Security and Communication Networks

Data-Driven Face Forensics and Security

Lead Guest Editor: Beijing Chen
Guest Editors: Guoying Zhao, Shunquan Tan, and Gouenou Coatrieux

Data-Driven Face Forensics and Security

Security and Communication Networks

Data-Driven Face Forensics and Security

Lead Guest Editor: Beijing Chen
Guest Editors: Guoying Zhao, Shunquan Tan, and Gouenou Coatrieux

Contents

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1325573, 12 pages
https://doi.org/10.1155/2021/1325573

Research Article

Multiplicative Watermarking Method with the Visual Saliency Model Using Contourlet Transform

Jinhua Liu,[1] **Jiawen Huang**,[1] **and Yuanyuan Huang**[2]

[1]*School of Mathematical and Computer Sciences, Shangrao Normal University, Shangrao 334001, China*
[2]*Department of Network Engineering, Chengdu University of Information Technology, Chengdu 610225, China*

Correspondence should be addressed to Jinhua Liu; liujinhua_uestc@126.com

Received 17 June 2021; Accepted 27 September 2021; Published 7 October 2021

Academic Editor: Beijing Chen

We have proposed an image adaptive watermarking method by using contourlet transform. Firstly, we have selected high-energy image blocks as the watermark embedding space through segmenting the original image into nonoverlapping blocks and designed a watermark embedded strength factor by taking advantage of the human visual saliency model. To achieve dynamic adjustability of the multiplicative watermark embedding parameter, the relationship between watermark embedded strength factor and watermarked image quality is developed through experiments with the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), respectively. Secondly, to detect the watermark information, the generalized Gaussian distribution (GGD) has been utilized to model the contourlet coefficients. Furthermore, positions of the blocks selected, watermark embedding factor, and watermark size have been used as side information for watermark decoding. Finally, several experiments have been conducted on eight images, and the results prove the effectiveness of the proposed watermarking approach. Concretely, our watermarking method has good imperceptibility and strong robustness when against Gaussian noise, JPEG compression, scaling, rotation, median filtering, and Gaussian filtering attack.

1. Introduction

Transmitting and sharing digital multimedia have become more convenient with the rapid development of the network. However, such phenomenon results in security issues, such as authentication, copyright protection, and fingerprinting [1–7]. Digital watermarking can be used as an effective method to address these problems. Generally, in the watermarking process, some useful information (e.g., watermark data) is embedded into an original signal while ensuring its quality. Furthermore, robustness and imperceptibility are the main factors in digital image watermarking. Many image watermarking algorithms have been presented in the literature. On the basis of the embedding method, most algorithms can be divided into three categories, namely, additive, quantization, and multiplication-based watermarking algorithms.

For the additive-embedding watermarking approach, the watermark information is directly added to the host image coefficients or image block of the same size. Generally, the coefficients can be obtained from some common transforms, including discrete wavelet transform (DWT), discrete cosine transform (DCT), and Fourier transform. The additive-embedding watermarking embeds the watermark information in the most important frequency domain of image perception, which is similar to the spread spectrum communication idea in the communication system. Cox et al. [8] first designed a digital watermarking method based on the idea of spread spectrum, which embedded watermark data in the important perception transformation coefficient of the host signal by applying the spread spectrum principle. Cox's spread spectrum watermarking algorithm has been considered a representative method. The only deficiency is that the digital watermarking algorithm requires participation of the original image when detecting watermark information, indicating that it is not a blind watermarking algorithm. Subsequently, Cheng et al. [9] proposed an additive watermarking approach, which detects the watermark by

using the generalized Gaussian distribution (GGD). Experiments show that this distribution can effectively control the detection error probability of the watermark. Liu et al. [10] transformed the test signal into the DCT domain. Moreover, a local optimal detection model that is suitable for any host signal was derived by conducting hypothesis testing analysis in this domain. Although these methods [9, 10] can detect watermark information effectively, their parameter estimation process is complex. To address this problem, Kwitt et al. [11] proposed a lightweight blind optimal detector for additive watermarking; it is expected to be useful in resisting watermark desynchronization. Zhang et al. [12] proposed a high-security additive watermarking algorithm by utilizing gyrator transform and matrix decomposition. A key innovation of this algorithm is to adopt an invariant integer wavelet transform, which transforms the image wavelet coefficients into integers, thereby enhancing the performance of the watermarking.

In the quantization-based watermarking method, the main procedure is to embed the watermark data into the host signal by designing a corresponding quantizer. The watermark data are detected according to the quantization interval of the image transform coefficient to extract watermark. Many watermarking methods with quantization scheme have been proposed in recent years. Chen et al. proposed a digital watermarking method with the quantization index modulation (QIM) scheme; it is the most representative quantization watermarking algorithm based on edge information coding [13]. QIM has the characteristics of high capacity, blind detection, and simple implementation. However, QIM watermarking has two main shortcomings. First, it is sensitive to amplitude scaling attacks; second, it is not robust to gain attacks. Researchers proposed corresponding improvement methods to address these problems. In view of the sensitivity of the QIM watermarking method to scaling attacks, researchers mainly improved it in accordance with the quantization step size. To solve the inconsistency of quantization step between the embedded end and the receiver end, as well as the adaptability problem of quantization step, several watermarking methods have been proposed, such as rational dither modulation [14] and adaptive QIM [15]. Furthermore, to enhance the robustness of the QIM watermarking against gain attack, the quantization watermarking [16], sample projection-based quantization [17], P-norm ratio-based quantization [18], angle quantization [19], complex wavelet domain l_1 norm quantization [20], and random projection-based quantization methods [21] have been proposed one after another. These quantization watermarking methods mainly aim to enable the watermark algorithm to obtain invariance to the scaling or gain attacks, and the watermark has strong robustness performance in resisting compression, filtering, and gain attacks. However, the performance of these quantization methods in desynchronization attacks is still inadequate. To further enhance the robustness of quantization watermarking, some researchers have designed corresponding quantization watermarking algorithms by combining the just noticeable distortion (JND) model, image texture complexity, and texture direction features, such as texture direction quantization [22], pair quantization based on extended JND [23, 24], and mixed modulation quantization using singular value decomposition [25]. These quantization methods are combined with image features; they can retain image orientation features and reduce image distortion after watermark embedding. However, these methods are generally vulnerable to noise attack.

The performance of the multiplicative embedding-based watermarking method is similar to that of the quantization watermarking method. The multiplicative watermarking algorithm is usually combined with the human visual perception model, and the embedded strength factor varies with the intensity of the original signal. Moreover, a good trade-off between imperceptibility and robustness can be achieved in the multiplicative watermarking algorithm. Akhaee et al. [26] developed an image watermarking method based on a "scaling" strategy by using the Watson entropy visual masking. The watermark data were embedded into the image block with high entropy to improve the invisibility of the watermark. The algorithm is robust against Gaussian filtering, Gaussian noise, and scaling attacks. However, the entropy value of the image block changes after embedding the watermark; this finding is inconsistent with the entropy of the image block prior to embedding the watermark, thereby reducing the robustness of the watermark against synchronization attacks. Subsequently, Akhaee et al. [27] proposed a scaling-based image watermarking method with contourlet transform in a noisy environment. Experiments demonstrated that the robustness of this watermarking method is good. However, the algorithm has high complexity. Different from the Watson entropy visual masking, Khalilian et al. [28] proposed a multiplicative watermarking algorithm by taking advantage of the visual saliency model. They designed an adaptive embedding factor by combining visual saliency and texture masking. On the one hand, the embedding factor should increase with the distance from the significant region of the image. On the other hand, the watermark embedding strength should be larger in regions with rich texture. This method improves the robustness of the watermarking when against some common image processing attacks. However, the performance of their watermarking still needs to be enhanced in terms of resisting antidesynchronization attack. Moreover, some visual attention-based watermarking methods have been presented in the last few years. For example, Bhowmik et al. [29] embedded high-strength and low-strength watermarks into significant and insignificant regions of vision, respectively, thereby improving the watermarking performance. Hernandez et al. [30] proposed a video watermarking algorithm that took full advantage of the video's spatiotemporal characteristics and minimized the perceived redundancy of the video. Thus, the trade-off between imperceptibility and robustness has been achieved in their method. Yadav et al. [31] developed an image watermarking algorithm by using an adaptive embedded factor, which only used image variance information to compute watermark embedded factor. However, the performance of this method is weak when resisting rotation attacks.

Inspired by literature [28], an image watermarking algorithm was developed based on the visual saliency model in the contourlet transform domain. The main contributions of our work are summarized as follows:

(1) An adaptive watermark embedded strength factor is exploited with a visual saliency model, which can achieve a good trade-off between the robustness and imperceptibility of the watermarking.

(2) The watermark information was embedded into the contourlet coefficients with high energy that can enhance the imperceptibility of watermarking.

The remainder of this paper is organized as follows. The belief concept of the contourlet transform is introduced in Section 2. Section 3 introduces the proposed watermark embedding and detection method. Section 4 shows the experimental results of the proposed watermarking and the comparative results with other watermarking approaches. Finally, the conclusions are summarized in Section 5.

2. Brief Introduction of Contourlet Transform

In 2005, Do et al. [32] proposed a "real" 2D representation of images, that is, the contourlet transform. It captures the segmented conic curves of an image by using different subband scales and frequencies, which have directivity and anisotropy, thereby enabling the contourlet transform to obtain a "sparser" representation. Thus, the contourlet transform has the characteristics of sparse representation at both spatial and directional resolutions. In contourlet transform, multiscale and directional analyses are performed separately. First, the image was transformed into one coarse version plus a set of band-pass images by the Laplacian pyramid (LP) method. Second, each LP band-pass image was decomposed into a number of subbands with 2D quincunx filtering and critical subsampling. Therefore, the contourlet transform can decompose images into multidirectional subbands at multiple scales. Figure 1 illustrates a diagram of the contourlet transform. Furthermore, we have utilized the contourlet toolbox to decompose the "Peppers" image [32]. Figure 2 shows the result of applying the contourlet transform on the "Peppers" image. The figure clearly shows that the contourlet transform can decompose the "Peppers" image into multidirectional subbands.

3. Watermark Embedding and Decoding

In this section, Figure 3 shows the proposed watermark embedding and watermark detection procedure. As shown in Figure 3, we embed the watermark data into the contourlet coefficients with high energy in our implementation. In addition, we utilize the visual saliency model to construct the watermark embedded strength factor; thus, a trade-off between the invisibility and robustness of the watermark can be achieved elegantly with the watermark embedded strength factor. In the watermark detection stage, we model the contourlet coefficients with GGD to detect the watermark due to the non-Gaussian property of the contourlet coefficients.

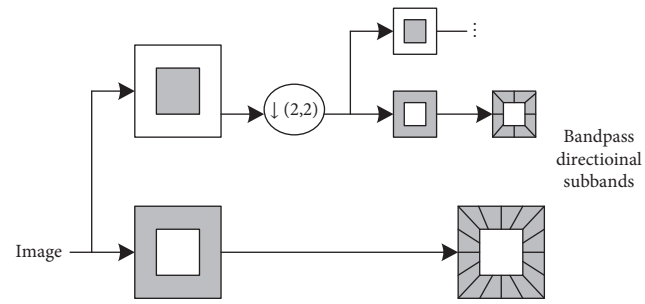


Figure 1: The diagram of the contourlet transform.

3.1. Proposed Watermark Embedding. The procedure of the proposed watermark embedding in Figure 3(a) can be generalized as follows:

Step 1: We segment the host image into $L \times L$ blocks and select the first N image blocks with high energy. The energy is calculated as the sum of the squares of the absolute values of the pixels of the image block. Consequently, the energy of block [28] can be computed by $E = \sum_{m=1}^{M} \sum_{n=1}^{N} \|B(m,n)\|^2$, where $M \times N$ denotes the size of the image block B and (m,n)represents the positions of image block. Generally, a larger value of the energy of image block implies that this image region contains more important coefficients and should be considered a significant image block in comparison with other image blocks. Therefore, to improve the robustness of the watermarking, the watermark is embedded into the image blocks with high energy.

Step 2: Then, we decompose each selected image block by using a two-level contourlet transform. Thus, we embed the watermark data into the coefficients of low-frequency subband. The host contourlet coefficient vectors are denoted as $x = [x_1, x_2, ..., x_n]$, and the watermarked contourlet coefficient vectors are denoted as $y = [y_1, y_2, ..., y_n]$. Suppose that the watermark is $w = [w_1, w_2, ..., w_n]$ with n components and$w_i \in \{-1, 1\}$; the watermark embedding process can be expressed as follows:

$$y = x(1 + \alpha w), \tag{1}$$

where α denotes the embedded strength factor and its value was calculated in Section 3.2.

Step 3: Repeat Step 2 for each image block.

Step 4: Two-level inverse contourlet transform on the watermarked image subband is performed, and it is combined with the image subbands, which are not embedded watermark information, to obtain the whole watermarked image.

3.2. Watermark Embedded Strength Factor. The JND threshold has been widely applied in the field of image processing. Its value is often higher in the image texture region [33]. On the basis of [33], the image texture region

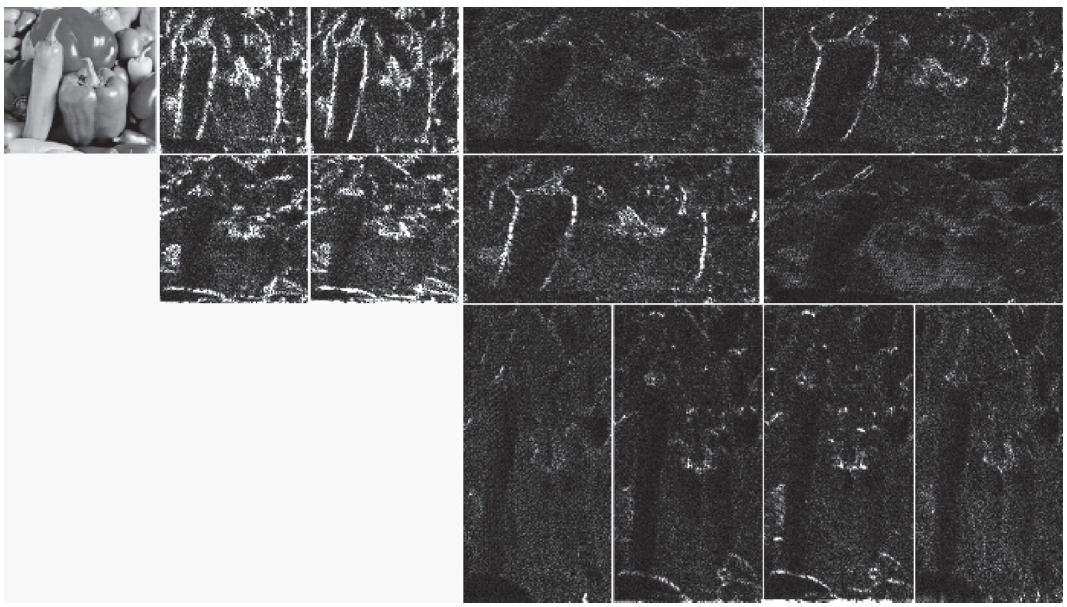


Figure 2: Contourlet transform of the "Peppers" image using two levels [32].

