International Journal of Computational Intelligence Systems (2023) 16:134
https://doi.org/10.1007/s44196-023-00312-8

RESEARCH ARTICLE

Two-Stream Xception Structure Based on Feature Fusion for DeepFake Detection

Bin Wang[1] · **Liqing Huang**[1,2,3] · **Tianqiang Huang**[1,2,3] · **Feng Ye**[1,2,3]

Received: 23 January 2023 / Accepted: 31 July 2023

Abstract
DeepFake may have a crucial impact on people's lives and reduce the trust in digital media, so DeepFake detection methods have developed rapidly. Most existing detection methods rely on single-space features (mostly RGB features), and there is still relatively little research on multi-space feature fusion. At the same time, a lot of existing methods used a single receptive field, which leads to models that cannot extract information of different scales. In order to solve the above problems, we propose a two-stream Xception network structure (Tception) that fused RGB spatial feature and noise-space feature. This network structure consists of two main parts. The first part is a feature fusion module, which can adaptively fuse RGB feature and noise-space feature generated by RGB images through SRM filters. The second part is the two-stream network structure, which utilizes a parallel structure of convolutional kernels of different sizes allowing the network to learn features of different scales. The experiments show that the proposed method improves performance compared to the Xception network. Compared to SSTNet, the detection accuracy of the Neural Textures is improved by nearly 8%.

Keywords Deep learning · Feature fusion · Two-stream structure

Abbreviation
Tception Two-stream Xception network

1 Introduction

The rapid development of DeepFake techniques has fueled the sharp increase of forgery face images and videos, and the fake images and videos created by these techniques are becoming increasingly realistic. Falsified content of videos and images raises various disconcerting problems within wide spread social media, such as fake news dissemination, and fraud. Therefore, there has been an explosive increase in the demand for DeepFake detection methods to counteract its impacts [1–3].

In fact, DeepFake detection is a challenging classification problem. The most important aspect of DeepFake detection is to find the differences between real and fake images. In this problem, artificial neural networks have made outstanding achievements especially convolutional neural networks (CNNs) [4–7]. However, most existing models (shown in Sect. 2.2) used RGB images for detection, which led to a limited amount of information in the final detector and makes it difficult to detect images or videos by different domains of information. At the same time, many of the existing models used a single receptive field neural network to classify images, which made it difficult to extract information at different scales. The use of multiple receptive fields to extract information at different scales had become an important method to improve the ability of the model.

To address the above shortcomings, we propose a two-stream Xception framework (namely Tception). The Tception structure could obtain a wider range of receptive field than the original Xception structure, which could improve the network performance. At the same time, to address the

✉ Liqing Huang
lqhuang@fjnu.edu.cn

Bin Wang
qsx20200690@student.fjnu.edu.cn

Tianqiang Huang
fjhtq@fjnu.edu.cn

Feng Ye
yefeng@fjnu.edu.cn

[1] College of Computer and Cyberspace Security, Fujian Normal University, Fuzhou 350117, Fujian, China

[2] Digital Fujian Institute of Big Data Security Technology, Fuzhou 350117, Fujian, China

[3] Fujian Provincial Engineering Research Center of Big Data Analysis and Application, Fuzhou 350117, Fujian, China

Published online: 16 August 2023

problem that existing networks mostly process images in RGB space only, we have considered fusing features from the original RGB spatial image with the Fourier transformed image, taking into account that forgery traces are mostly at the edges of the image. However, it is difficult to use a unified feature fusion method for integrating as the images in the frequency domain do not have a one-to-one correspondence location with the spatial image. Therefore, Tception structure used the SRM (Steganalysis Rich Model) space filter to process images, and then used the feature fusion module to adaptively fuse the RGB space with the feature maps in SRM space, so that the network obtains richer features and improves the network performance.

The main contributions of this paper can be summarized as follows:

(1) We propose a new two-stream Xception (Tception) structure. The Tception structure can expand the receptive field of the network to better perceive the nuances of real and fake images, resulting in better results.
(2) We design the feature fusion module to fuse the features in RGB space and SRM space. In this way, the network can obtain richer features for discrimination.
(3) We combine feature fusion module and Tception structure to let the network access more information. Experiments show that the proposed method has better performance.

2 Related Work

2.1 DeepFake Datasets

There are many datasets in the field of DeepFake tampering forensics, e.g., UADFV [8], Celeb-DF [9], DFDC [10], and FaceForensics++ [11]. Among them, FaceForensics++ is a popular dataset in the field of DeepFake due to its comprehensive video content and its classification according to video quality. Therefore, our work will be experimented on the FaceForensics++ dataset mainly.

FaceForensics++ contains videos that have been tampered with using different human face tampering methods such as DeepFake [12] (DF), Face2Face [13] (F2F), FaceSwap [14] (FS) and Neural Textures [15] (NT). Each video is available in RAW, HQ(c23) and LQ(c40) quality. A part of video cutouts from FaceForensics++ dataset are shown in Fig. 1.

The Celeb-DF dataset contains both real and DeepFake synthesized videos with similar video quality to those disseminated online. We also completed partial comparison experiments on the Celeb-DF dataset to demonstrate the good general applicability of our method.

