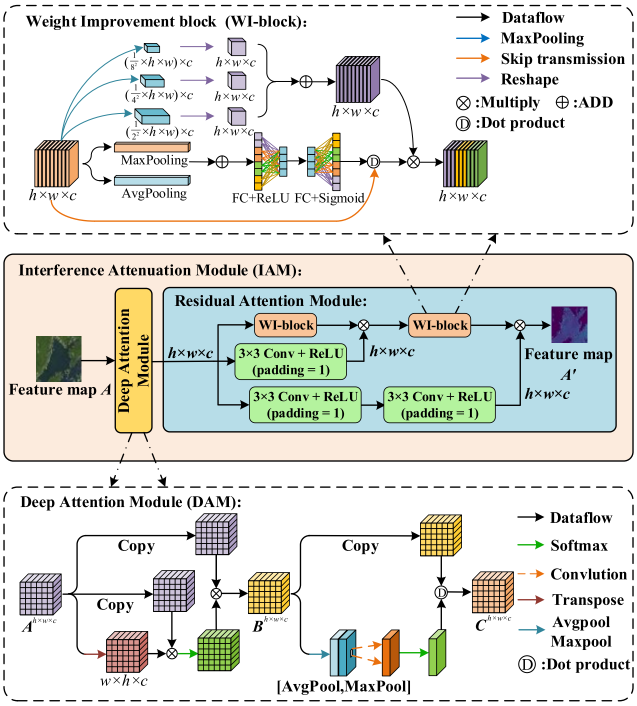


Fig. 4. Structure of the proposed IAM, which consists of DAM and RAM.

of feature fusion by fusing feature elements of different levels so that the features of the previous layer are fully utilized. The function of the pooling layer is to reduce the dimension of the extracted feature information so that the subsequent DS-block can obtain a larger receptive field, thus extracting deep-level semantic features. To enable CNN to extract rich texture information and contour information, we use average pooling and max pooling in the pooling layer.

C. Interference Attenuation Module

To alleviate the over-segmentation problem caused by nonlake water objects, an IAM is designed, the structure of which is shown in Fig. 4. The innovation of IAM is that it can not only model the semantic representation of key features from spatial and channel dimensions but also use a variety of representative global features to enhance key features. IAM consists of a deep attention module (DAM) and a residual attention module (RAM). Among them, the role of the DAM is to use the positional relationship to model the differential features between lakes and other ground objects. The role of the RAM is to give more weight to the key features along the feature channel direction and utilize multiscale global information to emphasize key information, thus diluting the feature representation of the interference area. The workflow of the DAM and the RAM will be described in detail in the following.

1) Deep Attention Module: First, the initial feature maps $A \in \mathbb{R}^{h\times w\times c}$ need to be transposed to generate transposed feature maps $A^{\mathrm{T}} \in \mathbb{R}^{w\times h\times c}$. After the transposition operation is completed, the A^{T} needs to be multiplied with the A, and the product result is sent to the Softmax layer. It is worth noting that A and A^{T} can be multiplied only when the H and W of the image are the same. If the H and W of the image are not the same, consider introducing data preprocessing operations to keep H and W consistent. Then, the output of the Softmax layer is multiplied back to the A to generate the intermediate layer feature maps $B \in \mathbb{R}^{h\times w\times c}$. The purpose of the above steps is to model the feature differences between lake water bodies and other ground objects based on the spatial positional relationship of information. After completing the modeling of the differential features, it is necessary to use the pooling operation and the convolution operation to extract the representative lake water features from the B. Then, use the Softmax layer to reassign the weight ratio of the differential features and use the dot product operation to assign the reassigned information weight to the B, thus generating the feature maps $C \in \mathbb{R}^{h\times w\times c}$ that contain less disturbing information.

2) Residual Attention Module: To further suppress the feature representation of nonlake objects, a RAM is set up. This module mainly utilizes two consecutive weight improvement blocks (WI-blocks) to process the input feature maps $C \in \mathbb{R}^{h\times w\times c}$. WI-block can learn the nonlinear interaction between channels through squeezing and excitation operations, thus strengthening the differential performance of key information and interference information on the channel. In addition to using global pooling and average pooling, WI-block also uses global information at various scales such as $(h/2) \times (w/2) \times c$, $(h/4) \times (w/4) \times c$, and $(h/8) \times (w/8) \times c$ to improve the sensitivity to key information on the feature channels. To enhance the learning ability of the WI-block for discriminative features, we use the skip residual [35], [36] containing 3×3 convolutions to fuse the feature maps C with the output of WI-block.

D. Multilevel Transformer

The role of the MT module is to capture the global dependencies used to extract lake water information and improve the coherence of lake water boundaries. This module is capable of self-attention modeling of multiscale features of different paths to obtain rich boundary information, and its structure is shown in Fig. 5.

Compared with the MT, most of the existing methods [37], [38] insert the transformer as a single module into the encoder of the semantic segmentation network. Although this combination method is simple to operate, it will limit the performance of the transformer. Because the composition

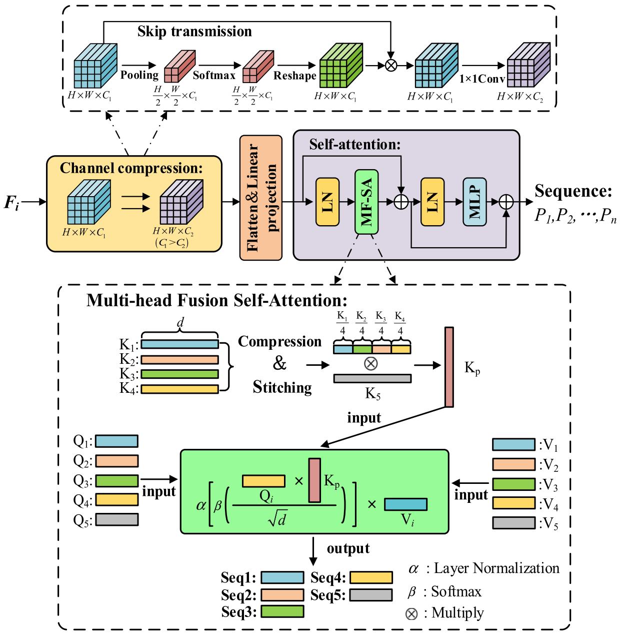


Fig. 5. Structure of the proposed MT module.

of images contains multiple levels of semantic features, this combination method can only model the global context of features at one level. Our research found that the initial images will generate feature maps with different semantic levels after passing through the CNN. If self-attention modeling is performed on feature maps at different levels, the spatial features and contextual semantic associations of target objects can be better captured.

Compared with the MT, most of the existing methods [37], [38] insert the transformer as a single module into the encoder of the semantic segmentation network. Although this combination method is simple to operate, it will limit the performance of the transformer. Because the composition of images contains multiple levels of semantic features, this combination method can only model the global context of features at one level. Our research found that the initial images will generate feature maps with different semantic levels after passing through the CNN. If self-attention modeling is performed on feature maps at different levels, the spatial features and contextual semantic associations of target objects can be better captured.

The novelty of the transformer is that it can perceive the distribution of lake boundaries in the global domain by modeling multiple levels of feature information without adding too many model parameters. In addition, it can use high-level semantic features to guide the modeling process of low-level texture features, thus improving the network's ability to identify detailed local features. The execution steps of the MT include channel compression and self-attention.

