A Dual-Branch Neural Network for DeepFake Video Detection by Detecting Spatial and Temporal Inconsistencies

Liang Kuang[1,2] • Yiting Wang[3] • Tian Hang[1] • Beijing Chen[1,4*] • Guoying Zhao[5]

Abstract It has become a research hotspot to detect whether a video is natural or DeepFake. However, almost all the existing works focus on detecting the inconsistency in either spatial or temporal. In this paper, a dual-branch (spatial branch and temporal branch) neural network is proposed to detect the inconsistency in both spatial and temporal for DeepFake video detection. The spatial branch aims at detecting spatial inconsistency by the effective EfficientNet model. The temporal branch focuses on temporal inconsistency detection by a new network model. The new temporal model considers optical flow as input, uses the EfficientNet to extract optical flow features, utilize the Bidirectional Long-Short Term Memory (Bi-LSTM) network to capture the temporal inconsistency of optical flow. Moreover, the optical flow frames are stacked before inputting into the EfficientNet. Finally, the softmax scores of two branches are combined with a binary-class linear SVM classifier. Experimental results on the compressed FaceForensics++ dataset and Celeb-DF dataset show that: (a) the proposed dual-branch network model performs better than some recent spatial and temporal models for the Celeb-DF dataset and all the four manipulation methods in FaceForensics++ dataset since these two branches can complement each other; (b) the use of optical flow inputs, Bi-LSTM and dual-branches can greatly improve the detection performance by the ablation experiments.

Keywords DeepFake video detection; optical flow; convolution neural network; long short-term memory network

1 Introduction

Digital images and videos have filled our lives and become an indispensable part of social network. They contain a large

✉ Beijing Chen
Tel.: +86 25 58 73 13 23
E-mail address: nbutimage@126.com

[1] School of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2] School of IoT Engineering, Jiangsu Vocational College of Information Technology, Wuxi 214153, China
[3] Warwick Manufacturing Group, University of Warwick, Coventry CV4 7AL, UK
[4] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China
[5] Center for Machine Vision and Signal Analysis, University of Oulu, Oulu 90014, Finland

amount of information and are easy to understand. However, the popularity of image editing software and technologies, especially for the development of deep learning, makes image tampering easier and easier [1-3]. Furthermore, the tampering can leave no obviously visible traces of any modification [4]. In particular, a new AI-based fake video generation methods known as DeepFake has attracted much attention recently [5]. It is a technique that can superimpose a face image of a target person to a video of a source person and then create a video of the target person doing or saying things that the source person does. DeepFake videos can be abused to fool the public or even cause political or religious tensions between countries [6-7]. It has been applied to create videos of some countries' leaders with fake speeches for falsification purposes, such as US President Obama and Trump. At the same time, DeepFake has also be used to exchange stars' faces on pornographic videos for illegal profits [6]. Accordingly, there is an urgent need for reliable and effective methods to expose DeepFake videos.

Until now, DeepFake video detection methods have relied on either spatial or temporal inconsistencies. From the perspective of the spatial inconsistency, each frame will inevitably have artifact when it is generated. Therefore, many recent works [8-13] have effectively detected DeepFake video via the artifact detection. However, with the development of the image forgery technology, such artifacts have been more and more challenging to capture [10]. In addition, the compression of the video in transmission makes it more difficult to detect because the image quality is significantly degraded [12]. Therefore, the temporal (inter-frame) inconsistency detection methods are proposed [14-23]. According to the studies in [21], the temporal inconsistency is easier to be captured and achieves the higher detection accuracy than the spatial inconsistency. However, the detection performance still to be improved.

So, both the spatial and temporal inconsistencies are considered in this paper with a dual-branch neural network. The major contributions of this paper are summarized as follows:

(a) A dual-branch (spatial branch and temporal branch) neural network is proposed to detect the inconsistency in both spatial and temporal for DeepFake video detection.

(b) A new temporal network model adopted as the temporal branch is constructed to capture the temporal inconsistencies between frames. This model considers optical flow as input, uses the EfficientNet [24] to extract optical flow features, and

utilizes Bidirectional Long-Short Term Memory (Bi-LSTM) network [25] to capture the temporal inconsistency of optical flow. Moreover, the optical flow frames are stacked before inputting into the EfficientNet.

The rest of the manuscript is organized as follows. Section 2 presents the related works about DeepFake video detection. Section 3 explains the proposed model. Experiment results and analysis are presented in Section 4. The findings are concluded in Section 5.

2 Related works

The research of DeepFake video detection has been primarily driven by the advances of image classification technologies. Currently, all the existing works focus on detecting the inconsistency in either spatial or temporal. The spatial inconsistency of videos can be found within a frame. For example, Yang et al. [8] used the head posture estimation to distinguish real faces from fake faces through the face marker estimation and central region estimation. Matern et al. [9] utilized artifacts of eyes and teeth to expose DeepFakes. On the other hand, some works [10-13] introduced deep learning to learn discriminative features or find manipulation traces within a frame. For instance, Afchar et al. and Rossler et al. proposed the well-known MesoNet model [10] and Xception model [11], respectively. Recently, Li et al. [12] presented an approach called face X-ray for DeepFake detection. They adopt a framework based on convolutional neural network (CNN) to extract the face X-ray of an input image and then to output the probability of the input image being real or blended. Moreover, Khalid et al. [13] proposed the OC-FakeDect, which uses a one-class Variational Autoencoder (VAE) to train only on real face images and detects DeepFake images by treating them as anomalies.

