Detecting Compressed Deepfake Videos in Social Networks Using Frame-Temporality Two-Stream Convolutional Network

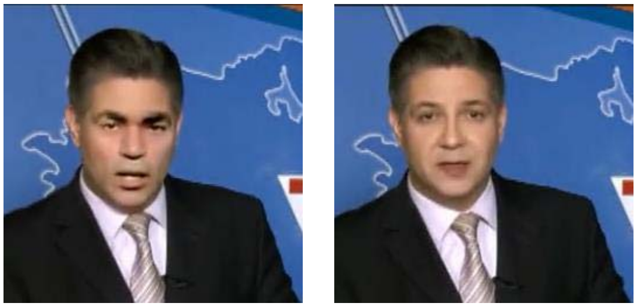
Juan Hu, Xin Liao, *Member, IEEE*, Wei Wang, *Member, IEEE*, and Zheng Qin, *Member, IEEE*

***Abstract*—The development of technologies that can generate Deepfake videos is expanding rapidly. These videos are easily synthesized without leaving obvious traces of manipulation. Though forensically detection in high-definition video datasets has achieved remarkable results, the forensics of compressed videos is worth further exploring. In fact, compressed videos are common in social networks, such as videos from Instagram, Wechat, and Tiktok. Therefore, how to identify compressed Deepfake videos becomes a fundamental issue. In this paper, we propose a two-stream method by analyzing the frame-level and temporality-level of compressed Deepfake videos. Since the video compression brings lots of redundant information to frames, the proposed frame-level stream gradually prunes the network to prevent the model from fitting the compression noise. Aiming at the problem that the temporal consistency in Deepfake videos might be ignored, we apply a temporality-level stream to extract temporal correlation features. When combined with scores from the two streams, our proposed method performs better than the state-of-the-art methods in compressed Deepfake videos detection.**

***Index Terms*—Video forensics, compressed Deepfake videos, frame-level stream, temporality-level stream.**

I. Introduction

WITHIN the past decade, Internet traffic has shifted dramatically from text pages to multimedia files [1]. Besides, the emergence of large-scale multimedia social software such as Instagram, Tiktok, and WeChat has brought about great changes in our lives. It can not only enrich the lives of people but also make people share their lives in more convenient ways. With the development of video generation technology, multimedia data can be used for various purposes. It makes positive in entertainment, artistic expression, and social interaction, but poses a threat to political security, public security, and personal privacy at the same time [2]. The combination of forgery techniques and AI technology dramatically improves the indistinguishability of digital media [3]. As we can see in the left of Fig. 1, the faces are forged, and we can hardly see anything unusual with our naked eyes. What is more, there are fake videos about Obama on the Internet. These videos show that Obama is altered to say false statements [4]. With the fake information transmission through social networks, the influence of forgery information can be magnified by 10 million times in an instant. People who view the fake video of Obama may fall for the video content, which causes a negative effect on politics. In addition, the appearance of forgery technology aggravates public distrust and causes a serious crisis of public trust. It may also result in the disclosure of personal privacy data, the telecommunications-fraud, and social justice harm [5].



(a) Fake (b) Original

Fig. 1. The example of an image (right) being forged (left) by using the technique named DeepFakes.

Nowadays, compressed videos are widely used in social networks. The reason is that the uncompressed videos take up a lot of storage, but our device has limited memory. Furthermore, if there is no high network bandwidth, the transmission speed of high definition videos will be quite slow. In social media, when a user uploads a video to Tiktok, the video will be compressed by the Tiktok. If a user sends a video with social software such as Wechat and Instagram, the size of the videos is so restrictive that the users have to compress it and upload it again. If criminals deliberately spread compressed fake videos, it will make it difficult for us to detect the

Manuscript received March 3, 2021; accepted April 14, 2021. Date of publication April 20, 2021; date of current version March 9, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61972142, Grant 61972395, and Grant 61772191; in part by the Hunan Provincial Natural Science Foundation of China under Grant 2020JJ4212; in part by the Key Lab of Information Network Security, Ministry of Public Security under Grant C20611; and in part by the Science and Technology Program of Changsha under Grant kq2004021. This article was recommended by Associate Editor J.-M. Guo. *(Corresponding author: Xin Liao.)*

Juan Hu and Zheng Qin are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: hujuan@hnu.edu.cn; zqin@hnu.edu.cn).

Xin Liao is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: xinliao@hnu.edu.cn).

Wei Wang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wwang@nlpr.ia.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3074259.

Digital Object Identifier 10.1109/TCSVT.2021.3074259

forgery video. In order to solve the problem that seeing is not believing, the forensic of compressed Deepfake videos becomes an important issue.

In the pioneering work, researchers presented methods for detecting removing objects in videos [6], copying objects in videos [7], moving objects in videos [8], interpolating frames in videos [9], deleting frames in videos [10], fake bitrate videos [11], and deinterlaced videos [12]. Since the Deepfake videos generate a face by using artificial intelligence without leaving obvious traces, their methods cannot be used for Deepfake videos detection. To detecting Deepfake videos, there are some frame-based methods [13]–[15] for video detection. However, these frame-based detection methods do not fully expose the characteristics of videos. The study in [16]–[19] combined temporal information and spatial information to improve the performance in video classification. Unfortunately, the method in [16] cannot be utilized to detect the fake videos with normal blink frequency, and other methods [17]–[19] cannot extract the tampered artifacts from the compressed Deepfake videos. It is because compressed Deepfake videos exist some artifacts produced by compression. These compression artifacts would be mixed with the artifacts left by tampering.

This paper proposes a method that detects compressed Deepfake videos by learning frame-level features and temporality-level features. On the one hand, the compressed videos add a lot of redundant information to frames such as compression artifacts. Our frame-level stream can prune the redundant connections to prevent the invalid connections from affecting the final prediction. Besides, we extract the I-frames instead of the whole frames of the video, which can reduce the training efficiency. On the other hand, since fake videos ignore the temporal consistencies during the synthesis process, the temporality-level stream is utilized to capture temporal features. The two streams will be trained independently and then be fused. The main contributions of this paper are as follows:

1) The frame-level stream extracts the features without being influenced by compression artifacts. The temporality-level stream is utilized to extract the time-dependent features introduced by the face-swapping process. By utilizing both frame-level stream and temporality-level stream to train the detection model, the proposed method detects both tempered artifacts and the inconsistency between the frames of compressed Deepfake videos.

2) The proposed frame-level stream significantly reduces the scale of training datasets, which may further reduce the time costs of model training. Specifically, we utilize the I-frames instead of the whole frames to train the frame-level stream, which significantly reduces the scale of the training dataset and thus reduces the training time costs without sacrificing the detection accuracy.

3) Our proposed method is evaluated over benchmarks of manipulated facial video detection datasets. Extensive experimental evaluations demonstrate that the proposed method performs well on Celeb-DF [20] and FaceForensics++ [21] datasets. Specifically, our proposed method is robust to the compression factor.

The rest of this paper is organized as follows. Section II illustrates the related works, including 3 aspects: AI-based video synthesis algorithms, video-based digital media forensics, and Deepfake videos forensics. Section III analysis the compressed Deepfake videos from frame-level and temporality-level. Section IV presents the details about the proposed compressed Deepfake videos detection method. Section V provides the implementation of our method and assesses our proposed method through a set of experiments. Finally, the conclusion and future works are made in Section VI.

II. Related Work

In this section, we describe a few conventional and deep learning techniques for generating Deepfake videos. Next, the previous related works on video-based digital media forensics and Deepfake videos forensic are introduced as follows.

A. Video Synthesis Algorithms

Previously, realistic videos were generated using detailed 3D computer graphics models. For example, Face2Face [22] is a graphics-based approach for real-time facial reenactment of a monocular target video sequence. At run time, the program uses a dense photometric consistency measure to track facial expressions of both source and the target video. Then, the faces of the source sequence are warped to produce an accurate fit with the target sequence. Finally, the synthesized target face is rendered to the corresponding video stream. Similar to Face2Face, FaceSwap [23] is a graphics-based approach to transfer the face region from a source video to a target video. First, it extracts the face region by using sparse detected facial landmarks. Then, using the textures of the input image to minimize the difference between the projected shape and the localized landmarks. Finally, the rendered model is blended with the image, and the color correction is applied. These steps are performed for all pairs of source and target frames until one video end. For the Neural Texture [24] synthesis method, 3D reconstruction of images is done in imperfect geometry conditions and produced at real-time rates.