1) Channel Compression: The role of channel compression is to reduce useless feature channels to reduce the number of parameters used for transformer calculations. The input $F_i \in \mathbb{R}^{H_i\times W_i\times C_i}(i = 1,2,3,4,5)$ of this process is the I_1, I_2, and I_3 from the IAM and the D_4 and D_5 from the encoder, respectively. The number of channels of I_1, I_2, I_3, D_4, and D_5 is 64, 128, 256, 512, and 1024 in sequence.

First, the F_i needs to undergo a pooling operation to extract the texture and contour information of the lake water bodies. The Softmax function and linear interpolation will reshape this part of the information to generate the key feature maps $F_i' \in \mathbb{R}^{H_i\times W_i\times C_i}(i = 1,2,3,4,5)$. Then, F' is multiplied with the F_i to generate the feature maps $E_i \in \mathbb{R}^{H_i\times W_i\times C_i}(i = 1,2,3,4,5)$ that contain rich lake's information. Finally, a 1 × 1 convolution kernel is utilized to extract the useful key information in E_i and discard the useless feature information, thus generating the feature maps $C_i \in \mathbb{R}^{H_i\times W_i\times c_i}(i = 1,2,3,4,5)$ for the self-attention process. The sizes of C_i are $(H/2) \times (W/2)$, $(H/4) \times (W/4)$, $(H/8) \times (W/8)$, $(H/16) \times (W/16)$, and $(H/32) \times (W/32)$ (H and W are the height and width of the input images), respectively. The channel numbers of C_i are 4, 16, 64, 256, and 1024, respectively.

2) Self-Attention: First, the feature maps $C_i \in \mathbb{R}^{H_i\times W_i\times c_i}(i = 1,2,3,4,5)$ need to be cut into patches of size P, $(P/2)$, $(P/4)$, $(P/8)$, and $(P/16)(P = 16)$, respectively. These patches can be mapped to different regions of the feature map from multiple scales. These patches are then subjected to flattening and location labeling operations to generate feature sequences for the self-attention process, as shown in the following equation:

$$S_i \in \mathbb{R}^{\left(P_i^2\times c_i\right)\times\frac{H\times W}{P_i^2}}, \quad (i = 1,2,3,4,5) \tag{1}$$

where S_i represents the feature sequences. $(H,\ W)$ is the dimension of the feature maps and P_i is the size of the patches. c_i represents the number of channels of F_i. $P_i^2 \times c_i$ represents the dimension of the feature sequences, and $(H \times W/(P_i^2))$ represents the number of feature sequences.

After the feature sequences are obtained, these feature sequences need to be linearly transformed to generate a vector representation for the self-attention process. The process of the linear transformation is shown in the following equation:

$$\begin{cases} Q_i(\text{query}) = S_i \cdot W_{\text{q}} \\ K_i(\text{key}) = S_i \cdot W_{\text{k}} \\ V_i(\text{value}) = S_i \cdot W_{\text{v}} \end{cases} \tag{2}$$

where W_{q}, W_{k}, and W_{v} are all trainable matrices.

Then, use compression and splicing operations to splice $K_i \in \mathbb{R}^{N\times d}(i = 1,2,3,4,5)$ into a public key value $K_p \in \mathbb{R}^{N\times d}$, where N represents the number of key values and d represents its dimension. The specific values are $N = 196$ and $d = 1024$, and the K_p generation process is shown in the following equation. The purpose of K_i stitching is to fuse high-level features with low-level features, so that high-level features can guide the modeling process of low-level features [39], thus improving the ability of the transformer to capture boundary details

$$K_p = \phi\left[\gamma\left(K_i\right)\right] \otimes K_5 = \left(\frac{K_1 + K_2 + K_3 + K_4}{4}\right) \otimes K_5 \tag{3}$$

where ϕ represents the splicing operation, γ represents the compression operation, and ⊗ represents the bitwise product operation.

After completing the above steps, the self-attention process can be performed, which is shown in (4)

$$\mathrm{SA}_i(Q, K, V) = \alpha\left[\beta\left(\frac{Q_i \cdot K_p^{\mathrm{T}}}{\sqrt{d}}\right)\right] \cdot V_i \tag{4}$$

$$\text{MF-SA}_i = \frac{\left(\mathrm{SA}_i^1 + \mathrm{SA}_i^2 + \cdots + \mathrm{SA}_i^h\right)}{h} \tag{5}$$

$$\mathrm{Out} = \mathrm{MLP}[\alpha(\text{MF-SA}_i)] \tag{6}$$

where $\alpha(\cdot)$ represents the layer normalization operation, and $\beta(\cdot)$ represents the Softmax function. h represents the number of heads, and MF-SA represents the output processed by multihead self-attention. MLP represents the multilayer perceptron, which consists of a fully connected layer, a dropout layer, and a Gaussian Error Linear Units (GELU) [40] activation function.

E. Decoder

The role of the decoder is to fuse the feature information from the MT and encoder. Then, the fused feature information is upsampled step by step to obtain a lake water segmentation image with the same size as the original image. In particular, to maintain the consistency of the feature dimension, the hidden feature sequence from the transformer needs to be reshaped to adjust the size from $(H \times W/(P_i^2)) \times c_i$ to $(H/P_i) \times (W/P_i) \times c_i$. In NT-Net, the decoder consists of five US_blocks. Each US_block contains a concatenate layer, a 1×1 convolution kernel, a Rectified Linear Unit (ReLU) activation function, and a $2\times$ upsampling operation.

F. $Loss_{seg}$ Loss Function

NT-Net uses the $\mathrm{Loss}_{\mathrm{seg}}$ loss function to train and optimize the node parameters of the neural network. The $\mathrm{Loss}_{\mathrm{seg}}$ loss function consists of a binary cross-entropy loss ($\mathrm{Loss}_{\mathrm{bce}}$) and a dice coefficient loss ($\mathrm{Loss}_{\mathrm{dice}}$), which can be defined as follows:

$$\mathrm{Loss}_{\mathrm{seg}} = \lambda_1 \mathrm{Loss}_{\mathrm{bce}} + \lambda_2 \mathrm{Loss}_{\mathrm{dice}} \tag{7}$$

where $\mathrm{Loss}_{\mathrm{bce}}$ represents the binary cross-entropy loss, and $\mathrm{Loss}_{\mathrm{dice}}$ represents the dice coefficient loss. λ_1 and λ_2 are the adjustable hyperparameters. In this article, we set $\lambda_1 = 1$ and $\lambda_2 = 1$.

The distribution of lake water samples in remote sensing images is extremely uneven, which makes the final training results dominated by negative samples with more pixels, resulting in inaccurate segmentation results. NT-Net uses Dice loss [41] for this problem, and its definition is shown in the following equation:

$$\mathrm{Loss}_{\mathrm{dice}}(P, G) = 1 - \frac{2\sum_{h=1}^{H}\sum_{w=1}^{W}|G_{h,w} \cap P_{h,w}|}{\sum_{h=1}^{H}\sum_{w=1}^{W}\left(|G_{h,w}| + |P_{h,w}|\right)} \tag{8}$$

where P is the prediction result and G is the ground truth in the label image. W and H represent the width and height of the image, respectively.