The temporal inconsistency is also an important clue for DeepFake video detection. Agarwal et al. [14] simplified each 10-second video clip into a 190-dimensional facial expression feature vector, which was used to classify video into real or fake by Support Vector Machine (SVM). With the breakthrough development in deep learning, some deep learning-based works have been studied. Ciftci et al. [15] presented the FakeCatcher, a fake portrait video detector based on biological signals. They generated video-based biological signal maps and employed a Convolutional Neural Networks (CNN) to detect synthetic content

using the generated maps. Amerini et al. [16] proposed to use the optical flow fields to exploit possible inter-frame dissimilarities. Optical flow frames are then used as inputs of a CNN-based model to detect fake faces. Some recent works [17-20] utilized the recurrent neural networks (RNNs) [26] combined with the CNN. For example, long-term recurrent CNN (LRCN) [27] was adopted to make eye-blinking detection in DeepFake videos [17]. Convolutional LSTM architecture was used to detect DeepFake videos in [18]. Furthermore, Sabir et al. [19] proposed to use Bi-LSTM combined with the DenseNet [28] to find inconsistent features across frames. Then, Chen et al. [20] improved the architecture in [19] by introducing a superpixel-wise binary classification unit, which is specifically designed to guide the backbone network to particularly focus on the differences between forged face with its surrounding regions. Recently, Li et al. [21] presented sharp multiple instance learning (S-MIL) for DeepFake video detection. They designed a new spatial-temporal instance to capture the inconsistency between faces, which can help to improve the accuracy of DeepFake detection. What's more, DeepFake video discriminators with a 3D convolutional network [29] is also introduced in [22, 23] since the 3D convolutional network is able to extract motion features encoded in adjacent frames in the video.

It can be seen from the above analysis that, on the one hand, the works by the spatial inconsistency are effective in still fake face image detection but not suitable to capture the variation in a DeepFake video. On the other hand, the works using temporal inconsistency is more suitable for capturing the variation but they pay less attention to subtle artifacts within a frame. Therefore, just like human beings, it is necessary to take both the spatial and temporal inconsistencies into consideration. The spatial anomaly can be used as an effective complementary clue of the temporal anomaly. With the abundant information from two aspects, the model should be more robust and effective.

3 Proposed model

In this section, we describe the proposed dual-branch neural network model for DeepFake video detection. Before the introduction of the proposed model, some pre-processing technologies are described in the subsection 3.1.

3.1 Pre-Processing

3.1.1 Face cropping

According to [19], face cropping is beneficial for classification. So, it is employed here as well. All the faces in the videos are cropped frame by frame. Inspired by its success in the face detection, the MTCNN [30] is used to locate the human face in the image. It leverages a cascaded architecture that consists of three parts: Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). P-Net is a deep convolutional network used to classify face and non-face in the frame by generating a candidate window, while the R-Net is a network to reject false candidates from the P-Net. Finally, O-Net is adopted to output five facial landmarks: the left and right mouth corners, the center of the nose, and the centers of the left and right eyes. After detecting the input frames with MTCNN, we crop all the regions where faces are detected and resize them to 224 × 224 pixels. Fig. 1 shows an example face cropped from a frame.

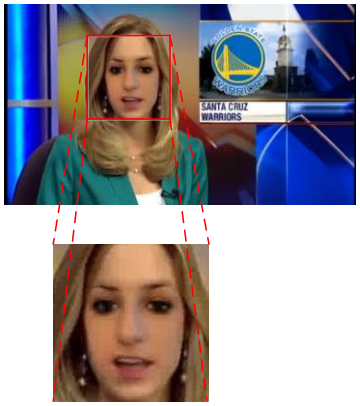


Figure 1. Extraction the specified face from a frame

3.1.2 Optical flow

Gibson [31] first introduced the optical flow method to extract foreground object movement information in the videos. Specially, the objective of the optical flow method is to find a disparity map u. The target of u is to minimize an image-based error criterion together with a regularization force as

$$\int_{\Omega}\{\lambda\Phi(I_0(x)-I_1(x+u(x)))+\varphi(u,\nabla u,...)\}dx \tag{1}$$

where I_0 and I_1 are the two frames, $\varphi(u,\nabla u,...)$ represents the regularization term inducing the shape prior,

$\Phi(I_0(x)-I_1(x+u(x)))$ is the image data fidelity, and λ is the weight between the regularization force and the data fidelity.

Currently, there are four types of optical flow algorithms in general [32]: frequency-based, phase-based, match-based and gradient-based. Compared to the other three types, the gradient-based algorithms are simple and easy to calculate. Moreover, the optical flow by the gradient-based algorithms can describe the motion trajectory more accurately. The TV-L_1 algorithm is a commonly-used gradient-based algorithm. It utilizes the L_1 norm and can calculate the large offset of frames. Specially, in TV-L_1 algorithm, two functions in (1) are chosen as $\Phi(x)=|x|$ and $\varphi(\nabla u)=|\nabla u|$, then (1) yields to

$$E=\int_{\Omega}\{\lambda|I_0(x)-I_1(x+u(x))|+|\nabla u|\}dx \tag{2}$$

The detailed solution of (2) can be found in [33].

3.2 Dual-branch architecture

The proposed dual-branch neural network model consists of a spatial branch and a temporal branch. The spatial branch is dedicated to detecting the artifact in the RGB frame, while the temporal branch is used to detect the temporal inconsistency through a series of optical flow frames. The overall architecture is shown in Fig. 2. As shown in Fig. 2, each branch performs detection on its own. The softmax scores of two branches are combined with a binary-class linear SVM classifier for the final classification.