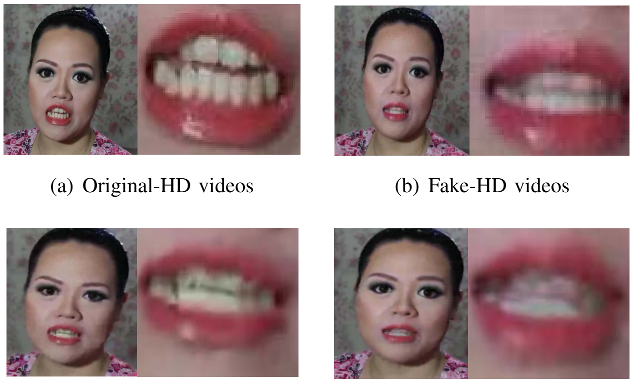
Recently, the development of new deep learning algorithms makes the synthesized video more realistic. DeepFakes [25] sets an encoder and two decoders, puts the extracted faces of two people into the model, the encoder learns the common features of the two people, and the two decoders are trained separately to regenerate the faces of the two people, respectively. Then, supervised learning is used to compare the difference between input and output, making the difference smaller. Subsequently, many general works have been proposed based on the idea of GAN [26]. Liu *et al.* [27] proposed an unsupervised image-to-image translation framework based on Coupled GANs, which is the basis for the DeepFakes algorithm. GAN typically consists of two networks: the generator network and the discriminator network. The first step in building GAN is to clarify the output and create a training data set for the generator. Once the generator starts to produce acceptable output, these videos are input to the discriminator. Then, let the discriminator distinguish the video. Every time the discriminator judges that the video clip is false, it will provide the generator with a loss. As the generator can generate more realistic videos, the discriminator will recognize them more and more accurately. On the contrary, as the discriminator's recognition ability increases, the level of the generator to synthesize fake videos will continue to increase.

B. Video-Based Digital Media Forensics

In recent years, various popular schemes are designed for video-based digital media forensics. Aloraini *et al.* [6] proposed a spatiotemporal filter based on sequential and patch analyses to detect object removal forgery. D'Amiano *et al.* [7] computed the feature on a spatio-temporal grid and singled out areas with coherent spatio-temporal displacement as candidate copy-moves. For automatic detection of object-based video forgery, Chen *et al.* [8] developed an approach for automatic identification and forged segment localization of object-based forged video encoded with advanced frameworks. By analyzing the distribution of residual energies within interpolated frames, Ding *et al.* [9] observed that there exist strong correlations between artifact regions and high residual energies. They proposed a robust motion-compensated frame interpolation (MCFI) detector to locate interpolated frames. Feng *et al.* [10] developed a feature based on frame motion residuals to identify frame deletion points. What is more, forgers can create fake bitrate videos by up-converting the bitrate of original videos with lower video quality to attract more viewers on video sharing websites. To tackle these disadvantages, He *et al.* [11] proposed a detection method for fake bitrate videos using a hybrid deep-learning network from recompression error. Besides, Wang and Farid [12] described two techniques for detecting traces of tampering in deinterlaced and interlaced video. For deinterlaced video, they quantified the correlations introduced by the camera or software deinterlacing algorithms. For interlaced video, they proposed an efficient way to measure the motion between fields of a single frame and across fields of neighbor frames. However, these video-based forensics methods cannot be used for Deepfake videos detection. It is because the Deepfake videos are generated by the artificial intelligence method without removing, copying, moving objects, which makes it can be quite challenging to detect.

C. Deepfake Videos Forensics

There are also several countermeasures based on frame-level that have been proposed to detect Deepfake videos. Typically, Li and Lyu [13] found there are face warping artifacts in fake videos, and they showed the performance of Deepfake videos detection by extracting face warping artifacts. Nguyen *et al.* [14] used part of the VGG19 network to extract the latent features and input features into a capsule network to detect forged images and videos. Afchar *et al.* [15] started their experiments with MesoNet, which can focus on the mesoscopic properties of videos with a low number of layers and train a neural network in a supervised fashion. While these techniques show great promise in high-definition videos, these frame-based methods ignore the characteristics of videos. Therefore, their detection accuracy is worth improving.



(a) Original-HD videos (b) Fake-HD videos

(c) Original-Compressed videos (d) Fake-Compressed videos

Fig. 2. The analysis of the frame-level. In the first line, the high definition videos are examples showing artifacts from imprecise geometry estimation. Unlike the teeth of the real face (a), the teeth of the fake face (b) are generated as a structureless white blob. In the second line, the video is compressed, and the compression artifacts are generated. Both the teeth of the real face (c) and fake face (d) are generated as a structureless white blob, which brings a lot of noise to frame-level features.

In addition to frame-level papers, there are also temporal-spatial methods. Li *et al.* [16] detected videos by judging the blink frequency of the person in the video. If the blink frequency of a video is lower than normal, the video is deemed to be fake. However, the continuous development of forgery techniques based on generative adversarial networks makes the techniques generate the fake video with normal blink frequency, which means that the method of blink frequency is invalid. Moreover, Güera and Delpv [17] proposed a temporal-aware pipeline to detect Deepfake videos automatically. They used a convolutional neural network to extract frame-level features. These features are then used to train a recurrent neural network. While the method of Güera and Delpv [17] showed promising performance, it has its drawback in compressed Deepfake videos detection. The possible reason is that the compressed videos bring a lot of compression noise, which influences the spatial feature extraction. When the spatial features with compression noise are input to the network, it will affect the detection performance.

III. The Analysis of Compressed Deepfake Videos

In this section, we analyze the compressed Deepfake videos from two aspects: frame-level and temporality-level. At the frame-level, compressed videos mainly increase some artifacts compared with high definition videos. At the temporality-level, temporal inconsistencies between frames would be created during the process of generating Deepfake videos.

A. The Analysis of Frame-Level

For the uncompressed Deepfake videos, we can spot artifacts arising from an imprecise estimation of the underlying geometry. Such artifacts are shown in the top row of Fig. 2. Unlike the real face, the fake face from the Deepfake videos appears as a single white blob instead of individual teeth. The difference between real and fake videos makes the forensics of uncompressed videos relatively easy.

TABLE I

Measure the Noise Level and Distortion by Calculating the Visual Quality of the Videos Before and After Compression. These Videos Contain 100 Real Videos and 100 Fake Videos That Are Randomly Selected From FaceForensic++

Index	PSNR	SSIM	VIF	RECO	DIQaM-FR	DeepQA
Value	30.10	0.89	0.35	0.54	0.61	0.74

When the videos are compressed, each frame of the image sequence is divided into multiple pixel blocks. Since pixel blocks are processed differently, the correlation between blocks would be eliminated. This will result in the appearance of the block boundary compression artifacts. As shown in the bottom row of Fig. 2, both the compressed real video and compressed fake video produce compression artifacts, which adds noise to the data and makes it difficult to detect.

In addition, the application of quantization and inverse quantization in the compression process will result in quantization noise and distortion. These factors can also cause visual artifacts. We use PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Metric) [28], VIF (Visual Information Fidelity) [29], RECO (Relative Polar Edge Coherence) [30], DIQaM-FR (Deep Image Quality Measure for FR-IQA) [31], and DeepQA (Deep Image Quality Assessment) [32] to measure noise levels. As shown in Table I, the PSNR and SSIM between the uncompressed video and the compressed video are 30.10 dB and 0.89, which means the compression adds noise information and has a certain distortion. The values of VIF and RECO indicate that there exists noise in the visual and edge. DIQaM-FR compares the frames before and after the compression and measures the distortion of compressed frames. The value of DIQaM-FR represents that there is indeed distortion in compressed frames. DeepQA can generate the perceptual error map as an intermediate result, which provides us with intuitive analysis of local artifacts for given distorted images. The value of DeepQA represents that there are redundant artifacts in compressed frames. Therefore, we use the convolution neural network with low-complexity and gradually prune the connections to avoid being influenced by redundant information.

B. The Analysis of Temporality-Level

The temporal weakness is inherent to the generation process of the Deepfake videos itself. As we can see in Fig. 3, face A is forged to a fake face by using the Deepfake technique. First, gathering the aligned faces of face A and face B from their videos respectively. Then, training the encoder to reconstruct the faces. The decoder will reconstruct a face based on the facial information in the video. It is worth pointing out that the face A and face B share the same encoder but use two different decoders. Once the optimization of the decoder and encoder is done, any image containing a face of A can be encoded through this shared encoder. After the forged face A is generated, these fake images are used to create a Deepfake video. Because the encoder generates the videos frame by frame, it is completely unaware of any previously generated face. This lack of temporal awareness is the source of multiple anomalies.

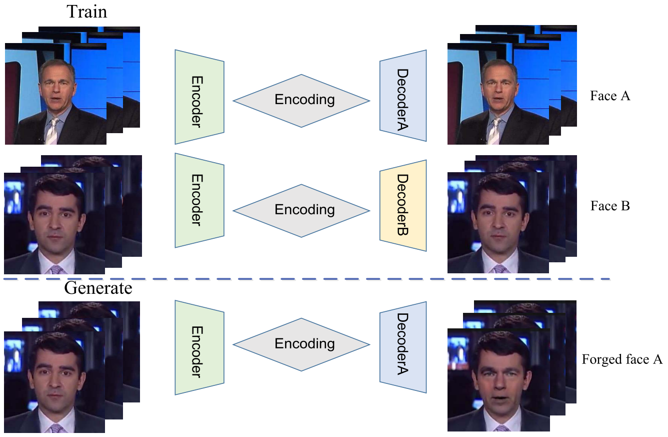


Fig. 3. The example from FaceForensics [21] shows that the process of generating fake videos is carried out frame by frame, which makes the temporal inconsistencies between frames introduced. These video anomalies can be exploited to detect if a video under analysis is a deepfake manipulation or not.