Dice is a region-dependent loss, and the loss of the current pixel is not only related to the predicted value of the current pixel, but also related to the predicted value of other points. When the proportion of positive samples in the image is small, Dice will overwhelm some negative samples, making training more inclined to mine foreground regions. However, when there are too many tiny targets, the convergence process is unstable, and even gradient saturation occurs in extreme cases. To optimize the training process of the network, we introduce a binary cross-entropy loss in NT-Net. Unlike Dice loss, binary cross-entropy loss is a pixelwise measure of the difference between the actual distribution and the predicted distribution. Since it directly utilizes the prediction of pixels to adjust the gradient, it is more stable to optimize the model's training process. Therefore, we use the cross-entropy loss to make up for the inadequacy of the Dice loss in model optimization, which is defined as shown in the following equation:

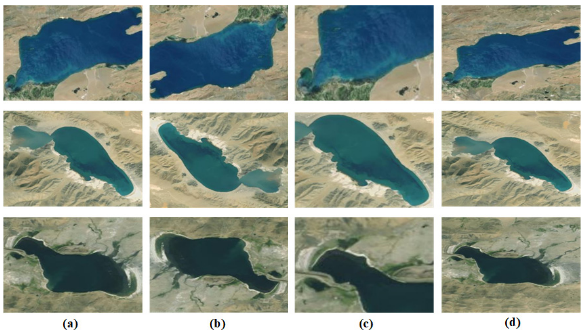


Fig. 6. Augmentation process of the dataset: (a) original images; (b) flipped; (c) randomly cropped; and (d) stretched.

$$\begin{aligned}&\mathrm{Loss}_{\mathrm{bce}}(P, G)\\ &= -\sum_{h=1}^{H}\sum_{w=1}^{W}\left[G_{h,w} \times \log P_{h,w} + \left(1 - G_{h,w}\right)\log\left(1 - P_{h,w}\right)\right].\end{aligned} \tag{9}$$

IV. Experiments

A. Experimental Details

NT-Net is built in the Pytorch deep learning framework and trained using five NVIDIA RTX 3080 (8G) GPUs. To achieve fast convergence of the network, we use the Stochastic Gradient Descent (SGD) optimizer and set momentum 0.9 and weight decay 1e-4 to optimize the back-propagation process of NT-Net. We extracted 5283 remote sensing images of lakes with a size of 448×448 using Google Earth and randomly selected 4021 images as a training dataset and 1262 images as a testing dataset. To improve the generalization ability of the network model to changes in lake morphology, we use rotation, stretching, and random cropping to perform data augmentation on the training dataset as shown in Fig. 6. The augmented training dataset contains 8061 remote sensing images of lake water bodies.

We use the following evaluation metrics to compare the performance differences of different semantic segmentation networks: intersection over union (IoU), Dice (Dice similarity coefficient), and Acc, as shown in the following equations:

$$\mathrm{IoU} = \frac{1}{N}\sum_{n=1}^{N}\frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{FN} + \mathrm{TP}} \times 100\% \tag{10}$$

TABLE I

Comparison Results of NT-Net and Classical Semantic Segmentation Network

Network	*IoU*/%	*Dice*/%	*Acc*/%
U-Net [42]	65.63	71.89	74.51
PspNet [43]	70.10	73.65	76.02
DeepLabV3+ [44]	72.47	76.85	80.81
Our (NT-Net)	**85.26**	**89.38**	**91.49**

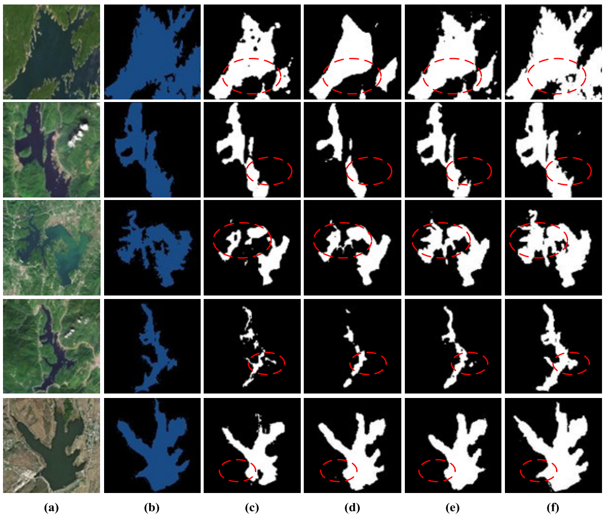


Fig. 7. Visual results of NT-Net and classical semantic segmentation networks: (a) images; (b) ground truth; (c) U-Net; (d) PspNet; (e) DeepLabV3+; and (f) NT-Net.

$$\text{Dice} = \frac{1}{N}\sum_{n=1}^{N}\frac{2\times\text{TP}}{\text{FP}+\text{FN}+2\times\text{TP}}\times 100\% \tag{11}$$

$$\text{Acc} = \frac{1}{N}\sum_{n=1}^{N}\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{FP}}\times 100\% \tag{12}$$

where N is the number of remote sensing images, TP is the number of pixels classified as lakes, TN is the number of pixels classified as background, FP is the number of pixels misclassified as lakes, and FN is wrongly classified as the background number of pixels.

B. Comparison With Classic Semantic Segmentation Networks

To ensure the fairness of the experiments, all networks use DS-block in the encoder. The experiments compare the performance differences between NT-Net and U-Net [42], PspNet [43], and DeeplabV3+ [44]. As shown in Table I, NT-Net achieves significant improvement on all evaluation metrics. The IoU of NT-Net is 85.26%, which is 12.79%–19.63% higher than other networks, which indicates that it is more accurate for lake localization and segmentation. The Dice and Acc of NT-Net are 89.38% and 91.49%, respectively, which are 12.53%–17.49% and 10.68%–16.98% higher than other networks, which means that NT-Net has better practical generalization ability in pixel recognition Acc.

Fig. 7 shows a visual comparison of NT-NET with other networks. As can be seen from Fig. 7, there are many information holes in the segmentation results of U-Net, because U-Net lacks an effective way to perceive the contextual information of images. Although the segmentation results of PspNet and DeepLabV3+ are relatively good, they lack the semantic judgment ability to deal with background interference, resulting in the inability to reproduce the boundary information of the lake. Compared with other networks, NT-Net is not only able to preserve the continuity of lakes, but also accurately segment tiny lakes at the boundaries. In the face of complex background information, NT-Net uses the IAM to suppress the feature performance of background information, making the boundary contour of the lake more refined. Based on the above analysis, it can be seen that NT-Net can accurately identify and segment lake water bodies in images and shows an all-around performance that is better than that of classical semantic segmentation networks.