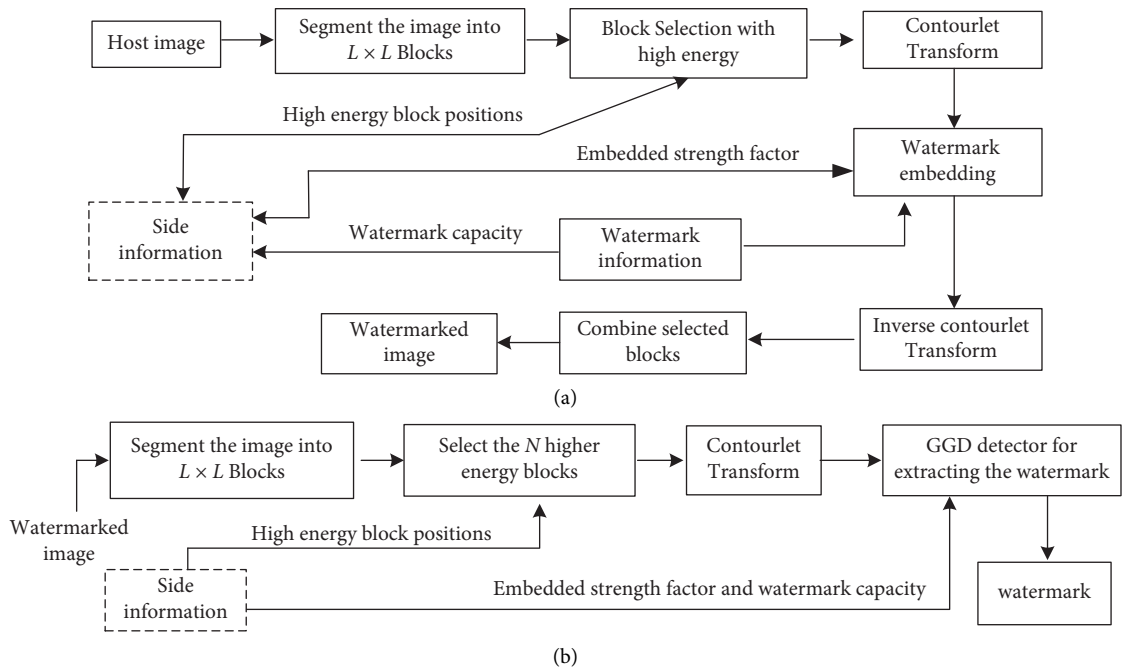


Figure 3: Block diagram of the proposed method. (a) Watermark embedding. (b) Watermark detection.

can hide more information without being perceived by human eyes. Therefore, the embedded strength factor can select a high value. Literature [27] used this fact to develop an image watermarking algorithm. Literature [34] shows that the human visual system tends to focus on the salient areas of an image. As a result, the image salient area hides more distortion, and the embedded strength factor can be enhanced. Therefore, to calculate the embedded strength factor, we take advantage of the texture masking and visual saliency model in this study. The calculation process is summarized as follows.

First, we use a two-level contourlet transform to decompose the host image, which obtains a low-frequency subband, four subbands, and eight subbands from the coarsest scale to the finest scale (Figure 2). Therefore, we compute the energy of directional subband of each block according to the property of the image texture masking. The calculation can be expressed as follows:

$$E_H = \sum_{i=1}^{12} E_{H_i}, \tag{2}$$

where E_{H_i} is the i-th directional subband's energy of each image block. Each image block has 12 directional subbands after the two-level decomposition with contourlet transform. Suppose that $\widetilde{E}_H$ denotes the average energy of twelve image blocks. When increasing the average energy, the watermark embedded strength factor could increase correspondingly. Hence, according to [27], the watermark embedded strength factor of the high-frequency part can be written as follows:

$$\alpha_{HF} = \eta - \rho \cdot e^{-\zeta \cdot \widetilde{E}_H}, \tag{3}$$

where η, ρ, and ζ are set to 1.025, 0.02, and ×, respectively. These parameters are determined by experimental simulation. In the right part of equation (3), for α_{HF}, the parameter η is set to 1.025 for larger image energy $\widetilde{E}_H$, when the exponential function vanishes. This parameter is set to 1.025 in our experiments mainly because it maintains the imperceptibility of the image when used in high image energy when the exponential term disappears. On the contrary, for small image energy, we set parameter ρ to 0.02. Parameter ζ has an important effect in the increasing rate of watermark embedded strength factor; its value is set to × mainly because it can achieve a good trade-off between the robustness and imperceptibility of the watermarking. Therefore, the parameter setting of the watermark embedded strength factor is mainly based on the size of image energy. The main reason is to embed the watermark information while maintaining the imperceptibility of the image watermark.

Then, inspired by [35], we modified the embedded strength factor, which is denoted by α_{HF} by applying visual saliency. Suppose that D_S represents the saliency distance of each block and $D_s^{\max}$ denotes the maximum saliency distance in all image blocks. Therefore, the watermark embedded strength factor can be expressed by $1 + 0.02/D_s^{\max} D_s$. Finally, the modified watermark embedded strength factor can be represented as follows:

$$\begin{aligned} \alpha &= \alpha_{\mathrm{HF}} \times \left(1 + \frac{0.02}{D_s^{\max}} D_s\right) - 1.0 \\ &= \left(\eta - \rho \cdot e^{-\zeta \cdot \widetilde{E}_H}\right) \times \left(1 + \frac{0.02}{D_s^{\max}} D_s\right) - 1.0. \end{aligned} \tag{4}$$

3.3. Watermark Decoding. In this section, we model the contourlet coefficients by the GGD. The probability density function of the GGD model is represented as follows:

$$p_X(x) = A e^{\left(-(\beta|x-\mu|)^c\right)}, \tag{5}$$

where $A = \beta c / 2\Gamma(1/c), B = 1/\sigma(\Gamma(3/c)/\Gamma(1/c))^{1/2}$, and μ, σ denote the mean value and variance, respectively. $\Gamma(\cdot)$ is the gamma function when $\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} \mathrm{d}t, z > 0$, and c denotes the shape parameter. Watermark detection can perform the detection and evaluation of signals. The hypothesis test can be drawn as follows, using the likelihood ratio test (LRT):

$$\begin{cases} H_0\colon \alpha = 0\,(\text{no watermark}) \\ H_1\colon \alpha > 0\,(\text{watermark}) \end{cases}, \tag{6}$$

where H_0 and H_1 are the null and alternative hypotheses. According to the statistical signal processing method, the maximum likelihood ratio can be represented as follows:

$$l(y) = \frac{p(y|H_1)}{p(y|H_0)} \approx \frac{p(y|H_1)}{p(y|0)}. \tag{7}$$

Proofs of (7) are as follows [36]:

$$\int_{-1}^{1} P_y(y_i|w_i)\mathrm{d}w_i = \int_{-1}^{1} \frac{1}{1+\alpha_i w_i} \times P_x\left(\frac{y_i}{1+\alpha_i w_i}\right)\mathrm{d}w_i. \tag{8}$$

Let $t = y_i/(1+\alpha_i w_i)$. Then, the integrand substitutes t for w_i; (8) can be rewritten as follows:

$$\int_{-1}^{1} P_y(y_i|w_i) dw_i = \int_{y_i/(1+\alpha_i)}^{y_i/(1-\alpha_i)} \frac{1}{\lambda_i t} \times P_x(t)\mathrm{d}t. \tag{9}$$

One order Taylor series of $(1/\alpha_i t P_x(t))$ around y_i is expanded as follows:

$$\frac{1}{\alpha_i t} P_x(t) = \frac{1}{\alpha_i y_i} P_x(y_i) + \frac{d}{\mathrm{d}t}\left(\frac{1}{\alpha_i t} P_x(t)\right)\Big|_{t=y_i} (t - y_i). \tag{10}$$

Therefore, (10) is rewritten as follows:

$$\int_{-1}^{1} P_y(y_i|w_i)\mathrm{d}w_i = \int_{y_i/(1+\alpha_i)}^{y_i/(1-\alpha_i)} \frac{1}{\lambda_i y_i} P_x(y_i)\mathrm{d}t + \frac{d}{\mathrm{d}t}\left(\frac{1}{\alpha_i t} P_x(t)\right)\Big|_{t=y_i} \int_{y_i/(1+\alpha_i)}^{y_i/(1-\alpha_i)} (t - y_i)\mathrm{d}t. \tag{11}$$

$\alpha_i \ll 1, \mathrm{d}/\mathrm{d}t\,(1/\alpha_i t P_x(t))|_{t=y_i} \int_{y_i/(1+\alpha_i)}^{y_i/(1-\alpha_i)} (t - y_i)\mathrm{d}t$ is approximately zero in (11). Therefore, equation (11) can be further expressed as follows:

$$\int_{-1}^{1} P_y(y_i|w_i)\mathrm{d}w_i \approx \int_{y_i/(1+\alpha_i)}^{y_i/(1-\alpha_i)} \frac{1}{\alpha_i y_i} P_x(y_i)\mathrm{d}t = \frac{1}{\alpha_i y_i} P_x(y_i) \frac{2\alpha_i y_i}{1-\alpha_i^2} = 2P_x(y_i). \tag{12}$$

Therefore, $P(y|H_0) \approx 1/2^N \prod_{i=1}^{N} (2P_x(y_i)) = P(y|0)$.

On the basis of the analysis, we can rewrite the maximum likelihood ratio by combining watermark embedding (1) and the GGD model as follows:

$$l(y) = \ln \frac{P(y|H_1)}{P(y|H_0)} \approx \ln \frac{P(y|H_1)}{P(y|0)} = \ln \frac{\prod_{i=1}^{N}\left(A/1 + \alpha_i w_i \exp\left(-|\beta_i y_i/1 + \alpha_i w_i|^c\right)\right)}{\prod_{i=1}^{N}\left(A \exp\left(-|\beta_i y_i|^c\right)\right)}. \tag{13}$$

Furthermore, equation (13) can be simply represented as follows:

$$l(y) = \sum_{i=1}^{N} \left(-\alpha_i w_i + c|\beta y|^c \alpha_i w_i\right). \tag{14}$$

Thus, we can write the watermark detector as follows:

$$T(y) = \frac{\partial l(y)}{\partial \alpha_i} = \sum_{i=1}^{N} \left(-w_i + c|\beta y_i|^c w_i\right). \tag{15}$$

Next, we can compute the watermark detection threshold. The Gaussian distribution characteristic of the watermark detector under the null hypothesis condition and its mean is zero. As a result, we can calculate the watermark detection threshold as follows:

$$\tau = \sigma_T Q^{-1}\left(P_f\right), \tag{16}$$

where τ denotes the watermark detection threshold, $\sigma_T = \sqrt{\sum_{i=1}^{N} c w_i^2}$ represents the variance, and $Q(x) = (1/\sqrt{2\pi}) \int_x^{+\infty} \exp(-t^2/2)\mathrm{d}t$ denotes the right-tail probability of the Gaussian distribution. $P_f = P(T(y) > \tau|H_0) = Q(\tau/\sigma_T)$ represents the false alarm probability.

Generally, false alarm is generated due to the existence of the watermark information detected in the unwatermarked image. A missed alarm is the phenomenon in which the watermark detector does not detect the watermark information in the watermarked image. Therefore, the receiver operating characteristic (ROC) curve of the watermarking can be derived as follows.

Suppose P_0 denotes the detection probability of watermark. Hence, $(1 - P_0)$ can represent the missed alarm probability. According to the statistical hypotheses and the central limit theorem, the mean and variance of the distribution of the host image and watermarked image can be estimated; they are denoted as μ_{T_0}, μ_{T_1} and $\sigma_{T_0}, \sigma_{T_1}$, respectively. As a result, P_0 can be written as follows:

$$P_d = Q\left(\frac{\sigma_{T_0} Q^{-1}\left(P_f\right) + \mu_{T_0} - \mu_{T_1}}{\sigma_{T_1}}\right), \tag{17}$$

where $\sigma_{T_0} \approx \sigma_{T_1}$, $\mu_{T_0} = 0$, and $\mu_{T_1} = (1/N \sum_{i=1}^{N} \alpha_i) \cdot \sigma_{T_0}^2$. Let $\mathrm{SNR} = \mu_{T_1}/\sigma_{T_1} = (1/N \sum_{i=1}^{N} \alpha_i) \cdot \sigma_{T_0}$.

Finally, the ROC relationship can be defined as follows:

$$P_d = Q\left(Q^{-1}\left(P_f\right) - \mathrm{SNR}\right). \tag{18}$$

4. Experimental Results

In this regard, to verify the effectiveness of the proposed watermarking method, several experiments have been performed, including the imperceptibility, robustness, and performance of watermark detection. We have compared the proposed watermarking with other related watermarking approaches. All experiments have been performed on a PC with 4.0 GHz Intel Core i7 CPU and 16 G RAM. The simulation software was MATLAB R2018a that ran in 64-bit Windows 10. In summary, the simulation settings are provided in Table 1.

4.1. Imperceptibility Test. We have tested eight standard images, which include Lena, Barbara, Bridge, Boat, Elaine, Mandrill, Peppers, and Man, to demonstrate the invisibility of the proposed method; the size of each standard image is 512×512. In our implementation, a two-level contourlet transform has been applied to decompose each image block. The filters are set to "Pivka." Figure 4 only shows the host images and their watermarked version made by applying our method with 16×16 blocks and a 512-bit watermark capacity due to the limited space. Figure 4 shows that the imperceptibility of our method is satisfied. Therefore, finding the difference between the original image and their watermarked version is difficult.

In addition, the embedded strength factor can be adapted and adjusted according to the watermark capacity to further enhance the performance of the proposed method. The relationship between the embedded parameter and watermarked image quality is developed through experiments, and the results are shown in Figures 5 and 6 . The performance is mainly measured by peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [37]. As shown in Figures 5 and 6, when the watermark embedded strength factor increases, the values of PSNR and SSIM decrease. The range of embedded strength factor can be set within 0.005 to 0.025 to balance the imperceptibility, robustness, and watermark capacity of watermarking.

Table 1: Experimental parameter settings.

Parameter name	Configuration
Experimental platform	Window 10, MATLAB R2018a
Test images	Lena, Barbara, Bridge, Boat, Elaine, Mandrill, Peppers, and Man
Image size	512×512
Wavelet filters of contourlet transform	Pivka
Watermark length (bits)	512
Decomposition level	Two-level
Performance evaluation	PSNR, SSIM, and bit error rate (BER)

Figure 4: Original and watermarked versions: Lena, Barbara, Bridge, Boat, Elaine, Mandrill, Peppers, and Man. For each image, the left and right parts denote the original image and the watermarked image, respectively.

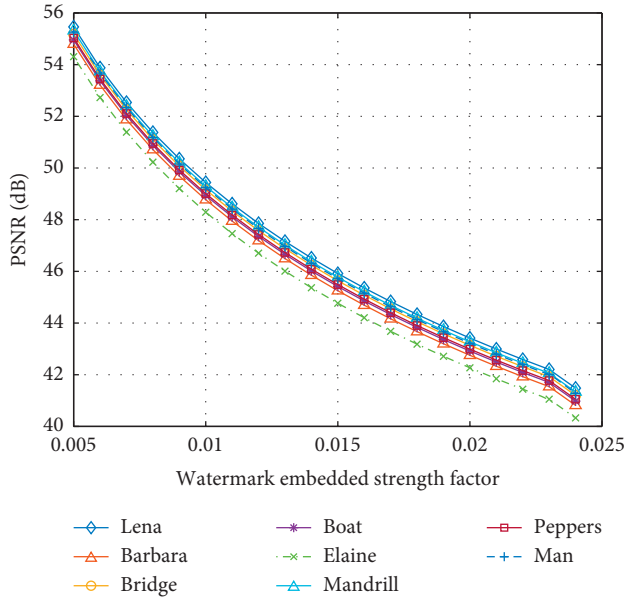


Figure 5: PSNR versus watermark embedded strength factor with watermark capacity 2048.

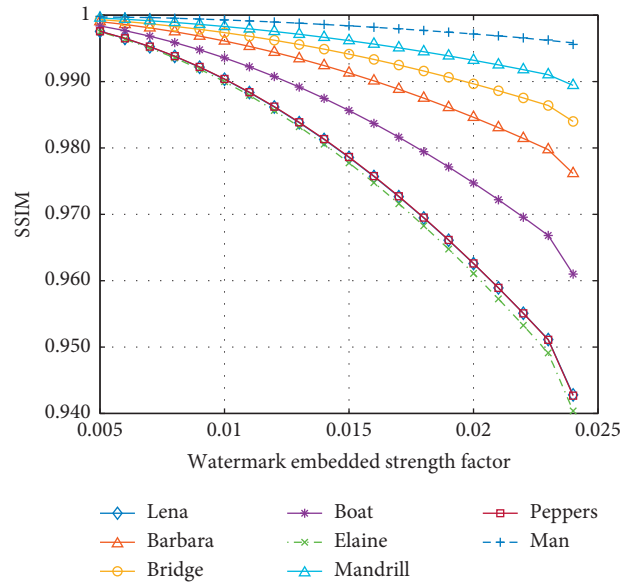


Figure 6: SSIM versus watermark embedded strength factor with watermark capacity 2048.