2.2 DeepFake Detection

DeepFake detection generally includes extracting features manually and extracting features automatically using deep networks. Extracting features manually are interpretable but often less accurate than extracting features automatically using deep networks, while methods using deep networks have improved detection accuracy at the expense of some interpretability. The accuracy of the methods using deep networks has been improved at the expense of some interpretability. In general, the automatic feature extraction methods based on neural networks are better than the manual feature extraction methods and are gradually becoming the mainstream methods of DeepFake detection.

Cozzolino et al. [16] proposed a residual-based local descriptor approach and allowed for better performance with fine-tuned networks on small datasets. Bayar and Stamm [17] proposed a method based on deep Siamese CNNs to detect not only the tampering traces but also the kind of tampering is taking place. Rahmouni et al. [18] proposed a novel method for classifying computer graphics and real photographic images that integrates a statistical feature extraction to a CNN framework and the method could find the best features for efficient boundary. Darius Afchar et al. [19] modified MesoNet that is a light-weight network specifically for face tampering detection and able to train better models on a relatively small number

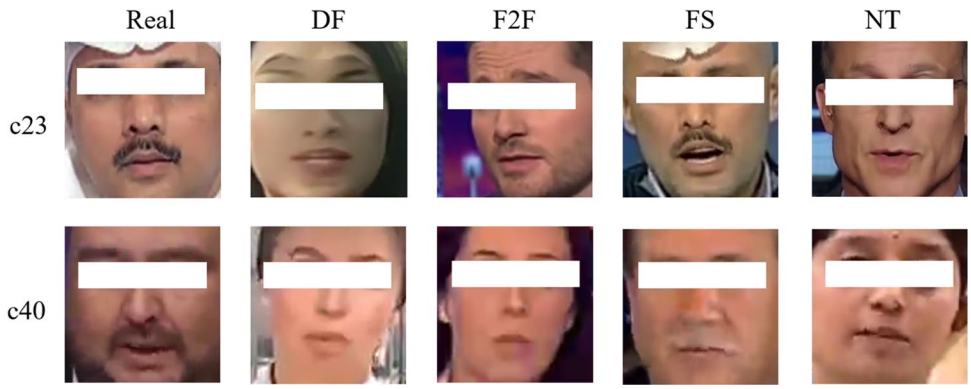

Fig. 1 Selected video cutouts from the FaceForensics++ dataset. FaceForensics++ contains videos that have been tampered with using different human face tampering methods such as DeepFake (DF), Face2Face (F2F), FaceSwap (FS) and Neural Textures (NT). Each video in turn contains a corresponding different quality

of network layers. Chai et al. [20] assembled the Xception network and it had become one of the baselines in the field because of its good performance. These methods used neural networks and achieve better detection results on the DeepFake detection task. However, these methods only used RGB spatial images for relevant feature extraction operations, which could only extract information within a single space, and the discriminator had a relatively limited basis for discrimination.

There are also many methods based on feature fusion for DeepFake detection. Zhao et al. [21] proposed frequency-aware discriminative feature learning for face forgery detection. Zekun Sun et al. [22] proposed an efficient and robust framework (LRNet) to detect DeepFake videos through temporal modeling of precise geometric features. Yuval Nirkin et al. [23] modified two-stream residual structures, as a new idea for improving networks. Li et al. [24] proposed a fusion of the spatial and frequency domains to perform forgery detection. Although these methods fused different feature space information and feed the fused information to a discriminator for discrimination, these methods used a single size of convolutional kernel for the network structure, making it difficult to extract information of different sizes.

One of the most inspiring aspects for our work is Xception network, a convolutional neural network architecture based entirely on deeply separable convolutional layers. It is based on the assumption that the mapping of cross-channel correlation and spatial correlation in the feature map of a convolutional neural network can be completely decoupled. Overall, the Xception architecture is a linear stack of deeply separable convolutional layers with residual connections. This makes the architecture very easy to define and modify.

3 Proposed Methodology

3.1 Overall Framework

The proposed overall framework is shown in Fig. 2. First, we processed the RGB space image by the SRM filters to produce a noise-space image (Sect. 3.2). Second, we used point convolution to fuse the RGB space information with the SRM noise-space information (Sect. 3.3). Finally, we fed the fused information to the proposed Tception structure (Sect. 3.4) to infer the input image being real or fake.

3.2 SRM Noise Space

Sometimes, RGB channels were not sufficient to solve all the different tampering situations. Since forged faces often produced differences at the edges of the face, detecting images that had been carefully modified after tampering is a challenge for the RGB stream.

Previous research found that most of the artifacts produced by forged faces were high-frequency noise at the pixel level, so we can effectively compensate for the disadvantage of highly correlated RGB spatial features with image content by extracting the high-frequency noise components of the image (noise residues) rather than its content.

SRM (steganalysis rich model) [25] filters were proposed to collect the underlying noise features, quantifies and truncates the output of these filters, and extracts nearby cooccurrence information as the final features. SRM has become a common method for extracting noise features.