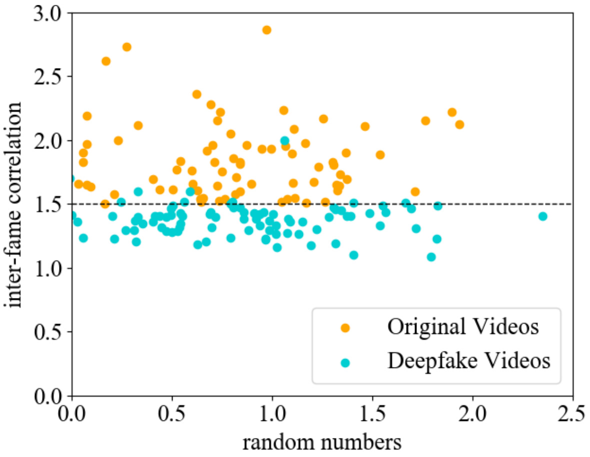


Fig. 4. The comparison of the inter-frame correlation between original and fake videos. Since the temporal correlation between frames is ignored in fake videos, the correlation of neighbor frames in fake videos is lower than that of the original videos.

Then, we analyze videos by calculating the inter-frame correlation information, which represents the temporal perception of a video sequence. The inter-frame correlation is represented by the Hamming distance between the current frame and the previous frame. The lower the Hamming distance of two adjacent frames, the stronger the relevancy between them. We randomly select 100 original videos and 100 fake videos manipulated by DeepFakes and Face2Face. The results are shown in Fig. 4. The ordinate represents the reciprocal after summing the Hamming distances of all adjacent frames in the video. Thus, the sequence with a higher inter-frame correlation has a higher ordinate. As expected, it can be found that the correlation of neighbor frames in the original video is higher than those of the altered video.

Since the inter-frame correlation of compressed Deepfake videos is different from that of compressed real videos, it is essential to extract time-dependent features. The residual features of the temporality-level stream are used for detecting compressed Deepfake videos. If the residual feature is not a time-independent feature, the inter-frame correlation feature cannot be extracted. Then, the temporality-level stream cannot detect the difference between original and fake video. Therefore, it is important to prove that the residual features are time-dependent features.

Let t be the current time, and P be the current P frame. (x, y) indicates the position of the macroblock in the P frame. Then, the information of the macroblock in the P frame can be expressed as:

$$P_t(x, y) = MV_t(x, y) + R_t(x, y) \tag{1}$$

where $P_t(x, y)$ represents the information of the macroblock in the P frame at the current time, $MV_t(x, y)$ represents the motion vector of the macroblock in the P frame at the current time, $R_t(x, y)$ represents the residual of the macroblock in the P frame at the current time. Since both the compression coding standards H.264 and H. 265 employ various sizes of the coding unit and flexible block partition, we use $(x-\alpha, y-\gamma)$ represents random position coding unit. According to the macroblock motion vector prediction method, we have:

$$MV_t(x, y) = P_t(x-\alpha, y-\gamma) \tag{2}$$

Combine Eq. (1) and Eq. (2):

$$\begin{aligned} P_t(x, y) &= P_t(x-\alpha, y-\gamma) + R_t(x, y) \\ &= P_t(0, 0) + \sum_{a=0}^{x-\alpha} R_t(a, y) + \sum_{b=0}^{y-\gamma} R_t(0, b) + R_t(x, y) \end{aligned} \tag{3}$$

Since the macroblock $P_t(0, 0) = 0$ at the position of (0, 0), Eq. (3) can be rewritten as:

$$P_t(x, y) = \sum_{a=0}^{x-\alpha} R_t(a, y) + \sum_{b=0}^{y-\gamma} R_t(0, b) + R_t(x, y) \tag{4}$$

H.264 and H.265 employ flexible block partition. If the time domain difference of the current macroblock in P frame is D_t, we have:

$$D_t = P_{t-\beta}(x', y') - P_t(x, y) \tag{5}$$

where $P_{t-\beta}(x', y')$ is the best matching macroblock of the current macroblock in reference frame $P_{t-\beta}$. Then combine Eq. (4) and Eq. (5):

$$\begin{aligned} D_t = &\sum_{a=0}^{x'-\alpha} R_{t-\beta}(a, y') + \sum_{b=0}^{y'-\gamma} R_{t-\beta}(0, b) + R_{t-\beta}(x', y') \\ &- (\sum_{a=0}^{x-\alpha} R_t(a, y) + \sum_{b=0}^{y-\gamma} R_t(0, b) + R_t(x, y)) \end{aligned} \tag{6}$$

The Eq. (6) has only two variables, $R_{t-\beta}$ and R_t. $t-\beta$ and t are variables representing time. Thus, we conclude that the residual of the macroblock has a strong relationship with the temporal information. Then, we use the residual features to capture the inconsistencies between frames.

IV. The Proposed Method of Compressed Deepfake Videos Detection

In this section, we firstly illustrate an overview of the proposed method. Then, the details of the frame-level stream and temporality-level stream are introduced, respectively. Furthermore, we describe the algorithm for the training procedure and explain the losses of two streams. Finally, the two-stream score fusion is introduced.

A. Overview of Our Proposed Method

Fig. 5 shows our proposed method of compressed Deepfake videos detection. The method includes two streams: frame-level stream and temporality-level stream. The frame-level stream is a CNN trained with a pruning module to classify whether the frame is tampered or authentic. First, we construct the input data by extracting the I-frames from the video, increasing the data diversity, and cleaning up the interference data. Next, the data are put to MesoNet [15], which can extract mesoscopic properties of images. To avoid the model fitting the compression noise, we prune MesoNet. It consists of three steps: train connectivity, prune connections, and train weights. Finally, we get the probability values of labels.

The compressed video contains not only I-frames but also P-frames. The I-frames are used for the frame-level stream. The temporality-level stream is trained on time-dependent features of videos, which are bound up with P-frames. It first cleans the videos to keep the data the same as the frame-level stream. According to the analysis of Section III, the inter-frame consistency is ignored when generating the fake video, and the residual feature is a time-dependent feature. Thus, the residual features are utilized for the detection of temporal inconsistency. Next, we use Resnet18 [34] to model the features. In the end, the model will output the probability values of positive and negative, respectively. To combine the two streams, we construct a series of strategies for final scores. Interestingly, we found a simple softmax of scores to work well (see Section V for details).

B. Frame-Level Stream

As we can see in Section III, the compression brings artifacts noise to the data. To prevent the model from fitting the noise in data, we choose a compact mesoscopic network and prune the network, which can reduce the complexity of the model and improve the generalization of the model. The method of the frame-level stream is explained in detail in the following.

For the frame-level stream S_1, the first step in our proposed method is to extract the frames from the compressed videos. We analyze the compressed videos and find the I-frames can retain the complete information during the encoding process. The compressed videos lose lots of high-frequency information, and we need to input as much information as possible into the network without affecting the training efficiency. Instead of grabbing images frame by frame, we put the I-frames into the network. To further enlarge the training diversity, we cropped the face of the I-frames in multiscale. After cropping, some images contain only faces, and others include