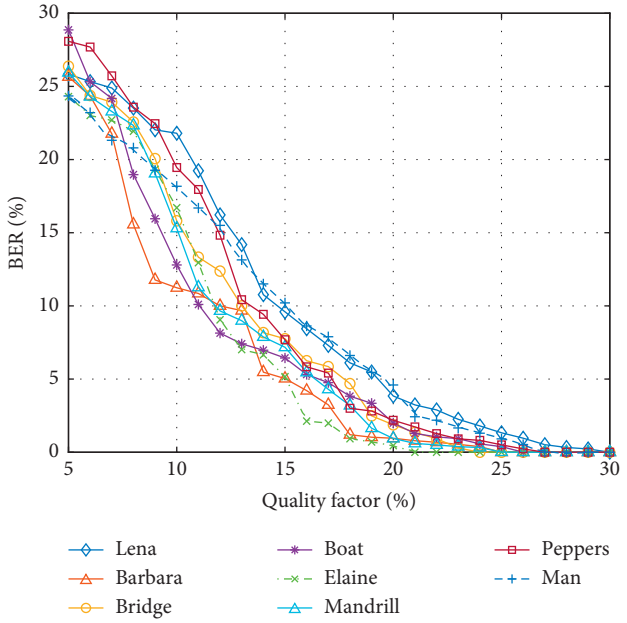


Figure 7: BER (%) results under JPEG compression attack.

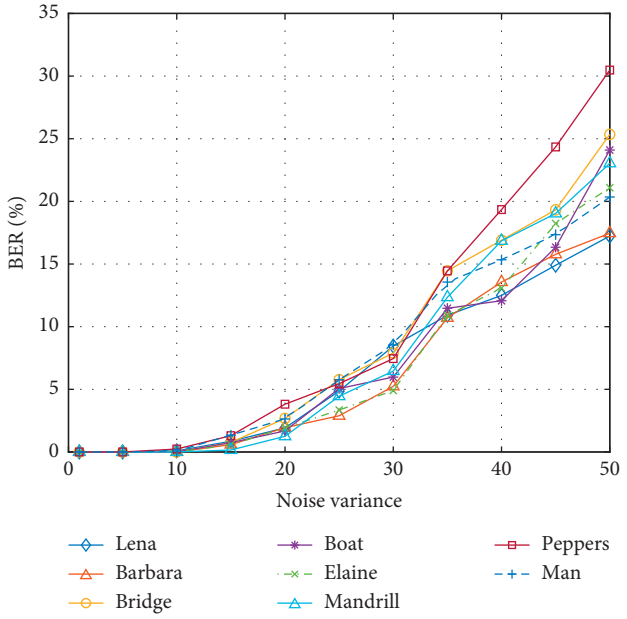


Figure 8: BER (%) results under Gaussian noise attack.

4.2. Robustness Test. In this section, to assess the robustness of the proposed watermarking, several experiments have been performed in common image processing and some geometric attacks. These attacks include additive whiten Gaussian noise, salt and pepper noise, median filtering, rotation, cropping, flipping, scaling, JPEG compression, and Gaussian filtering attack. Furthermore, to evaluate the effectiveness of our watermarking method, we have compared it with other related watermarking approaches, which include the methods in [27, 31] and [38]. Moreover, the robustness performance is measured through the bit error rate (BER).

We have performed two common experiments under JPEG compression and Gaussian noise attack. The result is shown in Figures 7 and 8. In this work, the watermark capacity is 512 bits. Figure 7 shows that our method has satisfying robustness on JPEG compression attack. Similarly, Figure 8 shows that our watermarking method has good robustness against Gaussian noise attacks. Moreover, Tables 2 and 3 show the comparison of the performance of our method with other methods under common image processing, geometric, and combined attacks. All watermarking methods, for the purpose of comparison, use the same watermark capacity. The watermark capacity of all methods is also 512 bits in Tables 2 and 3.

Table 2: BER (%) results of various watermarking methods under common attacks.

Image	Methods	Gaus.noi. 20	Salt.Pep.0.05	JPEG 20%	Gau.filt. 3 x 3	Med.filt. 3 x 3
Lena	Method [27]	3.02	3.43	5.63	—	—
	Method [31]	2.41	13.78	9.24	1.19	1.56
	Method [38]	6.29	—	20.76	1.83	4.95
	Proposed	2.25	11.24	4.83	2.86	3.72
Barbara	Method [27]	2.56	3.95	2.14	—	—
	Method [31]	1.89	10.26	8.30	0.82	1.65
	Method [38]	7.34	—	18.85	1.49	5.36
	Proposed	2.08	5.23	2.35	1.73	2.20
Bridge	Method [27]	2.94	9.44	6.79	—	—
	Method [31]	3.81	12.50	8.96	2.49	2.16
	Method [38]	6.93	—	19.50	1.68	5.18
	Proposed	2.59	9.69	2.61	3.80	5.43
Boat	Method [27]	3.12	10.17	5.29	—	—
	Method [31]	4.93	14.98	7.42	1.85	3.26
	Method [38]	6.68	—	18.20	1.56	5.64
	Proposed	2.34	12.05	2.58	2.23	6.69
Elaine	Method [27]	2.59	13.30	3.68	—	—
	Method [31]	2.08	15.19	5.89	2.79	4.92
	Method [38]	6.34	—	17.35	2.76	5.54
	Proposed	2.13	14.58	0.87	2.55	6.80
Mandrill	Method [27]	1.78	4.87	3.22	—	—
	Method [31]	2.34	10.48	6.45	1.38	2.64
	Method [38]	5.95	—	18.29	1.82	5.97
	Proposed	1.82	7.67	1.33	2.19	4.50
Peppers	Method [27]	5.70	12.79	6.86	—	—
	Method [31]	4.48	15.24	5.53	2.64	1.75
	Method [38]	8.60	—	19.42	1.76	6.08
	Proposed	4.27	12.87	2.73	2.39	1.44
Man	Method [27]	4.39	13.02	8.46	—	—
	Method [31]	3.66	10.82	11.25	4.89	6.88
	Method [38]	7.08	—	22.07	2.10	5.83
	Proposed	2.45	11.77	4.94	4.75	5.32

Table 3: BER (%) results of various watermarking methods under geometric attacks.

Image	Methods	Rot.10°	Scal. 0.75	Crop.50%	Rot.5 + Scal .5
Lena	Method [27]	10.22	27.34	29.45	30.24
	Method [31]	12.76	22.08	26.17	29.46
	Method [38]	17.49	20.89	22.32	34.58
	Proposed	9.68	19.97	20.80	21.73
Barbara	Method [27]	9.34	32.29	30.13	28.71
	Method [31]	11.92	27.44	26.49	25.82
	Method [38]	18.24	24.12	25.69	37.61
	Proposed	6.60	26.80	16.32	20.94
Bridge	Method [27]	7.38	9.65	32.74	31.18
	Method [31]	17.82	19.54	25.58	24.19
	Method [38]	19.23	23.47	27.48	39.69
	Proposed	9.56	16.89	24.22	23.71
Boat	Method [27]	9.51	18.36	28.67	28.40
	Method [31]	8.84	24.07	20.34	29.33
	Method [38]	15.02	28.73	26.85	38.87
	Proposed	5.97	25.99	22.76	22.25
Elaine	Method [27]	11.35	21.43	27.50	27.42
	Method [31]	16.24	28.30	19.38	25.78
	Method [38]	20.43	35.66	21.80	35.93
	Proposed	10.33	30.12	16.14	25.39
Mandrill	Method [27]	8.98	12.46	24.55	27.16

Table 3: Continued.

Image	Methods	Rot.10°	Scal. 0.75	Crop.50%	Rot.5 + Scal .5
	Method [31]	7.22	20.79	20.18	24.23
	Method [38]	22.06	29.78	24.57	36.94
	Proposed	6.50	17.43	12.84	19.59
Peppers	Method [27]	12.45	23.78	29.43	29.68
	Method [31]	13.39	18.76	21.80	30.49
	Method [38]	18.87	27.26	23.18	38.68
	Proposed	6.07	19.59	13.67	23.67
Man	Method [27]	9.94	14.41	27.86	27.45
	Method [31]	10.18	21.78	19.44	22.97
	Method [38]	21.50	26.15	23.07	36.34
	Proposed	7.83	20.42	10.29	21.80

Table 2 shows the results of the simulation experiments under common image processing attacks, which cover Gaussian noise with noise variance 20, Salt and Pepper noise with noise variance of 0.05, JPEG compression with a quality factor of 20%, Gaussian filtering with the windows of size 3×3, and median filtering with the windows of size 5×5. Table 3 shows the results of the simulation experiments under geometric attacks, including the rotation attack with 10° angle, amplitude scaling attack with factor 0.75, cropping with factor 50%, combination attack with rotation of 5° angle, and scaling with a factor of 0.50.

As shown in Tables 2 and 3, the proposed watermarking method has slightly better performance than the image watermarking methods. This finding is mainly due to the application of the following factors. First, we embed the watermark information into the image blocks with high energy in the contourlet transform domain. Second, the watermark embedded strength factor was constructed by taking advantage of the visual saliency model and texture masking. Thus, embedding the watermark can be adapted. As such, a good trade-off between the invisibility and robustness of the watermark can be achieved. Finally, the watermark detection performance can be improved by the GGD model.

However, the proposed watermark detector relies on partial original image feature information, such as positions of image blocks; thus, the proposed algorithm becomes semiblind. In the subsequent work, we will design a blind watermarking method.

4.3. Performance of Watermark Detection. The GGD is used to model the contourlet coefficients to further demonstrate the detection performance, and the ROC is utilized to measure the performance of watermark detection according to equation (18) of Section 3.3. Figure 9 shows the results and indicates that the detection performance of our method is satisfied. The main reason is that the contourlet coefficient distribution is highly nonlinear, and the GGD fits the contourlet coefficient effectively.

However, the proposed watermarking method performs weakly when resisting other attacks, including combinational attack amplitude scaling and JPEG compression, Salt and Pepper and Gaussian noise, and global affine transformation and histogram equalization attack. These problems will be addressed by developing some matrix decomposition-based watermarking methods or deep learning-based watermarking algorithms in our future work.

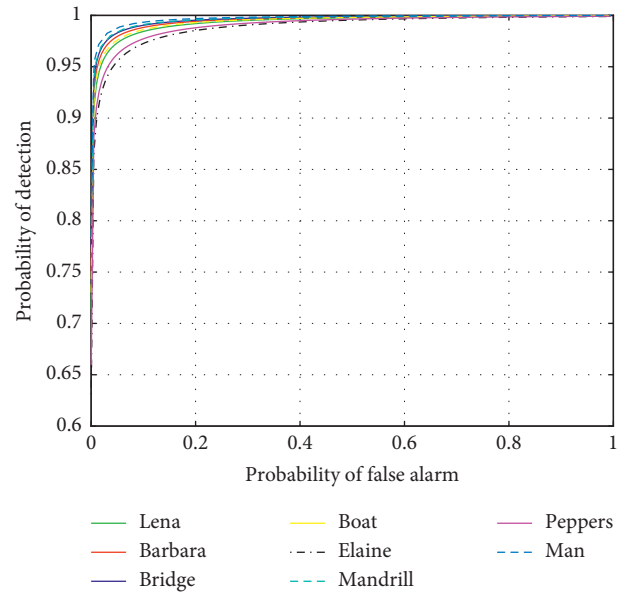


Figure 9: Performance of watermark detection for different images.

5. Conclusion

We have developed an image watermarking algorithm by using the visual saliency model in the contourlet domain. In watermark embedding, high-energy image blocks are selected for the watermark embedding space, and the watermark embedded strength factor is exploited by taking advantage of texture masking and visual salience. The watermark can be embedded into the contourlet coefficients adaptively by using this strategy. For watermark decoding, the GGD model is used to describe the contourlet coefficients, and the ROC has been derived by applying the statistic signal processing method. Finally, we have performed several experiments to demonstrate the proposed method. Simulation results show that our watermarking

method has satisfied imperceptibility and robustness. In the future work, a novel watermark detection approach will be designed using the deep learning or generative adversarial network method.

Data Availability

Eight standard grayscale images Lena, Barbara, Bridge, Boat, Elaine, Mandrill, Peppers, and Man are used as host images in the simulations, which are shown in Figure 4 in this paper. The results in this paper are entirely theoretical and analytical. The main steps of the demonstrations for each result are clearly reported in the text and the paper is fully consistent without the support of any additional data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangxi (no. 20192BAB207013). The authors would like to thank professor M.N. Do for providing the CODE to perform the contourlet transform.

References

[1] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2131–2153, 2018.

[2] H. Fang, W. Zhang, H. Zhou, H. Cui, and N. Yu, "Screen-shooting resilient watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1403–1418, 2019.

[3] S. Li and X. Zhang, "Toward construction-based data hiding: from secrets to fingerprint images," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1482–1497, 2019.

[4] Y. Huang, B. Niu, H. Guan, and S. Zhang, "Enhancing image watermarking with adaptive embedding parameter and PSNR guarantee," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2447–2460, 2019.

[5] D. Rajani and P. R. Kumar, "An optimized blind watermarking scheme based on principal component analysis in redundant discrete wavelet domain," *Signal Processing*, vol. 172, Article ID 107556, 2020.

[6] B. Chen, Y. Wu, G. Coatrieux, X. Chen, and Y. Zheng, "JSNet: a simulation network of JPEG lossy compression and restoration for robust image watermarking against JPEG attack," *Computer Vision and Image Understanding*, vol. 197-198, Article ID 103015, 2020.

[7] B. Xiao, J. Luo, X. Bi, W. Li, and B. Chen, "Fractional discrete Tchebyshev moments and their applications in image encryption and watermarking," *Information Sciences*, vol. 516, pp. 545–559, 2020.

[8] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[9] Q. Cheng and T. S. Huang, "An additive approach to transform-domain information hiding and optimum detection structure," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 273–284, 2001.

[10] W. Liu, L. Dong, and W. Zeng, "Optimum detection for spread-spectrum watermarking that employs self-masking," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 645–654, 2007.

[11] R. Kwitt, P. Meerwald, and A. Uhl, "Lightweight detection of additive watermarking in the DWT-domain," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 474–484, 2011.

[12] L. Zhang and D. Wei, "Image watermarking based on matrix decomposition and gyrator transform in invariant integer wavelet domain," *Signal Processing*, vol. 169, Article ID 107421, 2020.

[13] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[14] F. Perez-Gonzalez, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: a high-rate data-hiding method invariant to gain attacks," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3960–3975, 2005.

[15] Q. Li and I. J. Cox, "Using perceptual models to improve fidelity and provide resistance to valumetric scaling for quantization index modulation watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, pp. 127–139, 2007.

[16] N. K. Kalantari and S. M. Ahadi, "A logarithmic quantization index modulation for perceptually better data hiding," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1504–1517, 2010.

[17] M. A. Akhaee, S. M. E. Sahraeian, and C. Jin, "Blind image watermarking using a sample projection approach," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 883–893, 2011.

[18] M. Zareian and H. R. Tohidypour, "A novel gain invariant quantization-based watermarking approach," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1804–1813, 2014.

[19] N. Cai, N. Zhu, S. Weng, and B. Wing-Kuen Ling, "Difference angle quantization index modulation scheme for image watermarking," *Signal Processing: Image Communication*, vol. 34, pp. 52–60, 2015.

[20] J. Liu, Y. Xu, S. Wang, and C. Zhu, "Complex wavelet-domain image watermarking algorithm using $$L_1$$ L 1 -norm function-based quantization," *Circuits, Systems, and Signal Processing*, vol. 37, no. 3, pp. 1268–1286, 2018.

[21] M. Sadeghi, R. Toosi, and M. A. Akhaee, "Blind gain invariant image watermarking using random projection approach," *Signal Processing*, vol. 163, pp. 213–224, 2019.

[22] H. Fang, H. Zhou, Z. Ma, W. Zhang, and N. Yu, "A robust image watermarking scheme in DCT domain based on adaptive texture direction quantization," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8075–8089, 2019.