Inspired by [24], we exploit the local noise distribution of the image to provide additional features. Compared to RGB streams, noise streams are designed to focus more on noise rather than semantic image content, which gives

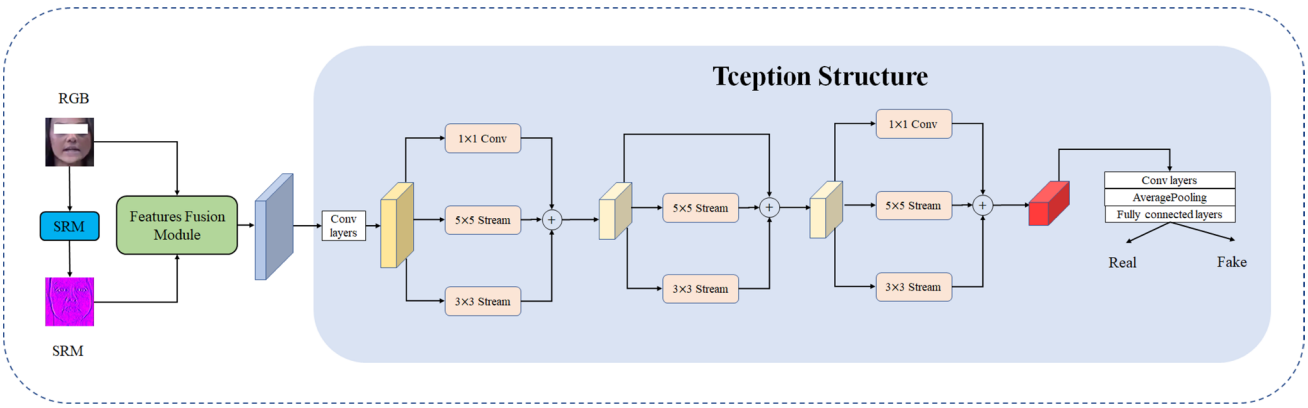

Fig. 2 Overview of our method. Our method is divided into two parts: first, the information in the corresponding noise space is obtained through the SRM filter, and the information in RGB space is fused with the information in SRM noise space by point convolution; subsequently, the fused information is fed into our proposed Tception structure, a network of Xception-like structures containing multiple sensory fields, for the final inference

the possibility to construct more sophisticated forgery detectors. We use SRM filters to extract local noise features from RGB images as the input for noise streams.

Our aim is to extract high-frequency noise from the image. However, we are unsure of the exact pattern of noise from the different tampering methods, we choose a set of filters that extract only the high-frequency noise at the edges of the image (the convolution kernel weights sum to 0) and are relatively symmetrical. The weights of our filters are shown in Eq. (1):

$$\frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix} \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

The 3 channels of RGB space were passed through 3 filters, respectively. Each channel produces a corresponding 3-channel feature map, and after this SRM transformation, a 9-channel feature map can be produced in the end. It emphasize local noise rather than image content and clearly reveal traces of tampering that may not be visible in the RGB channels.

3.3 Fusion of RGB Space and SRM Noise Space

The difference between a real face and a fake face is mainly in the edge part of the face. As mentioned in the previous section, the single RGB space has the limited feature information, while the noise space can better highlight the edge information of the image. Therefore, we propose the feature fusion module which can adaptively fuse RGB feature and SRM space feature.

We first use the SRM for each channel of RGB images to produce a 9-channel feature map. If the feature map in RGB space and the feature map in noise space are directly concatenated together, the network can pay more attention to the feature map in RGB space due to the RGB space having more information. Therefore, we transform the 9-channel feature map in SRM space into a 3-channel feature map using point convolution. Then, performed a point convolution operation on the 3-channel feature map in SRM space and the corresponding channel of the 3-channel feature map in RGB space. The output is still a 3-channel feature map as shown in Fig. 3.

The 3-channel feature map contains the information from the RGB space and the noise space, and we use this image as the input to the neural network.

3.4 Tception Structure

In the Tception structure, we add a 5×5 separable convolutional stream to the original Xception structure module, as shown in Fig. 4. In this way, the problem of insufficient receptive field of the original Xception can be effectively solved. At the same time, we retain the residual structure of the original Xception, thus the problem of gradient drop or gradient disappearance due to overly large and deep network. It also can retain the integrity of the information.

Specifically, the fused feature map is convolutionally transformed in two layers to produce a 64-dimensional feature map X. This feature map X is fed into a block of the entry flow, which use two separable convolutional flows of 3×3 and 5×5 and a 1×1 convolutional flow, respectively, and the three resulting feature maps are summed. As shown in Eq. (2),