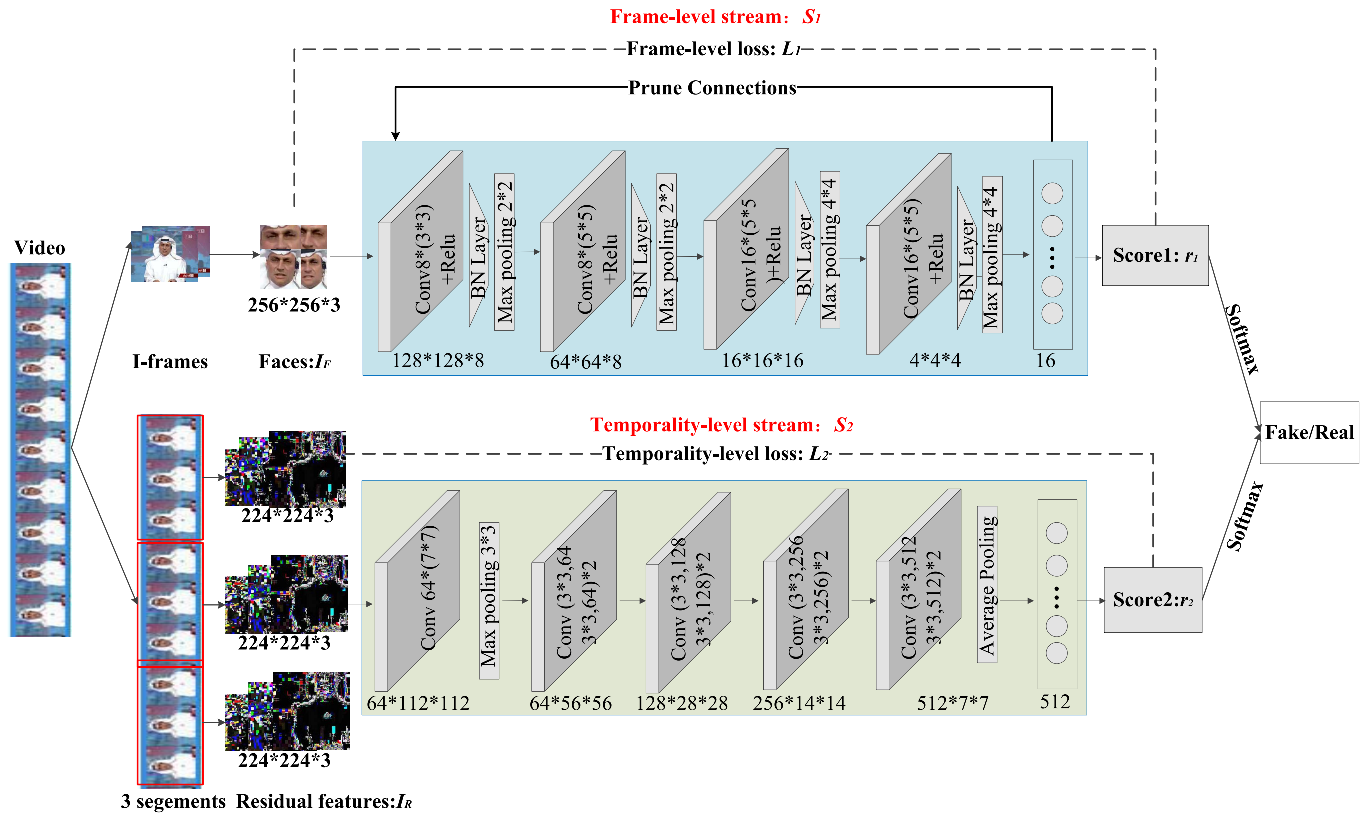


Fig. 5. The proposed two-stream method for detecting the compressed Deepfake videos. To avoid the model fitting the compression noise, our proposed frame-level stream uses a compact model with a pruning module. During generating the manipulated faces in Deepfake videos, intra-frame inconsistencies and temporal inconsistencies between frames are created. Thus, the temporality-level stream is utilized to capture temporal features between frames.

faces and surrounding areas. When cutting the face, there is a small number of data that is mistaken for the face. We clean up these interference data to prevent them from affecting the result.

Next, the faces I_F are put into MesoNet [15]. It focuses on the mesoscopic properties of videos, which are better than macro and micro features. Indeed, microscopic analyses cannot be applied in compressed videos where the video loses lots of information. The macroscopic analyses struggle to distinguish forged videos where the videos do not leave obvious visual traces. This network contains four convolutional layers and two fully-connected layers. Each convolutional layer uses RELU activation functions followed by a batch normalization unit and a maximum pooling layer. In this way, it can regularize the data, reduce the data dimensionality, and prevent the vanishing gradient effect. We use the kernel size of 4*4 in the third pooling layer instead of 2*2 for improving the robustness. We also analyze the impact of the kernel size in Section V.

Finally, we gradually prune the modified network. The quantization noise in the process of video compression results in a specific visual error called artifacts. The compressed videos have a lot of redundant noise, which increases the difficulty of detection. To avoid being influenced by the redundant artifacts noise, the modified network is gradually pruned. The pruning [35] plays an important role in the robustness of the model. First, pruning the unimportant connections can prevent the invalid connections from affecting the final prediction. Second, the pruning is an iterative process. Each iteration is a greedy search so that the classifier will select the most informative predictors and thus have much less prediction variance. Third, after pruning the network, the parameters become sparse, and the network gets a low complexity, which can reduce over-fitting.

C. Temporality-Level Stream

In addition to model the features of tampered frames, we also leverage informative clues of temporality for accurate compressed Deepfake videos detection. The compressed Deepfake videos lose lots of information, and we need to extract effective features from limited information. According to the previous analysis of Section III, we know that an important distinction between real and fake videos is that fake videos ignore the temporal consistency during the synthesis process. Since our frame-level stream focuses on the face region, our proposed temporality-level stream is supposed to focus on the whole frame for exploiting the temporal features between frames. Also, the residual is proved to be a temporal correlation feature. Therefore, the temporality-level stream utilizes the time-dependent residual features to extract the inconsistency between the frames and uses the Resnet18 for classification. The details of the temporality-level stream are explained as follows.

Firstly, we use 3 temporal segments [36] to capture features. Three segments mean that the whole video is divided into three parts. Using temporal segments is beneficial to capture long term dependency. Each segment in the video will produce its preliminary prediction of video detection. Then, a consensus among the segments will be derived as the video-level prediction. In the learning process, the scores of video-level predictions are averaged from those of snippet-level predictions. We also analyze the impact of the temporal segments in Section V.

Algorithm 1 The Procedure of Training the Frame-Level Stream S_1 and Temporality-Level Stream S_2

```
Input:
The faces $I_F$
The residual features $I_R$
The learning rate $\alpha_1$ is fixed to 0.0001 for $S_1$
The initial learning rate $\alpha_2 = 0.001$ for $S_2$, decayed by factor 0.1 every 10 epochs
The batch size $b = 8$
Output:
Trained frame-level network $\theta_{S_1}$
Trained temporality-level network $\theta_{S_2}$
while $Step < max\_steps$ do
    for $i = 1 \rightarrow num\_iter$ do
        $g_{\theta_{S_1}} \leftarrow \nabla_{\theta_{S_1}}(\frac{1}{b}\sum_{i=1}^{b}\mathcal{L}_1)$
        $g_{\theta_{S_2}} \leftarrow \nabla_{\theta_{S_2}}(\frac{1}{b}\sum_{i=1}^{b}\mathcal{L}_2)$
        $\theta_{S_1} \leftarrow \theta_{S_1} + \alpha_{S_1} \cdot \text{Adam}(\theta_{S_1}, g_{\theta_{S_1}})$
        $\theta_{S_2} \leftarrow \theta_{S_2} + \alpha_{S_2} \cdot \text{Adam}(\theta_{S_2}, g_{\theta_{S_2}})$
    end for
    if $Step\%5 == 0$ then
        $\theta_{S_1}.prune()$
    end if
end while
```

Then, we extract the residual features I_R from each segment. As the proof of Section III, the residual is a time-dependent feature that can be used to expose the inconsistency between the frames. Therefore, for the temporality-level stream S_2, we extract the residual features I_R of the video segments as the temporality-level feature and input them to the Resnet18. It can detect the inconsistency between the frames of compressed Deepfake videos.

D. Training and Losses

A detailed description of the algorithm is given in Algorithm 1 for training the models. We input the faces I_F to train frame-level stream S_1. The target of the classification task is to minimize the loss L_1. Furthermore, we apply the pruning in the model. The residual features I_R are input to train the temporality-level stream S_2. The target of the classification task is to minimize the loss L_2. The Adam is utilized to optimize the model. Then, we define the training losses: L_1 and L_2.

For the cross entropy loss function, as the error increases, the gradient of the parameter would be larger, which is advantageous to converge our detection model. Thus, the cross entropy loss is applied to train the frame-level stream. The input of the frame-level stream is I-frames, and these frames are modeled by the CNN. The target of the classification task is to minimize the loss:

$$L_1 = \frac{1}{N}\sum -[y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)] \tag{7}$$

where y_i represents the label of the frame and p_i is the probability of positive prediction of frame i.

The input of the temporality-level stream is the residual features from video segments. Then, the Resnet18 in the temporality-level stream utilizes the residual features for classifying the videos. After the formally, the loss for the Resnet18 network is defined as:

$$L_2 = -log\frac{exp(v[l])}{\sum_j exp(v[j])} \tag{8}$$

where v is the predictive value of the network which has been transformed into a one-hot encoding, l is the label of the videos. The j in the denominator represents the index of categories. Our task has two classes, that is, $j = \{0, 1\}$. Thus, the numerator is the probability of the target category, and the denominator is the probability of all categories. The greater the probability of the target category, the smaller the L_2. During hard negative mining, some pseudo negatives will be treated as true hard negatives, and the model will eventually train the data to minimize L_2.

E. Two-Stream Score Fusion

The classification results of videos are acquired from the two streams. For each stream, the last layer of network output scores of the labels. The final score for a video is obtained by combining the output scores of the two streams:

$$r = softmax(r_1) + softmax(r_2) \tag{9}$$

where r_1 represents the output of the frame-level stream, r_2 refers to the result of the temporality-level stream, and r is the final result. Since the detection is a binary classification task, the model will output two values that represent the probability value of the original video and the probability value of the fake video. Both r_1 and r_2 are vectors with the dimensions of 1*2. The $softmax(r_1)$ is a 1*2 vector that contains two probability values that represent the probability value of the original video and the probability value of the fake video, likewise for $softmax(r_2)$. The $softmax(r_1) + softmax(r_2)$ represents the corresponding addition of the prediction results of the two streams. After the addition, two values are obtained. The two values respectively represent the probability value of the original video and the probability value of the fake video. If the probability value of the original video is higher than the probability value of the fake video, the video is predicted to be an original video. Otherwise, the video is predicted to be a fake video.

V. Experiments and Discussions

In this section, we introduce the implementation details of the two streams. Then, evaluating the performance of our proposed method through a set of experiments and comparing its performance with state-of-the-art methods.