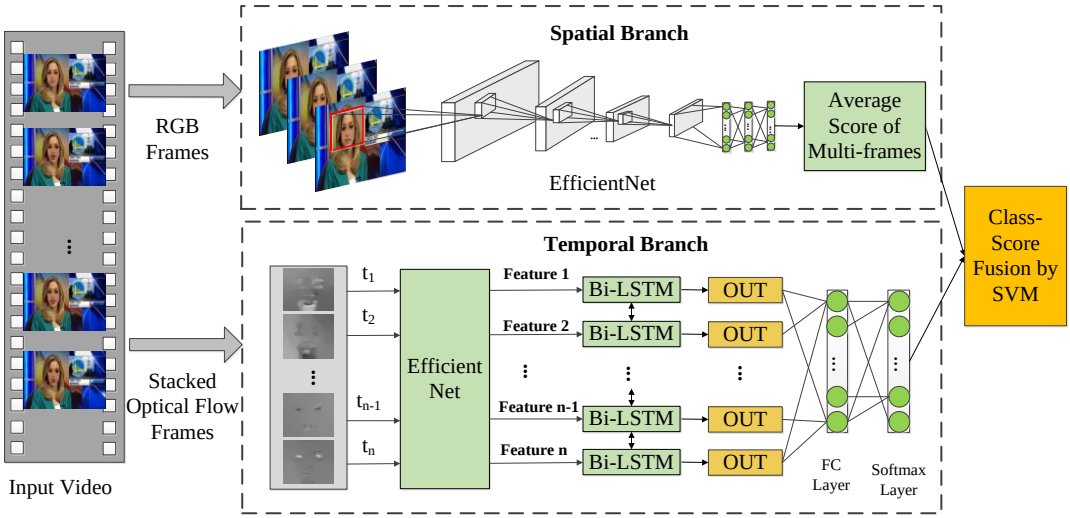


Figure 2. Structure of the dual-branch neural network

3.2.1 Spatial branch

The spatial branch takes a sequence of RGB frames of a video as input and outputs a sequence of the corresponding softmax scores. These scores are averaged as the final score of the video. Here, the EfficientNet model [24] is used to detect each frame because it transfers well and has achieved better performance in the DeepFake Detection Challenge (DFDC) [34] than some existing CNNs such as AlexNet [35], GoogleNet [36], and Xception [37]. The baseline EfficientNet-B0 network structure used in this paper is shown in Fig. 3. As we can see in Fig. 3, there are 16 MBConv layers, 2 conv layers, 1 pooling layer and 1 fully connected layer. The main building block is mobile inverted bottleneck convolution MBConv [38], which can reduce the computational cost by a factor proportional to the number of channel. The MBConv architecture is show in Fig. 4 as MBConv1, and MBConv6. The DWConv denotes depthwise conv, k3×3/k5×5 denotes kernel size, BN is batch norm and $H \times W \times F$ denotes the tensor shape (height, width, depth). It is transferred to our task by replacing the last fully connected layer with two outputs. Cross entropy is adopted as the loss function by,

$$L_{CE} = y \log \hat{y} + (1-y) \log(1-\hat{y}) \tag{3}$$

where $\hat{y}$ represents the final predicted probability, and y is set to 1 if the face image is manipulated, otherwise it is set to 0.

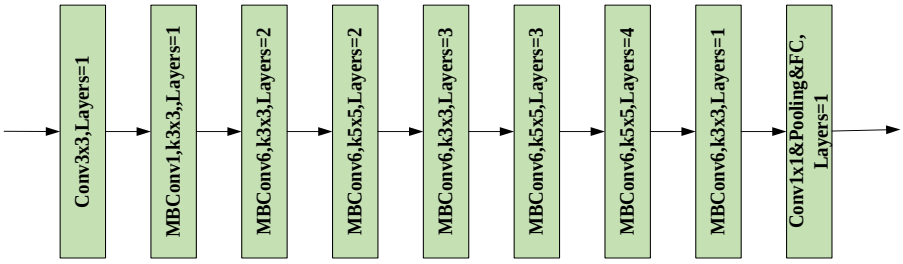
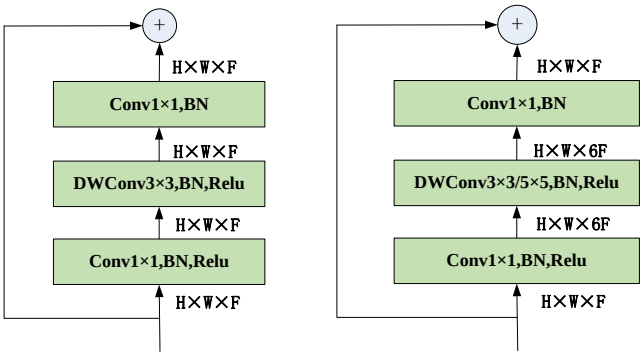


Figure 3. Structure of the EfficientNet-B0 network



(a) MBConv1 (b) MBConv6

Figure 4. MBConv architecture

To better understand how our spatial branch works, Gradient-weighted Class Activation Mapping (Grad-CAM) [39] is employed to compare and visualize the learned features in spatial branch. The Grad-CAM takes a simple RGB frames as input and outputs a coarse localization map, which highlights the important regions in the image for prediction after passing into the final layer. Some results are illustrated in the Fig. 5. It can be clearly observed that the EfficientNet effectively focuses on the important regions that people pay attention to, such as the eyes, nose, mouth, etc. The softmax score for each frame is shown at the bottom of the figure. The number in the red box represents the probability that the RGB frame is considered as fake, and the number in the green box represents the probability that the RGB frame is considered as real.

Figure 5. Hotmaps extracted with Grad-CAM from a frame sequence. The last row shows the softmax score of each frame.

