Hindawi
Security and Communication Networks
Volume 2022, Article ID 5169873, 13 pages
https://doi.org/10.1155/2022/5169873

Research Article

Face Forgery Detection Based on the Improved Siamese Network

Bo Wang[1], **Yucai Li**[1], **Xiaohan Wu**[1], **Yanyan Ma**[1], **Zengren Song**[2] and **Mingkan Wu**[1]

[1]*School of Information and Communication Engineering, Dalian University of Technology, Dalian, China*
[2]*National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China*

Correspondence should be addressed to Yucai Li; lyc124184@mail.dlut.edu.cn

Received 30 November 2021; Revised 28 December 2021; Accepted 8 January 2022; Published 5 February 2022

Academic Editor: Beijing Chen

Face tampering is an intriguing task in video/image genuineness identification and has attracted significant amounts of attention in recent years. In this work, we propose a face forgery detection method that consists of preprocessing, an improved Siamese network-based feature extractor (including a feature alignment module), and postprocessing (a voting principle). Roughly speaking, our method extracts the features in the grey space of face/background image pairs and measures the difference to make decisions. Experiments on several standard databases prove the effectiveness of our method, and especially on the low-quality subdataset of the FaceForensics++ , our method achieves a competitive result.

1. Introduction

In recent years, image/video tampering methods have developed rapidly [1], including Deepfake [2], Face2Face [3], FaceSwap [4], and Neural Textures [5]. These methods rely on advanced image/video processing algorithms and are embedded within many applications in the market. Because visual contents can be easily manipulated, the detection of tampered contents is of practical significance and readily attracts attention [6]. In this work, we are interested in face forgery detection.

Many methods have been proposed for detection of tampered face images and videos, and the accuracy mainly depends on the selection of features and classifiers. The state-of-the-art methods roughly consist of two stages: feature extraction and classification. Several methods segregate these stages as separate subproblems [7–10], while some methods integrate the two stages in sequence based on deep neural networks (DNNs) [1, 11–18]. Regarding face forgery detection, there are two main types of selections of features: one is based on single-image features [1, 7–29], while the other is based on between-frame feature differences in videos [30–39]. Note that various types of classifiers are used (e.g., SVM, CNN, RNN, and MLP) and that SVM and CNN are relatively more popular.

The existing methods have achieved excellent detection accuracy on public datasets, including [1, 23, 40–43]. However, there are still problems yet to be solved. The first problem is that most methods offer poor robustness. They can achieve satisfactory accuracy on uncompressed or lightly compressed images and videos, but for content that is compressed with high intensity, the detection accuracy is greatly reduced because the compression may significantly eliminate the traces of forgery. The quality of images and videos also decreases after rounds of postprocessing, which greatly compromises the performance of the existing methods. The second problem is that almost all of the methods use only the features of the facial area or the fusion boundary area of the face and background but discard the features of the background. Although normally only the facial area is tampered, it is worth noting that, for untampered images, the facial area and the background are consistent at a certain feature level, which stands in contrast with forged images. Therefore, in this work, we address the face-background difference-based features.

In this paper, we propose a method based on the improved Siamese network [44]. The Siamese network was originally proposed to learn a similarity metric with application to face verification. We use the Siamese network to

measure the similarity between the face area and the background of the video frames. Before being saved in memory, a captured video is processed through a series of steps, including quantization, denoising, color correction, gamma correction, filtering, white balance, and even JPEG compression [45]. This series of processing steps involves unique statistical characteristics, and in an untampered video, the face area and the background of the video frames exhibit high similarity. In a tampered video, the similarity between the face area and the background is low because they originate from different videos. It is worth noting that this specialty is video-level; that is, the similarity relationship between the face area and the background between different frames in the video conforms to this law, because all processing is carried out on the whole video, that is, all frames. Our improved Siamese network can measure the similarity in order to distinguish genuine and tampered images and videos. The general pipeline of our method is depicted in Figure 1, and our contribution can be roughly concluded as follows: First, we design a preprocessing module that obtains a large number of image patch pairs of face area and background. Next, we present our improved Siamese network, which consists of two submodules, i.e., feature extraction and feature alignment. In the feature extraction module, we grey the image patch pairs and then input the pairs to a two-stream convolutional neural network with shared weights to extract features in the grey space of the images. In the feature alignment module, we align the features to measure the similarity between the face area and the background of images and obtain the final authenticity judgement result. During testing, we define a voting principle to correct our results by cropping multiple pairs of face area and background from a video frame. Then, we define a voting principle to correct the classification results. Last, through experiments, we show that our method outperforms the state-of-the-art methods on challenging low-quality datasets.

2. Related Work

2.1. Face Forgery. The most widely used face tampering methods include Deepfake [2], Face2Face [3], FaceSwap [4], and Neural Textures [5]. Examples of these methods are depicted in Figure 2.

The core of the application of Deepfake to facial video tampering is the parallel training of two autoencoders with shared parameters. The production process has two stages: the training stage and the generation stage. In the training stage, two autoencoders with shared parameters extract the features of two faces that belong to different persons and then input two autodecoders with independent parameters. In the generation stage, the facial features extracted by the autoencoder are input into the autodecoder corresponding to another different face to generate a mixed face. Finally, the mixed face is blended with the rest of the image using Poisson image editing [46]. Face2Face is a technology that can modify the expression and mouth shape of the target character. The main advancement of Face2Face lies in deforming various algorithms, including improvements in RGB tracking algorithms, transfer functions, and the establishment of mouth models. FaceSwap is used to transfer the face area from the source video to the target video. For the source video, the method first extracts the facial area of the source video and its corresponding facial landmarks and then fits a 3D model. For the target video, the method uses the same approach to fit the 3D model, which is rendered by the texture coordinates obtained from the 3D model of the source video to produce the final face-changing video. Neural Textures uses expression migration to modify the texture map of the target actor's face to match the expression of the source actor. This texture map is used to sample the neural texture of the target character. Then, the method inputs the sampled neural texture map to the delayed neural renderer and outputs the final reproduction result after end-to-end training.

2.2. Detection of Face Forgery. With the development of face tampering technology, the forged images and videos produced are close to genuine, which has aroused concerns and attracted attention to research on detection technology for face tampering. Existing detection methods can be roughly divided into two types: detection for tampered images and videos.