TABLE II

Comparison Results of NT-Net and Mainstream Lake Segmentation Networks

Network	*IoU*/%	*Dice*/%	*Acc*/%
Lae-Net [45]	75.86	77.64	81.57
HA-Net [46]	77.21	79.02	83.38
SR-SegNet [47]	82.05	83.25	85.46
MFA-Net [14]	80.95	82.49	85.12
FEW-Net [21]	82.83	84.33	87.25
Our (NT-Net)	**85.26**	**89.38**	**91.49**

C. Comparison With the Lake Segmentation Networks Based on Pure CNN Structure

The experiments compare the performance differences between NT-Net and Lae-Net [45], HA-Net [46], SR-SegNet [47], MFA-Net [14], and FWE-Net [21]. As can be seen from Table II, NT-Net achieves the best values on all evaluation metrics. Compared with other networks, NT-Net improves IoU, Dice, and Acc by 2.43%–9.40%, 5.05%–11.74%, and 4.24%–9.92%, respectively. In the above data, the improvement of NT-Net on Dice is particularly obvious, which indicates that NT-Net can more accurately locate the specific location of lakes in the image.

Fig. 8 shows the visualization results of NT-Net and the comparison networks. As shown in Fig. 8, the networks based on pure CNN structure cannot accurately locate the boundary position of the lakes, resulting in low segmentation Acc of the lake boundary. For example, in the last row of Fig. 8, the contrastive networks ignore the boundary information of the middle region of the image. The reason for the above phenomenon is that the receptive field of these networks is small and cannot perceive the contextual information of the boundary. Compared with other networks, the lakes obtained by NT-Net not only have more coherent boundary contours, but also reproduce tiny lakes at the boundary. This is because the MT used by NT-Net can model the contextual distribution of boundary information from the global scope of the images, thus preserving the integrity of boundary contours. Since the MT can utilize high-level semantic features to assist the modeling process of detailed low-level features, NT-Net can further

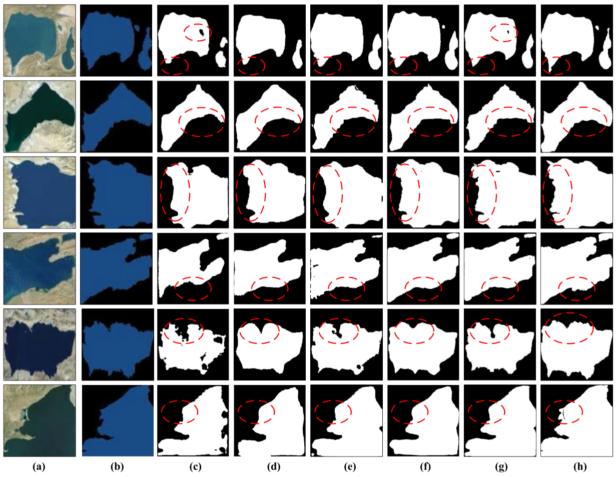


Fig. 8. Visual results of NT-Net and mainstream lake segmentation networks: (a) images; (b) ground truth; (c) Lae-Net; (d) HA-Net; (e) SR-SegNet; (f) MFA-Net; (g) FEW-Net; and (h) NT-Net.

TABLE III
COMPARISON RESULTS BETWEEN NT-NET AND SEGMENTATION NETWORKS BASED ON PURE TRANSFORMER STRUCTURE

Network	*IoU*/%	*Dice*/%	*Acc*/%
BA-Net [30]	78.01	82.44	84.33
Swin-UNet [48]	80.20	83.59	86.15
TransUNet [37]	78.89	82.57	85.35
TransFuse [38]	81.65	84.11	86.67
STransFuse [31]	82.13	85.97	88.52
Our (NT-Net)	**85.26**	**89.38**	**91.49**

refine the edge texture of lakes. Through the experiments in this section, it can be verified that NT-Net can produce more accurate results than the network models based on the pure CNN structure.

D. Comparison With the Segmentation Networks Based on Pure Transformer Structure

The experiments compare the performance differences between NT-Net and BA-Net [30], Swin-UNet [48], TransUNet [37], TransFuse [38], and STransFuse [31]. It can be seen from Table III that NT-Net shows the best performance on the three evaluation indicators of IoU, Dice, and Acc. Compared with TransFuse, NT-Net improves IoU, Dice, and Acc by 3.61%, 5.27%, and 4.82%, respectively; compared with STransFuse, it improves by 3.13%, 3.41%, and 2.97%, respectively. At the same time, it is also significantly higher than BA-Net, SwinUNet, and TransUNet on all evaluation metrics. The above results show that NT-Net can produce better segmentation results than other network models in the task of segmenting lake water bodies.

Fig. 9 shows the visualized experimental results. From the fifth and sixth rows of Fig. 9, it can be seen that the contrastive networks incorrectly classify some background pixels into the lake category, which leads to the occurrence of the over-segmentation problem. This is because contrastive networks cannot suppress the feature representation of background information. Unlike other networks, NT-Net effectively alleviates the over-segmentation problem by using the IAM to weaken the adverse effects of background objects on the segmentation results. Apart from the over-segmentation problem, the contrastive networks cannot recover the detailed texture of the lake. For example, the segmentation results obtained by BA-Net are significantly different from the ground truth. The reason for the above problem is that BA-Net only performs self-attention processing on the feature maps of the high level, while ignoring the utilization of the low-level feature maps. Because the low-level feature maps contain a lot of feature information for scene interpretation, the network cannot accurately reconstruct the detailed features at the boundary. Compared with BA-Net, NT-Net can use self-attention modeling for multiple levels of features and utilizes high-level features to guide the modeling process of low-level features, resulting in a finer edge segmentation effect.

TABLE IV
RESULTS OF THE MT MODULE ABLATION EXPERIMENTS. "—" REPRESENTS THE ELIMINATION OF RELATED MODULES

Network	*IoU*/%	*Dice*/%	*Acc*/%	*Recall*%
NT-Net	**85.26**	**89.38**	**91.49**	**93.82**
NT-Net−MT	82.15	84.47	85.19	82.04

Through the experiments in this section, it can be proved that NT-Net can obtain more accurate segmentation results compared with the current transformer-based network model. In addition, by observing Figs. 7–9, it can be found that compared with other networks, NT-Net still obtains higher segmentation Acc when facing lakes with different shapes. This shows that NT-Net benefits from the advantages of model structure and can obtain stronger generalization ability under the condition of data augmentation.

E. Ablation Studies

To evaluate the performance of the MT module and the IAM, ablation experiments on NT-Net are conducted. In the experiment, the Recall and Precision evaluation indicators are added to evaluate the effectiveness of each module more intuitively, as shown in following equations:

$$\text{Recall} = \frac{1}{N}\sum_{n=1}^{N}\frac{\text{TP}}{\text{FN}+\text{TP}} \times 100\% \tag{13}$$

$$\text{Precision} = \frac{1}{N}\sum_{n=1}^{N}\frac{\text{TP}}{\text{FP}+\text{TP}} \times 100\% \tag{14}$$

where N is the number of remote sensing images, TP is the number of pixels classified as lakes, FP is the number of pixels misclassified as lakes, and FN is wrongly classified as the background number of pixels. The function of Recall is to evaluate the severity of the under-segmentation problem. The lower the value, the more serious the under-segmentation problem is. The role of Precision is to evaluate the severity of the over-segmentation problem. The lower the value, the more serious the over-segmentation problem is.