[23] J. Wu, L. Li, W. Dong, G. Shi, W. Lin, and C.-C. J. Kuo, "Enhanced just noticeable difference model for images with pattern complexity," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2682–2693, 2017.

[24] W. Wan, J. Wang, J. Li et al., "Pattern complexity-based JND estimation for quantization watermarking," *Pattern Recognition Letters*, vol. 130, pp. 157–164, 2020.

[25] H.-T. Hu, L.-Y. Hsu, and H.-H. Chou, "An improved SVD-based blind color image watermarking algorithm with mixed modulation incorporated," *Information Sciences*, vol. 519, pp. 161–182, 2020.

[26] M. A. Akhaee, S. M. E. Sahraeian, B. Sankur, and F. Marvasti, "Robust scaling-based image watermarking using maximum-likelihood decoder with optimum strength factor," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 822–833, 2009.

[27] M. A. Akhaee, S. M. E. Sahraeian, and F. Marvasti, "Contourlet-based image watermarking using optimum detector in a noisy environment," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 967–980, 2010.

[28] H. Khalilian and I. V. Bajic, "Video watermarking with empirical PCA-based decoding," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4825–4840, 2013.

[29] D. Bhowmik, M. Oakes, and C. Abhayaratne, "Visual attention-based image watermarking," *IEEE Access*, vol. 4, pp. 8002–8018, 2016.

[30] A. Cedillo-Hernandez, M. Cedillo-Hernandez, M. Nakano Miyatake, and H. Perez Meana, "A spatiotemporal saliency-modulated JND profile applied to video watermarking," *Journal of Visual Communication and Image Representation*, vol. 52, pp. 106–117, 2018.

[31] N. Yadav and K. Singh, "Robust image-adaptive watermarking using an adjustable dynamic strength factor," *Signal, Image and Video Processing*, vol. 9, no. 7, pp. 1531–1542, 2015.

[32] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[33] X. Yang, W. Lin, Z. Liu, E. Ongg, and S. Yao, "Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 745–752, 2005.

[34] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[35] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1304–1318, 2004.

[36] J. Wang, G. Liu, Y. Dai, J. Sun, Z. Wang, and S. Lian, "Locally optimum detection for barni's multiplicative watermarking in DWT domain," *Signal Processing*, vol. 88, no. 1, pp. 117–130, 2008.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[38] F. Ernawan and M. N. Kabir, "A robust image watermarking technique with an optimal dct-psychovisual threshold," *IEEE Access*, vol. 6, pp. 20464–20480, 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8661083, 10 pages
https://doi.org/10.1155/2021/8661083

Research Article

Dual-Tree Complex Wavelet Transform-Based Direction Correlation for Face Forgery Detection

Shichao Gao, Ming Xia, and Gaobo Yang

College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

Correspondence should be addressed to Gaobo Yang; yanggaobo@hnu.edu.cn

Received 17 June 2021; Accepted 28 August 2021; Published 29 September 2021

Academic Editor: Beijing Chen

With the rapid development of face synthesis techniques, things are going from bad to worse as high-quality fake face images are unnoticeable by human eyes, which has brought serious public confidence and security problems. Thus, effective detection of face image forgeries is in urgent need. We observe that some subtle artificial artifacts in spatial domain can be easily recognized in transformation domain, and most facial features have an inherent directional correlation, and generative models would ruffle this kind of distribution pattern. Inspired by this, we propose a two-stream dual-tree complex wavelet-based face forgery network (DCWNet) to expose face image forgeries. Specifically, dual-tree complex wavelet transform is exploited to obtain six directional features ($\pm 75^{\circ}$, $\pm 45^{\circ}$, $\pm 15^{\circ}$) of different frequency components from original images, and a direction correlation extraction (DCE) block is presented to capture the direction correlation. Then, the direction pattern-aware clues and the original image are taken as two complementary network inputs. We also explore how specific frequency components work in face forgery detection and propose a new multiscale channel attention mechanism for features fusion. The experimental results prove that the proposed DCWNet outperforms the state-of-the-art methods in open datasets such as FaceForensics++ and achieves high robustness against lossy image compression.

1. Introduction

In recent years, various deep learning technologies such as FaceSwap [1], Deepfake [2], and Face2Face [3] have presented for facial image manipulations which change the attributes of face images. Besides, some generative adversarial network- (GAN-) [4] based works can even create fake faces without target images. As shown in Figure 1, these artificial products seem scarily real that it is difficult to find fake face images from real ones by naked eyes. This brings great threats to public information security. For example, these techniques might be used to produce pornographic videos or scams. Thus, how to distinguish real and fake face images has attracted more attentions in the community of image content security.

Many works have been proposed to use artificial intelligence (AI) to fight with AI, namely, using deep learning methods to differentiate real images from fake ones. Among them, some sophisticated convolutional neural network (CNN) structures [7–10] were proposed or they were combined with hand-crafted features [11–13] to achieve better performance. However, what makes CNNs be much more perceptive than humans? Some researchers tried to provide some explanations to this from frequency domain [14–17]. Nevertheless, the conventional frequency-domain transformation methods, such as FFT [18] and DCT [19], do not keep well the spatial information of the original image. That is, the images with distinct visual contents might have the same spectral amplitudes. Thus, vanilla CNN structures might be inapplicable. In [16], the frequency features extracted by frequency-aware decomposition (FAD) and local frequency statistics (LFS) were combined with sliding window DCT (SWDCT) to preserve the spatial structure of the image to some extent.

(a)

(b)

Figure 1: Real and fake face images. (a) Real face images. (b) From left to right, fake face images generated by Deepfake, FaceSwap, Face2Face, Neural Textures [5], and StyleGAN [6].

Wavelet transform has been widely used in various image applications such as denoising, compression, and texture classification. Compared with fast Fourier transform (FFT) and other transforms, wavelet transform preserves well multiscale image spatial structure, which makes it to be known as textual microscope. This motivates us that wavelet transform might be compatible with CNN for face forgery detection tasks.

The direction-related details such as facial contour, wrinkles, and light-shadow cross lines are intuitive yet effective for face image forensics. Dual-tree complex wavelet transformation (DTCWT) was proposed to overcome the translation sensitivity, which has higher directional selectivity than traditional wavelets [20]. We exploit the DTCWT to reveal the correlation between facial features in different directions. Moreover, wavelet transformation decomposes the original image into multiple scales. Among them, the low-level features provide richer details, whereas the high-level features provide more semantics information. It is well-known that both low-frequency and high-frequency information is useful for image classification tasks [21]. Is it the same for face image forensics? If so, what is the role each component plays in face forgery detection and how can we fuse multiscale features?

In this work, we propose a novel two-stream deep network for face image forgery detection. One stream exploits DTCWT to learn multiscale directional features. In Figure 2, we show the results of the two-stage DTCWT on the original face image. Each stage contains six different directional features. The other stream takes the original image as input which provides low-frequency and pixel-level information for the network. Moreover, to fully exploit different frequency components, we propose a multiscale channel attention (MSCA) mechanism to fuse multiscale frequency-domain features from direction correlation extraction (DCE) block. The main works and contributions are three-fold: (1) DTCWT is combined with CNN for face image forensics. It addresses face forgery detection from a new perspective, in which a novel DCE block is proposed to extract the correlation features. (2) A MSCA mechanism is proposed to improve feature fusion efficiency. (3) We demonstrate that face image forensics is different from image classification, and the influence of various frequency components on face forgery detection is well studied.

The remainder of this paper is organized as follows: Section 2 summarizes the related works. Section 3 presents the proposed DCWNet. Section 4 reports the experimental results, and conclusion is given in Section 5.

2. Related Work

The recent AI-enabled face forgeries can generate fake face images without any noticeable artificial artifacts. CNNs have achieved great success compared with the earlier works which exploit hand-crafted features [22, 23]. Many face forgery detection works have been presented for better accuracy or interpretability.

2.1. Pixel-Level Forgery Detection. The most widely used method is to input the original images into CNN, either in RGB or HSV color space. In [24], Dang et al. proposed a CNN-based approach integrated with an attention mechanism to improve the feature maps. Inspired by image steganalysis, Nataraj et al. proposed to combine pixel cooccurrence matrices with CNN for face forgery detection [13]. The model was trained on the dataset generated by CycleGAN [25] and had an extra test on face images generated by different GAN structures (StarGAN [26]). The experimental results showed that their work has good generalization capability. Afchar et al. proposed to use two existing networks, namely, Meso-4 and Meso-Inception-4, to exploit the mesoscopic properties of the images [27]. They achieved an accuracy of the ACC up to 98.4%. Guo et al. proposed an adaptive manipulation trace extraction network (AMTEN) [14]. It predicts manipulation traces by an adaptive convolution layer, which are also reused to

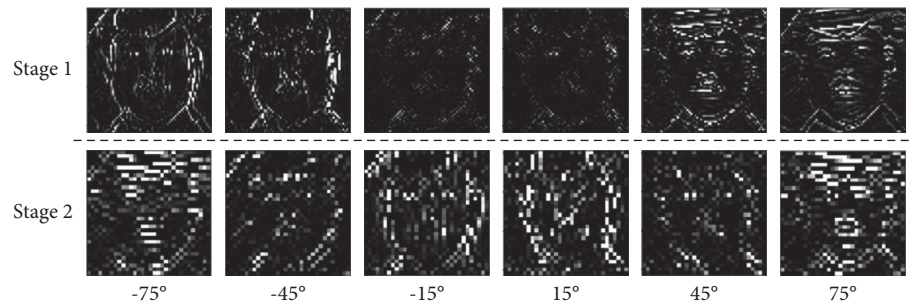


Figure 2: Results of two-stage dual-tree complex wavelet transform. Stage 1 is the result of the first wavelet transform on the original image and stage 2 is the second. Each stage contains six different direction features (±75°, ±45°, ±15°).

maximize manipulation artifacts. For various face forgeries, AMTEN achieved an average accuracy of 98.52%. Nirkin et al. thought that Deepfake methods produce discrepancies between faces and their context. Their approach involved two networks and used the recognition signals from these two networks to detect such discrepancies [28]. In addition, recurrent neural network (RNN) was also exploited by considering face images with temporal properties [29–31]. Some other works exploited visual artifacts such as 3D head poses incoherence for better explanations [32–34]. Chen et al. proposed an improved Xception model for GAN-generated faces [35]. They removed the four residual blocks of Xception to avoid the overfitting problem, and the dilated convolution is used to replace the common convolution layer. The proposed model performed well on their locally GAN-based generated face (LGGF) dataset.

2.2. Frequency-Based Forgery Detection. Image transformation refers to transforming an original image from the spatial domain to other domains such as frequency. The common image transformations include discrete cosine transform [19], fast Fourier transform [18], and wavelet transform [36], which are widely used in various image applications such as edge enhancement, image smoothing, and texture analysis.

In recent years, transform domain processing has been introduced into face forensics. Qian et al. proposed a novel F^3-Net [16], which exploits frequency-aware decomposed image components and local frequency statistics. F^3-Net performs well on the FaceForensics++ dataset, especially for low-quality images. Liu et al. found that the phase spectrum is more sensitive to the up-sample operation than the amplitude spectrum and proposed to expose the up-sample traces by exploiting the phase spectrum [37]. Gong et al. exploited 2D DCT for each RGB channel of the original image and then used AutoGAN [38] to synthesize GAN artifacts in any image without pretrained model [15].

2.3. Attention Mechanism. The attention mechanism generates a set of weighting coefficients, which are often adaptively weighted to strengthen interested regions and suppress irrelevant background regions. There are three common attention mechanisms. The first one is the channel attention. In SENet [39], global average pooling is used to obtain the mean value of the channels as the input of the following fully connected layer. In ECANet [40], 1×1 convolutions replace the fully connected layer to pay more attention to the relationship between adjacent channels. The second one is the spatial attention mechanism which reinforces local areas in each channel. One of the most outstanding works is CBAM [41]. The third one is the self-attention [42], which models the global context through the self-attention mechanism and effectively captures long-distance feature dependencies.

3. Our Approach

3.1. Direction Correlation Extraction Block. Face images have rich directional information such as wrinkles, facial contours, and light and shadow boundaries. They have distribution patterns under specific facial movements. That is, there are spatial correlations among them. The AI-generated fake faces might have weak relevances. This can be used as the clue for face forensics, which motivates us to design a DCE block to expose this, as shown in Figure 3. Conv means convolution operation, BN represents batch normalization, and ReLu is the activation function.

Directional correlation contains two parts: (1) local correlation inside each direction map. (2) Correlation among different direction maps. For local features, we applied 3×3 convolutions on each type of directional feature maps, respectively.

$$f_{n,i} = I_n * \left[C_{1,i}, C_{2,i}, \ldots, C_{m,i}\right], \quad n = \{1, 2, \ldots, m\}, \; i = \{1, 2 \ldots k\}, \tag{1}$$

where I_n are the face feature maps of the nth direction obtained by DTCWT; C_i denotes the convolution kernels; and $f_{n,i}$ represents the features extracted with C_i in direction n. In this work, both m and k are set to 6. For each input, we obtain the feature maps of six channels, which are concatenated to obtain F_{local}.

$$F_{\text{local}} = \text{concat}\left(\left[f_{1,1} \ldots f_{1,k}\right], \ldots \left[f_{m,1} \ldots f_{m,k}\right]\right). \tag{2}$$

The SE block [39] is an existing channel attention method. The input multichannel feature maps are taken into the global average pooling to obtain the weight array. Considering the characteristics of the wavelet coefficients,

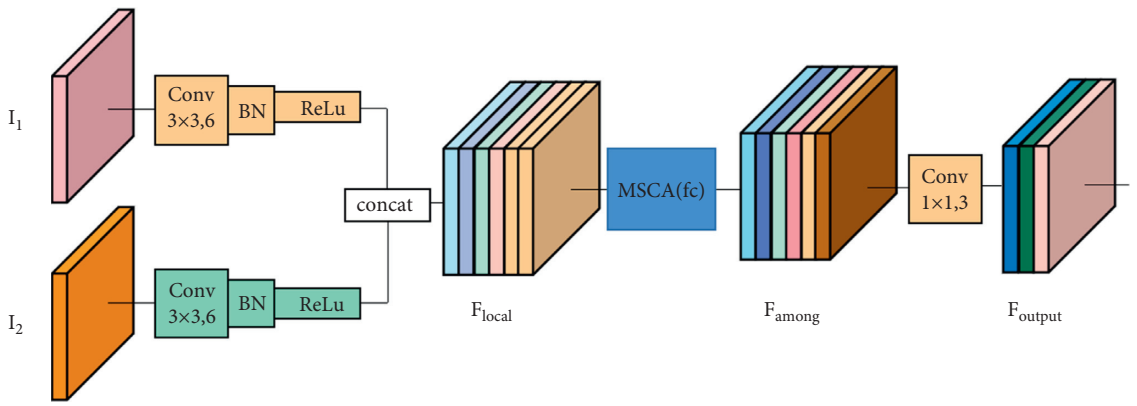


Figure 3: Directional correlation extraction block.

MSCA is adopted to extract features among directional channels (we will demonstrate MSCA in Subsection 3.2.2).

$$F_{\text{ouput}} = c_{1\times1} * \text{MSCA}_{fc}(F_{\text{local}}). \tag{3}$$

Note that the original 1×1 convolution in MSCA is replaced with a fully connected layer (MSCA_{fc}). The reason behind this is that the 1×1 convolution pays more attention to the correlation among adjacent channels. In contrast, the fully connected layer is a point-to-multipoint relationship, which comprehensively describes the relationship between interval channels. Besides extracting the correlation between channels, the MSCA_{fc} block also reduces redundant information in local features. Thus, DCE focuses on directional components. Then, we apply a 1×1 convolution operation $C_1 \times_1$ to further exploit interchannel correlation. In this manner, the same directional features share the convolution kernel in wavelet transform.

3.2. Attention-Based Multiscale Feature Fusion. In essence, multiscale wavelet transform is the stepped dichotomization of the original image frequency. How each frequency component works for face forensics task and how to effectively fuse the directional features obtained from the multiscale wavelet transform? Thus, we proposed a new attention-based feature fusion method.