2.2.1. Detection Methods for Tampered Images. This type of method aims to extract the features of the single image for classification. Some traditional manual features such as speeded up robust features (SURF) [7], photo response nonuniformity (PRNU) [8], local binary pattern (LBP) [9], image quality measures (IQM) [10], etc., can be used to detect tampered images. However, the accuracy of these methods is not competitive on large datasets. With the rapid development of deep learning, face forgery detection has also made extensive use of deep learning. Deep neural networks (DNN) are used to extract the features of a single image or as classifiers. Some methods use DNN to extract the frequency features of the images [19–22]. For example, Luo et al. [20] found that current CNN-based detectors tend to overfit to method-specific color textures and thus fail to generalize, so they proposed to utilize the high-frequency noises for face forgery detection by devising three functional modules observing image noises remove color textures and expose discrepancies between authentic and tampered regions. Besides, the unique biological features of face images are used as classification features by some methods [23–26]. Matern et al. [24] proposed a method to detect Deepfake videos based on the visual features of eyes, teeth, and facial contours. However, this method has certain requirements for the test images, such as that the images need to include clear eyes or teeth. References [27–29] effectively used the texture or boundary features of the images and had a certain improvement in cross-database detection performance. There are some methods that use specific neural networks to detect tampered images with end-to-end training [1, 11–15, 39] and some methods [16–18] also introduce attention mechanism on this basis. These methods rely on the powerful adaptive learning ability of the neural network and the focus of the methods is therefore on the construction

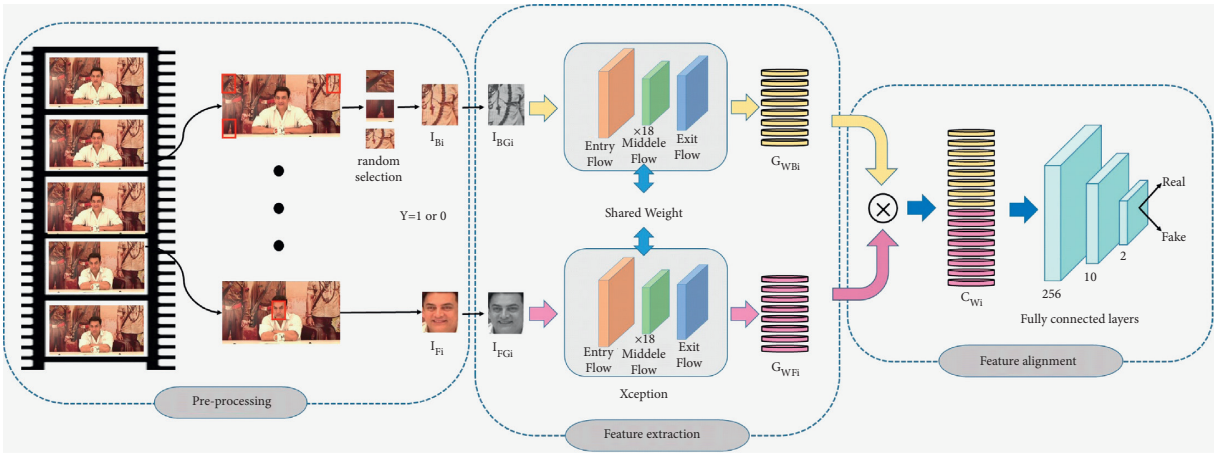

FIGURE 1: Overview of the proposed method. Our detection framework includes three modules. The preprocessing module is used to crop face area patches and background patches of video frames, where I_{Bi} and I_{Fi} ($i = 1, 2, \ldots, N$) represent the face patch and the background patch, respectively. N represents the number of videos. The feature extraction module converts patch pairs into greyscale, i.e., I_{BGi} and I_{FGi}, and then extracts features in the grey space of the pairs, i.e., G_{WBi} and G_{WFi}, using a two-stream network with shared weight. The feature alignment module mines their similarity by concatenating features from different areas and obtains the final classification result. C_{Wi} denotes aligned features. ⊗ represents the concatenation operation. The final classification result is obtained after C_{Wi} goes through three fully connected layers.

(a) (b) (c) (d) (e)

FIGURE 2: Examples of genuine images and four tampering methods. (a) Genuine images. (b) Deepfake. (c) Face2Face. (d) FaceSwap. (e) Neural Textures.

of the backbone network or attention network and good performance has been achieved. It is emphasized that methods based on features of the individual image can also be used to determine the authenticity of the videos.

2.2.2. Detection Methods for Tampered Videos. This type of method mainly uses the continuity and consistency of various features between video frames to determine authenticity. Therefore, it relies on the timing of the video

frames, and the detection object can only be a video, not a single image [30–38]. Haliassos et al. [38] proposed a detection called LipForensics which targets high-level semantic irregularities in mouth movements, which are common in many generated videos. But it requires a large-scale labelled dataset for pretraining. Zheng et al. [32] explored taking full advantage of the temporal coherence for video face forgery detection utilizing a novel end-to-end framework, which consists of two major stages. The temporal consistency of video frames is also used in [30–33]. Li et al. [36] proposed a long-term recurrent convolutional network (LRCN) to detect the blinking frequency of people in the videos and to compare it with the blinking frequency of normal people to distinguish between genuine and tampered videos. However, because the blinking frequency in high-quality tampered videos is almost the same as that of normal people, the prospective application of this method is not ideal. Agarwal et al. [37] used an open-source facial behavior analysis toolkit, Openface, to model the faces of five political celebrities in order to distinguish the authenticity of the videos. However, because there are not as many genuine and tampered videos for ordinary people as for politicians, this method has limited applications.

2.3. Siamese Network. The Siamese network is used to learn a function that maps the inputs into a target space such that the L_1 norm in the target space approximates the semantic distance in the input space. The details of the architecture are given in Figure 3. X_1 and X_2 are the inputs shown to the network, W is the shared parameter vector between CNNs, and G_W (X_1) and G_W (X_2) are the two points in the low-dimensional space that are generated by mapping X_1 and X_2. E_W is a function that measures the compatibility between X_1 and X_2.

3. Proposed Method

As shown in Figure 1, our method consists of three modules, i.e., preprocessing (Subsection III-A), feature extraction (Subsection III-B), and feature alignment (Subsection III-C). In addition, we introduce the voting principle in Subsection III-D.

3.1. Preprocessing. The feature extraction module takes image patch pairs as input, so we need to crop each video frame into image patch pairs. For each video in the datasets, we first use the software package dlib [47] to detect the face area of each frame in the video, and we crop a fixed-size face image patch according to the center of the face. We crop three corner background patches of the image to the same size as face image patches, excluding the lower right corner. It should be noted that the three corners are selected to facilitate cropping and improve the efficiency of preprocessing. In fact, it can be cropped anywhere on the background of images. And the number of cropped background patches can also be any odd number which is convenient for the voting principle (which will be introduced in Subsection *D*) other than three. The three

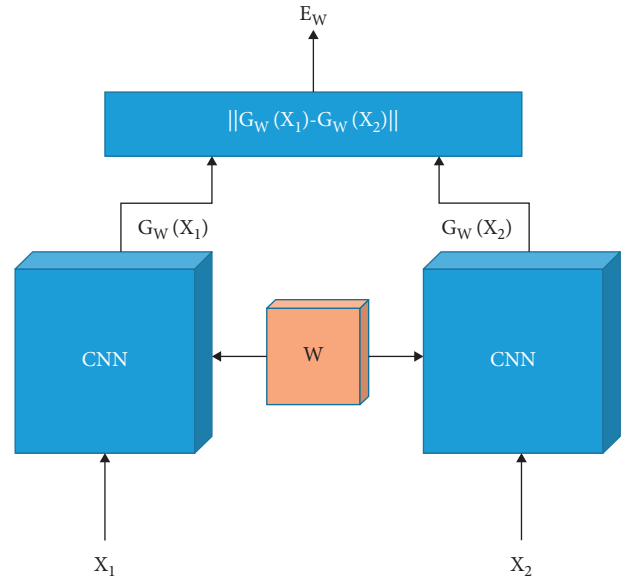

Figure 3: The architecture of the Siamese network.

background patches are later used by the voting principle to calibrate our test results. In the preprocessing stage, we finally process the videos in each dataset $F_i(i = 1, 2, \ldots, N)$, where N represents the number of videos, into face patches I_{Fi} and background patches I_{Bi}, as described in detail in Algorithm 1.