1) Multilevel Transformer: It can be seen from Table IV that without using the MT module, the network model has a relatively large reduction in the evaluation indicators of IoU, Dice, Acc, and Recall, decreasing by 3.12%, 4.91%, 6.30%,

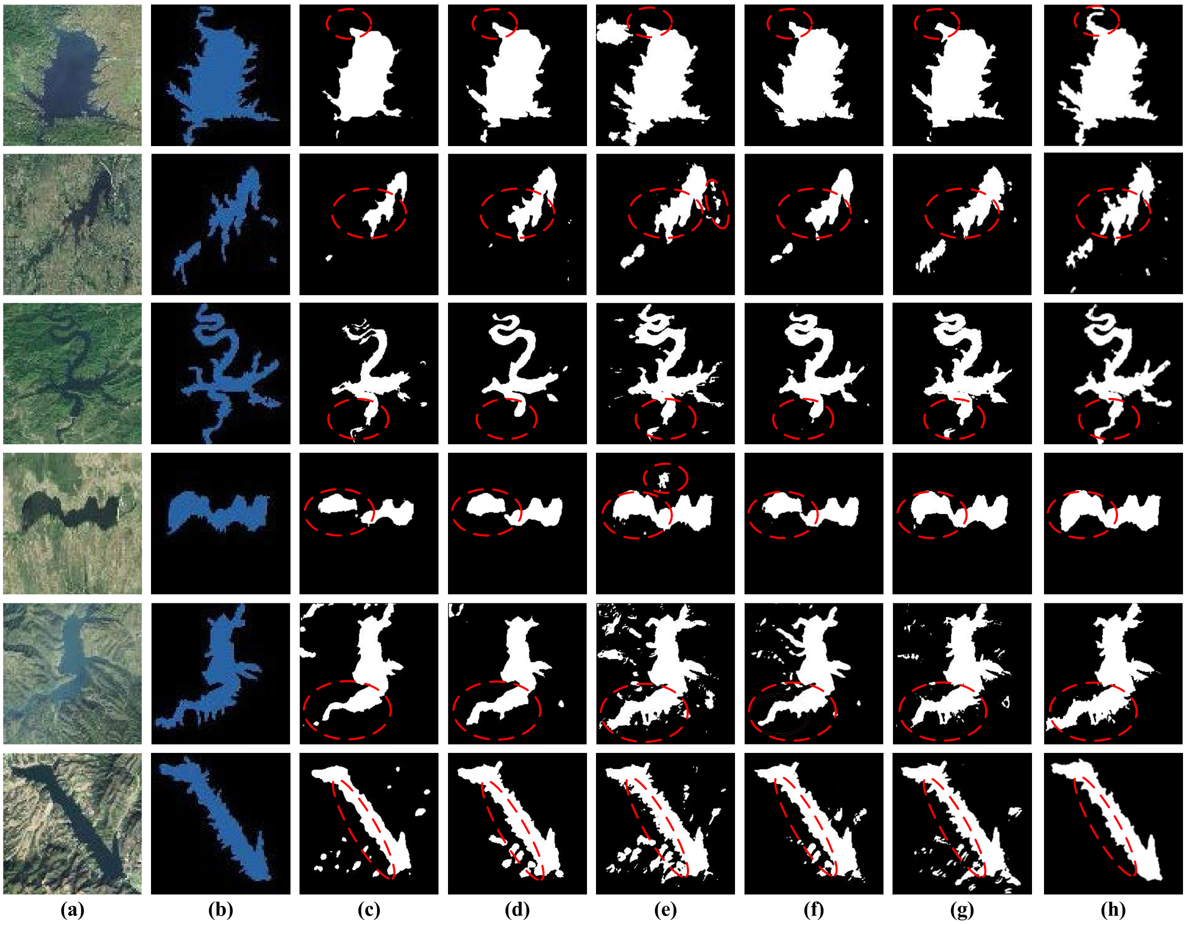


Fig. 9. Visualized segmentation results for the comparative experiments: (a) images; (b) ground truth; (c) BA-Net; (d) Swin-UNet; (e) TransUNet; (f) TransFuse; (g) STransFuse; and (h) NT-Net.

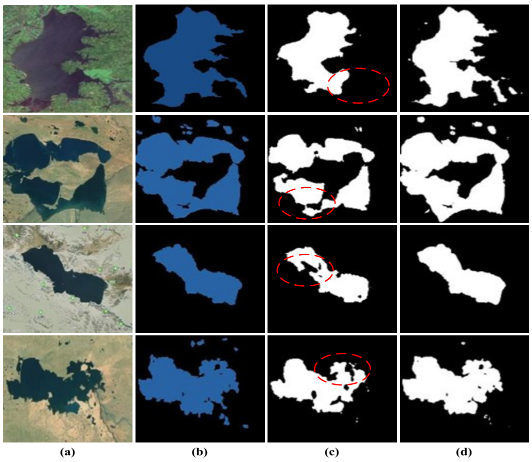


Fig. 10. Visualization results of the MT ablation experiment: (a) images; (b) ground truth; (c) NT-Net-MT; and (d) NT-Net.

and 11.78%, respectively. Among them, the decrease in the Recall value is the most obvious. The above results show that the lack of MT module will cause NT-Net to fail to accurately locate lake water bodies in images, resulting in under-segmentation problems.

Fig. 10 shows the visualization results of the experiment. As shown in Fig. 10, when the MT module is missing, many information holes appear at the boundary of the lake, destroying the integrity of the boundary contour. This is because NT-Net can only use ordinary convolution to aggregate the contextual information of local regions after the lack of the MT module. Since fragmented local context information cannot restore the distribution of boundary information in global space, it destroys the integrity and continuity of boundary contours. After using the MT module, the boundary of the lake water body is clearer and the outline is more complete. This is because MT can capture the semantic dependencies of objects from a global perspective and refine the modeling results of detailed low-level features. Through the experiments in this section, it can be verified that the MT module can show performance advantages to be trusted in improving the integrity of the lake water boundary.

TABLE V
RESULTS OF THE IAM ABLATION EXPERIMENTS

Network	***IoU*/%**	***Dice*/%**	***Acc*/%**	***Precision*/%**
Baseline	80.52	83.37	84.43	84.62
Baseline + DAM	82.05	85.91	87.74	88.51
Baseline + RAM	82.96	86.63	87.25	89.12
Baseline + IAM	**85.26**	**89.38**	**91.49**	**93.29**

2) Interference Attenuation Module: In this section, we conduct ablation experiments on the IAM, as well as the DAM

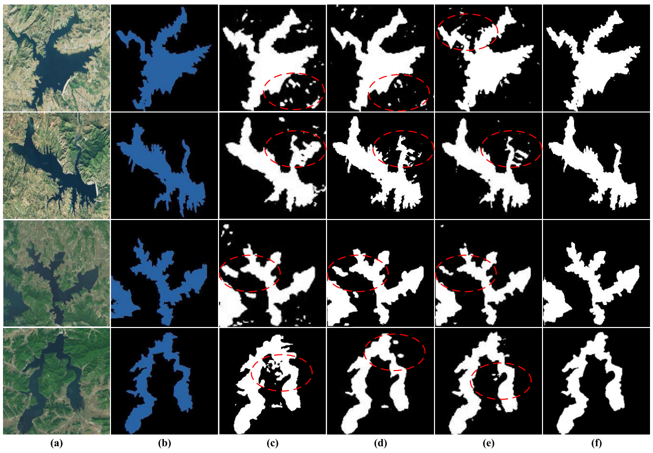


Fig. 11. Visualized segmentation results of the IAM ablation experiment: (a) images; (b) ground truth; (c) baseline; (d) baseline + DAM; (e) baseline + RAM; and (f) baseline + IAM.

and RAM contain. As can be seen from Table V, the baseline network with the IAM removed has the worst performance. Compared with the baseline network, the network using DAM or RAM showed significant improvement in IoU, Dice, Acc and Sen evaluation metrics. This shows that the DAM and RAM used in IAM can effectively suppress the feature representation of nonlake objects. Compared with other networks, the network using IAM achieves the best performance in terms of IoU, Dice, Acc and Sen evaluation indicators, and the improvement of Precision is the most obvious. This result shows that the IAM can improve the anti-interference ability of the network, thus solving the problem of over-segmentation in the segmentation results.