3.2.1. The Impacts of Frequency Components on Face Forensics. Face forgery detection is different from the traditional image classification tasks. As claimed in [21], the deep network models for image classification exploit both low-frequency and high-frequency information, both contribute to final classification. We conduct a preliminary experiment by selecting 10k face images in which real and fake ratios are half. The fake face images are generated by four face image forgeries. ResNet18 is exploited for experiments. These images are reconstructed by FFT with r as the radius to keep the centre frequency component (Figure 4(a)). The training and testing processes are recorded in Figure 4(b). The horizontal axis is the number of epochs trained, and the vertical axis is the ACC. r is the radius of masking. The larger the r is, the more the high-frequency components are retained. From it, we can observe the following: (1) for low-frequency images, the network converges much quickly, and three epochs are enough. (2) The initial accuracy is continuously improved with the increasing of the high-frequency components. (3) With the introduction of higher frequency components, the network benefits less, and even the accuracy drops.

From the above observations (1) and (2), the network should learn some features from low-frequency components. Note that the frequency components are exploited in parallel, which is different from the conventional image classification [21]. Actually, this is also consistent with our common sense. As we know, image classification is usually of semantic level, whereas face tampering detection is a fine-grained classification task. From the observation (3), since the image often contains some noises that usually exist in the high-frequency components, the accumulation of high-frequency components also brings some difficulties to network learning.

3.2.2. Multiscale Channel Attention. Wavelet transform can provide multiscale image description due to diverse frequency components. Both high-frequency and low-frequency components benefit for face forgery detection. Thus, fusing features is a key issue. The weights of the conventional channel attention mechanisms are based on the mean values of channels, e.g., SENet [39]. Although they work, yet ignore some important local information in the subimportant feature channels. This drawback inhibits wavelet transform from exerting its capability of detail representation. Inspired by the receptive field of human visual cortex neurons, we propose a multiscale channel attention (MSCA) mechanism, which considers the importance of local features and minimizes the side effect of noises. Figure 5 shows the proposed MSCA. C_n denotes different DCE feature maps. They are concentrated as C_a.

$$C_a = \text{concat}(C_1, C_2, \ldots, C_n). \tag{4}$$

We perform maximum pooling with the kernels of 3×3, 5×5, and 7×7 on C_a. For each pooling, we get a 1×1 channel array by global average pooling.

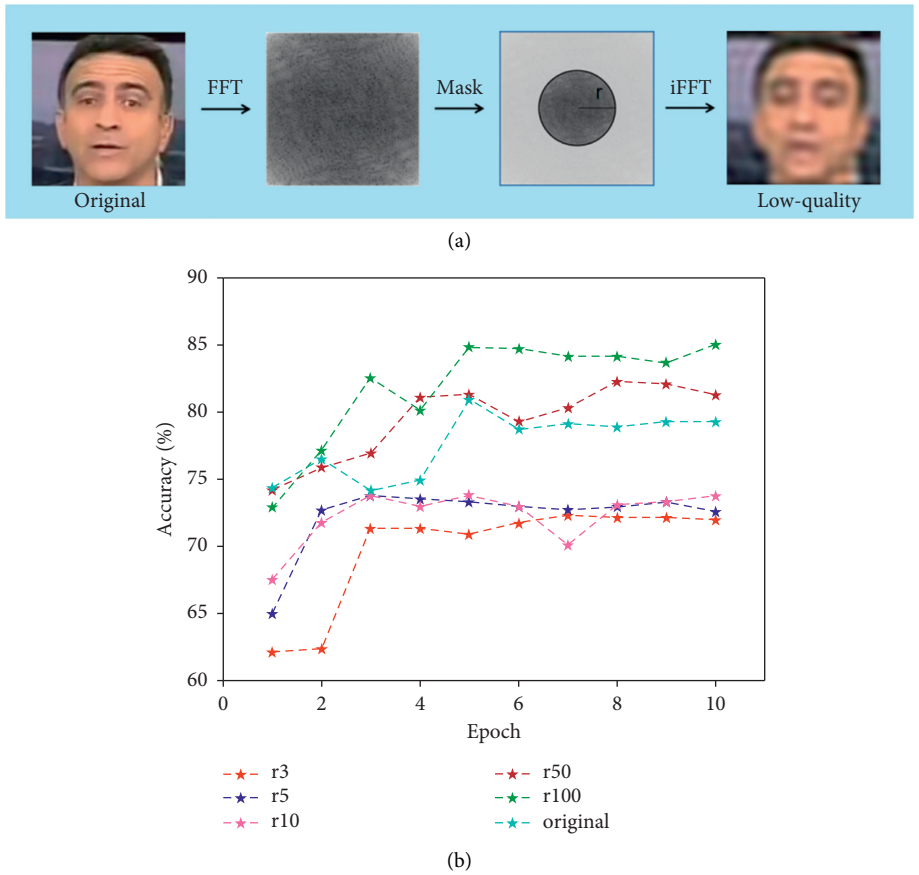


FIGURE 4: Exploring the effectiveness of different frequencies. (a) The original image is transformed by FFT, and we retain and reconstruct frequencies within the circle of radius (r). (b) The accuracy variation when using different frequency components during training.

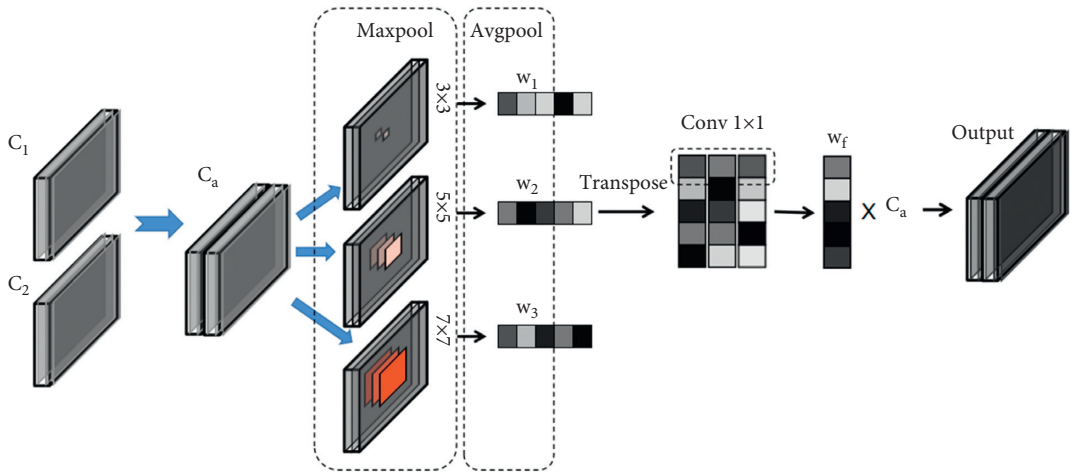


FIGURE 5: Multiscale channel attention (MSCA).

$$w_s = \mathrm{Avg}(C_a * \mathrm{Maxpool}_{s\times s}), \quad s = \{3, 5, 7\}. \tag{5}$$

Next, we transpose and concentrate them to 3×1 channels, then we use a 1×1 convolutional operation ($C_{1\times1}$) to obtain w_f. The final output is obtained by multiplying C_a with w_f.

$$w_f = \mathrm{concat}(w_3, w_5, w_7) * C_{1\times1}, \tag{6}$$

$$\mathrm{output} = w_f \odot C_a. \tag{7}$$

The maximum pooling strategy strengthens local features, while average pooling highlights global information.

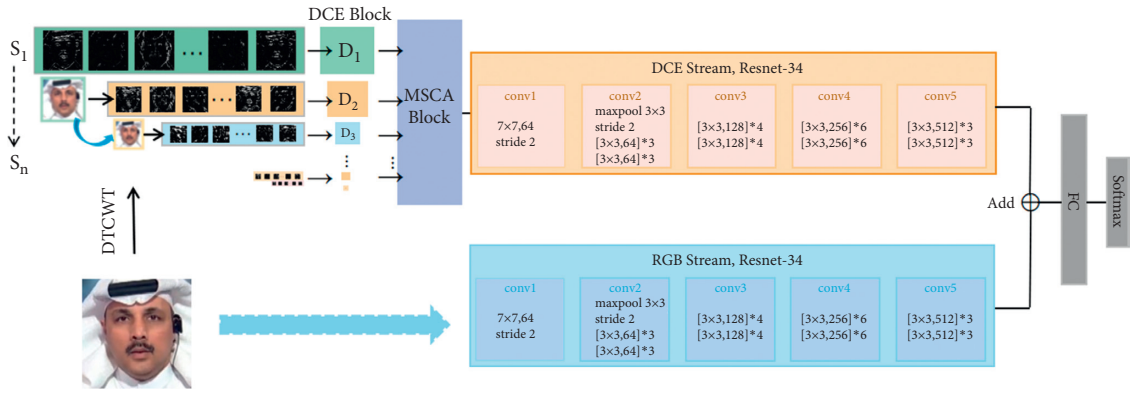


Figure 6: The framework of the proposed DCWNet.

Thus, the assignment of the weights for each channel is comprehensively considered by using MSCA. Please note that the directional features use high-frequency components. The experiment in Subsection 3.2.1 proves that the low-frequency components also play a role in the model training. Thus, we use a two-stream network to exploit the low-frequency information and pixel-level features simultaneously.

Based on the above methods, we proposed our DCWNet, and Figure 6 shows the framework of the complete work.

4. Experimental Results and Analysis

4.1. Experimental Setting. Image Dataset. FaceForensics++ is the most recent face manipulation dataset, which has been widely used in existing works [33, 43]. It is expanded from the FaceForensics dataset with three quality levels, namely, RAW (raw), HQ (high quality), and LQ (low quality). For the FaceForensics dataset, each level includes 1,000 videos, which are directly collected from YouTube without tampering. The same amounts of fake videos are generated by four face forgeries including Deepfake, Face2Face, FaceSwap, and Neural Textures. In addition, the FaceForensics++ dataset also contains 363 real videos from 28 actors under 16 scenes. Thus, the FaceForensics++ dataset has 1,363 real videos and 4,000 fake videos for each quality. We extract 60 frames for each real video at equal interval and 16 frames for each fake video. The MTCNN [44] is used to crop the face images. Thus, we have 63k fake face images and 63k real face images, totally 126k face images. We divide them into 85k, 35k, and 6k face images as the training set, the testing set, and the validation set, respectively. In addition, the DFDC preview [45] dataset, which is a preview dataset of the Deepfake Detection Challenge, is also used for experiments. It contains 1131 real videos and 4119 fake videos. We obtain 120k face images from the DFDC preview dataset.

Evaluation Metrics. To evaluate the effectiveness of our model, we exploit two widely used metrics, namely, classification accuracy (ACC) and area under receiver operating characteristic curve (AUC). The closer the ACC is to 100% and the AUC is to 1, the better the performance the network achieves.

Experiment Details. The ResNet34, which was pretrained on ImageNet [46], is exploited as the backbone for two streams. The Kaiming Batch Normalization is used for initialization. The networks are optimized via SGD with 0.9 as the momentum and 0.0005 as the weight decay. We set the base learning rate as 0.02 and use StepLR as the learning rate scheduler with half the learning rate per step. The batch size is 64 and we train the model for about 14k iterations. The whole work is completed upon PyTorch 1.1.0 with two Nvidia GeForce GTX 1080 Ti GPUs. To speed up the training process, we save the results of wavelet transform into local disk in NumPy format.

4.2. Comparisons with the Existing Works. The proposed DCWNet is tested on different quality image datasets that consist of fake images produced by different image tampering methods. Experimental comparisons are made among the proposed approach and the existing works. For the FaceForensics++ dataset, the experimental results are shown in Table 1. Apparently, the proposed DCWNet achieves a pretty high ACC (98.73%) and AUC (0.999) on the FaceForensics++ (HQ) dataset.

For the LQ dataset, DCWNet also achieves desirable results with the ACC of 97.91% and the AUC of 0.994. Compared to the baseline networks (ResNet34), DCWNet achieves the improvement of ACC about 2.05%. This proves that the DCE block is effective. Figure 7 reports the ROC curves for different face forgery detection methods. We also conduct the experiments on the DFDC preview dataset with the same experimental setting. Table 2 reports the experimental results.

For different face manipulations, we also test our model. Specifically, there are four face manipulations for the fake images in the FaceForensics++ dataset. Each face manipulation has 31k images. Among them, 22k, 8k, and 1k are used for training, testing, and validation, respectively. Similar experimental results are obtained, which are reported in Table 3.

Table 1: Results on the FaceForensics++ dataset with LQ and HQ.

Methods	ACC (LQ) (%)	AUC (LQ)	ACC (HQ) (%)	AUC (HQ)
Meso-4 [27]	54.38	0.542	60.63	0.660
Meso-Incep [27]	58.30	0.694	64.49	0.734
HP-CNN [11]	62.59	0.683	64.09	0.712
Constrained Conv [47]	80.05	0.883	83.40	0.920
AMTEN [14]	83.76	0.868	85.69	0.917
XceptionNet [9]	88.04	0.974	92.29	0.985
ResNet34 [8]	93.93	0.753	96.68	0.803
DCWNet(ResNet34)	97.91	0.994	98.73	0.999

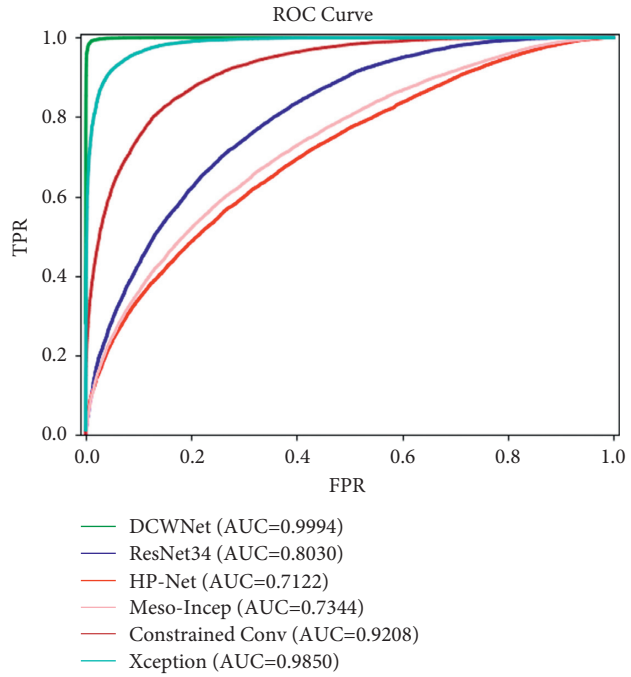


Figure 7: ROC curve for different face forgery detection methods.

Table 2: The experimental results on the DFDC preview dataset.

Methods	ACC (%)	AUC
Meso-4 [27]	53.71	0.553
Meso-Incep [27]	58.16	0.654
HP-CNN [11]	61.49	0.675
Constrained Conv [47]	81.01	0.877
AMTEN [14]	88.83	0.892
XceptionNet [9]	89.37	0.969
ResNet34 [8]	94.52	0.736
DCWNet(ResNet34)	97.31	0.920

4.3. Ablation Study

4.3.1. The DCE Block. To prove the contribution of the proposed DCWNet, ablation study is conducted. We first explore the influence of the number of directions, and the experimental results are recorded in Table 4. Even with features from one direction, the DCE stream achieves high ACC and AUC. This proves that the DCE block is powerful for local feature representation. With more features from multiple directions, the detection accuracies improve greatly. This implies that the features extracted from different directions are complimentary to each other. We also compare the effect of the FC layer and 1×1 convolution used in MSCA. We observe that with the using of more directions, FC is better than 1×1 convolution.

Figure 8 shows some feature maps extracted from the DCE block. We can notice that the attention responses of the fake images are distracted, whereas those of the real images are compact. The reason behind this is that the directional features are not strongly correlated in fake face images, while they are more uniform for real face images.

Table 3: Detection results for different face manipulations.