3.2. Feature Extraction. Artefacts may be left on videos due to hardware and software differences and manufacturing imperfections. In one genuine video, the artefacts are consistent and continuous in general, such that the facial area and the background have high similarity, while in the tampered video (e.g., generated by Deepfake and FaceSwap), the similarity between the face area and the background is lower. The tampered videos generated by Face2Face and Neural Textures only modify the facial expression and some attributes and are not directly derived from different videos, but the tampering still impacts the consistency of the artefacts.

To this end, we use the improved Siamese network to measure the similarity between the face area and the background of the video frames. We employ the Xception network [48] as the backbone of the Siamese network. The Xception network is currently one of the most effective and widely used networks, as in [1, 19, 20, 22], for face forgery detection. The advantage of deep learning lies in its powerful computing ability and autonomous learning ability. Through end-to-end training and supervised learning, the convolutional neural network extracts the suitable and effective features in the grey space of the images.

After the preprocessing module, we obtain a pair of image patches. In the feature extraction module, we first convert the pair of image patches to greyscale. Since the semantic content of the face patch and the background patch is very different, greying the pair of patches can reduce the impact of the semantic content so that the network can

Input: Videos of the dataset: F_i, $i = 1, 2, \ldots, N$, N: The number of videos
Output: Images patches I_B and I_F, I_{Bi}: background patches, $I_{F\ i}$: face patch
for *each* F_i **do**
  if F_i *is a real video* **then**
    Assign F_i to the set F_r
  else
    Assign F_i to the set F_t
  crop **face** patch I_{Fi} which is assigned to the set F_f and three background patches I_{Bi} which are assigned to the set F_b

Algorithm 1: Preprocessing.

concentrate more on the low-level features with better generalization performance. Then the pair of patches are given to the Xception networks with shared weights to get the 512-dimension feature maps in the grey space. Sharing weights ensures that the two streams of the network mine the features of the same space, and at the same time it is equivalent to enriching the feature data of each stream, making the network more efficient. And the feature maps can be regarded as features of the noise distribution of the image patches.

3.3. Feature Alignment. After obtaining the features of the face patch G_{WFi} ($i = 1, 2, \ldots, N$), where N represents the number of videos, and the features of the background patch G_{WBi} in the grey space, it is significant to measure their similarity in order to distinguish whether they are from genuine images or tampered images. The most direct way to accomplish this goal is to perform a residual operation on two feature maps, similar to what the original Siamese network does, but this is not suitable for image patches with large differences in semantic content. Thus, in the feature alignment module, we concatenate G_{WFi} and G_{WBi} and acquire the aligned features, which are 1024-dimensional feature maps, i.e., C_{Wi}, defined as

$$C_{Wi} = G_{WFi} \otimes G_{WBi}. \tag{1}$$

$\otimes$ represents concatenating G_{WFi} and G_{WBi}. C_{Wi} is then input to the fully connected layers behind. There are three fully connected layers that have 256, 10, and 2 nodes in sequence. The aligned features retain all the feature information of the image patch pair so that the following fully connected layers can fully mine the similarity between them and make the learning process more stable and robust in order to achieve more satisfactory performance.

The aligned features are very robust for classification. Limited by current technical conditions, no matter what kind of face tampering technology is employed, the focus is on the continuity of semantic content, and damage to the continuity of the noise artefacts in certain feature spaces is inevitable. Therefore, compared with the genuine videos, even if tampered videos undergo a variety of postprocessing operations, the similarity between the face patch and the background patch remains at a relatively low level. Extracting the features in the grey space of the images and measuring the similarity by concatenating features greatly reduce the influence of the semantic content of the images. This approach enables our method to maintain satisfactory detection performance for tampered images and videos with high compression factors.

Under the supervised and end-to-end training, the feature alignment module can measure the similarity between the face patch and the background patch and produce the final classification result. We train our network by minimizing the cross-entropy loss function, which is defined as

$$\text{Loss} = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - (\hat{y}))]. \tag{2}$$

y represents the labels of image patches and $\hat{y}$ represents the classification results output by the network. The fully connected layers of the feature alignment module act as a classifier.

Algorithm 2 describes the entire training process in detail.

3.4. Voting Principle. To obtain more accurate classification results, we define a voting principle in the test stage to modify them. The difference from the training is when we randomly select an image patch, we will select three patches from the same frame as the face patch and make them form three patch pairs by copying the face image patch with the three background patches. The three pairs of patches are then input into our trained feature extraction module and feature alignment module, and three binary predicted labels are obtained. Finally, according to the voting principle that the minority obeys the majority, the predicted label, that is, the classification result of the image to which the face patch and background patches belong, is obtained. At the same time, as we emphasized in Subsection III-A, the number of background patches can also be any odd number other than three. The details of the voting principle can be found in Figure 4. Let I_F be the face patch and let I_{B1}, I_{B2}, and I_{B3} be the three corresponding background patches. Let Y_1, Y_2, Y_3, and Y_t be the prediction labels of three patch pairs and the final prediction label, respectively. $Y_t = 1$ means that the image is genuine and $Y_t = 0$ means that image is tampered. Table 1 illustrates the voting principles between Y_t and the labels of the three patch pairs.

4. Experiments

In this section, we first introduce the datasets that we used in the experiment, and then, we introduce our experimental setup and detailed training process. Finally, we report the

Input: The pair of image patches: I_{Fi} and I_{Bi}, $i = 1, 2, \ldots, N$
N: The number of videos
Output: The prediction label Y_{ti}, I_{Bi}: background patches, $I_{F\ i}$: face patch
while epoch ≤ 30 **do**
  for each *pair of* I_{Fi} *and* I_{Bi} **do**
    if I_{Fi} *and* I_{Bi} *are from* F_r **then**
      label $l = 1$
    else
      I_{Fi} and I_{Bi} are from F_t, label $l = 0$
    Greying I_{Fi} to I_{FGi} and I_{Bi} to I_{BGi}
    Mapping I_{FGi} to $G_{W\ Fi}$ and I_{FGi} to $G_{W\ Bi}$ with shared weights W
    Concatenating G_{WBi} and G_{WFi} to C_{Wi}
    Mapping C_{Wi} to get label Y_{ti}
return Siamese network model

Algorithm 2: Training.