A. Data Preparation

We apply our proposed method on Celeb-DF [20] and FaceForensics++ [21] datasets. The final videos of Celeb-DF are in MPEG4.0 format. The videos of Faceforensic++ use the H.264 codec, which is used by social networks or video-sharing websites. The FaceForensics++ datasets comprise four types of forgery videos namely: DeepFakes, Face2Face, FaceSwap, and NeuralTextures. The Celeb-DF datasets are created by a refined version of the DeepFakes generation algorithm. Besides, the FaceForensics++ datasets contain compressed videos with compression factor 40. Then, we compress the videos from Celeb-DF datasets with the same compression factor. According to the manipulation types, these datasets are divided into DeepFakes, FaceSwap, Face2Face, NeuralTextures, and Celeb-DF datasets.

Frame-Level Stream: For the data preparation of the frame-level stream, we extract the I-frames by using FFmpeg [37]. Since the datasets only manipulated the face of videos, we focus on the faces and use dlib [38] to do the face detection, landmark extraction, and face alignment. When using dlib to extract human faces, the dlib sometimes treats circular objects as human faces but does not recognize human faces. For example, in the video 760_611.mp4 of FaceForensics++, the dlib mistakenly extracts the circular traffic sign as a face. When the dlib extracts the circular objects rather than one face in a single frame, the order of the faces will be scrambled. Then, the out-of-order frames cannot be used to build the model of temporality-level. In this case, if the dlib does not recognize the correct face, we will remove the video. Besides, we crop two sets of data with a size of 256*256. In the first set, we crop the I-frames leaving only the face area. Then, cropping the face and its surroundings to form a second set of data.

Temporality-Level Stream: The temporality-level stream performs the same data cleaning as the frame-level stream. Each video is divided into 3 segments. We use CoViAR [33] to extract residual features from each segment. The dimension of residual features is 3*224*224. Since the video has 3 segments, we get residual features with size 3*3*224*224 from each video. We also analyze the impact of the number of temporal segments in Section V part D.

B. Model Training

For the FaceForensics++ datasets, each category contains 1000 tampered videos generated from YouTube videos. Thus, there are a total of 1000 pristine and 4000 forged videos in FaceForensics++ datasets. 80% of all videos are used for training the model, and 20% for testing. For the Celeb-DF datasets, there are 890 original videos in Celeb-DF. To balance positive and negative samples, 890 fake videos are randomly selected to carry out the experiments. 80% of these videos are trained the model, 20% for testing. The two-stream experiments are conducted in PyTorch on NVIDIA Titan Xp.

Frame-Level Stream: The Adam optimizer [39] with 10^{-3} learning rate is used in the frame-level stream. The pruning threshold is chosen as a quality parameter multiplied by the standard deviation of a layer weights. We prune MesoNet using three steps. Firstly, learning the connectivity by training network to learn which connections are important. Secondly, if the weight is lower than the threshold, we remove the unimportant connections. Thirdly, we retrain the network so that the remaining connections can compensate for the connections that have been removed. Define the pruning rate as the remaining connections divided by the total connections. This pruning process will not stop until the pruning rate converges. Then, the models of DeepFakes, FaceSwap, Face2Face, NeuralTextures, and Celeb-DF are saved after the training.

TABLE II

THE DETECTION ACCURANCY (%) OF DIFFERENT KERNEL SIZES IN THE FRAME-LEVEL STREAM ON DETECTING THE COMPRESSED DEEPFAKE VIDEOS

Datasets	2*2	3*3	4*4	5*5
DeepFakes	88.01	91.32	**93.88**	87.76
FaceSwap	77.92	79.70	**82.99**	80.20
Face2Face	79.24	79.49	**80.76**	79.75
NeuralTextures	76.26	77.52	**78.03**	77.77
Celeb-DF	77.87	77.01	**78.16**	76.15

TABLE III

THE DETECTION ACCURACY (%) OF THE DIFFERENT NUMBERS OF TEMPORAL SEGMENTS IN THE TEMPORALITY-LEVEL STREAM ON DETECTING THE COMPRESSED DEEPFAKE VIDEOS

Datasets	1 segment	2 segments	3 segments	4 segments	5 segments
DeepFakes	75.26	76.02	**84.25**	82.14	81.12
FaceSwap	66.24	71.07	**77.16**	77.06	74.11
Face2Face	67.09	70.63	**76.00**	75.19	72.15
NeuralTextures	62.63	63.13	**70.00**	67.42	68.69
Celeb-DF	67.82	72.70	**79.78**	75.00	73.28

Temporality-Level Stream: For the temporality-level stream, we use temporal segments to capture long term dependency. Features at each step are the average of features across 3 segments during training. These features of each segment are put into Resnet18 to detect videos. During training, the learning rate is initialized to 10^{-3}, and it is divided by 10 when the accuracy plateaus.

C. Impacts of the Kernel Size in Frame-Level Stream

Mesonet is utilized in the proposed frame-level stream. Afchar *et al.* [15] used Mesonet with a kernel size of $2 * 2$ in the third pooling layer. To reduce overfitting, we change the kernel size of the third pooling layer into $4*4$. It is important to evaluate the effect of the kernel size on performance. We apply our proposed method on DeepFakes, FaceSwap, Face2Face, NeuralTextures, and Celeb-DF datasets and vary the value of the kernel size. The results are shown in Table II, and the performance of detecting the compressed Deepfake videos is improved after changing the size. When the kernel size is 4*4, we get the best result. The performance would be decreased in other kernel sizes. Therefore, we set $4 * 4$ as the kernel size of the third pooling layer in our experiments.

D. Impacts of the Number of Temporal Segments in Temporality-Level Stream

The temporal segment is utilized in the proposed temporality-level stream. Using temporal segments is

TABLE IV

THE DETECTION ACCURACY (%) OF FUSION STRATEGIES. s_1 IS SET AS THE WEIGHTS OF FRAME-LEVEL STREAM, AND s_2 REPRESENTS THE WEIGHTS OF TEMPORALITY-LEVEL STREAM

Fusion Strategies	DeepFakes	FaceSwap	Face2Face	NeuralTextures	Celeb-DF
s_1 = 0.1, s_2 = 0.9	86.48	79.70	80.25	70.20	76.45
s_1 = 0.2, s_2 = 0.8	90.31	82.49	82.53	78.28	78.45
s_1 = 0.3, s_2 = 0.7	92.60	84.77	83.29	79.55	79.31
s_1 = 0.4, s_2 = 0.6	92.86	85.03	85.06	79.80	80.46
s_1 = 0.6, s_2 = 0.4	**95.05**	**85.27**	86.08	**80.05**	80.17
s_1 = 0.7, s_2 = 0.3	94.13	84.77	83.54	79.55	79.02
s_1 = 0.8, s_2 = 0.2	93.62	84.77	82.53	79.29	79.02
s_1 = 0.9, s_2 = 0.1	93.37	84.01	82.03	79.29	77.87
Softmax Fusion	94.64	**85.27**	**86.48**	**80.05**	**80.74**
Additive Fusion	94.64	**85.27**	86.08	**80.05**	80.46

TABLE V

THE COMPARISONS OF THE CROSS-COMPRESSION DETECTION ACCURACY (%) WITH THAT OF THE STATE-OF-THE-ART METHODS ON DEEPFAKES, FACESWAP, FACE2FACE, NEURALTEXTURES, AND CELEB-DF DATASETS. C40-C23 REPRESENTS THAT THE MODEL IS TRAINED ON C40 VIDEOS AND TESTED ON C23 VIDEOS. C23-C40 REPRESENTS THAT THE MODEL IS TRAINED ON C23 VIDEOS AND TESTED ON C40 VIDEOS

Datasets	Nguyen et al. [14]		Afchar et al. [15]		Güera et al. [17]		Carreira et al. [18]		Tran et al. [19]		Rössler et al. [21]		Ours	
	C23-C40	C40-C23	C23-C40	C40-C23	C23-C40	C40-C23	C23-C40	C40-C23	C23-C40	C40-C23	C23-C40	C40-C23	C23-C40	C40-C23
DeepFakes	67.75	79.27	78.75	78.76	71.93	76.15	57.04	81.71	79.55	83.66	63.96	**84.61**	**88.78**	84.20
FaceSwap	54.50	73.90	59.75	60.36	52.01	54.08	50.00	58.97	54.52	75.06	55.63	79.85	**77.66**	**84.19**
Face2Face	55.75	80.00	69.75	71.03	50.95	51.98	56.82	54.02	69.10	78.11	50.83	63.47	**79.95**	**86.75**
NeuralTextures	53.75	63.66	53.25	60.05	51.76	58.21	60.57	52.50	59.80	64.63	55.84	73.15	**77.78**	**80.52**
Celeb-DF	50.58	69.89	57.25	66.70	51.70	66.07	72.63	62.54	64.55	79.17	50.72	64.30	**79.89**	**80.96**

beneficial to capture long term dependency. To analyze the impact of the number of temporal segments, we vary the number into 1 segment, 2 segments, 3 segments, 4 segments, and 5 segments. Then, applying our proposed temporality-level stream on the different numbers of temporal segments and showing the results in Table III. It is shown that using 3 segments performs better than that of 1 segment. That is because that using temporal segments is capable of modeling long-range temporal structure over the whole video. Further, when the number of temporal segments is 3, we get the best result, and the performance would be decreased in other values. If the number of temporal segments is too large, the individual segment will be too short. Then, the short video clip is not conducive to extracting time-related features. Therefore, we set 3 temporal segments in the temporality-level stream.