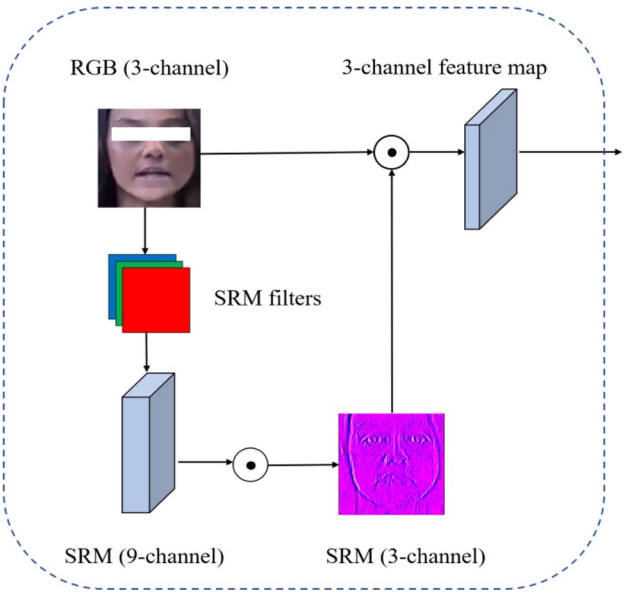

Fig. 3 Feature fusion module. The RGB space feature map is first filtered with three SRM filters to transform it into a 9-channel feature map in SRM space, then it is transformed into a 3-channel feature map using point convolution. Then, perform a point convolution operation on the 3-channel feature map in SRM space and the corresponding channel of the 3-channel feature map in RGB space. The output is a 3-channel feature map with fused features

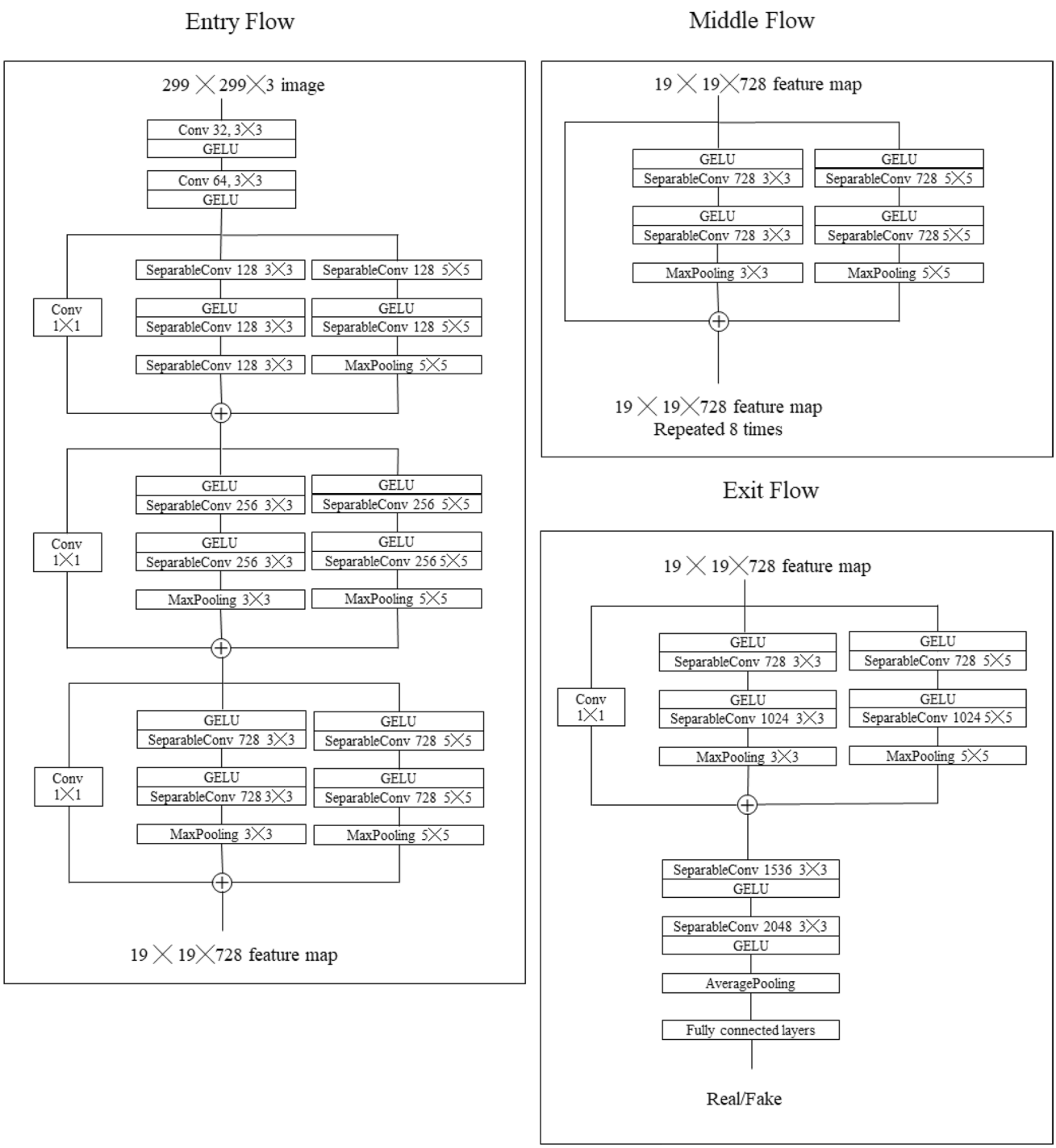

Fig. 4 Tception structure we proposed. Similar to the Xception structure, our proposed Tception structure is still divided into Entry Flow, Middle Flow and Exit Flow. We keep the separable convolution and residual structure of the Xception structure. Different from the Xception structure, we add a 5×5 separable convolutional stream to increase the perceptual field of the network structure and modify the activation function in the network to improve the performance of the network

$$Z_1 = F_1(X) + F_3(X) + F_5(X) \tag{2}$$

where $F_1(\cdot)$ is the result of a 1×1 convolution, $F_3(\cdot)$ is the result of a 3×3 convolution flow and $F_5(\cdot)$ is the result of a 5×5 convolution flow. In addition, the Z_1 is the output of the entry flow, which is a 728-dimensional feature map.