3.2.2 Temporal branch

Unlike the spatial branch, the input of the temporal branch is a sequence of consecutive optical flow frames. The proposed temporal branch is based on the EfficientNet [24] and Bi-LSTM network [25]. The EfficientNet is used to extract the optical flow feature. The Bi-LSTM network is used to capture the temporal inconsistency introduced by the face swapping process.

In the temporal branch, we adopt the modified EfficientNet B0 by removing the fully-connected layer to output the feature vector of each frame directly. Notice that the EfficientNet should be pre-trained with optical flow data set. Then the representations, 1280-dimensional feature vectors, are used as the sequential input of Bi-LSTM. Finally, the Bi-LSTM is followed by a 1024 fully-connected layer with 0.5 chance of dropout. The cross entropy loss in (3) and the softmax activation function are also considered in the temporal branch.

(a) Stacked optical flow frames

A dense optical flow can be seen as the horizontal displacement vector fields d_τ^U and vertical displacement vector fields

d_τ^V between the pairs of consecutive frames τ and $\tau + 1$. By averaging $d_\tau^U(x, y)$ and $d_\tau^V(x, y)$, we denote the final displacement vector $d_\tau(x, y)$ at the pixel (x, y). The optical flow frames from the corresponding video frames are shown in Fig. 6. As we can see from the second row, the optical flow frames are single-channel grayscale images in where some areas highlight the moving parts on the frames. However, motion representation may not always be obvious in optical flow frames due to the face slight movement. So, every L consecutive optical flow frames (e.g. L=5) are stacked to one frame. The objective is to fuse the adjacent optical flows frames to represent motion information over time. Specially, let w and h be the width and height of an optical flow frame, the input volume $D_\tau \in R^{w \times h}$ of the EfficientNet for an arbitrary stacked optical flow frame τ is constructed as,

$$D_\tau(u, v) = \max\{d_\tau(u, v), d_{\tau+1}(u, v), ..., d_{\tau+L-1}(u, v)\},\ u = 1, 2, ...w,\ v = 1, 2, ...h. \quad (4)$$

The pixel values of optical flow frame $D_\tau(u, v)$ range from 0-255. The greater the value, the larger the motion. The optical flow stacking method shown in (4) is quite different from that in [40].

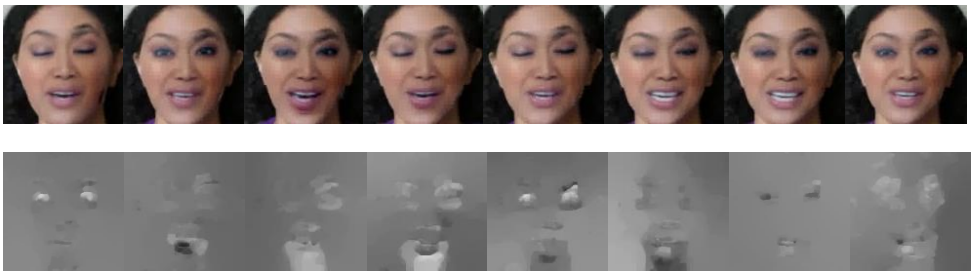


Figure 6. Optical flow frames from the corresponding video frames. The first row is the consecutive video frames, and the bottom is the corresponding optical flow frames.

Fig. 7 presents the stacked (L = 5) and non-stacked (L = 1) optical flow frames. It can be observed from the Fig. 7 that the stacked optical flow frames contain far more sufficient information than the non-stacked ones.

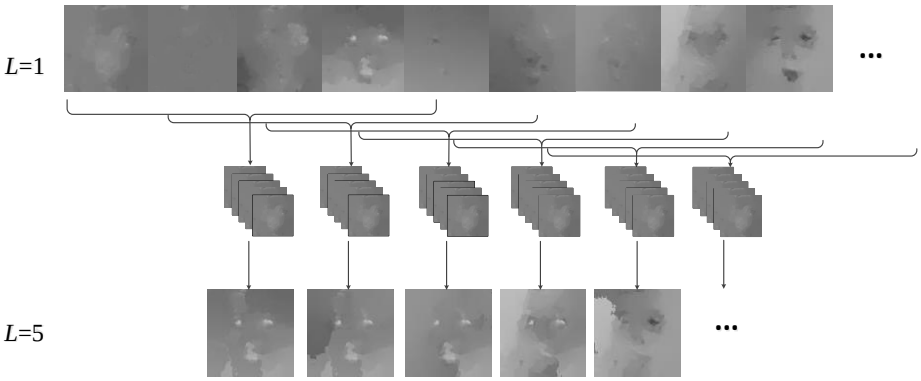


Figure 7. Stacked and non-stacked optical flow frames. The first row is the non-stacked frames (L=1), and the bottom is the stacked frames (L=5)

(b) Bi-LSTM

The RNN is a neural network that is specialized for processing sequential inputs. It is well-suited to process non-linear dynamics and temporal information. So, the Bi-LSTM, a special RNN, is adopted in our temporal branch to capture temporal

inconsistency. The LSTM network is a typical type of RNNs improved by adding cell state to the hidden layer. The cell state is preserved by three unique structures: forgetfulness gate, input gate, and output gate. Further more, as an improved version of the LSTM, the Bi-LSTM has two LSTMs stacked on the top of each other. One RNN goes in the forward direction, while the other one goes in the backward direction. Then, the outputs of the two RNNs are combined. Fig. 8 shows the architecture of the Bi-LSTM.