Fig. 11 shows the visualization results of the experiment. As can be seen from Fig. 11(c), when the IAM is removed, some background regions in the result are incorrectly segmented, especially at the boundary of the lake water body, there is an obvious over-segmentation problem. When the network uses DAM or RAM, although the problem of over-segmentation is effectively alleviated, the effect is not obvious. This is because interfering objects not only have lake-like characteristics, but also have some unique positional association with surrounding pixels. Therefore, the feature difference between lakes and disturbing objects cannot be modeled using only positional changes or channel associations. When the network uses the IAM, the over-segmentation problem is effectively alleviated. This is because IAM can take the advantage of the differences in deep semantics between lakes and other ground objects, thus effectively suppressing the feature representation of interfering objects and giving the target more discriminative feature weights. The experiments in this section demonstrate that the module's ability to suppress background interference is reliable.

3) Discussion of the λ_1 and λ_2 in Loss$_{seg}$: Loss$_{\text{seg}}$ uses two hyperparameters $\lambda_1 = 1$ and $\lambda_2 = 1$ to balance the advantages and disadvantages of Dice loss and binary cross-entropy loss. To verify the rationality of the hyperparameter settings in this article, the impact of different λ_1 and λ_2 on the network segmentation Acc is compared. The training and testing process of the experiment uses remote sensing images with a positive sample ratio of less than 35% to compare the experimental results more intuitively, as shown in Fig. 12.

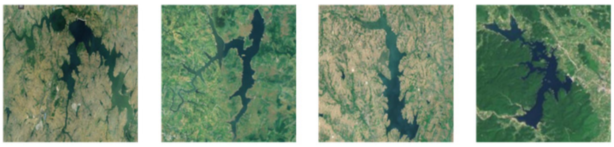


Fig. 12. Remote sensing images of lake water bodies with a positive sample ratio of less than 35%.

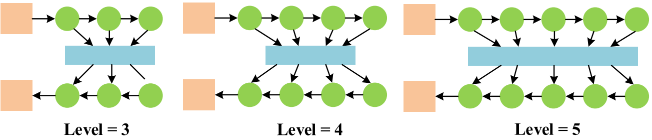


Fig. 13. Schematic of NT-Net at different sampling layers.

It can be seen from Table IV that when λ_2 representing Dice loss is 0.5 and 0.7, the segmentation Acc of the network is low, especially when λ_2 is 0.5, the IoU and Acc of the network have dropped significantly by 3.22% and 5.67%, respectively. The above phenomenon shows that when the number of foregrounds and background pixels is seriously unbalanced, the decrease of the Dice ratio will hinder the training process of the network and reduce the fitting degree of the parameters to the samples. When λ_1 representing the cross-entropy loss is 0.5 and 0.7, the segmentation Acc of the network is also lower. This is because the cross-entropy loss can maintain a high gradient state, improve the stability of the network convergence process, and update the parameters in a direction conducive to accurate segmentation. Therefore, the proportional decrease of the cross-entropy loss will affect the final segmentation Acc of the network. This article also compares the cases of $\lambda_1 = 1$, $\lambda_2 = 0.9$ and $\lambda_1 = 0.9$, $\lambda_2 = 1$, and finds that the segmentation Acc of the network differs very little. Therefore, considering comprehensively, we set λ_1 and λ_2 to be 1 uniformly.

V. Discussion

A. Network Depth of the NT-Net

NT-Net performs five downsampling operations and five upsampling operations on the input image, respectively. Its purpose is to enable the network to fully extract the semantic features of lake water without generating too many redundant features. To verify the rationality of the above settings, we compared NT-Net with 3, 4, 5, and 6 sampling layers (as shown in Fig. 13) and verified the variation of the Acc (Dice) with the number of iterations. To facilitate the training of NT-Net with six layers, the dataset size used in the experiment is 512^2.

As shown in Fig. 14, when the sampling layers of NT-Net are 3 and 4, the verification Acc is low because the shallow network cannot fully extract the feature information in the image, which limits the performance of the network. When

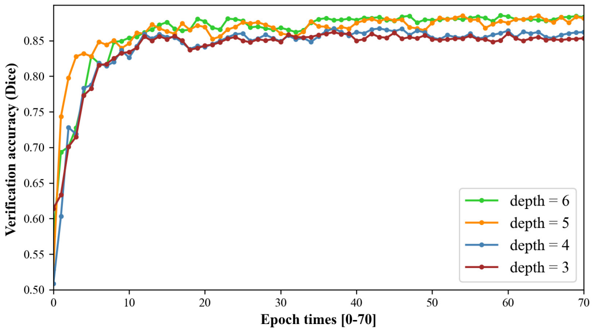


Fig. 14. Effect of the number of sampling layers on the validation Acc.

TABLE VI

EFFECT OF DIFFERENT λ_1 AND λ_2

Parameters	*IoU*/%	*Acc*/%
$\lambda_2 = 1,\ \lambda_1 = 0.7$	81.24	85.36
$\lambda_2 = 1,\ \lambda_1 = 0.5$	80.97	83.94
$\lambda_1 = 1,\ \lambda_2 = 0.7$	80.46	82.73
$\lambda_1 = 1,\ \lambda_2 = 0.5$	79.35	81.05
$\lambda_1 = 1,\ \lambda_2 = 1.0$	**82.57**	**86.72**

TABLE VII

EFFECT OF NETWORK LAYERS

Depth	*IoU*/%	*Dice*/%	*Acc*/%
Layer=3	82.69	85.03	88.75
Layer=4	84.24	85.58	89.63
Layer=5	**85.26**	**89.38**	**91.49**
Layer=6	85.48	89.07	92.12

the number of sampling layers of NT-Net is 6, its verification Acc is almost the same as that of NT-Net with five layers, and both are higher than NT-Net with depths 3 and 4. However, a too deep network will lead to an increase in computational complexity and also produce redundant features. To further demonstrate the impact of the difference in the number of sampling layers on the segmentation Acc, we compared NT-Nets with sampling layers of 3, 4, 5, and 6, and the results are shown in Table VII.

As shown in Table VII, when the number of sampling layers is 5, the network can show certain advantages in various evaluation indicators. This proves that when the number of sampling layers is 5, NT-Net can give full play to its performance and produce more accurate segmentation results.