E. Impacts of Fusion Strategies

The outputs of the frame-level stream and temporality-level stream will have different properties. To combine them, we tried various fusion strategies, including weighted fusion, additive fusion, and softmax fusion. The results are as shown in Table IV. The weighted fusion means giving each stream a different weight. The results show that the weighted fusion performs well on DeepFakes datasets. The additive fusion performs well on FaceSwap and NeuralTextures datasets. In most cases, softmax fusion gets the best performance by using the softmax function to normalize all scores. Therefore, we use the softmax fusion to detect the compressed Deepfake videos. It gives us a model that is easy to train and flexible for inference.

TABLE VI

ABLATION STUDY - THE DETECTION ACCURACY (%) OF TWO-STREAM FUSION

Datasets	Frame-level stream	Temporality-level stream	Two-stream method
DeepFakes	93.88	84.25	**94.64**
FaceSwap	82.99	77.16	**85.27**
Face2Face	80.76	76.00	**86.48**
NeuralTextures	78.03	70.00	**80.05**
Celeb-DF	78.16	79.78	**80.74**

F. Impacts of Compression Factor

In order to evaluate the robustness of the proposed method, we carry out the cross-compression experiments and show results in Table V. When we apply the C40 videos for training but using the C23 videos for testing, the detection accuracy drops slightly. It is because that the videos with compression factor C40 may have much more artifacts, which cause the model cannot fit the C23 videos. If we use C40 videos to test the model which is trained with C23 videos, it leads to a slight decline in classification performance. The possible reason is that C40 videos lose lots of information during

TABLE VII

ABLATION STUDY - THE FORGERY DETECTION ACCURACY (%) OF MESONET, MESONET-4*4, AND PRUNING MESONET-4*4 ON DEEPFAKES, FACESWAP, FACE2FACE, NEURALTEXTURES, AND CELEB-DF DATASETS. THE MESONET-4*4 REPRESENTS THAT USING THE KERNEL SIZE OF 4*4 IN THE THIRD POOLING LAYER

	DeepFakes	FaceSwap	Face2Face	NeuralTextures	Celeb-DF
Mesonet	87.50	80.50	81.25	70.50	57.50
Mesonet-4*4	90.05	77.92	78.50	77.53	74.71
pruning Mesonet-4*4	**93.88**	**82.99**	**80.76**	**78.03**	**78.16**

TABLE VIII

ABLATION STUDY - THE FORGERY DETECTION ACCURACY (%) WITH AND WITHOUT DATA AUGMENTATION IN THE TRAINING PROCESS

	DeepFakes	FaceSwap	Face2Face	NeuralTextures	Celeb-DF
without data augmentation	88.78	73.60	80.25	77.78	75.00
with data augmentation	**93.88**	**82.99**	**80.76**	**78.03**	**78.16**

TABLE IX

THE COMPARISONS OF THE DETECTION ACCURACY (%) WITH THAT OF THE STATE-OF-THE-ART METHODS ON DEEPFAKES, FACESWAP, FACE2FACE, NEURALTEXTURES, AND CELEB-DF DATASETS

Datasets	Li et al. [13]	Nguyen et al. [14]	Afchar et al. [15]	Güera et al. [17]	Carreira et al. [18]	Tran et al. [19]	Ours
DeepFakes	78.50	87.50	82.75	81.70	74.87	85.10	**94.64**
FaceSwap	67.25	80.50	76.25	66.83	66.33	72.11	**85.27**
Face2Face	66.75	81.25	73.25	70.85	65.58	73.12	**86.48**
NeuralTextures	64.75	70.50	63.75	69.10	63.89	60.30	**80.05**
Celeb-DF	54.45	57.50	54.25	77.28	76.08	78.67	**80.74**

the compression. When the model is trained on C40 and tested on C23, Rössler *et al.* [21] performs well on DeepFakes datasets. In other cases, the cross-compression detection accuracy of the proposed method is higher than other methods [14], [15], [17]–[19]. That is because our proposed frame-level stream uses a compact network with iterative pruning to avoid over-fitting, and the temporality-level extracts the time-dependent features that are not influenced by the compression factor.

G. Ablation Study

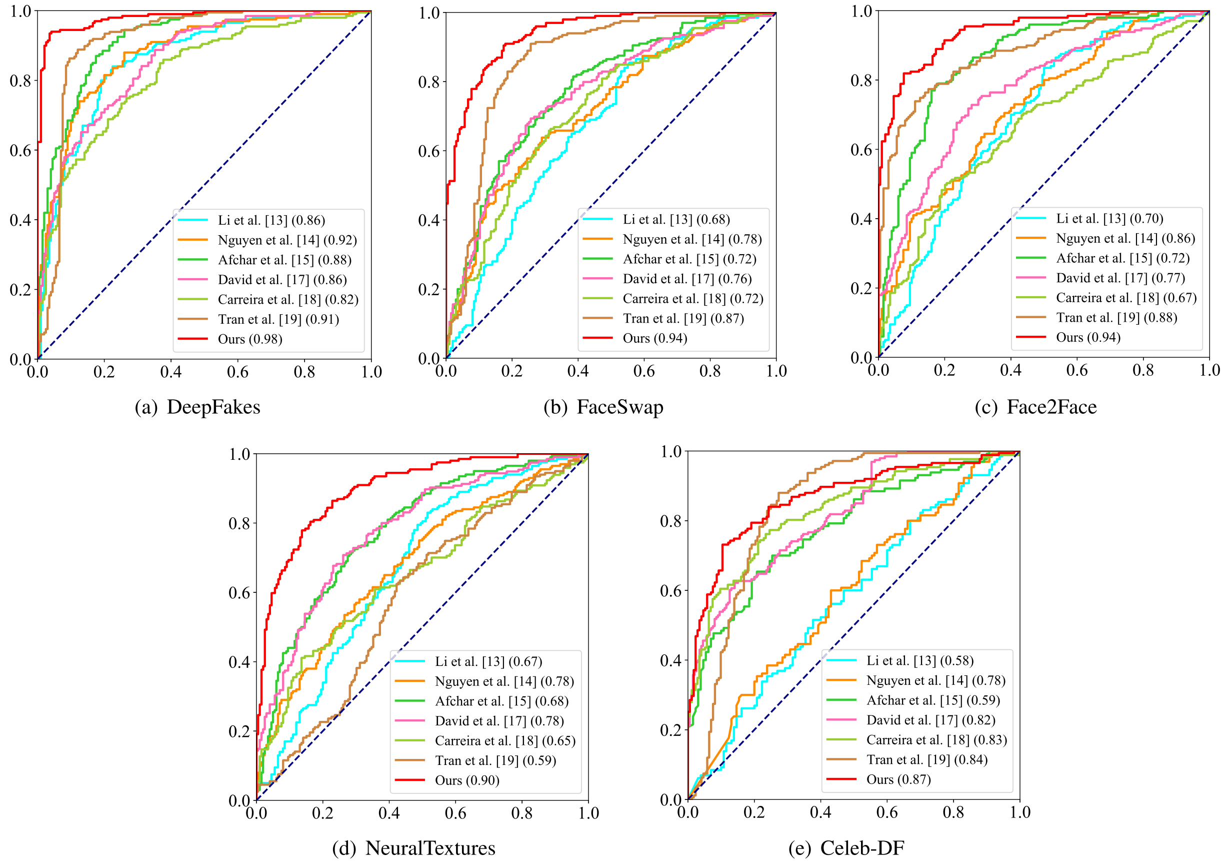
To show the performance of each stream separately, we compare the single stream with our proposed two-stream method. First, we detect the videos by using a frame-level stream. The classification results of frames are acquired from the last fully-connected layer of Mesonet. Then, we extend the results of frames to videos. For example, if the classification results of frames are positive and the frame is extracted from the real video, we consider the classification results to be correct. The results of the frame-level stream are shown in the first column of Table VI, and the detection accuracy is lower than our proposed two-stream method. Note that we use only I-frames as full images, which is a small subset of all frames, yet the frame-level stream achieves good performance. Second, we only implement the temporality-level stream to detect videos. As shown in the second column of Table VI, the performance is also worse than the combined result. In the third column of Table VI, we can see that combining the two streams gets the best result. The fusion of two streams could reveal both tampering artifacts and temporal features, contributing to better performance than a single-stream network. Therefore, the fusions of the two streams can improve the performance of compressed Deepfake videos detection.

Then, we perform ablation studies on the MesoNet, modified MesoNet, and pruning the modified MesoNet, and the results are shown in Table VII. It shows that Mesonet without modifying cannot perform well on the compressed Deepfake videos. To reduce overfitting, we change the kernel size of the third pooling layer into 4*4. The experimental results show that there is an advantage to modify Mesonet. To show the importance of pruning, we evaluate our proposed frame-level stream before and after pruning modified MesoNet. The results are shown in the second line and the third line of Table VII. It demonstrates that pruning modified Mesonet improves the detection accuracy. Since Deepfake videos add a lot of redundant information to frames, pruning modified Mesonet can prune the redundant connections to prevent the invalid connections from affecting the final prediction.