Next, the output Z_1 enters the middle flow residual block. Unlike the block in the entry flow, the residual

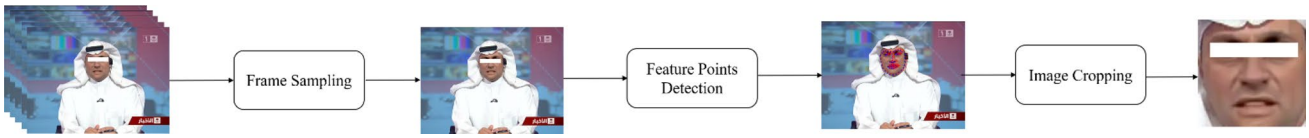

Fig. 5 Data pre-processing. First, we sample the frames of the video to obtain the image in the video. Then, we cut the image with the aid of Dilb detection of faces to obtain an image containing only faces

block does not perform 1×1 convolution operation, but uses residual concatenation, as shown in Eq. (3):

$$Z_2 = Z_1 + F_3(Z_1) + F_5(Z_1) \tag{3}$$

where Z_2 is the output of the middle flow, which is a 728-dimensional feature map.

Finally, in exit flow, after a block similar to the block in entry flow, two separable convolution layers are passed to obtain 2048-dimensional features. After averaging pooling, it can be discriminated.

It is worth mentioning that our proposed Tception structure does not use the ReLU activation function of the original structure. Instead, we use the GELU activation function, which adopted the idea of stochastic regularization. Compared to the ReLU function, the GELU function has another non-zero gradient in the negative region, thus avoiding the problem of dead neurons. In addition, GELU is smoother around 0 than ReLU, so it is easier to converge during the training process.

4 Experiments

4.1 Experiments' Setting

4.1.1 Datasets

We use the videos from HQ (c23) and LQ (c40) of the four tampering methods DeepFake (DF), Face2Face (F2F), FaceSwap (FS) and Neural Textures (NT) in the FaceForensics++ dataset after pre-processing to produce the dataset. We also complete partial comparison experiments on the Celeb-DF dataset to demonstrate the good general applicability of our method.

4.1.2 Data Pre-processing

First, the videos in the dataset are sampled every 16 frames to convert the video information into image information. Next, the 64 feature points of the face in the image are identified using the Dilb library, and the face image is truncated using these 64 feature points. The specific processing method is shown in Fig. 5.

Table 1 Comparative experiments

Methods	DF	F2F	FS	NT
Steg. features + SVM [25]	77.12	74.68	79.51	76.94
Cozzolino et al. [16]	81.78	85.32	85.69	80.6
Bayar and Stamm [17]	90.18	94.93	93.14	86.04
Rahmouni et al. [18]	82.16	93.48	92.51	75.18
MesoNet [19]	95.26	95.84	93.43	85.96
Xception [22]	**99.17**	98.60	98.63	93.12
Optical flow [24]	98.10	–	–	–
CNN + GRU + STN [26]	96.90	94.35	96.30	–
Tception (ours)	98.82	**98.80**	**99.08**	**94.17**

Compare our model on HQ quality videos. The evaluation metric is ACC. Bold represents the best result

Table 2 Comparative experiments

Methods	DF	F2F	FS	NT
Steg. features + SVM [25]	65.58	57.55	60.58	60.69
Cozzolino et al. [16]	68.26	59.38	62.08	62.42
Bayar and Stamm [17]	80.95	77.30	76.83	72.38
Rahmouni et al. [18]	73.25	62.33	67.08	62.59
MesoNet [19]	89.52	84.44	83.56	75.74
Xception [22]	95.01	86.67	89.39	90.50
I3D [27]	95.13	90.27	90.21	80.50
OpticalFlow [28]	–	81.6	–	–
SSTNet [29]	95.33	**90.48**	**94.04**	83.75
Tception (ours)	**95.92**	87.12	91.37	**91.68**

Compare our model on LQ quality videos. The evaluation metric is ACC. Bold represents the best result

4.1.3 Implementation Detail

He proposed model implement using the PyTorch framework and trained using the Adam optimizer (the default parameter). The learning rate is set to 0.001. A NVIDIA

Table 3 Comparative experiments

Methods	DF	F2F	FS	NT
Xception [22]	**99.84**	99.83	99.80	95.71
Tception (ours)	**99.84**	**99.91**	**99.93**	**97.83**

Compare our model on HQ quality videos. The evaluation metric is AUC. Bold represents the best result

Table 4 Comparative experiments

Methods	DF	F2F	FS	NT
Xception [22]	**98.73**	93.63	93.11	95.13
Tception (ours)	98.65	**94.10**	**95.13**	**97.40**

Compare our model on LQ quality videos. The evaluation metric is AUC. Bold represents the best result

Table 5 Experiments on the mixed dataset

Methods	ACC
Steg. Features + SVM [25]	70.97
Cozzolino et al. [16]	78.45
Bayar and Stamm [17]	82.97
Rahmouni et al. [18]	79.08
MesoNet [19]	83.10
Xception [22]	84.11
Tception (ours)	**87.61**

The evaluation metric is ACC. Bold represents the best result

Table 6 Comparative experiments in Celeb-DF

Methods	ACC
Xception [22]	97.31
Tception (ours)	**97.47**

The evaluation metric is ACC. Bold represents the best result

Tesla V100 GPU is used to the experiments. In our experiments, we use cross-entropy loss.