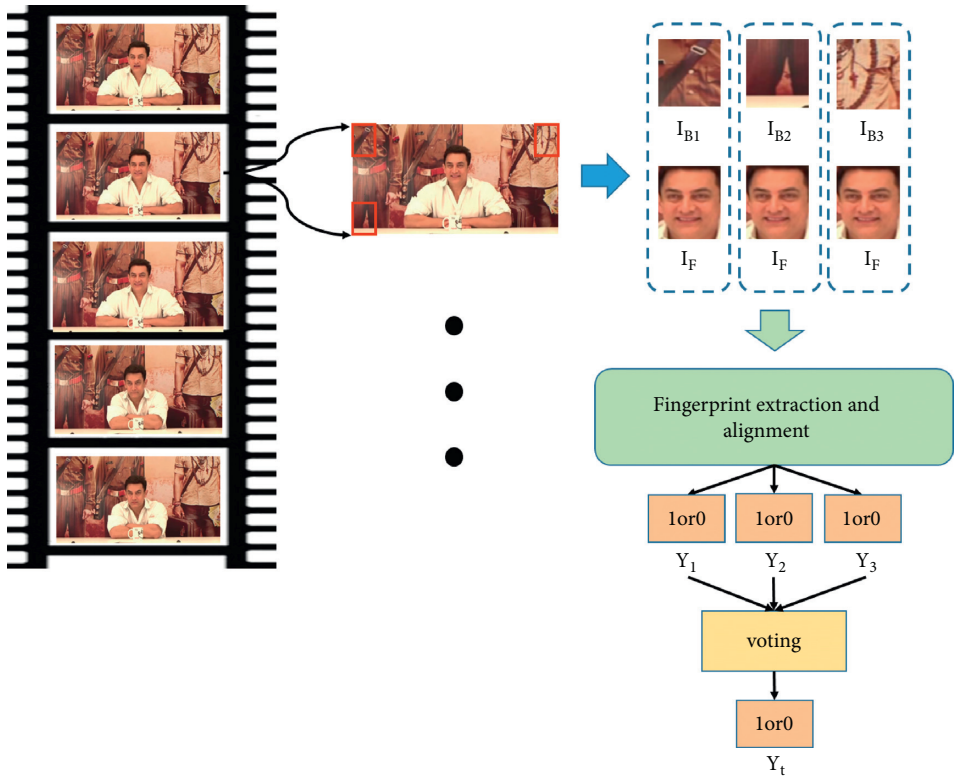

Figure 4: The details of the voting principle. It is used in the test stage to modify the classification results. I_F is the face patch, while I_{B1}, I_{B2}, and I_{B3} are the three corresponding background patches. Y_1, Y_2, and Y_3 are the prediction labels of three patch pairs, and the final predicted label, Y_t, is obtained according to the voting principle from Y_1, Y_2, and Y_3.

Table 1: The voting principles between Y_t and Y_1, Y_2, and Y_3. $Y_t = 1$ means that the image is genuine and $Y_t = 0$ means that the image is tampered.

Y	Binary label							
Y_1	1	1	1	0	1	0	0	0
Y_2	1	1	0	1	0	1	0	0
Y_3	1	0	1	1	0	0	1	0
$\boldsymbol{Y_t}$	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**

The bold results are the final prediction label.

performance of our proposed method and analyze the experimental results in detail.

4.1. Datasets. We used three datasets in our experiments: the FaceForensics++ dataset [1], the Celeb-DF(v2) dataset [40], and the UADFV dataset [23]. FaceForensics++ is a forensics dataset consisting of 1000 original video sequences that have been manipulated with four automated face manipulation methods: Deepfake (DP), Face2Face (F2), FaceSwap (FS),

and Neural Textures (NT); i.e., it contains four subdatasets. The data have been sourced from 977 YouTube videos, and all videos contain a trackable mostly frontal face without occlusions, which enables automated tampering methods to generate realistic forgeries. All videos have three resolutions, i.e., raw quality without compression, high quality with a light compression using a quantization of 23, and low quality with a heavy compression using a quantization of 40.

The UADFV dataset contains 98 videos, with 49 genuine videos and 49 tampered videos. All tampered videos are generated by the method of Deepfake. Each video has one subject and lasts approximately 11 seconds, with a typical resolution of 294 500 pixels. The Celeb-DF(v2) dataset is a large-scale challenging dataset for Deepfake forensics. It includes 590 original videos collected from YouTube, with subjects of different ages, ethnic groups, and genders, and 5639 high-quality Deepfake videos generated using an improved synthesis process. The overall visual quality of the synthesized Deepfake videos in the Celeb-DF dataset is greatly improved when compared to other datasets, with significantly fewer notable visual artefacts. In addition, the genuine video shows a wide range of changes in the subject's face size, orientation, lighting conditions, and background.

4.2. Implementation Details. In our experiment, we used the software package dlib [47] to detect faces in the frames of the videos and extract the face area, but we decided to eliminate some videos in the datasets for which the face extraction failed. For every subdataset of the FaceForensics++ dataset, we select 976 tampered videos, among which 681 videos were used as the training set, 145 videos are used as the validation set, and the other 145 videos are used as the test set. For the UADFV dataset, we selected 43 tampered videos, of which 31 videos are used as the training set, 6 videos are used as the validation set, and the other 6 videos are used as the test set. The number of genuine videos is the same as the number of tampered videos. In each video, we randomly select 50 pairs of face patches and background patches for training and 150 pairs of patches for validation and testing due to the need for the voting principle.

For the Celeb-DF(v2) dataset, because the number of genuine videos is far less than that of tampered videos, we use two methods to divide the dataset in order to ensure the balance of genuine and tampered data during the training process. One method is to divide the data according to the quantity balance; that is, for both genuine and tampered videos, 400 videos are selected for training, 50 videos for validation, and 50 videos for testing, and 50 pairs of patches are randomly selected in each video for training and 150 pairs of patches are selected for validation and testing. The other method is based on the proportional balance; that is, the genuine videos are divided in the same way as the previous method, but for tampered videos, 4,000 are selected for training, 500 for validation, and 500 for testing, while only 5 pairs of patches for training and 15 pairs of patches for validation and testing in each video are randomly selected in order to keep the quantities of tampered data and genuine data the same. The precise numbers of the patch pairs for each dataset can be found in Table 2.

Table 2: Precise numbers of patch pairs for training, validation, and testing of the three datasets.

Dataset	Number of pairs		
	Training	Validation	Test
FaceForensics++ [1]	68600	43500	43500
UADFV [23]	3100	1800	1800
Celeb-DF(v2) [40]	40000	15000	15000

All networks have been implemented with Python 3.7 using PyTorch. Weight optimization of the network is achieved with successive batches of 16. The sizes of face patches and background patches are both 256 256. The networks are optimized via Adam [49] with default parameters ($\beta 1 = 0.9$ and $\beta 2 = 0.999$). We adjust the learning rate by combining warm-up and stepwise methods. We set the base learning rate as 0.0001. Every training process contains 30 epochs: 10 are used to warm-up, 10 are maintained at the base learning rate, and then, the learning rate is divided by 10 every 5 epochs.