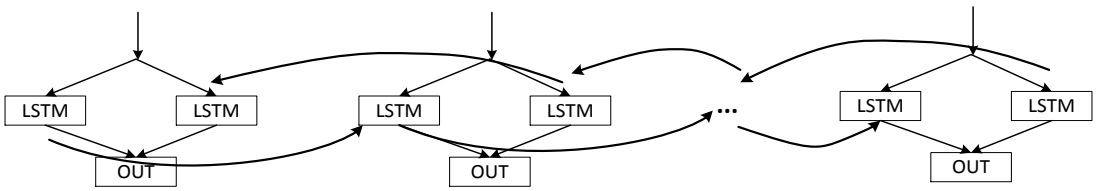


Figure 8. Bi-LSTM architecture

3.2.3 Pseudo-code

In order to make the proposed model clear and help the reader to implement it, Table 1 shows the pseudo-code of the proposed model.

Table 1. Pseudo-code of the proposed model

Algorithm 1: Pseudo-code of the proposed model
Input: Video V
1: Extract RGB frames from DeepFake videos as $I_{RGB} = Ext(V)$;
2: Crop the frames to extract specified faces as $I_{C\text{-}RGB} = Crop(I_{RGB})$;
3: Calculate optical flow frames as $I_{C\text{-}OF} = TV\text{-}L1(I_{C\text{-}RGB})$;
4: if step == Train do
5: Calculate the softmax scores of each frame in spatial branch as S_{S_score} = Spatial_branch_model(I_{C_RGB});
6: Calculate the loss of spatial branch as $Ls\_{CE} = CrossEntropy\_Loss(S_{S\_score}, Label)$;
7: Minimize Ls_{CE} and update the network parameters by back propagation;
8: Obtain the spatial branch model;
9: Calculate the softmax scores of the consecutive optical flow frames in temporal branch as S_{T_score} = Temporal_branch_model ($I_{C\text{-}OF}$);
10: Calculate the loss of the temporal branch as $L_{T\_CE} = CrossEntropy\_Loss(S_{T\_score}, Label)$;
11: Minimize L_{T_CE} and update the network parameters by back propagation;
12: Obtain the spatial branch model;
13: Average the softmax scores in spatial branch as $S_{S\_score\_avg}=Avg(\sum S_{S\_score})$;
14: Fuse the decision of two branches as $S_{fused} = SVM(S_{S\_score\_avg}, S_{T\_score})$;
15: Calculate the fused loss of the dual branches as $L_{SVM} = Hinge(S_{fused}, Label)$;
16: Minimize L_{SVM} and update the SVM network parameters by back propagation;
17:end if
18: if step == Test do
19: Calculate the softmax scores of each frame in the spatial branch with the trained spatial branch model as S_{S_score} = Spatial_branch_model(I_{C_RGB});
20: Obtain the average softmax scores of all the frames in spatial branch as $S_{S\_score\_avg} = Avg(\sum S_{S\_score})$;
21: Calculate the softmax scores of the temporal branch with the trained temporal branch model as S_{T_score} = Temporal_branch_model(I_{C_OF});
22: Predicted result $S_{fused} = SVM(S_{S\_score\_avg}, S_{T\_score})$);
23: end if
Output: *Predicted result* (real or fake)

4 Experimental results and analysis

In this section, the proposed DeepFake video detection model is evaluated. We first introduce the overall experimental setups and then present some experiments to prove the superiority of our model. In addition, two ablation experiments are conducted to prove the validity of both temporal branch and two branches fusion.

4.1 Experimental setups

4.1.1. Experimental datasets

The widely used datasets FaceForensics++ (FF++) and Celeb-DF are considered here for evaluation. A brief description of each dataset is provided below.

FaceForensics++ (FF++). There are 1000 original videos and 4000 fake videos in the raw sub-dataset without compression. The fake videos are generated by four different manipulation methods: DeepFake, Face2Face, FaceSwap, and Neural Texture (NT). Each of them has 1000 fake videos. Moreover, every video is compressed with two different compression ratios, obtaining two compressed sub-datasets (C23 and C40). The higher the compression ratio, the more difficult the detection. In this paper, we only evaluate the sub-dataset with the highest compression ratio (C40) because the accuracies of the raw sub-dataset and C23 sub-dataset have been greater than 98% by the Xception model shown in [11]. For the experiments, the C40 dataset is split into three sets (training, validation, and test sets). The three sets are constitute of 720,140, and 140 videos, respectively. Moreover, 32 frames are sampled from each video and conducted the face cropping. Some examples of cropped faces are shown in Fig. 9.

Celeb-DF. It is a new DeepFake dataset generated by using a refined synthesis algorithm that can reduce the visual artifacts effectively. It contains 590 real videos and 5639 synthesized videos. The videos have themes of different ages, races and genders. For experiments, the 590 real videos are randomly cut into 1000 clips. Each clip also contains 32 frames. So do the 5639 synthesized videos. Then, the total 2000 clips are divided into three sets (training, validation, and test sets) with the same ratio as the FF++ dataset. Some examples of cropped faces are also provided in Fig. 9.