Finally, we perform ablation studies for the benefits of using data augmentation. Table VIII shows the performance of with and without data augmentation in the training process. It can be seen that if we train the model without data augmentation, the performance would be decreased. Thus, it is significant to do the data augmentation in the training process.

H. Comparisons to State-of-the-Art Methods

In this subsection, we compare the performance of our proposed method with state-of-the-art methods on detecting compressed Deepfake videos.



(a) DeepFakes (b) FaceSwap (c) Face2Face

(d) NeuralTextures (e) Celeb-DF

Fig. 6. ROC (receiver operating characteristic) curves for the state-of-the-art compressed Deepfake videos detection methods on different public datasets: (a) DeepFakes, (b) FaceSwap, (c) Face2Face, (d) NeuralTextures, and (e) Celeb-DF datasets.

TABLE X

AUC (Area Under the Curve) for the State-of-the-Art Compressed Deepfake Videos Detection Methods on DeepFakes, FaceSwap, Face2Face, NeuralTextures, and Celeb-DF Datasets

Datasets	Li et al. [13]	Nguyen et al. [14]	Afchar et al. [15]	Güera et al. [17]	Carreira et al. [18]	Tran et al. [19]	Ours
DeepFakes	0.86	0.92	0.88	0.86	0.82	0.91	**0.98**
FaceSwap	0.68	0.78	0.72	0.76	0.72	0.87	**0.94**
Face2Face	0.70	0.86	0.72	0.77	0.67	0.88	**0.94**
NeuralTextures	0.67	0.78	0.68	0.78	0.65	0.59	**0.90**
Celeb-DF	0.58	0.78	0.59	0.82	0.83	0.84	**0.87**

We compare the detection accuracy of our works with that of the frame-based methods and spatiotemporal methods. As shown in Table IX, when detecting the compressed videos, our proposed method outperforms Li and Lyu [13], Nguyen *et al.* [14], and Afchar *et al.* [15] on DeepFakes, FaceSwap, Face2Face, NeuralTextures, and Celeb-DF datasets. Especially, for Celeb-DF datasets, though most of these frame-based methods [13]–[15] cannot detect it very well, the detection accuracy of our proposed method is higher than theirs. When detecting Deepfake videos, Güera and Delpv [17] combined spatial and temporal information. Carreira and Zisserman [18] and Tran *et al.* [19] classified the videos by using 3d convolutional network. Though these spatiotemporal methods can also address the Deepfake videos detection problem as shown in Table IX, our method performs better than Güera and Delpv [17], Carreira and Zisserman [18], and Tran *et al.* [19] on the compressed Deepfake videos detection.

We evaluate the overall detection performance using the ROC (receiver operating characteristic) curve, and the results are shown in Fig. 6. The abscissa values represent the FPR (False Positive Rate), and the ordinate values represent the TPR (True Positive Rate). Our curve is closer to the top left hand corner, representing our proposed method is better than the state-of-the-art methods on the compressed DeepFake videos detection.

To evaluate the overall detection performance, we calculate the AUC (area under the curve) score and show the results in Table X. The AUC is the area under the ROC (receiver operating characteristic) curve, and the maximum of the AUC score is 1. As shown in Table X, the AUC of our proposed

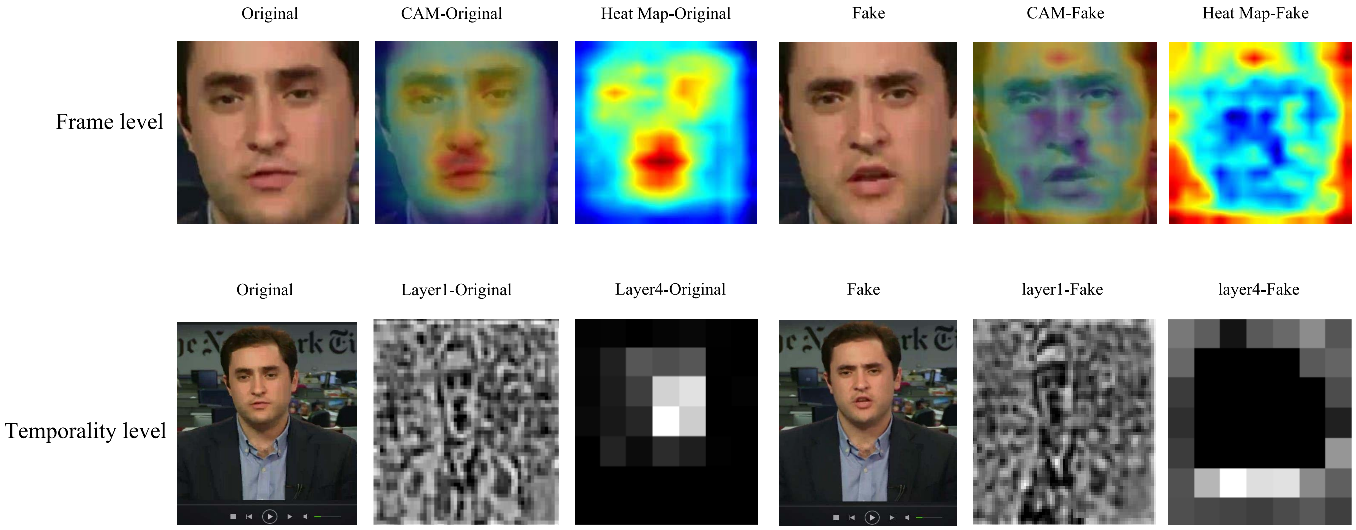


Fig. 7. The visualization of our proposed two-stream method. In the first line, we show the Class Activation Maps (CAMs) and heat map of the original face and the corresponding fake face obtained from the frame-level steam. The second line shows the layer1 and layer4 of original videos and fake videos obtained from the temporality-level stream.

TABLE XI

THE DETECTION ACCURACY (%) OF THE PROPOSED METHOD ON CELEB-DF. CELEB-DF_v2′ REPRESENTS THE SELECTED LARGE AMOUNT OF VIDEOS

	C40	C23	C40-C23	C23-C40
Celeb-DF	80.74	89.19	80.96	79.89
Celeb-DF_v2′	88.83	89.56	86.43	85.70

method is higher than other methods. These higher results indicate that our proposed method performs better on the compressed Deepfake videos detection.

Li and Lyu [13] fed all frames of the video to the CNN based model and then average the top third of the output score as the overall output of the video. Nguyen *et al.* [14] focused on the frames and used the first ten frames of the videos to detect videos. For our method, we combine the frame-level stream and temporality-level stream to detect videos, which is superior to a single stream. Since Afchar *et al.* [15] trained the model by using the frames of the entire videos, it may over-fit during the training of the Deepfake videos. Compressed videos bring a lot of compression noise, which influences the spatial feature extraction of Güera and Delpv [17]. When the spatial features with compression noise are input to RNN, it will affect the detection performance of Güera and Delpv [17]. The methods of Carreira and Zisserman [18] and Tran *et al.* [19] focused on the action of the face but ignored artifacts. The proposed frame-level stream prunes the network to prevent the model from fitting the compression artifacts, and the temporality-level stream extracts the inconsistency between frames. By combing the two streams, the proposed method performs well on various datasets.

I. Discussions

a) Interpretability of Our Proposed Method: To understand what visual clues the two streams rely on to detect the compressed Deepfake videos, we visualize the feature map in Fig. 7. Following the method used in [40] to generate Class Activation Maps (CAMs) of the frame-level stream. As shown in the first line of Fig. 7, we can observe the activation of original-weighted neurons display images with high probability in faces, while fake-weighted ones display low probability. That is understandable as Deepfake-generated faces tend to lack details compared to the background. The temporality-level stream extracts the proved time-dependent residual features and inputs the features to the Resnet18. We visualize the features from layer1 and layer4 of Resnet18. In the second line of Fig. 7, the faces are the most detailed part of the real video while it is the background in the forged video. During generating the manipulated faces in Deepfake videos, resulting in intra-frame inconsistencies and temporal inconsistencies between faces. Therefore, the inconsistency between the faces is exploited by our proposed temporality-level stream. On the one hand, this visualization indicates that our frame-level stream can learn reasonable features. On the other hand, the visualization of the temporality-level stream shows the different temporal features between the fake video and the original video. Therefore, our method is useful for compressed Deepfake videos detection.

b) Testing More Data for Celeb-DF_v2: Celeb-DF_v1 includes 408 real videos and 795 synthesized videos. Celeb-DF_v2 includes 482 real videos and 4844 synthesized videos. Thus, there are 890 original videos and 5639 fake videos in Celeb-DF datasets. All of the real videos are used for our experiments. To balance positive and negative samples, 890 fake videos are randomly selected to carry out the experiments. 80% of these videos are trained the model, 20% for testing, and the results are shown in the first line of Table XI. To fully evaluate the proposed method, we increase the test amount of Celeb-DF_v2 to 3000 videos. These 3000 videos are preprocessed and cleaned up as described in Section IV. Then, the trained model is used for testing the selected videos, and the results are shown in the second line of Table XI. As we can see from the table, our proposed