4.3. Evaluation Metrics. We apply the accuracy score (Acc) and the area under the receiver operating characteristic (ROC) curve (AUC) values that are commonly used in face forgery detection as our evaluation metrics. In addition, we apply precision (P), recall (R), and the F1 score on the challenging low-quality data from the FaceForensics++ dataset [1] to better evaluate the performance of our method.

4.4. Results. We first compare the performance of our network with the three most widely used networks based on the four subdatasets of the FaceForensics++ dataset with different quality. The results are listed in Table 3.

As these results show, except for the subdataset of Neural Textures (NT) with high quality, our method outperforms all the reference methods and different face manipulation methods with respect to all quality settings. It is worth noting that our method achieves Acc values of 84.14%, 97.97%, 98.88%, and 98.21% on the subdatasets of Deepfake (DP), Face2Face (F2), FaceSwap (FS), and Neural Textures (NT) with low quality, respectively. The performance of our method far exceeds that of the reference methods; in particular, the performance becomes even better after use of the voting principle to correct the results, with values of 84.14%, 96.62%, 99.49%, and 98.90% achieved. Moreover, compared to the results on the same subdataset with raw quality and high quality, the Acc scores of reference methods have significantly declined. However, except on the DP subdataset, the performance of our method on low-quality datasets is close to that of raw quality. The previous methods for face forgery detection can mine the differences in feature distribution between genuine and tampered images to find the traces of tampered images. The image compression eliminates the forgery traces to a certain extent so that the differences in the feature distribution of genuine and tampered images are reduced. Therefore, the performance of the network will also be reduced accordingly. However, our method determines the authenticity of the images by

Table 3: Acc score on the FaceForensics++ dataset. LQ represents low quality, HQ represents high quality, and Raw represents raw quality.

Method	Dataset											
	LQ				HQ				Raw			
	DP	F2	FS	NT	DP	F2	FS	NT	DP	F2	FS	NT
Meso4 [11]	77.68	83.65	79.92	77.74	89.77	94.25	95.50	78.70	96.37	97.95	98.17	93.30
MesoInception4 [11]	74.20	78.75	79.72	67.94	83.74	91.48	94.34	75.06	88.34	97.65	97.81	92.52
Xception [48]	83.70	87.21	83.17	87.90	95.15	97.07	95.96	**87.99**	98.31	97.75	98.10	96.45
Our method	**84.14**	97.97	98.88	98.21	**95.79**	97.11	97.37	84.69	**98.72**	97.91	98.75	97.33
Our method (voting)	**84.14**	**98.62**	**99.49**	**98.90**	95.77	**97.12**	**97.37**	84.71	**98.72**	**97.92**	**98.77**	**98.18**

The bold results show the best.

comparing the similarity between the face area and the background of the images, which greatly enhances the robust performance of the network so that postprocessing similar to image or video compression has a relatively small impact on the performance. In addition, from the results in Table 3, it can be concluded that the voting principle does not achieve better results on the DP subdataset; specifically, on the datasets with raw and low quality, the Acc scores are equal to those of the method not employing the voting principle, and on the high-quality dataset, the score even becomes slightly lower. Overall, however, the voting principle is still beneficial to the results.

We then evaluate our method on the UADFV and Celeb-DF(v2) datasets. The results are shown in Table 4. The proposed method achieves 99.94% Acc performance on the UADFV dataset, and the score even reaches 1.00 by voting, although this is only a small improvement compared to the Xception network. With respect to the ways that the Celeb-DF(v2) dataset is divided according to the proportional balance and the quantity balance, our method achieves 92.61% and 94.94%, respectively, exhibiting remarkable improvement compared to the reference methods. These results prove the superiority of our method.

To better evaluate the performance of our method on the low-quality datasets, we calculate precision (P), recall (R), and the F1 score of all methods, as shown in Table 5, and generate ROC curves of different methods as shown in Figure 5 on the FaceForensics++ dataset with low quality. It can be seen from the results in Table 5 that, compared with the reference methods, our method has achieved better performance with respect to all evaluation metrics on the four subdatasets. The AUC values of the proposed method, i.e., the area values in Figure 5, are far ahead, with the exception that the results on the DP subdataset are close to those of Xception.

4.4.1. Comparison with Recent Works on the Low-Quality Datasets of FaceForensics++ [1]. In order to demonstrate the competitive results of our method on low-quality datasets, we compared our results with recent methods [14, 19, 20, 22, 25, 28, 39, 50–52]. Since the experimental sets between us and others are almost the same, we directly used the results in these papers. The results are shown in Table 6.

Accuracy scores marked in bold represent the highest accuracy scores. The Acc of our method in some categories exceeds all the reference methods, i.e., F2, FS, NT. These results fully demonstrate that our method exhibits very good and robust performance and generalization ability on challenging low-quality datasets and that the impact of compression processing is very small, which is extremely important for the practical application and promotion of the detection methods.

Table 4: Acc score on the UADFV and Celeb-DF(v2) datasets. C P and C Q represent the way in which the Celeb-DF dataset is divided according to the proportional balance and the quantity balance, respectively.

Method	Dataset		
	UADFV [23]	Celeb-DF(v2) [40]	
		C_P	C_Q
Meso4 [11]	82.67	87.10	83.75
MesoInception4 [11]	96.33	88.10	70.15
Xception [48]	99.33	90.78	89.64
Our method	99.94	92.61	**94.94**
Our method (voting)	**100.00**	**92.62**	94.93

4.5. Discussion on Other Influencing Factors

4.5.1. Effect of Size of Image Patches. To evaluate the impact of image patch size on network performance, we used the patch sizes of 256×256, 192×192, and 128×128 to conduct ablation tests on the FaceForensics++ dataset with low quality. The results are shown in Table 7. The size of the image patches exerts an obvious influence on the performance of our method. For the size of 256×256, our method including the voting principle achieves the leading performance on all datasets, but for the sizes of 192×192 and 128×128, our method offers better performance on only three datasets. The impact of the size also differs according to different tampering methods. For Deepfake, the result for the size of 192×192 is the best, but for the other three methods, the results are best for the size of 256×256. In general, our method performs best for the size of 256×256.

4.5.2. Effects of Different Tampering Methods. From Table 3, it can be concluded that our method has a higher accuracy rate for the tampered images generated by FaceSwap with different quality. This is because FaceSwap has a simpler production principle and process than the other three methods. The most difficult tampering methods to detect for our method are Deepfake, Neural Textures, and Face2Face on the datasets of

Table 5: Precision (P), Recall (R), and $F1$ score on the FaceForensics++ dataset with low quality.