4.2 Comparative Experiments

We conduct experiments on the FaceForensics++ dataset and used accuracy (ACC) the evaluation metric. We conduct experiments at different compression rates of c23 and c40, respectively. The final experimental results are shown in Tables 1 and 2.

Our method has improved detection on the F2F, FS and NT methods on HQ quality. Our method offers a nearly 8% improvement in NT forgery detection compared to SSTNet, with similar detection accuracy of other forgery methods on LQ quality. We also used AUC as the evaluation metric. The results of the experiment are shown in Tables 3 and 4.

To further validate the robustness of proposed model, we test our model on a dataset with a mixture of the four tampering methods. We still use the FaceForensics++ dataset, where the real faces are keep constant and the forged faces account for about 1/4 of each of the four forgery methods. The results of the experiment are shown in Table 5.

Our method offers a nearly 3% improvement compared to Xception on the mixed dataset.

We also conduct experiments on the Celeb-DF dataset. The results are shown in Table 6. Our method has similar results compared to Xception.

Through comparative experiments, we find that our proposed method has a degree of improvement in both ACC and AUC metrics compared to other methods for image detection tasks generated by different tampering methods, and our method performs better on mixed datasets, reflecting the better robustness of our proposed method.

Table 7 Experimental setup of our ablation study

	Feature fusion module	Two-stream structure
X	–	–
XF	√	–
T	–	√
TF	√	√

Table 8 Results (HQ) of the ablation study

Methods	DF	F2F	FS	NT
X	**99.17**	98.60	98.63	93.12
XF	98.87	98.72	99.03	93.99
T	98.82	98.43	98.95	93.65
TF(ours)	98.82	**98.80**	**99.08**	**94.17**

The evaluation metric is ACC. Bold represents the best result

Table 9 Results (LQ) of the ablation study

Methods	DF	F2F	FS	NT
X	95.01	86.67	89.39	**91.93**
XF	94.84	87.24	89.97	87.76
T	95.53	**88.96**	91.07	87.15
TF(ours)	**95.92**	87.12	**91.37**	91.68

The evaluation metric is ACC. Bold represents the best result

4.3 Ablation Study

In this section, we perform a number of ablation studies to better understand the contribution of each component in our Tception structure. We set up the following experimental groups. X denotes Xception without Feature fusion module. XF denotes Xception with Feature fusion module. T denotes Tception without Feature fusion module. TF denotes Tception with Feature fusion module. The specific experimental setup is shown in Table 7. The experimental results are shown in Tables 8 and 9.

We find that when our structure containing both Feature fusion module and Two-stream structure has mostly achieved the highest accuracy. No matter which part is missing, the effect will decrease to varying degrees, which verified the rationality of our method.

We also find that the dual-stream network module is more effective than the feature fusion module in improving the original model, probably because the information in SRM space fused by the feature fusion module is obtained by transforming the information in RGB space, which is some kind of information enhancement of the information in RGB space, and the source of information is the same, and the information provided to the neural network learning may still be limited; the parallel structure of multi-sensory convolutional kernels can extract information at different scales, which is relatively more beneficial to the algorithm.

4.4 Comparison Experiments with Different Receptive Fields

The results of our experiments using different combinations of perceptual fields (convolutional kernel sizes) on the FF++ mixed dataset are shown in Table 10.

Table 10 shows that the Tception network consisting of two branches with 3×3 and 5×5 convolutional kernels is the best overall for face forgery detection. At the same time, we find that the overall network performance may not be satisfactory when the size of the convolutional kernels differs significantly. The reason may be the fact that after the convolution operation is performed, a padding operation is often required in order to unify the size of the feature maps. For networks consisting of branches composed of two convolutional kernels with large differences, the difference in the area filled by the Padding operation is also larger than the difference in the position of the unified features in the feature map, and the feature map may cause a feature shift when the Add operation is performed, thus reducing the detection performance of the network.

Table 10 Comparison experiments with different convolutional kernel sizes, Tception 3_5 in the table represents two branches using two convolutional kernels of 3×3 and 5×5, the others are similar

Methods	ACC
Tception 3_5	**87.61**
Tception 3_7	86.28
Tception 3_9	81.94
Tception 5_7	87.05
Tception 5_9	86.51

The evaluation metric is ACC. Bold represents the best result

4.5 Comparison Experiments with Different High-Pass Filters

We conduct experiments on the FF++ hybrid dataset using different high-pass filters and the results are shown in Table 11.

We find that each high-pass filter actually contributes to the performance improvement. The SRM filter works relatively well.