Methods	Deepfake (%)	Face2Face (%)	FaceSwap (%)	Neural Textures (%)
Meso-4 [27]	53.31	61.80	62.08	50.33
Meso-Incep [27]	76.01	71.12	71.69	50.30
HP-CNN [11]	86.03	81.48	89.30	77.07
Constrained Conv [47]	82.39	81.63	88.57	79.15
AMTEN [14]	86.56	84.76	80.12	76.07
XceptionNet [9]	97.51	97.24	97.11	79.41
ResNet34 [8]	98.32	98.35	97.90	95.90
DCWNet(ResNet34)	99.54	99.55	98.84	96.24

Table 4: Ablation study of the DCE block for different number of directions.

Direction	Conv 1 × 1		FC	
	ACC (%)	AUC	ACC (%)	AUC
(+15°)	89.24	0.908	90.03	0.898
(+15°, +45°)	91.19	0.932	90.42	0.906
(+15°, +45°, +75°)	92.88	0.929	92.74	0.938
(+15°, +45°, +75°, −15°)	92.87	0.923	93.28	0.942
(+15°, +45°, +75°, −15°, +45°)	93.40	0.946	94.38	0.948
(+15°, +45°, +75°, −15°, −45°, +75°)	93.77	0.956	95.34	0.962

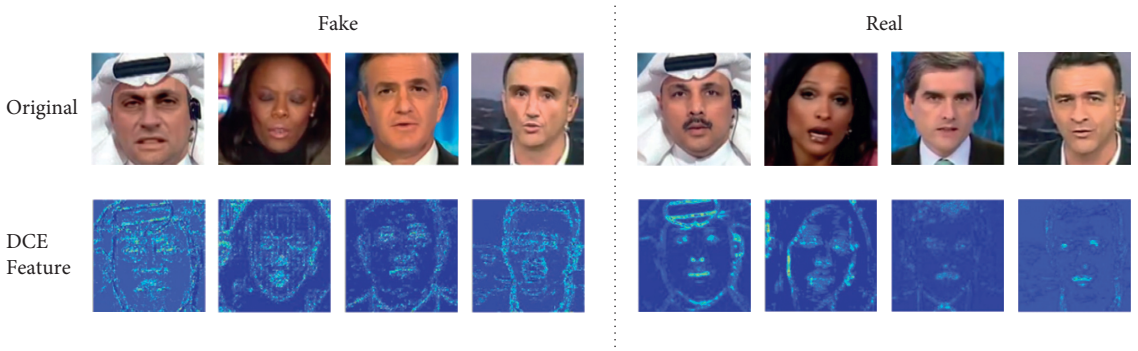


Figure 8: Feature maps from DCE block.

Table 5: Comparisons among three feature fusion methods.

Components	S1 (%)	S2 (%)	Addition (S1, S2) (%)	SE (S1, S2) (%)	MSCA (S1, S2) (%)
ACC	95.34	94.98	95.28	95.46	96.81

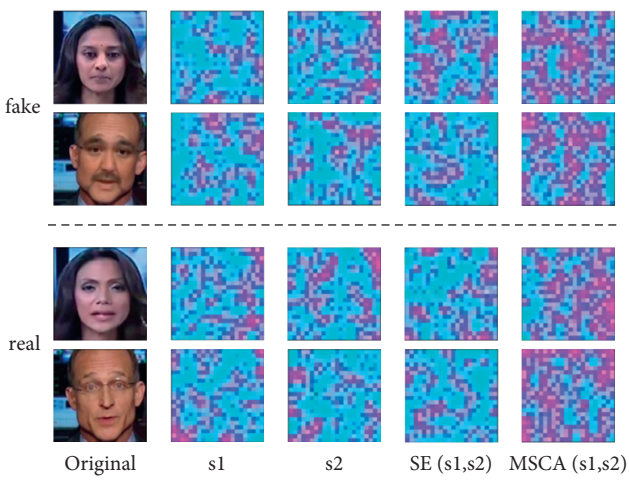


Figure 9: The feature maps from different fusion methods.

4.3.2. MSCA. To prove the effectiveness of MSCA, we use different feature fusion methods for the DCE feature maps. The experimental results are reported in Table 5. Specifically, we conduct experiments for the first (S1) and second (S2) stages of the wavelet transform, respectively. The element-wise addition, self-attention (SE), and MSCA are used for feature fusion. From Table 5, the MSCA achieves the best feature fusion. Figure 9 also compares the feature maps from the DCE stream between SE and MSCA.

5. Conclusion

In this work, we propose a two-stream DCWNet for face forgery detection. One stream uses the DCE block to exploit the multiscale directional correlation. To fuse the DCE feature maps of different scales, MSCA is proposed. The other stream uses the original image as input. The experimental results showed that DCWNet achieves desirable

results on the FaceForensics++ and DFDC preview datasets. From the ablation study, we observe that real and fake faces have different feature maps that learned from the DCE block. This proves that the correlation of direction distribution is valuable for face forgery detection. Moreover, the effectiveness of the proposed MSCA is verified by comparisons with existing feature fusion methods. We also explore how different frequency components contribute to face forgery detection, which provides some interpretability for face forensics.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] FaceSwap: https://github.com/MarekKowalski/FaceSwap.

[2] Deepfakes github: https://github.com/deepfakes/faceswap.

[3] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2387–2395, San Francisco, CA, USA, August 2016.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, June 2014.

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

[6] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, Los Angeles, USA, June 2019.

[7] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," Available: https://arxiv.org/abs/1704.04861, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, San Francisco, CA, USA, August 2016.

[9] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, Honolulu, HI, USA, July, 2017.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, Honolulu, HI, USA, July, 2017.

[11] H. Mo, B. Chen, and W. Luo, "Fake faces identification via convolutional neural network," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, New York, USA, June 2018.

[12] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," Available: https://arxiv.org/abs/1811.00656, 2018.

[13] L. Nataraj, T. M. Mohammed, and B. S. Manjunath, "Detecting GAN generated fake images using co-occurrence matrices," *Journal of Electronic Imaging*, vol. 5, pp. 1–7, 2019.

[14] Z. Guo, G. Yang, J. Chen, and X. Sun, "Fake face detection via adaptive manipulation traces extraction network," *Computer Vision and Image Understanding*, vol. 204, Article ID 103170, 2021.

[15] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," in *Proceedings of the 11th IEEE International Workshop on Information Forensics and Security (WIFS)*, Delft, The Netherlands, December 2019.

[16] 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, Glasgow, UK, August 2020.

[17] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: frequency channel attention networks," Available: https://arxiv.org/abs/2012.11879, 2020.

[18] D. F. Elliott and K. R. Rao, *Fast Fourier Transform and Convolution Algorithms*, Springer-Verlag, New York, 1982.

[19] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Signal Processing Magazine*, vol. 100, no. 1, pp. 90–93, 1974.

[20] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Transactions on Computers*, vol. 22, no. 6, pp. 123–151, 2005.

[21] H. Wang, X. Wu, and Z. Huang, "High-frequency component helps explain the generalization of convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694, Seattle, WA, USA, June 2020.

[22] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[23] X. Pan, X. Zhang, and S. Lyu, "Exposing image splicing with inconsistent local noise variances," in *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pp. 1–10, Seattle, WA, USA, April 2012.

[24] H. Dang, F. Liu, and J. Stehouwer, "On the detection of digit manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, June 2018.

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

[28] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "DeepFake detection based on discrepancies between faces and their context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 1, p. 99, 2020.

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

[30] D. Güera and E. J. 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, IEEE, Auckland, New Zealand, November 2018.

[31] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *Interfaces*, vol. 3, no. 1, pp. 80–87, 2019.

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

[33] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proceedings of the 2009 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92, IEEE, Snowbird, UT, USA, December 2009.

[34] N. Yu, L. Davis, and M. Fritz, "Attributing fake images to gans: analyzing fingerprints in generated images," 2018, https://arxiv.org/abs/1811.08180.

[35] B. Chen, X. Ju, B. Xiao, W. Ding, Y. Zheng, and V. H. C. De Albuquerque, "Locally GAN-generated face detection based on an improved Xception," *Information Sciences*, vol. 572, pp. 16–28, 2021.

[36] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.

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

[38] X. Gong, S. Chang, Y. Jiang, and Z. Wang, "Autogan: neural architecture search for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3224–3234, Los Angeles, USA, June 2019.

[39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[40] Q. Wang, B. Wu, P. Zhu, L. Peihua, Z. Wangmeng, and H. Qinghua, "ECA-Net: efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.

[41] S. Woo, J. Park, J. Y. Lee, and S. I. Kweon, "Cbam: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Munich, Germany, September 2018.

[42] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake City, UT, USA, June 2018.

[43] 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 IEEE 10th International Conference on Biometrics Theory, Applications and Systems*, pp. 1–8, Tampa, USA, 2019.

[44] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[45] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, https://arxiv.org/abs/1910.08854.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the I2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, June 2009.

[47] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pp. 5–10, Web Tokyo, Japan, 2016.

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

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1627486, 15 pages
https://doi.org/10.1155/2021/1627486

Research Article

F3SNet: A Four-Step Strategy for QIM Steganalysis of Compressed Speech Based on Hierarchical Attention Network

Chuanpeng Guo,[1] **Wei Yang**,[1] **Mengxia Shuai**,[2] **and Liusheng Huang**[1]

[1]*School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China*
[2]*School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China*

Correspondence should be addressed to Chuanpeng Guo; guocp@mail.ustc.edu.cn

Received 10 June 2021; Accepted 8 September 2021; Published 22 September 2021

Academic Editor: Beijing Chen

Traditional machine learning-based steganalysis methods on compressed speech have achieved great success in the field of communication security. However, previous studies lacked mathematical modeling of the correlation between codewords, and there is still room for improvement in steganalysis for small-sized and low embedding rate samples. To deal with the challenge, we use Bayesian networks to measure different types of correlations between codewords in linear prediction code and present F3SNet—a four-step strategy: embedding, encoding, attention, and classification for quantization index modulation steganalysis of compressed speech based on the hierarchical attention network. Among them, embedding converts codewords into high-density numerical vectors, encoding uses the memory characteristics of LSTM to retain more information by distributing it among all its vectors, and attention further determines which vectors have a greater impact on the final classification result. To evaluate the performance of F3SNet, we make a comprehensive comparison of F3SNet with existing steganography methods. Experimental results show that F3SNet surpasses the state-of-the-art methods, particularly for small-sized and low embedding rate samples.

1. Introduction

As an effective way to secretly transfer information over the Internet, steganography uses the redundancy of digital carriers to accomplish secret information embedding. In recent years, due to the pervasiveness of streaming media technologies, VoIP steganography and their countermeasures have become one of the hot topics in information hiding [1–3].

Among many VoIP applications for band-limited channels and wireless communication, speech coders such as G.729, G.713.1, Adaptive Multirate (AMR), and Enhanced Full Rate (EFR) have become essential components in mobile and wireless communication. How to exploit the redundancy existing in the encoding process to achieve steganography is a new research hotspot. Some methods which embed secret messages into the bitstream during the encoding process have been proposed, such as quantization index modulation (QIM) steganography [4–6], fixed codebook (FCB) steganography [7–9], and pitch modulation (PM) steganography [10, 11].

As the counterpart of steganography, steganalysis is not only to ensure that steganography is not maliciously abused but also a key technique for evaluating the performance of steganography algorithms. Machine learning algorithms, especially support vector machine (SVM), have been widely used in the field of steganalysis of both traditional media and VoIP streams. For QIM steganography, S. Li et al. proposed a variety of detection methods [12, 13]. In [12], they presented a statistical model to extract the quantitative feature vectors of the index distribution characteristics (IDC). In another work, Li et al. [13] further presented a model called the quantization codeword correlation network (QCCN) to quantify the correlation characteristics of the vertices in the correlation network. For FCB steganography, Miao et al. [14] first presented a Markov Transition Probabilities- (MTP-) based detection method and an entropy-based detection method to detect the steganography of compressed speech.

To improve the performance, Ren et al. [15] used the statistical probability of Same Pulse Position (SPP) in the same track to accurately distinguish covers from stegos. For PM steganography, Liu et al. [16] extracted the statistics of the high-frequency spectrum and the mel-cepstrum coefficients of the second-order derivative for detecting audio steganography. Li et al. [17] proposed a network model to quantify the correlation characteristics of the adaptive codebook. Undoubtedly, steganalysis of compressed speech based on machine learning has made great progress.

However, such methods mentioned above are facing some challenges. Firstly, as steganography becomes more sophisticated [7, 8, 18], the extracted statistical features for steganalysis are evolving from low dimensions and simplicity to high dimensions and complexity [19]. Secondly, information hiding technology is gradually developing towards randomization and fine granularity, that is, within the allowable range of carrier distortion, secret information is first divided into small segments, and then, carriers of different lengths are randomly selected to achieve fine-grained steganography with different embedding rates. Nevertheless, most existing steganalysis methods do not perform well [14, 15], especially for small-sized and low embedding rate samples.

Fortunately, the emergence of neural networks (NNs) has brought hope to deal with these challenges. In 2018, Lin et al. [20] first introduced neural networks (NNs) to the steganalysis of compressed speech. They proposed Recurrent Neural Network- (RNN-) based steganalysis model (RNN-SM) to detect the disparities in codeword correlations caused by QIM steganography. In 2019, Chen et al. proposed a steganalytic scheme by combining RNN and Convolutional Neural Network (CNN) for FCB steganography. However, sequence coding based on CNN or RNN is still a local coding method, and it models the local dependency of input information. In [21], Vaswani et al. argued that the attention mechanism can completely replace LSTM and convolutional neural networks. Inspired by their work, we integrate the attention mechanism and RNN and propose a deep network model to mine information that reflects changes in the correlation between codewords before and after steganography.

In this paper, we introduce F3SNet, a four-step strategy for QIM steganalysis based on hierarchical encoding representations. In F3SNet, the RNN encoder is used to keep much more information by being distributed among all its vectors, and the attention mechanism is used to decide which vectors should be paid more attention to. The practice has proved that F3SNet is very sensitive to the weak signal changes brought by steganography, especially for small-size and low embedding rate samples.

In summary, this work makes the following contributions:

(1) We first use the Bayesian network (BN) to establish a framework for uncertainty knowledge expression and reasoning and then calculate the link strength between different nodes as a measure of the strength of the codeword correlation. The process of quantification analysis serves as an essential step towards effective detection using a deep learning framework.

(2) We present F3SNet, a four-step strategy for QIM steganalysis method based on the hierarchical attention network. Through a four-step strategy, we encode the numerical codeword vectors into multiple memory vectors, then select a set of vectors that have the greatest impact on the classification result to prevent information overload, and finally achieve efficient steganography classification, even in special cases, such as small size and low embedding rate.

(3) To evaluate the performance of F3SNet, we perform comprehensive experiments on detection accuracy (ACC), false positive rate (FPR), and false negative rate (FNR) of the algorithm under different lengths and different embedding rates. Furthermore, we compare F3SNet with several existing algorithms, such as IDC [12], QCCN [13], RNN-SM [20], and FCEM [22] methods under different embedding rates and different lengths. The experimental results show that our algorithm is superior to other state-of-the-art algorithms.

The rest of the paper is structured as follows. Section 2 reviews related work on existing steganography and steganalysis of compressed speech. Section 3 provides an overview of linear prediction analysis and QIM steganography. Section 4 discusses correlations using the Bayesian network. Section 5 details the design and implementation of F3SNet, followed by experiments and discussions in Section 6. Finally, we conclude the paper and discuss future work in Section 7.

2. Related Works

In 2010, Ding and Ping [23] used the histogram features of the pulse position parameter to train the SVM classifier to distinguish cover and stego speech. In 2011, Huang et al. [24] employed the second detection and regression analysis not only to detect the hidden message but also to estimate the length of embedded messages. However, their method is a relatively dedicated steganography method. Li et al. [12] designed statistical models to extract the quantitative feature vectors of these characteristics for detecting QIM steganography using the SVM classifier. Furthermore, Li et al. [13] built a QCCN model, extracted feature vectors from split quantization codewords, and then trained a high-performance SVM classifier.

In addition, for FCB steganography, Miao et al. [14] used the Markov property of speech parameters to propose a detection method based on MTP and entropy in 2013. Ren et al. [15] proposed an AMR steganalysis algorithm based on the probability of the same pulse position in the same track in 2015. For better performance, in 2016, Tian et al. [19] characterized AMR speech exploiting the statistical properties of pulse pairs and presented a steganalysis of AMR speech based on the multidimensional feature selection mechanism. For pitch modulation steganography, Li et al.