Method	P				R				$F1$			
	DP	$F2$	FS	NT	DP	$F2$	FS	NT	DP	$F2$	FS	NT
Meso4 [11]	77.88	82.71	78.07	80.13	77.32	85.08	83.21	73.77	77.60	83.88	80.56	76.82
MesoInception4 [11]	80.91	87.91	92.49	86.48	63.35	66.68	64.69	42.54	71.06	75.83	76.13	57.03
Xception [48]	82.96	86.01	81.41	85.02	84.82	88.88	85.97	92.01	83.88	87.42	83.63	88.38
Our method	**83.36**	**98.15**	**98.85**	**98.34**	**85.84**	**97.78**	**98.91**	**97.91**	**84.58**	**97.97**	**98.88**	**98.12**

The bold results show the best.

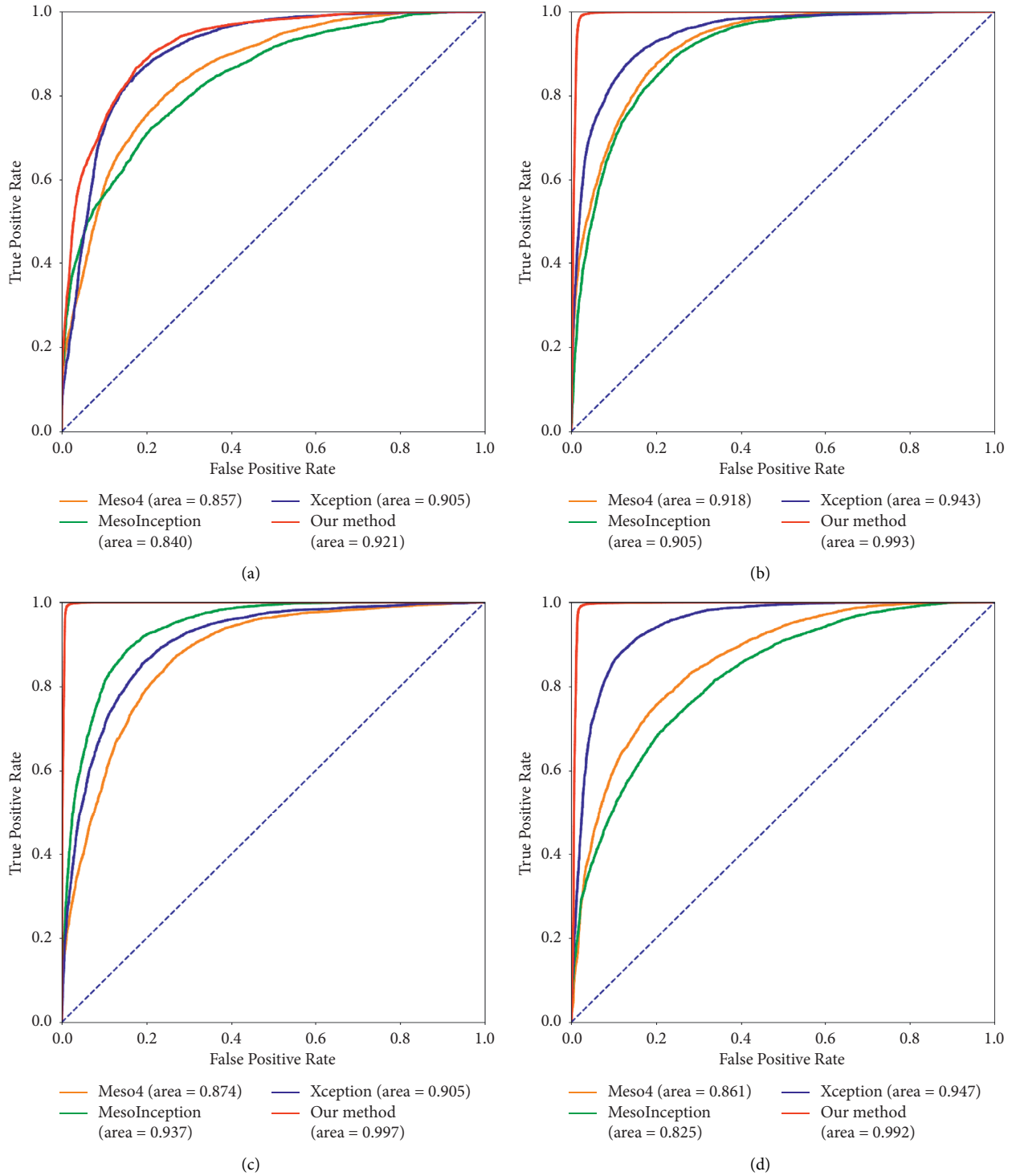

Figure 5: ROC curves of different methods based on the FaceForensics++ dataset with low quality. (a) Deepfake. (b) Face2Face. (c) FaceSwap. (d) Neural Textures.

Table 6: Comparative analysis of detection performance with recent methods on the low-quality datasets of FaceForensics++ [1]. The performances of [19, 25, 28, 39], [50, 51, 52] are obtained from [28], and others are from the original papers, respectively.

Method	Dataset			
	DP	*F2*	FS	NT
Durall et al. [50]	71.69	65.66	65.43	59.34
DSP-FWA [25]	93.60	91.77	90.73	83.15
Liu et al. [51]	92.39	90.67	91.99	84.69
Qian et al. [19]	96.01	93.62	94.33	86.37
Bondi et al. [52]	94.95	91.33	94.26	87.79
Bonettini et al. [39]	96.13	92.93	94.09	88.15
Khalid et al. [14]	88.40	71.20	86.10	97.50
Liu et al. [22]	93.48	86.02	92.26	76.78
Luo et al. [20]	**98.60**	95.70	92.90	—
Yang et al. [28]	97.88	96.85	96.87	88.47
Our method	84.14	97.97	98.88	98.21
Our method (voting)	84.14	**98.62**	**99.49**	**98.90**

The bold results show the best.

Table 7: Acc scores of different sizes of image patches based on the FaceForensics++ dataset with low quality.

Method	Dataset											
	256 × 256				192 × 192				128 × 128			
	DP	F2	FS	NT	DP	F2	FS	NT	DP	F2	FS	NT
Meso4 [11]	77.68	83.65	79.92	77.74	56.16	55.54	61.98	51.81	57.59	56.64	56.26	50.06
MesoInception4 [11]	74.20	78.75	79.72	67.94	76.23	64.25	63.46	71.40	67.97	64.15	70.22	64.41
Xception [48]	83.70	87.21	83.17	87.90	78.47	**67.84**	71.95	82.86	**77.26**	61.75	73.86	63.59
Our method	**84.14**	97.97	98.88	98.21	**84.74**	66.33	78.52	94.01	76.05	**65.18**	74.06	79.31
Our method (voting)	**84.14**	**98.62**	**99.49**	**98.90**	**84.74**	66.34	**78.54**	**95.74**	76.06	65.17	**74.15**	**80.72**

The bold results show the best.

low quality, high quality, and raw quality, respectively. Therefore, different tampering methods should be tested with different preprocessing operations in practical applications.