4.6 Comparison Experiments with Different SRM Filters

We conduct experiments on the FF++ hybrid dataset using different SRM filters and the results are shown in Table 12.

In Table 12, SRM_1 and SRM_2 used SRM filters with a single convolutional kernel size, and SRM_3 and SRM_4 used SRM filters with different convolutional kernel sizes. The specific filter weights are as followed.

The weights of SRM_1 are shown in Eq. (4):

Table 11 Comparison experiments with different high-pass filters

Methods	ACC
Sobel	87.05
DoG	86.48
SRM	**87.61**

The evaluation metric is ACC. Bold represents the best result

Table 12 Comparison experiments with different SRM filters. The evaluation metric is ACC

Methods	ACC
SRM_1	85.94
SRM_2	83.47
SRM_3	86.48
SRM_4	87.15
SRM_use	**87.61**

Bold represents the best result

$$\frac{1}{4}\begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} \tag{4}$$

The weights of SRM_2 are shown in Eq. (5):

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix} \tag{5}$$

The weights of SRM_3 are shown in Eq. (6):

$$\frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

The weights of SRM_4 are shown in Eq. (7):

$$\frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

We find that the filter combinations we used are slightly better than the other filter combinations. We also find hat using filters with different convolutional kernel sizes is generally better than using filters with a single convolutional kernel size. The reason for this may be that filters with different convolutional kernel sizes could extract features of different fineness.

4.7 Comparison Experiments with Different Activation Functions

We conducted experiments on the FF++ hybrid dataset using different activation functions and the results are shown in Table 13.

In Table 13, we find an improvement of about 2.5% on the FF++ mixed data set using the GELU activation function compared to the ReLU activation function.

Table 13 Comparison experiments with different activation functions

Methods	ACC
ReLU	85.04
GELU	**87.61**

The evaluation metric is ACC. Bold represents the best result

4.8 Visualization of Result

To further demonstrate the validity of our model, we give the CAM heat maps on a subset of the test samples to investigate the discriminatory basis of the neural network, and the results are shown in Fig. 6.

Through visual analysis, we find that our model can activate a wider range of features compared to the original Xception, resulting in a more well-founded and effective discriminant.

5 Conclusions

We propose a Tception structure that builds on Xception and expands the receptive field of the network by adding convolutional kernels of different sizes. At the same time, RGB streams and noise streams are used to learn rich features for image tampering detection. We extract noise

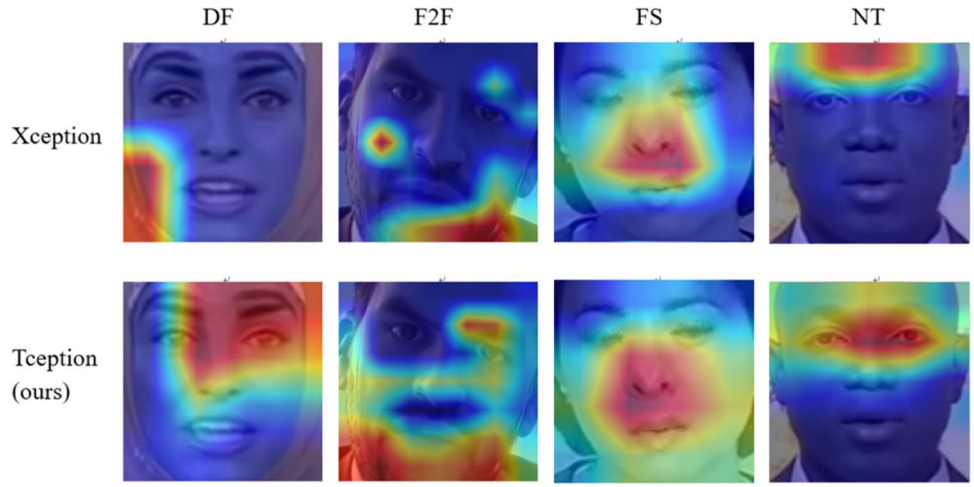

Fig. 6 The CAM heat map shows that our model can activate a much wider range of features than Xception

feature through an SRM filter layer to extract noise features and fuse them with features in RGB space adaptively, retaining features in RGB space and introducing features in noise space to achieve better results. The experiments show that our proposed method has improved performance compared to the Xception network. Compared to SSTNet, the detection accuracy of the Neural Textures is improved by nearly 8%. In the future, we will continue to investigate other feature fusion methods and carry out related work in other more complex cases (e.g., higher compression rates). It is worth noting that our proposed method does not perform best on all data generated by the falsification method and the generality of the model still needs to be improved.

Acknowledgements This paper was supported by the National Natural Science Foundation of China (No. 62072106), General Project of Natural Science Foundation in Fujian Province (No. 2020J01168), University industry Cooperation Project of Fujian Science and Technology Department (No. 2021H6004) and Open Project of Fujian Key Laboratory of Severe Weather (No. 2020KFKT04).

Author Contributions Conceptualization, BW and LH; methodology, LH; software, BW; validation, BW, LH, FY and TH; writing—original draft preparation, BW; writing—review and editing, TH and LH; visualization, FY; all the authors have read and agreed to the published version of the manuscript.