B. Number of Layers of the MT

This article sets up an eight-layer MT module in NT-Net. This setting expands the attention relationship between pixels, thus enhancing the global modeling ability of NT-Net for images. To verify the rationality of the above settings, we conducted the following experiments: using the training set data to train NT-Nets with 4, 6, 8, 10, and 12 layers of MT modules, respectively. Then, the change of IoU value with the number of iterations during network model training was compared (every 20 epochs are counted). The experimental results are shown in Fig. 15.

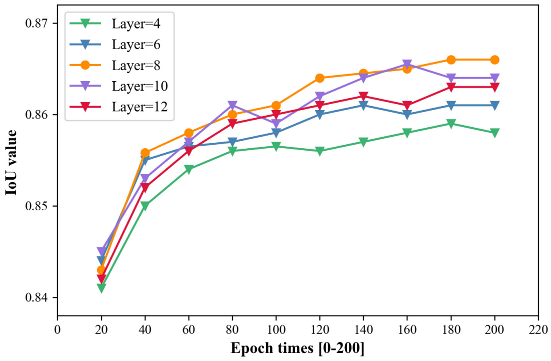


Fig. 15. IoU values vary with the number of transformer layers.

TABLE VIII

EFFECT OF INPUT RESOLUTION

Input size	*IoU*/%	*Dice*/%	*Acc*/%	*Time*/min
384×384	83.72	86.51	88.30	693
448×448	**85.26**	**89.38**	**91.49**	**1157**
512×512	86.31	89.75	91.60	1734

It can be found from Fig. 15 that when the number of MT modules is 8, NT-Net can achieve the highest segmentation Acc. The explanation for this phenomenon is that the eight-layer MT module helps to model accurate node features. In addition, setting up an eight-layer MT also helps NT-Net to expand the spatial connection of feature elements in the context area. In conclusion, if the number of layers is too small, it will reduce the ability of the network to capture nonlocal semantic dependencies. If there are too many layers, it will cause much redundant computation, hindering the optimization process of the network model. Therefore, from the perspective of algorithm operation efficiency, we set up an eight-layer MT module in NT-Net to produce more advantageous segmentation results.

C. Effect of Input Resolution

In NT-Net, we set the input resolution to 448 × 448. To verify the rationality of this setting, we test results with 384 × 384, 448 × 448, and 512 × 512 resolutions as NT-Net input is given, as shown in Table VIII.

As shown in Table VI, while keeping the number of patches fixed, the segmentation Acc of NT-Net increases with the input resolution. Taking the Dice value as an example, the segmentation Acc of NT-Net gradually increased from 86.51% to 89.75%. Although the segmentation Acc of NT-Net improves slightly when the input resolution size is increased from 448 × 448 to 512 × 512, the training time of the network increases significantly. Therefore, considering the computational overhead and the running efficiency of the algorithm, this article sets the input resolution of NT-Net to 448 × 448.

D. Model Complexity of the NT-Net

To demonstrate that the excellent performance of NT-Net is due to its efficient structural advantage rather than the huge number of parameters, the following experiments are set up.

TABLE IX

COMPARISON OF DIFFERENT NETWORKS IN TERMS OF PARAMETER SIZE, TRAINING TIME, AND ACC. THE ENCODER PARTS OF BOTH U-NET AND DEEPLABV3+ USE THE DS-BLOCK DESIGNED IN THIS ARTICLE. (CNN: U-NET, DEEPLABV3+ TRANSFORMER: BA-NET, STRANSFUSE)

Network	Input size	*IoU*/%	Parameters/M	Time/min
U-Net	448×448	65.63	36.54	544
DeepLabV3+	448×448	72.47	68.42	735
BA-Net	448×448	78.01	115.91	1084
STransFuse	448×448	81.65	108.32	973
Our	**448×448**	**85.26**	**127.06**	**1127**

The experiments compared U-Net, DeepLabV3+, BA-Net, STransFuse, and NT-Net in terms of parameter size, model size, training time, and IoU. The results are shown in Table IX.

As shown in Table IX NT-Net is significantly larger than CNN-based network models in terms of model parameters and model size. The reason is that the execution of the transformer involves complex matrix operations. Compared with transformer-based BA-Net, the number of parameters of NT-Net has increased by 11.15M, and the segmentation Acc has increased by 7.25%. Compared with STransFuse, the number of parameters of NT-Net has increased by 18.74M, and the segmentation Acc has increased by 3.61%. The above results show that although NT-Net achieves higher Acc, it does not lead to a large increase in model parameters and training time. Unlike BA-Net, which has a large number of parameters and low Acc, NT-Net achieves a good tradeoff between model complexity and segmentation Acc despite its complex model structure. Therefore, the experiments in this section can prove that the excellent performance of NT-Net mainly depends on the effectiveness of modules.

VI. CONCLUSION

In this article, we design a semantic segmentation network NT-Net that fuses transformer and CNN for the accurate extraction of lake water bodies in remote sensing images. NT-Net takes full advantage of the inductive bias ability of CNN in modeling information correlation and the modeling ability of transformer in global context information. Benefiting from the effectiveness of the structure, NT-Net can not only extract rich semantic features from parallel multipaths containing multiple levels, but also effectively express the contextual connections of semantic features. We use remote sensing images of lakes extracted from Google Earth to test the performance of NT-Net and other semantic segmentation networks. The experimental results show that NT-Net can obtain segmentation Acc better than other classical semantic segmentation networks, which proves the advancement and effectiveness of NT-Net. In addition, we also verify the excellent performance of the IAM and the MT module through ablation experiments. Overall, our study provides a new method for accurate and efficient extraction of lake water bodies from remote sensing images. Because of the high labor cost for the production of high-quality labels, we will try to transform NT-Net into a semisupervised model-based semantic segmentation network in future work. The network can extract lake water bodies in remote sensing images by using a small number of training samples.

REFERENCES

[1] X. Luo *et al.*, "Research on change detection method of high-resolution remote sensing images based on subpixel convolution," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1447–1457, 2021.

[2] M. M. Al-Khaldi *et al.*, "Inland water body mapping using CYGNSS coherence detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7385–7394, Sep. 2021.

[3] A. El-Alem, K. Chokmani, I. Laurion, S. E. El-Adlouni, S. Raymond, and C. Ratte-Fortin, "Ensemble-based systems to monitor algal Bloom with remote sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7955–7971, Oct. 2019.

[4] Y. Liu *et al.*, "A classification-based, semianalytical approach for estimating water clarity from a hyperspectral sensor onboard the ZY1-02D satellite," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[5] S. Wang, M. V. Peppa, W. Xiao, S. B. Maharjan, S. P. Joshi, and J. P. Mills, "A second-order attention network for glacial lake segmentation from remotely sensed imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 189, pp. 289–301, Jul. 2022.

[6] S. Lu, B. Wu, N. Yan, and H. Wang, "Water body mapping method with HJ-1A/B satellite imagery," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 13, no. 3, pp. 428–434, Jun. 2011.

[7] Y. Zheng, X. Zhang, B. Hou, and G. Liu, "Using combined difference image and *k*-means clustering for SAR image change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 691–695, Mar. 2014.

[8] S. Klemenjak, B. Waske, S. Valero, and J. Chanussot, "Automatic detection of rivers in high-resolution SAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1364–1372, Oct. 2012.