4.5.3. Effects of Different Image Modes. In our basic experiment, we have processed all image patches into greyscale mode. To compare the impacts of different image modes on the classification performance, we used the image patches of the RGB mode to conduct a comparative experiment. The experiment is performed on the FaceForensics++ dataset with low quality. Figure 6 shows the results of the comparison experiment. It can be determined that the classification performance in the greyscale mode is better than that in the RGB mode for each subdataset, and it is even more superior than Face2Face and FaceSwap. This result shows that our method can find a more suitable feature distribution in the grey space to distinguish between real and tampered images. And the reason may be that the grey domain reduces the relevant semantic features produced by colors compared to the RGB domain, so that our network can find more general features.

4.5.4. Effect of Concatenating Features. We use feature subtraction instead of concatenation in the feature alignment module to conduct a comparative experiment, and the experiment is performed on the FaceForensics++ dataset with low quality. The results are shown in Figure 7. It is

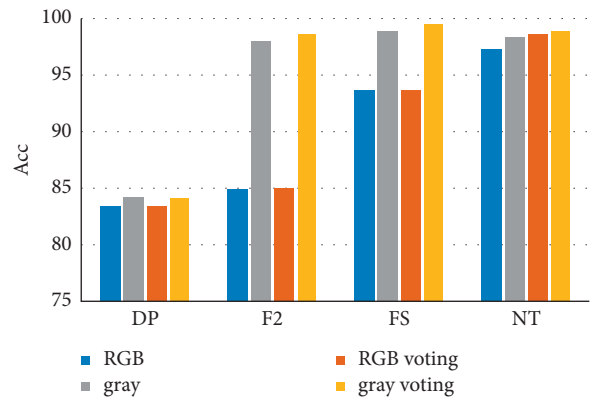

Figure 6: Results for the impacts of different image modes on classification performance. The classification performance in the greyscale mode is better than that in the RGB mode on all subdatasets, especially for Face2Face and FaceSwap.

obvious that concatenating features is more effective. In fact, the subtraction operation is more suitable for use in face recognition tasks with image pairs including similar semantic content. In our task, the face area and the background area are divergent in semantic content, the effect of the subtraction operation is greatly reduced, and the effect is almost completely lost for Face2Face and FaceSwap. The concatenation operation allows the fully connected layer to be classified under richer feature conditions, resulting in better performance.

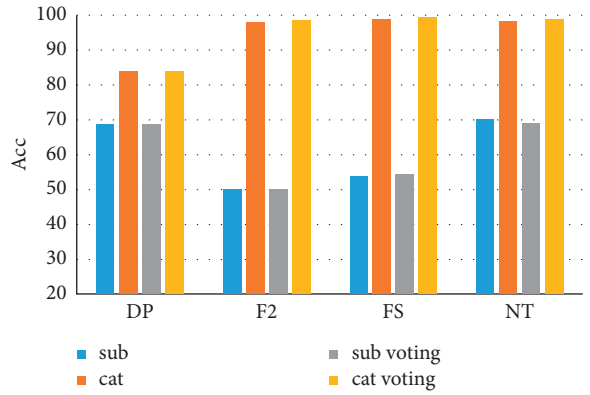

Figure 7: The results of the impact of concatenating features on classification performance. In the figure, cat represents concatenation and sub represents subtraction. The classification performance of the method with feature concatenation far exceeds that of the method with features subtraction on all subdatasets.

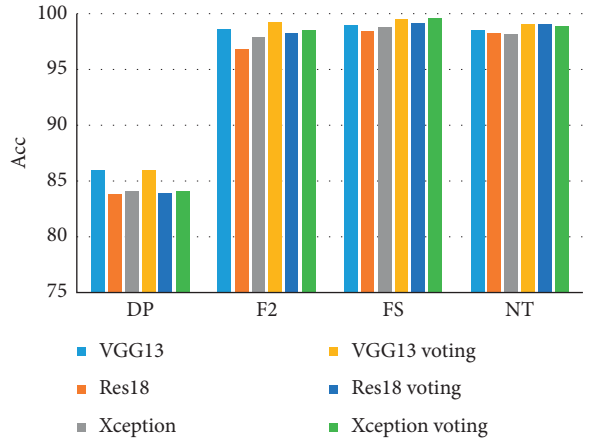

Figure 8: The results of the impacts of different backbones of the feature extraction module on classification performance. The classification performance of the framework used in VGG13 has achieved better results. These results illustrate the universality of our method for different classification networks.

4.5.5. Effects of Different Backbones of the Feature Extraction Module. We chose Xception [48], which is currently the most widely used network in the field of face forgery detection, as the backbone of the feature extraction module. However, the backbone of our feature extraction module based on the Siamese framework can also be some general classification networks. To explore the universality of our method, we use VGG13 [53] and ResNet18 [54] to conduct a comparative experiment: the experiment is performed on the FaceForensics++ dataset with low quality. As shown in Figure 8, the overall performance of the detection framework using Xception because of the backbone is slightly better than that of ResNet18, but slightly worse than VGG13. This finding shows to a certain extent that our method still has the potential to continue to improve and that it can be adapted to some general classification networks.

Through these ablation experiments, we explore the impacts of different conditions on our methods. At the same time, it can also be learned that, for images and videos with different resolution and those generated by different forgery methods, we should use the framework with different details to achieve the best results. The generalization performance of the method will be the focus of future work.

5. Conclusion

The development of deep learning has significantly improved the quality and efficiency of generating forged face images and videos. In this paper, we propose an innovative face forgery detection framework based on the improved Siamese network, which extracts and aligns the features of the face area and the background of the image and then mines the similarity between them to determine the authenticity of the image. This framework not only offers great robustness and generalization performance but also makes full use of the feature information of the image background. We evaluate our method on several different datasets, thus proving its effectiveness in practice, especially that it achieves impressive results on low-quality datasets.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. U1936117, 62106037, 62076052, and 61772111), the Science and Technology Innovation Foundation of Dalian (no. 2021JJ12GX018), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (no. 202100032), and the Fundamental Research Funds for the Central Universities (DUT21GF303, DUT20TD110, and DUT20RC(3)088).

References

[1] A. Ro¨ssler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: learning to detect manipulated facial images," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–11, Seoul, South Korea, January 2019.

[2] Deepfakes, "Deepfakes," 2018, https://github.com/deepfakes/.

[3] J. Thies, M. Zollho¨fer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: real-time face capture and reenactment of rgb videos," *Communications of the ACM*, vol. 62, pp. 96–104, 2019.

[4] Faceswap, "Faceswap," 2018, https://github.com/.

[5] J. Thies, M. Zollho¨fer, and M. Nießner, "Deferred neural rendering: image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[6] B. Chen, W. Tan, G. Coatrieux, Y. Zheng, and Y.-Q. Shi, "A serial image copy-move forgery localization scheme with source/target distinguishment," *IEEE Transactions on Multimedia*, vol. 23, pp. 3506–3517, 2021.