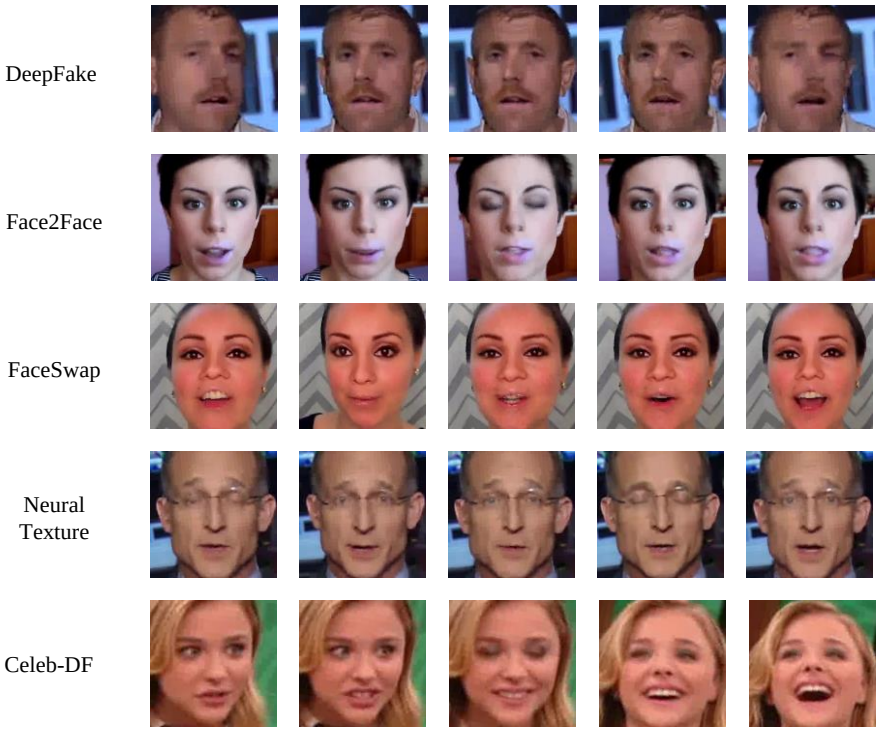


Figure 9. Examples of cropped faces in FF++ dataset and Celeb-DF dataset. From top to bottom are generated by DeepFake, Face2Face, FaceSwap, Neural Texture and Celeb-DF, respectively.

Notice that in the spatial branch all the 32 cropped faces of a video are taken as inputs separately and the average softmax score of 32 face frames is regarded as the score of the spatial branch. In the temporal branch, it takes 32 consecutive stacked optical flow frames as input.

4.1.2. Parameters

In the spatial branch, the batch size is set to 16. The ADAM optimizer is used with the initial learning rate 1.0e-4, which decays to half of original if the accuracy is not improved in 5 epochs. The maximum number of iterations is set to 200, and the early stop is also adopted.

In the temporal branch, the batch size of is set to 4 and the ADAM optimizer is used with the initial learning rate 1.0e-5. We set the time step to 32, the number of optical flow frames. The total number of iterations is set to 1000. Regarding the EfficientNet used in both branches, we consider EfficientNet-B0 and re-train it on the basis of the pre-trained model in ImageNet.

4.2 Evaluation the improvements on the temporal branch

The following metric *Accuracy* is used to measure the performance of DeepFake video detection,

$$Accuracy = (TP + TN) / (TN + FP + TP + FN) \times 100\%, \quad (5)$$

where *TP* denotes the number of correctly predicted fake video cases, *FP* is the number of the normal cases that are misclassified

as fake video, *TN* represents the number of the normal cases that are correctly classified, and *FN* denotes the number of the fake video cases that are misclassified as normal cases.

Firstly, in order to verify the improvements on the temporal branch, an ablation study is conducted by comparing the basic EfficientNet with three improved versions: EfficientNet+LSTM (L=1), EfficientNet+Bi-LSTM (L=1), and EfficientNet+Bi-LSTM (L=5). The ablation experimental results are shown in Table 2. Notice that, for the EfficientNet, the softmax scores of optical flow frames are averaged as the final video-level prediction. It can be observed from the Table 2 that: (a) the models combining with LSTM greatly improves the performance of the model with the EfficientNet only because the LSTM can effectively capture the temporal inconsistency in DeepFake videos; (b) the Bi-LSTM performs better than the LSTM due to the bidirectional detection of Bi-LSTM; (c) using the stacked optical flow frames can achieve the better performance than using the non-stacked optical flow frames because the stacked frames carry more motion-dependent information than the non-stacked ones.

Table 2. Ablation experimental results in the temporal branch by the detection accuracy (%)

Models	FF++				Celeb-DF
	DeepFake	Face2Face	FaceSwap	NT	
EfficientNet	70.83	75.00	72.50	66.67	71.5
EfficientNet+LSTM (L=1)	88.21	86.07	83.57	75.71	82.14
EfficientNet+Bi-LSTM (L=1)	90.00	86.43	89.29	76.43	84.29
EfficientNet+Bi-LSTM (L=5)	96.43	90.0	93.21	85.00	96.43

Then, in order to test the performance of the proposed temporal branch, it is also compared with three existing temporal models, i.e., C3D [23], CNN-LSTM [19], and Sharp Multiple Instance Learning (S-MIL) [21]. Moreover, the input of each compared model is a sequence of consecutive stacked optical flow frames. The comparative results are presented in Fig. 10. The results show that our model is much better than C3D and S-MIL due to the use of the EfficientNet and Bi-LSTM. The average accuracy of our model reaches 92% for the two datasets. Even for the most difficult NT subdataset, the accuracy is still 85%. The results also demonstrate that our model outperforms the CNN-LSTM based on DenseNet because of the effective of the EfficientNet.