[17] proposed a network model to quantify the correlation between the adaptive codebook. The SVM classifier was used in the above three papers.

In recent years, with the application of different types of deep learning, many novel algorithms have been proposed for steganalysis and forgery based on image, audio, and video [25–27]. Compared with the conventional methods with handcrafted features [13, 19, 28, 29], the algorithms based on deep learning can significantly improve the detection performance. In 2015, Qian et al. [30] proposed a customized CNN for image steganalysis. The model could capture the complex dependencies in images and achieve better detection performance than the Spatial Rich Model (SRM). Xu et al. [31, 32] proposed a CNN architecture that is more suitable for image steganalysis and enhanced it by improving the statistical model in the subsequent layers and preventing overfitting. Ye et al. [33] proposed a CNN-based image steganalysis method, which uses an activation function called truncated linear unit (TLU), and improved the steganalysis ability by incorporating the knowledge of selection channel. In 2016, Paulin et al. [34] presented an audio steganalysis method using deep belief networks (DBN). Compared with SVM and Gaussian mixture model (GMM), the proposed DBN-based steganalysis method could get higher classification accuracy. In 2017, Chen et al. [35] designed a novel CNN to detect audio steganography in the time domain. However, due to different signal characteristics, these algorithms are difficult to directly apply to compressed speech.

In 2018, Lin et al. [20] proposed the codeword correlation model based on RNN. They used a supervised learning framework to train RNN-SM. Experiments showed that RNN-SM achieved better detection results regardless of short sample length or low embedding rate. In 2019, Chen et al. [36] proposed a steganalytic scheme by combining RNN and CNN. They utilized RNN to extract higher level contextual representations of FCBs and CNN to fuse spatial-temporal features for the steganalysis. Experiments results validated that their method outperforms the existing state-of-the-art methods. In 2019 and 2020, Hao et al. [22, 37] successively proposed hierarchical representation network and multihead attention-based network to extract correlation features for QIM steganalysis. Both methods significantly improve the best result especially in detecting both short and low embedded speech samples. Inspired by their work, we proposed a new model called F3SNet based on the hierarchical attention network to model the spatial and temporal characteristics of the quantization index in LPC and further improve the accuracy of detecting CNV steganography [4].

3. Background

3.1. Linear Prediction Analysis. As the basis of low-rate speech coding, the basic idea of linear predictive analysis (LPA) is to use the correlation of the speech signal to approximate the sample value at the current moment with the linear combination of several past speech samples. Linear predictive coding is mainly divided into three processes: LPA, line spectrum pair (LSP) analysis, and vector quantization (VQ). First, the speech signal can be regarded as the output produced by an input sequence $\mu(n)$ exciting an all-pole system $H(z)$. The transfer function of the system is

$$H(z) = \frac{G}{1 - \sum_{i=1}^{p} \alpha_i z^{-i}}, \tag{1}$$

where G is a constant, p is the order of the model, and α_i is a real number. The p prediction coefficients form a p-dimensional vector, which is the linear prediction coefficient.

However, the LPC coefficient fluctuates greatly, and the error of a certain LPC coefficient will make a greater impact on the entire frequency domain. Therefore, the LPC coefficient is not suitable for direct quantization and needs to be further transformed into the line spectrum frequency parameter LSF (line spectrum frequency). To further balance the bit rate and quantization accuracy, vector quantization technology is used to search the codebook for the codeword vector $\overrightarrow{C_k}$ that is closest to the vector $\overrightarrow{p}$ to be quantized in a certain distance, and the sequence number k of the codeword vector is obtained as the quantization result.

3.2. QIM Steganography. The intrinsic essence of QIM steganography is that there is redundancy in the quantization codebook, and the suboptimal codebook parameters caused by steganography have little impact on the speech quality.

Chen et al. first proposed a steganography method suitable for QIM of static digital carriers such as image, text, audio, and video [38]. Assume that the secret information to be transmitted is from the set $S = \{s_k | 1 \le k \le n\}$. The sender wants to hide secret information s_k. First, the codebook D is divided into n disjoint subsets $C = \{c_k | 1 \le k \le n\}$. Then, he (or she) establishes the mapping: $f: s_k \longrightarrow c_k$. For the input vector X to be quantized, only the codeword closest to X is searched in subcodebook $f(s_k)$. The receiver extracts secret information by checking which part of the codebook the codeword belongs to.

In 2009, Xiao et al. [4] combined the QIM method with VQ in the encoding process of compressed speech and proposed a novel steganography algorithm based on complementary neighbor vertices (CNV). Given N codewords, every codeword is m-dimensional. Xiao et al. used graph theory to establish a graph $G(V, E)$ in the code space, which can be defined as follows:

$$\begin{cases} V = \{V_i | 0 \le i \le N, |V_i| = m\}, \\ E = \left\{ \langle v_i, v_j \rangle | d(V_i, V_j) = \sqrt{\left(\sum_{i=1}^{m} (x_i - y_i)^2 \right)} \right\}, \end{cases} \tag{2}$$

where V_i is the ith codeword in the codebook. Each edge represents a certain relationship between codewords, and the weight of the edge is defined as the Euclidean distance between any two codewords. In Xiao's paper, he gave a graph construction algorithm and proved that the graph can be two-colorable. In the process, the vertices of the same color

were assigned to the same subset. The dyeing operations were repeated until all vertices have been assigned, to obtain different partitioned subsets of the codebook. Finally, each codeword is in the opposite part to its nearest neighbor. Suppose X is the input value to be quantized. In this case, the additional quantization distortion caused by CNV steganography can be given:

$$\mathscr{L}(X, \widehat{Y}) = d(X\widehat{Y}) - d(XY). \tag{3}$$

It can be proved that the algorithm can minimize the signal distortion and significantly improve the undetectability and robustness of CNV steganography. This paper implements steganalysis for the CNV algorithm.

4. Codewords Correlation Modeling and Analysis

To fully describe the correlation between codewords in LPC, we use the BN to model the codewords and then analyze the correlation. BN can be represented as a 2-tuple $\langle G, \theta \rangle$, where $G = (V, E)$ denotes a directed acyclic graph and θ denotes a set of conditional probabilities, called network parameters.

Suppose there are S frames, each of which contains N codewords. V and E represent the set of vertices and the set of edges in the directed graph G, respectively, which can be expressed as follows:

$$\begin{cases} V = \{V_1[m], V_2[m], \ldots, V_N[m]\}, \quad m \in \{0, 1, \ldots, S-1\}, \\ E = \left\{\left\{\langle v_i, v_j \rangle\right\} | v_i \in \{V_1[i], V_2[i], \ldots, V_N[i]\}, v_j \in \{V_1[j], V_2[j], \ldots, V_N[j]\}\right\}, \end{cases} \tag{4}$$

where $L(0 \le L \le (S-1))$ denotes the relative distance of different frames. If $j - i = 0$, $\langle v_i, v_j \rangle$ stands for the edge in the interframe. If $j - i \ge 1$, $\langle v_i, v_j \rangle$ stands for the edge in the intraframe. Once the vertices and edges of the directed graph G are determined, the network parameters θ can be computed to characterize the dependencies between the vertices. Therefore, the following formula can be established:

$$\Theta = \{P(\Lambda_i | V_i), i = 1, 2, \ldots, N\}, \tag{5}$$

where V_i is the set of parent nodes of node Λ_i. The construction of BN includes structure learning and parameter learning, and parameter learning depends on structure learning. Structure learning refers to finding a network structure that is as similar as possible to the data for any given dataset $D = \{D_1, D_2, \ldots, D_n\}$. In the paper, the K2 algorithm based on Bayesian scoring rules is used to find the network with the largest probability under a given dataset. According to the Bayesian formula,

$$P(G|D) = \frac{P(G)P(D|G)}{P(D)}, \tag{6}$$

where $P(G)$ is the prior knowledge of the network structure G and the dataset D is known information and is independent of the network structure, and we have

$$\max \arg_G P(G|D) = \max \arg_G P(G)P(D|G). \tag{7}$$

Since $P(G)P(D|G) \propto \log P(G) + \log P(D|G)$, the Bayesian score is defined as follows:

$$Score(G, D) = \log P(G) + \log P(D|G). \tag{8}$$

Assuming that the prior distribution of the parameter Θ obeys the Dirichlet distribution, let r_i represent the number of values of the ith variable, q_i represent the number of possible values of the parent node of the ith variable, m_{ijk} represent the number of samples whose parent node is the jth value when the ith node in the Bayesian network takes the kth value, and α_{ijk} is a hyperparameter, and $\alpha_{(ij*)} = \sum_k \alpha_{ijk}, m_{ij*} = \sum_k m_{ijk}$; then,

$$\text{Score}(G, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\log \frac{\Gamma(\alpha_{ij*})}{\Gamma(\alpha_{ij*} + m_{ij*})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})} \right], \tag{9}$$

where $\Gamma(\cdot)$ is the gamma function and n represents the number of variables. It has been proved that the K2 algorithm can almost learn the Bayesian network when the node priority is completely correct.

To verify the effectiveness of BN, we select a 40-second speech segment, compress it with a G.729 vocoder, and then extract 4000 sets of quantized codewords. In the experiment, we construct the BN with 9 vertices and then perform parameter learning. Using the above K2 algorithm, the learned network structure is shown in Figure 1. The intraframe codeword correlation is mainly reflected between codeword l_1 and codeword l_2 and between codeword l_1 and codeword l_3, and the interframe correlation is mainly reflected in the first codewords of the two consecutive frames. How to measure and visualize the link strength between different codewords? For that purpose, Imme [39] proposed a measurement method for discrete Bayesian networks based on mutual information and conditional mutual information. In his method, X and Z are both the parent nodes of Y, and $P(y|x, z)$ is given by the conditional probability table of y; given x and z, link strength is defined as

$$LS_{\text{blind}}(X \longrightarrow Y) = \widehat{E}(Y|Z) - \widehat{E}(Y|X, Z), \tag{10}$$

where

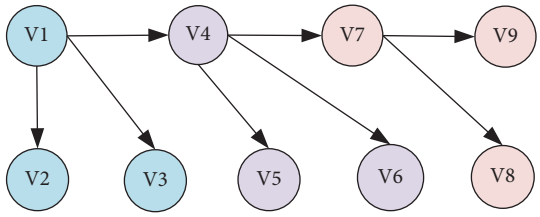


Figure 1: 9-node Bayesian network structure.

$$\widehat{E}(Y|Z) = \frac{1}{\#(X)\#(Z)} \sum_{x,y,z} P(y|x,z)\log_2 \frac{\#(X)}{\sum_x P(y|x,z)},$$

$$\widehat{E}(Y|X,Z) = \frac{1}{\#(X)\#(Z)} \sum_{x,y,z} P(y|x,z)\log_2 \; P(y|x,z), \tag{11}$$

where $\#(X)$ denotes the number of discrete states of X. Conveniently, the LinkStrength package in MATLAB's Bayes Net Toolbox (BNT) provides functions to calculate and visualize entropy, connection strength, and link strength for discrete Bayesian networks. For simplicity, we only use link strength in this paper. Figure 2 shows blind average link strength.

In the link strength graph, the value of the link strength is indicated by the number next to the arrow. As indicated by the blind average link strength in Figure 2, most links are quite strong. Especially, the link strengths between the first codewords of two consecutive frames are 3.472 and 3.582, respectively, which are the two connections with the largest value. This demonstrates that the correlation between consecutive frames is the strongest. Next, it can be observed that in three consecutive frames, the link strength between the first codeword and the third codeword is greater than the link strength between the first codeword and the second codeword. For example, in the first frame, the former value is 1.996 and the latter value is 1.953, which is 4.3 % higher. This implies that the correlation between the first and the third codeword is stronger than that between the first and the second codeword. Furthermore, the absence of links between other vertices does not mean that there are no correlations between them. It is just that the correlations are too weak and optimized by the learned model. Of course, the weak links can be measured by manually adding the link relationship in the graph.

As can be seen, the correlations between codewords in LPC are complex. The correlation measure proposed in [20] uses conditional probability, provided that it is based on the Markovian modeling of the codeword sequence. However, our method is based on Bayesian networks, which are closer to the true distribution of the codeword sequence. Thus, it is necessary to find a novel method to improve the traditional detection method. Steganalysis based on deep learning can automatically extract the intrinsic features of the carrier, avoiding the complexity of establishing the model. Therefore, we propose a steganalysis method that utilizes the advantages of RNN and attention mechanism.

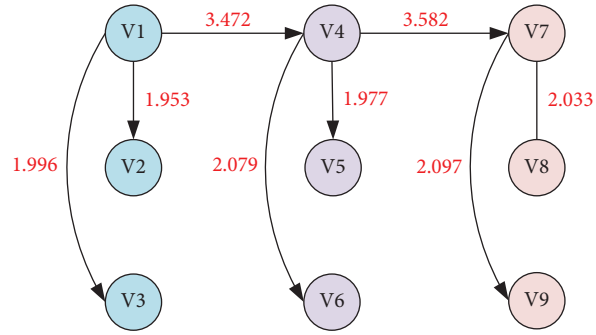


Figure 2: Link strengths using blind average.

5. Proposed Method

Till now, we can formally present our F3SNet, which is an architecture based on a hierarchical attention network. The structure is shown in Figure 3. It includes an embedding layer, multilayer attention layer, and a classifier. Among them, the multilayer attention layer adopts a two-layer structure and includes a single codeword encoder, a codeword attention layer, a codeword sequence encoder, and a codeword sequence attention layer.

The steganography classification is briefly summarized as follows. Simply feed in an input array and get the codeword vectors and codeword sequence matrices. The codeword vectors are taken as the input and sent to the first attention layer. The compressed vector representations of the codewords are provided by LSTM, and then, some important vectors that can reflect the correlation of the codeword are extracted by the attention mechanism. Simultaneously, these codeword sequence matrices enter the second attention layer. After the same operation, a sequence-level expression that summarizes all information in the entire speech is obtained. Finally, the obtained representations are further used as classification features to achieve steganography classification by a fully connected network. For the convenience of verification, we choose keras as the steganalysis framework. Below we describe the details of different components.

5.1. Input. As we know, speech has a hierarchical structure similar to that of a document, which can be divided into different sentences, and each sentence contains a corresponding number of words. As a result, one speech can be divided into codeword sequences and codewords. Each codeword sequence and codeword contains unique information. To fully mine this information, we use a hierarchical attention network to model the structure of the quantized codewords. Here, two types of input data with different shapes are required.