[7] Y. Zhang, L. Zheng, and V. L. L. Thing, "Automated face swapping and its detection," in *Proceedings of the 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pp. 15–19, Singapore, November 2017.

[8] M. Koopman, A. Macarulla Rodriguez, and Z. Geradts, "Detection of deepfake video manipulation," in *Proceedings of the 20th Irish Machine Vision and Image Processing Conference (IMVIP)*, pp. 133–136, Belfast, Ireland, 2018.

[9] A. Khodabakhsh, R. Raghavendra, K. Raja, P. Wasnik, and C. Busch, "Fake face detection methods: can they be generalized?" in *Proceedings of the 2018 Interna- tional Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–6, Germany, September 2018.

[10] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recogni- tion? assessment and detection," 2018, https://arxiv.org/abs/1812.08685.

[11] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a com- pact facial video forgery detection network," in *Proceedings of the2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, Hong-Kong, China, December 2018.

[12] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *Proceedings of the 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8, Tampa, FL, USA, 2019.

[13] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: using capsule networks to detect forged images and videos," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2307–2311, Brighton, UK, May 2019.

[14] H. Khalid and S. S. Woo, "Oc-fakedect: classifying deepfakes using one-class variational autoencoder," in *Proceedings of the 2020 IEEE/CVF Conference on Com- puter Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2794–2803, June 2020.

[15] P. Zhou, X. Han, V. Morariu, and L. Davis, "Two-stream neural networks for tampered face detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1831–1839, Honolulu, HI, USA, July 2017.

[16] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi- attentional deepfake detection," in *Proceedings of the 2021 IEEE/CVF Conference on Com- puter Vision and Pattern Recognition (CVPR)*, pp. 2185–2194, 2021.

[17] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 918–1014 927, Nashville, TN, USA, June 2021.

[18] B. Chen, X. Liu, Y. Zheng, G. Zhao, and Y.-Q. Shi, "A robust gan- generated face detection method based on dual-color spaces and an improved xception," in *Proceedings of theIEEE Transactions on Circuits and Systems for Video Technology*, November 2021.

[19] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: face forgery detection by mining frequency-aware clues," in *Proceedings of the European Conference on Computer Vision*, pp. 86–103, Springer, Glasgow, UK, 2020.

[20] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16 312–316 321, Nashville, TN, USA, June 2021.

[21] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, "Frequency-aware discrim- inative feature learning supervised by single-center loss for face forgery detection," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6454–6463, 2021.

[22] H. Liu, X. Li, W. Zhou et al., "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 772–781, 2021.

[23] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265, Brighton, UK, May 2019.

[24] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proceedings of the 2019 IEEE Winter Applica- tions of Computer Vision Workshops (WACVW)*, pp. 83–92, Waikoloa Village, HI, USA, February 2019.

[25] 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, IEEE, Long Beach, CA, USA, June 2019.

[26] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Li, "Face forgery detection by 3d decomposition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 46–52, Nashville, TN, USA, June 2021.

[27] 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 (CVPR)*, pp. 5000–5009, Seattle, WA, USA, March 2020.

[28] J. Yang, A. Li, S. Xiao, W. Lu, and X. Gao, "Mtd-net: learning to detect deepfakes images by multi-scale texture difference," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4234–4245, 2021.

[29] J. Yang, S. Xiao, A. Li, G. Lan, and H. Wang, "Detecting fake images by identifying potential texture difference," *Future Generation Computer Systems*, vol. 125, pp. 127–135, 2021.

[30] D. Guera and E. Delp, "Deepfake video detection using recurrent neural networks," in *Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, February 2018.

[31] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," in *Proceedings of the CVPR Workshops*, Long Beach, CA, USA, June 2019, https://arxiv.org/abs/1905.00582.

[32] G. Jia, M. Zheng, C. Hu et al., "Inconsistency-aware wavelet dual-branch network for face forgery detection," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 308–319, 2021.

[33] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, "Exploring temporal coherence for more general video face forgery detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15 044–115 054, Montreal, Canada, October 2021.

[34] I. Amerini, L. Galteri, R. Caldelli, and A. D. Bimbo, "Deepfake video detection through optical flow based cnn," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1205–1207, Seoul, South Korea, March 2019.

[35] U. A. Ciftci and I. Demir, "Fakecatcher: detection of synthetic portrait videos using biological signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, 2020.

[36] Y. Li, M. C. Chang, and S. Lyu, "In ictu oculi: exposing ai created fake videos by detecting eye blinking," in *Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, Hong Kong, China, December 2018.

[37] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," *In CVPR Workshops*, 2019.

[38] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: a generalisable and robust approach to face forgery detection," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5037–5047http://arxiv.org/abs/2012.07657, Nashville, TN, USA, June 2021.

[39] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of cnns," in *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, pp. 5012–5019https://arxiv.org/abs/2004.07676, Milano, Italy, January 2021.

[40] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: a large-scale challenging dataset for deepfake forensics," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3204–3213, 2020.

[41] B. Dolhansky, J. Bitton, B. Pflaum et al., "The deepfake detection challenge dataset," 2020, https://www.arxiv-vanity.com/papers/2006.07397/.

[42] L. Jiang, W. Wu, R. Li, C. Qian, and C. C. Loy, "Deeperforensics-1.0: a large-scale dataset for real-world face forgery detection," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2886–2895, Seattle, WA, USA, June 2020.

[43] B. Zi, M. Chang, J. Chen, X. Ma, and Y. G. Jiang, "Wilddeepfake: a challenging real-world dataset for deepfake detection," in *Proceedings of the 28th ACM International Conference on Multimedia*, New York; NY, 2020.

[44] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, San Diego, CA, USA, June 2005.

[45] H. Farid, "Image forgery detection – a survey," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, 2009.

[46] P. Pe´rez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, pp. 313–318, 2003.

[47] D. King, "Dlib-ml: a machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[48] F. Chollet, "Xception: deep learning with depthwise separable con- volutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807https://arxiv.org/abs/1610.02357, Honolulu, HI, USA, July 2017.

[49] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2015, https://arxiv.org/abs/1412.6980.

[50] R. Durall, M. Keuper, F.-J. Pfreundt, and J. Keuper, "Unmasking deepfakes with simple features," 2019, https://arxiv.org/abs/1911.00686.

[51] Z. Liu, X. Qi, J. Jia, and P. H. S. Torr, "Global texture enhancement for fake face detection in the wild," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8057–8066https://arxiv.org/abs/2002.0013, Seattle, WA, USA, June 2020.

[52] L. Bondi, E. D. Cannas, P. Bestagini, and S. Tubaro, "Training strategies and data augmentations in cnn-based deepfake video detection," in *Proceedings of the 2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6https://arxiv.org/abs/2011.07792, New York City, NY, USA, December 2020.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, https://arxiv.org/abs/1409.1556.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778https://arxiv.org/abs/1512.03385, Las Vegas, NV, USA, June 2016.