TABLE XII
THE COMPARISONS OF THE AMOUNT OF INPUT FRAMES WITH THAT OF THE STATE-OF-THE-ART METHODS

Datasets	Li et al. [13]	Nguyen et al. [14]	Afchar et al. [15]	Güera et al. [17]	Carreira et al. [18]	Tran et al. [19]	Ours
DeepFakes	42064	16012	807931	64000	102400	25600	**8168**
FaceSwap	42064	16013	807928	64000	102400	25600	**7374**
Face2Face	42064	16008	807930	63960	102400	25600	**8215**
NeuralTextures	42064	16016	807929	63920	102400	25600	**7585**
Celeb-DF	30169	12070	390622	28240	91136	22784	**5340**

method performs well on the selected large amount of videos.

c) Robustness of the Model: There are many algorithms for generating the Deepfake videos. Finding a common model to detect video is particularly important. When we use the model that trained with videos from NeuralTextures to detect the videos from Deepfakes, the detection accuracy is only 70.56%. Here, the best performance of Li and Lyu [13], Nguyen *et al.* [14] and Afchar *et al.* [15] is 55.82%, 56.25% and 50.50% on mismatch datasets. The probable cause is that CNN focused on two classes during training, but the video features of other datasets are different. Therefore, it is difficult to accurately detect other datasets without learning them. This prompts us to focus on improving the generalization of the model in the next stage.

d) Effectiveness of Our Proposed Method: Suppose a convolutional layer have kernels of size $k_w \times k_h$, the numbers of input channel and output channel are C_{in} and C_{out}, the size of the output feature map is $f_w \times f_h$. The multiplication computation of this convolutional layer is $(k_w \times k_h \times C_{in}) \times f_w \times f_h \times C_{out}$, the addition computation is $(k_w \times k_h \times C_{in} - 1) \times f_w \times f_h \times C_{out}$, and the bias computation is $f_w \times f_h \times C_{out}$. Thus, the whole computation (multi-adds) of the convolutional layer is $2 \times (k_w \times k_h \times C_{in}) \times f_w \times f_h \times C_{out}$. According to the above calculation algorithm, we could obtain the multi-adds of the proposed method. Here, the multi-adds of the frame-level stream is 55.11 million, and the multi-adds of the temporality-level stream is 362.81 million. The multi-adds of Refs. [18], [19] is 222.30 billion and 77.09 billion, respectively. Moreover, we use only I-frames in the frame-level stream, which is a small subset of all frames. Table XII shows that the number of frames we input is much lower than that of other methods. Thus, the computational complexity of our proposed method is acceptable.

VI. CONCLUSION

This work focus on compressed Deepfake videos with low-quality factor to account for situations typically encountered on social networks. In view of the noise that the compression brings to the frame, we apply the frame-level stream with a low complexity network and prune the model to refrain from fitting the noise. To detect the temporal features of compressed videos, the temporality-level stream is used to extract the inconsistency between frames. The two streams extract the frame-level features and temporality-level features of compressed videos. We carry out our proposed two-stream method on DeepFakes, FaceSwap, Face2Face, NeuralTextures, and Celeb-DF datasets, and the result performs better than the existing work. Last but not least, the experimental results of cross-compression detection accuracy show our proposed method is robust to the compression factor. Future work will focus on improving the robustness of the model so that we can identify multiple types of manipulated facial videos by using a general model.

REFERENCES

[1] H. V. Zhao, W. S. Lin, and K. J. R. Liu, "Behavior modeling and forensics for multimedia social networks," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 118–139, Jan. 2009.

[2] V. G. Ivanov and Y. R. Ignatovskiy, "Deepfakes: Prospects for political use and threats to the individual and national security," *RUDN J. Public Admin.*, vol. 7, no. 4, pp. 379–386, Dec. 2020.

[3] L. Verdoliva, "Media forensics and DeepFakes: An overview," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 910–932, Aug. 2020.

[4] *Deep Fakes: A Looming Crisis for National Security, Democracy and Privacy?*. Accessed: Feb. 4, 2021. [Online]. Available: https://www.lawfareblog.com/deep-fakes-looming-crisisnational-/security-democracy-and-privacy

[5] S. Lyu, "Deepfake detection: Current challenges and next steps," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6.

[6] M. Aloraini, M. Sharifzadeh, and D. Schonfeld, "Sequential and patch analyses for object removal video forgery detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 917–930, Mar. 2021, doi: 10.1109/TCSVT.2020.2993004.

[7] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A PatchMatch-based dense-field algorithm for video copy–move detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 669–682, Mar. 2019.

[8] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic detection of object-based forgery in advanced video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2138–2151, Nov. 2016.

[9] X. Ding, N. Zhu, L. Li, Y. Li, and G. Yang, "Robust localization of interpolated frames by motion-compensated frame interpolation based on an artifact indicated map and tchebichef moments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1893–1906, Jul. 2019.

[10] C. Feng, Z. Xu, S. Jia, W. Zhang, and Y. Xu, "Motion-adaptive frame deletion detection for digital video forensics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2543–2554, Dec. 2017.

[11] P. He, H. Li, B. Li, H. Wang, and L. Liu, "Exposing fake bitrate videos using hybrid deep-learning network from recompression error," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4034–4049, Nov. 2020.

[12] W. Wang and H. Farid, "Exposing digital forgeries in interlaced and deinterlaced video," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 438–449, Sep. 2007.

[13] Y. Li and S. Lyu, "Exposing Deepfake videos by detecting face warping artifacts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, vol. 2, pp. 46–52, 2019.

[14] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2307–2311.

[15] A. Darius, N. Vincent, Y. Junichi, and E. Isao, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[16] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing AI created fake videos by detecting eye blinking," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[17] D. Guera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.

[18] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[20] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3204–3213.

[21] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[22] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2387–2395.

[23] (2018). *FaceSwap Github Non Official Project Based on Original Deepfakes Thread*. [Online]. Available: https://github.com/MarekKowalski/FaceSwap/

[24] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, 2019.

[25] (2018). *DeepFakes Github Non Official Project Based on Original Deepfakes Thread*. [Online]. Available: https://github.com/deepfakes/faceswap

[26] J. Ian Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[27] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[28] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, 2003, pp. 1398–1402.

[29] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[30] V. Baroncini, L. Capodiferro, D. Elio Di Claudio, and G. Jacovitti, "The polar edge coherence: A quasi blind metric for video quality assessment," in *Proc. 17th Eur. Signal Process. Conf.*, Aug. 2009, pp. 564–568.

[31] S. Bosse, D. Maniry, K.-R. Muller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 206–219, Jan. 2018.

[32] J. Kim and S. Lee, "Deep learning of human visual sensitivity in image quality assessment framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1969–1977.

[33] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Compressed video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6026–6035.

[34] 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.

[35] H. Song, P. Jeff, T. John, and D. William, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.

[36] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[37] Y. Cheng, Q. Liu, C. Zhao, X. Zhu, and G. Zhang, "Design and implementation of mediaplayer based on FFmpeg," in *Software Engineering and Knowledge Engineering: Theory and Practice* (Advances in Intelligent and Soft Computing), vol. 115, Y. Wu, Ed. Berlin, Germany: Springer.

[38] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Dec. 2009.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

Juan Hu received the B.S. degree from the College of Electronic Information Science and Technology, Nanjing Agricultural University, Nanjing, China, in 2017, and the M.S. degree from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2019, where she is currently pursuing the Ph.D. degree. Her current research interests include multimedia forensic and artificial intelligence.

Xin Liao (Member, IEEE) received the B.E. and Ph.D. degrees in information security from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2012, respectively. In 2012, he joined Hunan University, Changsha, China, where he is currently an Associate Professor. He was a Visiting Scholar with the University of Maryland, College Park, MD, USA, from 2016 to 2017. His current research interests include multimedia forensics, watermarking, and steganography. He is also a Senior Member of the China Computer Federation and the Chinese Institute of Electronics. He is also a member of the Technical Committee (TC) on Multimedia Security, the Forensics of Asia-Pacific Signal and Information Processing Association, the TC on Computer Forensics of Chinese Institute of Electronics, and the TC on Digital Forensics and Security of China Society of Image and Graphics.

Wei Wang (Member, IEEE) received the B.Sc. degree in computer science and technology from North China Electric Power University, in 2007, and the Ph.D. degree in pattern recognition from the Institute of Automation, Chinese Academy of Sciences (CASIA) in 2012. He is currently an Associate Professor with the National Laboratory of Pattern Recognition (NLPR), CASIA. His current research interests include artificial intelligence and its security problem, image/video forensics and steganalysis, and information content security. He is also a member of CCF (China Computer Federation) and CSIG (China Society of Image and Graphics). He is also a member of the Technical Committee (TC) on Computer Vision of CCF and the TC on Digital Forensics and Security of CSIG.

Zheng Qin (Member, IEEE) received the Ph.D. degree in computer software and theory from Chongqing University, China, in 2001. From 2010 to 2011, he served as a Visiting Scholar with the Department of Computer Science, University of Michigan. He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, where he serves as the Vice Dean. He also serves as the Director of the Hunan Key Laboratory of Big Data Research and Application and the Vice Director of the Hunan Engineering Laboratory of Authentication and Data Security. His main interests are network and data security, privacy, data analytics and applications, machine learning, and applied cryptography. He is also a member of the China Computer Federation (CCF).