Funding This paper was supported by the National Natural Science Foundation of China (No. 62072106), General Project of Natural Science Foundation in Fujian Province (No. 2020J01168), University industry Cooperation Project of Fujian Science and Technology Department (No. 2021H6004), Open Project of Fujian Key Laboratory of Severe Weather (No. 2020KFKT04), Natural Science Foundation of Fujian Province (No. 2022J01190) and Education Department Young and Middle-aged Teachers Project of Fujian Province (No. JAT210053).

Availability of Data and Material The data that support the findings of this study are openly available, reference number [9, 11].

Declarations

Conflict of Interest None of the authors of this paper has a financial or personal relationship with other people or organizations that could inappropriately influence or bias the content of the paper.

Ethical Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Fernando, T., Fookes, C., Denman, S., et al.: Detection of fake and fraudulent faces via neural memory networks. IEEE Trans. Inf. Forensics Secur. **16**, 1973–1988 (2020)
2. Kong, C., Chen, B., Li, H., et al.: Detect and locate: exposing face manipulation by semantic-and noise-level telltales. IEEE Trans. Inf. Forensics Secur. **17**, 1741–1756 (2022)
3. Yang, J., Li, A., Xiao, S., et al.: MTD-Net: learning to detect deepfakes images by multi-scale texture difference. IEEE Trans. Inf. Forensics Secur. **16**, 4234–4245 (2021)
4. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., et al.: Deepfakes and beyond: a survey of face manipulation and fake detection. Inf. Fusion **64**, 131–148 (2020)
5. Qian, Y., Yin, G., Sheng, L., et al.: Thinking in frequency: face forgery detection by mining frequency-aware clues. In: European Conference on Computer Vision, pp. 86–103. Springer (2020)
6. Wang, J., Wu, Z., Ouyang, W., et al.: M2tr: multi-modal multi-scale transformers for deepfake detection. In: International Conference on Multimedia Retrieval, pp. 615–623. Springer (2022)
7. Zhang, X., Karaman, S., Chang, S. F.: Detecting and simulating artifacts in gan fake images. In: IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2019)
8. Yang, X., Li, Y., Lyu, S.: Exposing deep fakes using inconsistent head poses. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8261–8265 (2019)
9. Li, Y., Yang, X., Sun, P., et al.: Celeb-df: a large-scale challenging dataset for deepfake forensics. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3207–3216 (2020)
10. Dolhansky, B., Howes, R., Pflaum, B., et al.: The deepfake detection challenge (dfdc) preview dataset. arXiv preprint arXiv:1910.08854 (2019)
11. Rossler, A., Cozzolino, D., Verdoliva, L., et al.: Faceforensics++: learning to detect manipulated facial images. In: IEEE/CVF International Conference on Computer Vision, pp. 1–11 (2019)
12. Deepfakes github. https://github.com/ deepfakes/faceswap(2018)
13. Thies, J., Zollhofer, M., Stamminger, M., et al.: Face2face: real-time face capture and reenactment of rgb videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2387–2395 (2016)
14. Faceswap. https://github.com/MarekKowalski/FaceSwap (2018)
15. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Trans. Graph. (TOG) **38**(4), 1–12 (2019)
16. Cozzolino, D., Poggi, G., Verdoliva, L.: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 159–164 (2017)
17. Bayar, B., Stamm, M. C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10 (2016)
18. Rahmouni, N., Nozick, V., Yamagishi, J., et al.: Distinguishing computer graphics from natural images using convolution neural networks. In: IEEE Workshop on Information Forensics and Security (WIFS), pp 1–6 (2017)
19. Afchar, D., Nozick, V., Yamagishi, J., et al.: Mesonet: a compact facial video forgery detection network. In: IEEE International Workshop on Information Forensics and Security (WIFS), pp 1–7 (2018)
20. Zhao, H., Zhou, W., Chen, D., et al.: Multi-attentional deepfake detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2185–2194 (2021)

21. Sun, Z., Han, Y., Hua, Z., et al.: Improving the efficiency and robustness of deepfakes detection through precise geometric features. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3609–3618 (2021)
22. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
23. Nirkin, Y., Wolf, L., Keller, Y., et al.: DeepFake detection based on discrepancies between faces and their context. IEEE Trans. Pattern Anal. Mach. Intell. (2021). https://doi.org/10.1109/TPAMI.2021.3093446
24. Li, J., Xie, H., Li, J., et al.: Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6458–6467 (2021)
25. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **7**(3), 868–882 (2012)
26. Sabir, E., Cheng, J., Jaiswal, A., et al.: Recurrent convolutional strategies for face manipulation detection in videos. arXiv preprint arXiv: 1905.00582 (2019)
27. Wang, Y., Dantcheva, A.: A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes. In: IEEE International Conference on Automatic Face and Gesture Recognition (FG), pp. 515–519 (2020)
28. Amerini, I., Galteri, L., Caldelli, R., et al.: Deepfake video detection through optical flow based CNN. In: IEEE/CVF International Conference on Computer Vision Workshops (2019)
29. Wu, X., Xie, Z., Gao, Y. T., et al.: Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2952–2956 (2020)