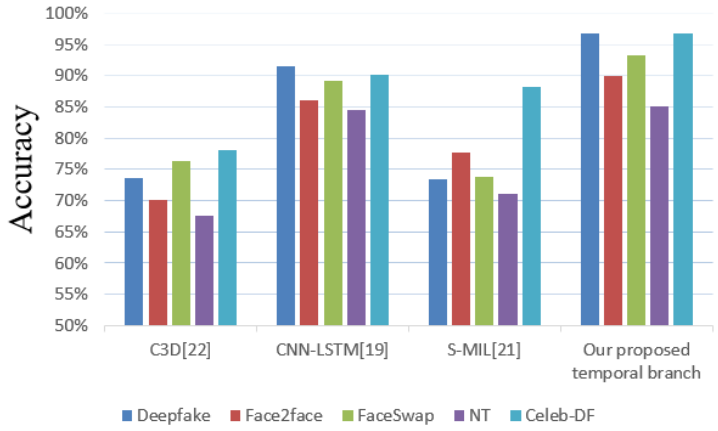


Figure 10. Accuracies (%) of different temporal models on the FF++ dataset generated by four different manipulation methods and Celeb-DF dataset

4.3 Comparison with some existing models

Firstly, in order to verify the improvements of dual branches, the second ablation experiment is conducted here. The proposed dual-branch model is compared with two models with each branch only. Table 3 shows the comparative results. It can be observed from the Table 3 that: (a) the temporal branch achieves a better overall performance than the spatial branch because the temporal inconsistency is usually more obvious than the spatial inconsistency; (b) the proposed dual-branch model obtains the accuracies over 90% for all the five types of manipulation methods and is superior to the other two models with each branch only due to the consideration of both the temporal inconsistency and spatial inconsistency.

Table 3. Ablation experimental results in dual-branch fusion by the detection accuracy (%)

Models	FF++				Celeb-DF
	DeepFake	Face2Face	FaceSwap	NT	
Spatial branch	91.43	92.50	92.86	80.71	95.36
Temporal branch	96.43	90.00	93.21	85.00	96.67
Proposed dual-branch	98.21	95.00	93.57	90.71	98.57

Then, the proposed dual-branch model is compared with seven state-of-the-art models. The compared seven models contains three spatial ones (MesoNet [10], Xception [11], and OC-FakeDec [13]) and four temporal ones (Optical Flow Features [16], C3D [22], CNN-LSTM [19], and S-MIL [21]). Notice that here the inputs of three temporal models (C3D, CNN-LSTM, and S-MIL) are the RGB frames instead of optical flow frames. The comparative results are given in Table 4. The results for the OC-FakeDect and S-MIL models are taken from their corresponding literatures [13] and [21], respectively. The results in the Table 4 show our proposed dual branches model outperforms others in most cases, achieving the overall best performance among eight compared models. In addition, the temporal models are usually better than the spatial models, which is consistent with the results given in the Table 3. Besides, eight compared models achieves a better performance on Celeb-DF dataset than FF++

dataset. This is because the FF++ dataset we actually used is C40 with the highest compression ratio, which is more challenging than Celeb-DF. The satisfactory results of the proposed model are attributed to the following main reasons: (a) both the spatial and temporal inconsistency are considered, and they complement each other; (b) the optical flow frames are used as the input for the temporal branch; (c) the effective models EfficientNet and Bi-LSTM are applied into two branches.

Table 4. Comparison results with other state-of-the-art models by the detection accuracy (%)

Models		FF++				Celeb-DF
		DeepFake	Face2Face	FaceSwap	NT	
Spatial	MesoNet [10]	84.64	67.50	74.29	67.86	85.71
	Xception [11]	92.14	90.71	92.50	80.71	93.57
	OC-FakeDect [13]	88.35	71.20	86.05	97.45	89.03
Temporal	Optical Flow Features [16]	72.86	76.43	69.29	64.64	75.00
	C3D [22]	90.71	83.21	91.79	75.71	93.21
	CNN-LSTM [19]	94.64	89.29	94.29	81.43	95.36
	S-MIL [21]	97.14	91.07	96.07	86.79	98.84
Spatial + Temporal	Proposed dual-branch	98.21	95.00	93.57	90.71	98.93

5 Conclusions

In this paper, we propose a dual-branch network model to detect DeepFake video by the inconsistency in both spatial and temporal. The experimental results and analysis on the FF++ dataset and Celeb-DF dataset show that the spatial and temporal inconsistency can complement each other so that the proposed model achieves a better performance than some existing spatial models and temporal ones. Moreover, the combination of EfficientNet and Bi-LSTM can capture the temporal inconsistency more effectively. For the future work, since all the current works including our work focus on DeepFake video detection in plaintext, we will try to detect the encrypted DeepFake videos for privacy protection.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62072251, Natural Science Research Project of Jiangsu Universities under Grant 20KJB520021, Higher Vocational Education Teaching Fusion Production Integration Platform Construction Projects of Jiangsu Province under Grant No. 2019(26), the PAPD fund.

References

[1] R. Tolosana, R.Vera-Rodriguez, J. Fierrez, et al., “Deepfakes and beyond: A survey of face manipulation and fake detection,” Information Fusion, vol. 64, pp. 131-148, 2020.

[2] B. Chen, W. Tan, and G. Coatrieux, et al., “A serial image copy-move forgery localization scheme with source/target distinguishment,” IEEE Transactions on Multimedia. 2020. DOI: 10.1109/TMM.2020.3026868.

[3] L. Verdoliva, “Media forensics and deepfakes: an overview,” IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 5, pp. 910-932, 2020.

[4] D. Zhang, X. Chen, F. Li, et al., “Seam-carved image tampering detection based on the cooccurrence of adjacent LBPs,” Security and Communication Networks, pp. 1-12, 2020.

[5] Y. Li, and S. Lyu, “Exposing deepfake videos by detecting face warping artifacts,” in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 46-52, 2018.

[6] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, et al., “Deep learning for deepfakes creation and detection,” 2019, https://arxiv.org/abs/1909.11573.