[9] S. K. McFeeters, "The use of the normalized difference water index (NDWI) in the delineation of open water features," *Int. J. Remote Sens.*, vol. 17, no. 7, pp. 1425–1432, May 1996.

[10] C. B. Obida, G. A. Blackburn, J. D. Whyatt, and K. T. Semple, "River network delineation from Sentinel-1 SAR data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 83, Nov. 2019, Art. no. 101910.

[11] L. Wu *et al.*, "Discrimination of algal-Bloom using spaceborne SAR observations of great lakes in China," *Remote Sens.*, vol. 10, no. 5, p. 767, May 2018.

[12] Q. Zhao, J. Liu, Y. Li, and H. Zhang, "Semantic segmentation with attention mechanism for remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.

[13] F. Chen, "Comparing methods for segmenting supra-glacial lakes and surface features in the Mount Everest region of the Himalayas using Chinese GaoFen-3 SAR images," *Remote Sens.*, vol. 13, no. 13, p. 2429, Jun. 2021.

[14] K. Hu, M. Li, M. Xia, and H. Lin, "Multi-scale feature aggregation network for water area segmentation," *Remote Sens.*, vol. 14, no. 1, p. 206, Jan. 2022.

[15] Z. Wang, X. Gao, Y. Zhang, and G. Zhao, "MSLWENet: A novel deep learning network for lake water body extraction of Google remote sensing images," *Remote Sens.*, vol. 12, no. 24, p. 4140, Dec. 2020.

[16] J. Li, C. Wang, L. Xu, F. Wu, H. Zhang, and B. Zhang, "Multitemporal water extraction of Dongting lake and Poyang lake based on an automatic water extraction and dynamic monitoring framework," *Remote Sens.*, vol. 13, no. 5, p. 865, Feb. 2021.

[17] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2016.

[18] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.

[19] Z. Zhang, M. Lu, S. Ji, H. Yu, and C. Nie, "Rich CNN features for water-body segmentation from very high resolution aerial and satellite imagery," *Remote Sens.*, vol. 13, no. 10, p. 1912, May 2021.

[20] M. Xia, Y. Cui, Y. Zhang, Y. Xu, J. Liu, and Y. Xu, "DAU-Net: A novel water areas segmentation structure for remote sensing image," *Int. J. Remote Sens.*, vol. 42, no. 7, pp. 2594–2621, Apr. 2021.

[21] J. Wang *et al.*, "FWENet: A deep convolutional neural network for flood water body extraction based on SAR images," *Int. J. Digit. Earth*, vol. 15, no. 1, pp. 345–361, Dec. 2022.

[22] H.-F. Zhong, H.-M. Sun, D.-N. Han, Z.-H. Li, and R.-S. Jia, "Lake water body extraction of optical remote sensing images based on semantic segmentation," *Appl. Intell.*, vol. 2022, pp. 1–16, Apr. 2022.

[23] G. Wu *et al.*, "A boundary regulated network for accurate roof segmentation and outline extraction," *Remote Sens.*, vol. 10, no. 8, p. 1195, Jul. 2018.

[24] Y. Wang, W. Ding, R. Zhang, and H. Li, "Boundary-aware multitask learning for remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 951–963, 2021.

[25] Z. Miao, K. Fu, H. Sun, X. Sun, and M. Yan, "Automatic water-body segmentation from high-resolution satellite images via deep networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 4, pp. 602–606, Apr. 2018.

[26] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–22.

[27] V. Ashish *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 1–11.

[28] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang, "SG-Net: Syntax guided transformer for language representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3285–3299, Jun. 2022.

[29] H. Li, J. Xiao, M. Sun, E. G. Lim, and Y. Zhao, "Transformer-based language-person search with multiple region slicing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1624–1633, Mar. 2022.

[30] L. Wang, R. Li, D. Wang, C. Duan, T. Wang, and X. Meng, "Transformer meets convolution: A bilateral awareness network for semantic segmentation of very fine resolution urban scene images," *Remote Sens.*, vol. 13, no. 16, p. 3065, Aug. 2021.

[31] L. Gao *et al.*, "STransFuse: Fusing Swin transformer and convolutional neural network for remote sensing image semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10990–11003, 2021.

[32] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[33] Z. Xu, W. Zhang, T. Zhang, Z. Yang, and J. Li, "Efficient transformer for remote sensing image segmentation," *Remote Sens.*, vol. 13, no. 18, p. 3585, Sep. 2021.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[36] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[37] J. Chen *et al.*, "TransUnet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[38] Y. Zhang, H. Liu, and Q. Hu, "TransFuse: Fusing transformers and CNNs for medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, vol. 12901, 2021, pp. 14–24.

[39] X. Ma, X. Zhang, and M.-O. Pun, "A crossmodal multiscale fusion network for semantic segmentation of remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 3463–3474, 2022.

[40] S. Lei, Z. Shi, and W. Mo, "Transformer-based multistage enhancement for remote sensing image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[41] F. Pan, Z. Wu, Q. Liu, Y. Xu, and Z. Wei, "DCFF-Net: A densely connected feature fusion network for change detection in high-resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11974–11985, 2021.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[44] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 11211, 2018, pp. 833–851.

[45] W. Liu *et al.*, "LaeNet: A novel lightweight multitask CNN for automatically extracting lake area and shoreline from remote sensing images," *Remote Sens.*, vol. 13, no. 1, p. 56, Dec. 2020.

[46] Z. Wang, X. Gao, and Y. Zhang, "HA-Net: A lake water body extraction network based on hybrid-scale attention and transfer learning," *Remote Sens.*, vol. 13, no. 20, p. 4121, Oct. 2021.

[47] L. Weng, Y. Xu, M. Xia, Y. Zhang, J. Liu, and Y. Xu, "Water areas segmentation from remote sensing images using a separable residual SegNet network," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, p. 256, Apr. 2020.

[48] H. Caoet *et al.*, "Swin-Unet: Unet-like pure transformer for medical image segmentation," 2021, *arXiv:2105.05537*.

Hai-Feng Zhong was born in Shandong, China, in 1996. He received the B.S. degree from the Shandong University of Traditional Chinese Medicine, Jinan, China, in 2019. He is currently pursuing the M.S. degree with the Shandong University of Science and Technology, Qingdao, China.

His research interests include image processing and deep learning.

Qing Sun received the B.S. and M.S. degrees in computer science from the Shandong University of Science and Technology, Qingdao, China, in 1995 and 2008, respectively.

He is currently an Associate Professor with the Dongfang College, Shandong University of Finance and Economics, Taian, China. His research interests include artificial intelligence, computer vision, and big data processing.

Hong-Mei Sun received the B.S. and M.S. degrees in computer science from the Shandong University of Science and Technology, Qingdao, China, in 1995 and 2005, respectively.

She is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, and is the Leader of the Key Research and Development Projects of Shandong Province, China. She has authored ten publications and coauthored 30 publications. Her research interests include microseismic monitoring technology and software engineering.

Rui-Sheng Jia is currently a Full Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China, and is the Leader of the Humanities and Social Science Fund of Ministry of Education of the People's Republic of China. He has authored more than 30 publications and coauthored more than 40 publications. His research interests include artificial intelligence, computer vision, big data processing, information fusion, microseismic monitoring, and inversion.