[7] B. Chen, X. Ju, B. Xiao, et al., “Locally GAN-generated face detection based on an improved Xception,” Information Sciences, vol. 572, pp. 16-28, 2021.

[8] X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses,” in: Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2019), pp. 8261-8265, 2019.

[9] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations, ” in: Proceedings of the 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW2019), pp. 83-92, 2019.

[10] D. Afchar, V. Nozick, J. Yamagishi, et al., “Mesonet: a compact facial video forgery detection network,” in: Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS2018), pp. 1-7, 2018.

[11] A. Rossler, D. Cozzolino, L. Verdoliva, et al., “Faceforensics++: learning to detect manipulated facial images, ” in: Proceedings of the 2017 IEEE/CVF International Conference on Computer Vision, pp. 1-11, 2019.

[12] L. Li, J. Bao, T. Zhang, et al., “Face x-ray for more general face forgery detection,” in: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2020), pp. 5001-5010, 2020.

[13] H. Khalid, and S. S. Woo, “OC-FakeDect: classifying deepfakes using one-class variational autoencoder, ” in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, vol. 656-657, 2020.

[14] S. Agarwal, H. Farid, Y. Gu, et al., “Protecting world leaders against Deep Fakes," in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 38-45, 2019.

[15] U. A. Ciftci, I. Demir, and L. Yin, “FakeCatcher: detection of synthetic portrait videos using biological signals,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. DOI: 10.1109/TPAMI.2020.3009287.

[16] I. Amerini, L. Galteri, R. Caldelli, et al., “Deepfake video detection through optical flow based CNN,” in: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshops, pp. 1205-1207, 2019.

[17] Y. Li, M. Chang, and S. Lyu, “Exposing ai created fake videos by detecting eye blinking,” in the 2018 IEEE International Workshop on Information Forensics and Security (WIFS2018), pp. 1-7, 2018.

[18] D. Guera, and E. J. Delp, “Deepfake video detection using recurrent neural networks,” in: Proceedings of the 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS2018), pp. 1-6, 2018.

[19] E. Sabir, J. Cheng, A. Jaiswal, et al., “Recurrent convolutional strategies for face manipulation detection in videos,” in: Proceedings of the 2018 IEEE/CVF International Conference on Computer Vision Workshops, pp. 80-87, 2019.

[20] P. Chen, J. Liu, T. Liang, et al., “FSSPOTTER: spotting face-swapped video by spatial and temporal clues,” in: Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME2020), pp. 1-6, 2020.

[21] X. Li, Y. Lang, Y. Chen, et al., “Sharp multiple instance learning for deepfake video detection,” in: Proceedings of the 28th ACM International Conference on Multimedia, vol. 1864-1872, 2020.

[22] I. Ganiyusufoglu, L. M. Ngô, N. Savov, et al., “Spatio-temporal features for generalized detection of deepfake videos,” 2020, https://arxiv.org/abs/2010.11844.

[23] O. D. Lima, S. Franklin, S. Basu, et al., ”Deepfake detection using spatiotemporal convolutional networks,” 2020, https://arxiv.org/abs/2006.14749.

[24] M. Tan, and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks, “ in : Proceedings of 2019 International Conference on Machine Learning, pp. 6105-6114, 2019.

[25] M. Schuster, and K.K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[26] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” 2014, https://arxiv.org/abs/1409.2329.

[27] J. Donahue, L. A. Hendricks, S. Guadarrama, et al., “Long-term recurrent convolutional networks for visual recognition and description,” in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), pp. 2625-2634, 2015.

[28] G. Huang, Z. Liu, L. V. D. Maaten, et al., “Densely connected convolutional networks,” in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017), pp. 2261-2269, 2017.

[29] D. Tran, H. Wang, L. Torresani, et al., “A closer look at spatiotemporal convolutions for action recognition,” in: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2018), pp. 6450-6459, 2018.

[30] K. Zhang, Z. Zhang, Z. Li, et al., “Joint face detection and alignment using multitask cascaded convolutional networks,” IEEE Signal Processing Letters, vol.23, no. 10, pp. 1499-1503, 2016.

[31] J. J. Gibson, “The perception of the visual world,” Houghton Mifflin, 1950.

[32] J. L. Barron, D. J. Fleet, S. S. Beauchemin, et al., “Performance of optical flow techniques,” International Journal of Computer Vision, 1992.

[33] C. Zach, T. Pock, and H. Bischof, “A duality based approach for realtime TV-L1 optical flow,” in: Proceedings of the 29th DAGM Conference on Pattern Recognition, pp. 214–223, 2007.

[34] DeepFake Detection Challenge (DFDC). https://ai.facebook.com/datasets/dfdc/

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional Neural Networks,” Neural Information Processing Systems, vol.25, pp. 1097-1105, 2012.

[36] C. Szegedy, W. Liu, Y. Jia, et al., “Going deeper with convolutions.” in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), pp. 1-9, 2015.

[37] F. C. Google, “Xception: deep learning with depthwise separable convolutions,” in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017), pp. 1800-1807, 2017.

[38] M. Sandler, A. Howard, M. Zhu, et al., “MobileNetV2: inverted residuals and linear bottlenecks,” in: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2018), pp. 4510–4520, 2018.

[39] R. R. Selvaraju, M. Cogswell, and A. Das, “Grad-cam: visual explanations from deep networks via gradient-based localization,” in: Proceedings of the 2017 IEEE International Conference on Computer Vision (CVPR2017), pp. 618-626, 2017.

[40] K. Simonyan, A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” Advances in Neural Information Processing Systems, vol. 27, pp. 568–576, 2014.