Assume that there are S frames in a given speech sample of duration $L(s)$. We extract the codeword index and pack all indices of a speech sample into a vector X with size $(S \times 3)$. X_1 is the first layer input, and the format is as follows:

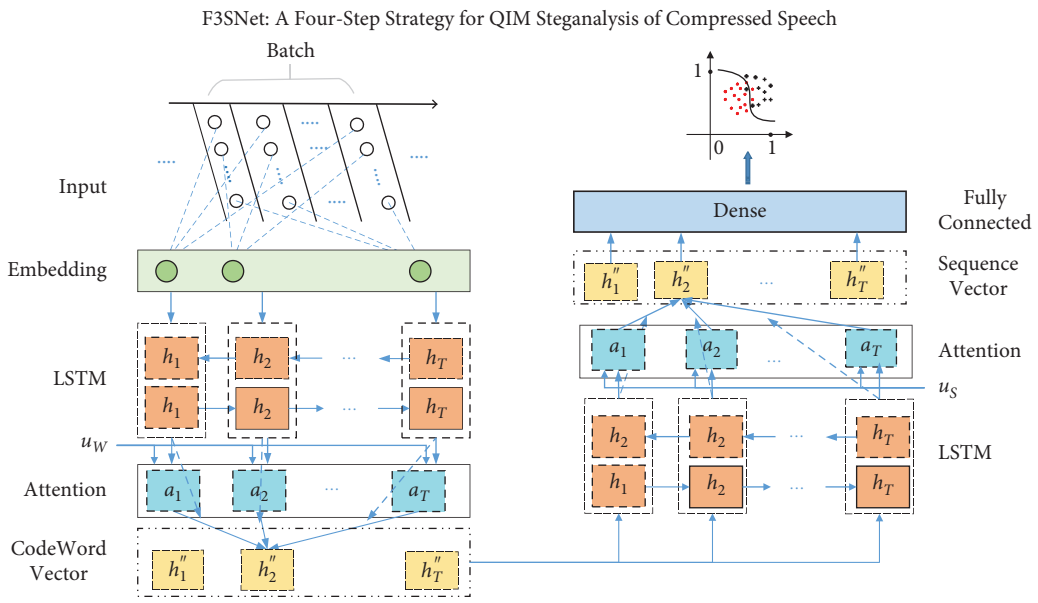


Figure 3: The model based on the hierarchical attention network.

$$X_1 = \left[l_{00}, l_{10}, l_{20}, l_{01}, l_{11}, l_{21}, \ldots, l_{0(S-1)}, l_{1(S-1)}, l_{2(S-1)}\right], \quad (12)$$

where $l_{ij}\,(0 \le i \le 2, 0 \le j \le S-1)$ denotes the ith index in the jth frame. For the second layer input, we take the L-len speech as a unit and pack the codeword indices of the S frame into a matrix as

$$X_2 = \begin{bmatrix} l_{00} & l_{01} & \cdots & l_{0(S-1)} \\ l_{10} & l_{11} & \cdots & l_{1(S-1)} \\ l_{20} & l_{21} & \cdots & l_{2(S-1)} \end{bmatrix}. \quad (13)$$

5.2. Embed. The embedding layer is used as the first hidden layer in our model, which converts the quantized codeword index sequence (QIS) into a fixed-size vector sequence. Through the embedding layer, a continuous, distributed QIS representation can be obtained and can effectively characterize the correlations between different codewords. In principle, a set of two-dimensional tensors with shape $(\text{batchsize}, S \times 3)$ is fed into the embedding layer. And, they are used as 'indices' to select a permutation of inner trainable weights matrix $W_{\text{Max\_num}\times D}$, where D represents the output dimension of the embedding layer.

In our experiment, matrix $W_{\text{Max\_num}\times D}$ is initialized randomly, which is regarded as a part of the deep learning model, and updated during the model learning process. After multiple epochs, the entire correlations between codewords are correctly expressed. Using this learned weight, the final outputs are a batch of 3-dimensional tensors with shape $(\text{batchsize}, S \times 3, D)$, which are the encoded representations.

As can be seen in Section 6.3, the comparison between model #1 and #4 shows that the embedding layer can significantly improve the classification accuracy.

5.3. Encode. The embedding layer is followed by the LSTM coding layer. LSTM mainly processes the encoded sequence from left to right through three-gated logics (forgetting gate, input gate, and output gate) and returns an ordered list of hidden states $\{h_1, h_2, \ldots, h_T\}$ as well as an ordered list of output vectors $\{y_1, y_2, \ldots, y_T\}$. As shown in Figure 4, the LSTM cell remembers values over arbitrary time intervals, while the three gates regulate the flow of information into and out of the cell.

There are eight groups of parameters that need to be learned throughout the LSTM network, which are the weight matrices and the corresponding bias terms of the three gates. The parameters are defined as follows: forgotten gate weight matrix W_f and its bias term b_f, input gate weight matrix W_i and its bias term b_i, output gate weight matrix W_o and its bias term b_o, and cell state weight matrix W_c and its bias term b_c. For clarity, the four weight matrices are further subdivided into W_{if}, W_{hf}, W_{ii}, W_{hi}, W_{io}, W_{ho}, W_{ic}, and W_{hc}. Taking the forget gate as an example, the calculation process of giving the control factor and retaining how much memory is given. In each LSTM cell, the two weight matrices connecting the input node to the hidden node are, respectively, the input weights (W_{if}) and the hidden node feedback weights (W_{hf}). First, the network output h_{t-1} at time $t-1$ is combined with the current network input x_t and then linearly transformed to obtain u_f^T. The mathematical process is briefly described as follows:

$$u_f^t = \begin{bmatrix} W_{if} & W_{hf} \end{bmatrix} \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_f. \quad (14)$$

Then, u_f^T is mapped to $0 \sim q(1)$ by the nonlinear activation function to obtain the control factor of the forget gate, which can be described as

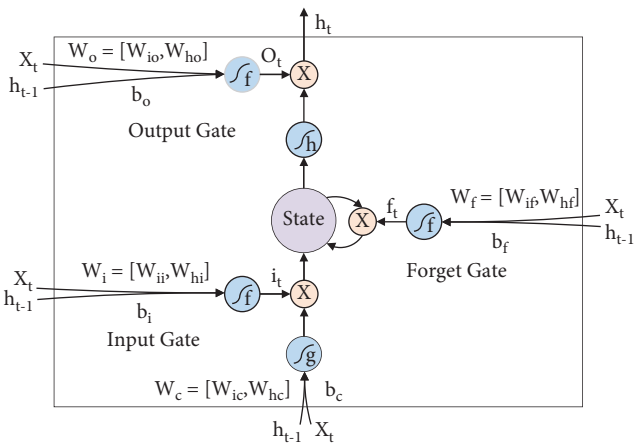


Figure 4: Internal structure diagram of an LSTM cell.

$$f_t = f\left(u_f^T\right). \tag{15}$$

In a similar way, the control factor i_t of the input gate and the control factor o_t of the output gate can be calculated. At each time step t, LSTM cell outputs two vectors: the memory c_t from the current block and the output h_t of the current block, i.e.,

$$\begin{cases} c_t = f_t \cdot c_{t-1} + i_t \cdot f\left(u_c^t\right), \\ h_t = f\left(u_o^t\right) \cdot f\left(c_t\right), \end{cases} \tag{16}$$

where the symbol $f(\cdot)$ represents the activation function, two types of activation functions ReLU or Tanh are used in the LSTM cell, and symbol "·" means multiplication by elements. Finally, LSTM will give an output sequence of dimension $L \times P \times Q$ ($Q = M \times D$), where L is the length of the samples, P is the batch size, M is the hidden size, and D is the network direction ($D = 1$ indicates a one-direction network; $D = 2$ indicates a bidirection network). In the work, the output vectors $H_T^{LSTM} = [h_1', \ldots, h_T']$ of the LSTM layer further serve as input for the attention layer.

5.4. Attend. As mentioned above, the encoder is able to keep much more information by distributing it among all its vectors. Moreover, not all vectors contribute equally to the final classification. Hence, the attention mechanism (AM) is introduced to extract such vectors that are important to the steganalysis and aggregate the representation of those informative vectors to form the feature vectors. As illustrated in Figure 5, attention can be divided into two steps. One is to calculate the attention distribution based on all input information; the other is to calculate the weighted average of the input information based on the attention distribution.

Given the input sequence H_T^{LSTM}, and then it is passed to a dense layer with activation tanh. A set of intermediate vectors is obtained:

$$\begin{aligned} U &= [u_1, \ldots, u_T] = \tan h\left(W H_T^{LSTM}\right) \in \mathscr{R}^{D_u \times N} \\ &= \tan h\left(W\left[h_1', \ldots, h_T'\right]\right), \end{aligned} \tag{17}$$

where W is the parameter matrix of the dense layer. The attention distribution can be then derived by comparing the output u_t of the dense layer with a trainable context vector u and normalizing with a softmax:

$$\alpha_{nt} = \frac{\exp\left(s\left(u_t, u_n\right)\right)}{\sum_k \exp\left(s\left(u_j, u_n\right)\right)}. \tag{18}$$

Using the scaled dot product model, the scoring function is obtained, denoted as $s(u_t, u_n) = u_t^T u_n / \sqrt{D}$ (D is the dimension of the input vector). Let α_{nj} represent the weight of the j-th input concerned by the n-th output. For each input vector, get the weighted average output vector h_n'':

$$h_n'' = \sum_{t=1}^{T} \alpha_{nt} h_t', \tag{19}$$

where $n, t \in [1, T]$ is the position of the output and input vector sequence. Finally, the output vector sequence $H_{ATT} = [h_1'', \ldots, h_T'']$ containing the most information is obtained, which is used as a classification feature for steganalysis.

5.5. Classify. After several neural network layers, high-level reasoning in F3SNet is done via a fully connected (FC) classifier. The classifier is shown in Figure 3. The FC layer calculates the probability that the speech sample belongs to a normal set and stego set. No matter how many FC layers are passed, it is still regarded as a linear transformation, which implements the conversion from the $P \times Q$ feature matrix to the $P \times 2$ classification result matrix. Assume that the parameters in the FC layers of our network, namely, the weights and bias terms, are denoted by W_F (size, $2 \times Q$) and b_F (size 2), respectively. Note that each batch of samples shares the same set of parameters. The output array y (size $P \times 2$) can be calculated as

$$y = \sigma\left(h_t W_F + b_F\right), \tag{20}$$

where σ is the sigmoid function.

In a nutshell, there are three reasons why F3SNet is effective for small samples and low embedding rate samples. Firstly, the embedding layer is more conducive to expressing the correlation between codewords. Secondly, and most importantly, the integration of multilayer RNN and AM facilitates the extraction of speech spatiotemporal features. Thirdly, similar to words, sentences, and paragraphs in NLP that can express information of different dimensions, more features can be extracted from the two dimensions of codewords and sequences. The following experiment can well prove the effectiveness of F3SNet.

6. Experiments and Discussions

6.1. Experimental Setup. To the best of our knowledge, there is no public database in speech steganography and steganalysis to date. Previous works used self-generated speech samples for experimentation. To facilitate the comparison of algorithm performance, we use the speech sample set published by Lin et al. on GitHub (https://github.com/fjxmlzn/NN-SM/). In this paper, we divide the original

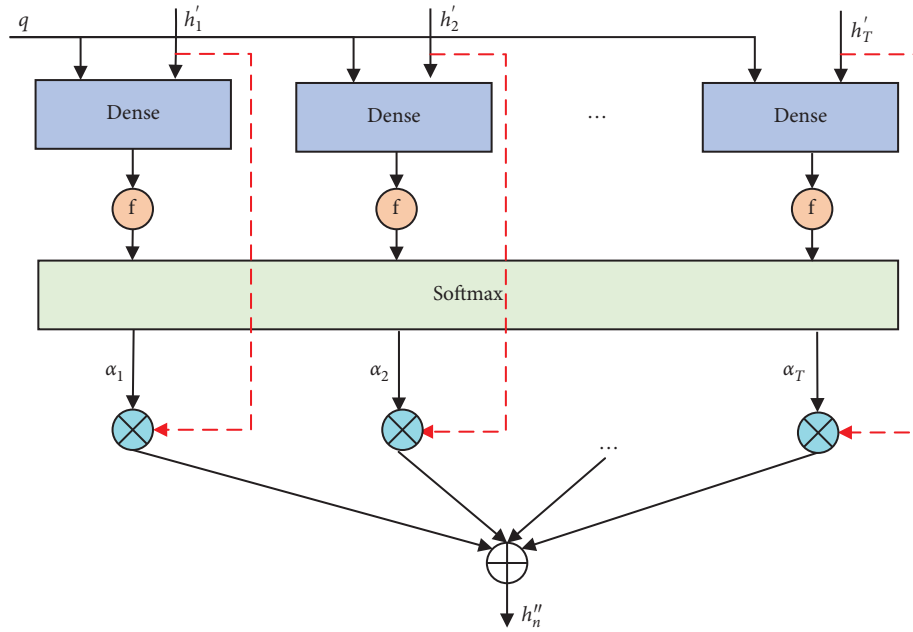


Figure 5: The attention mechanism.

samples into 5-second samples of equal length and then convert the audio into PCM format with 8 KHz sampling rate, 16 bits per sample, and stereo by Cool Edit Pro 2.1. Finally, a cover database with a total of 5120 different speech samples isestablished.

As described in Section 3.2, the steganography method was involved in the experiment, namely, CNV steganography [4]. For each sample in the cover database, several bits of randomly generated secret data are separately embedded into the cover speech. The actual number of embedded bits depends on sample length and embedding rate. At the same time, different sample lengths and different embedding rates also have a direct impact on the detection accuracy of the proposed steganalysis algorithm. Additionally, the normal signals are assigned to the negative category, and the stego samples were selected from the positive and negative categories to construct a training set and a test set, respectively. To evaluate the performance of F3SNet, three statistical indicators are used to measure the classification efficacy of F3SNet, i.e., false positive rate (FPR), false negative rate (FNR), and accuracy (ACC).

Firstly, to evaluate the effect of different sample lengths on the performance of F3SNet, we give the sample lengths of 0.1 , 0.2 s, 0.4 s, 0.6 s, 0.8 s, 1 s, 2 s, 4 s, and 5 s with 20% and 40% embedding rate, respectively. As mentioned before, many existing algorithms have good detection accuracy for large-sized samples, but they do not perform well for small-sized samples. Therefore, we focus on how well F3SNet performs for small-sized samples.

Then, to evaluate the effectiveness of F3SNet at different embedding rates, the normal signals and the stego signals with different embedding rates (ER) are grouped. Therefore, embedding rates in the experiment are chosen to be 100%, 80%, 60%, 40%, 20%, and 10%, respectively. At the same time, we focus on the performance of F3SNet for small-sized samples. The length of the sample is set to 0.2 s and 1 s in the experiment.

Thirdly, as described above, for steganography based on compressed speech, researchers have successively developed a variety of steganalysis methods. Among them, the typical algorithms are IDC [12], QCCN [13], RNN-SM [20], and FCEM [22]. Below we will compare the performance of these state-of-the-art algorithms and F3SNet using different lengths and different embedding rates.

6.2. Determining Hyperparameters of F3SNet. The hyperparameters in our model involved include the output dimension of the embedding layer, the number of LSTM hidden units, the recurrent layers of LSTM, the dropout rate, batch size, epoch, and so on. All these hyperparameters are determined by cross-validation on the training set and validation set.

For a given network model, hyperparameters such as the dimension of the embedding layer, the number of LSTM hidden unit, and the recurrent layers of LSTM are determined by cross-validation on the training set and validation set. Taking into account classification accuracy and training time, we collect a total of 102, 400 speech samples with a length of 1 s (cut from the above database) and then divide them into the training set and validation set in 7: 3 ratio. To optimize the tuning process of the model, the Adam optimizer was used for model training. The learning rate is done in the default way.

In our implementation, the programs run on a single GPU in the deep learning server, which has "Intel (*R*) Xeon (*R*) CPU E5-2620 V4 @ 2.10 GHZ ," 64 GB memory, and 4 NVIDIA GeForce GTX 2080 Ti GPUs. Moreover, the memory size and processing power of the GPU are 11 GB and 11.3 TFLOPS in double precision, respectively.

Normally, it has the ability to accommodate most of the implementation in deep learning architecture. Thus, based on the GPU server resources in our lab, the final parameters are as follows. Batch size was set to 128. The dimension of the embedding layer is 100. The dimension of word LSTM is 100. The dimension of sentence LSTM is 50. The recurrent layer of LSTM is 1. It is worth mentioning that the current parameter values are not necessarily optimal, and one may find a more balanced point of accuracy and time cost through experiments.

6.3. Comparison with Different Network Model. Different models have different learning capabilities. Generally speaking, the more complex the model, the stronger the deep learning capabilities, but the greater the resource overhead. Here, we use classification accuracy and training time as evaluation metrics to compare six types of models, as shown in Table 1. As can be seen from the above, F3SNet uses a hierarchical attention model. Models #2, #3, and #4 are variants obtained by modifying the proposed model in the paper. For example, model #2 only considers a single-layer attention structure, model #3 does not use a LSTM layer, and model #4 does not use an embedding layer. In addition, model #5 and #6 are the two deep learning models proposed before [20, 22], and both are compared here.

For the classification accuracy metric, 1 s speech samples are selected, and the embedding rate starts from 0.1 and increases at a growth rate of 10%. After 10 iterations, the maximum accuracy is plotted on the Y-axis and the embedding rate is plotted on the X-axis, as shown in Figure 6. We can find that, as the embedding rate increases, the classification accuracy of all models is significantly improved, and F3SNet is the best among all embedding rates, which shows that the model has excellentsteganography feature learning capabilities. It can be said that the embedding layer and multilayer attention mechanism make F3SNet show better performance. However, it can be seen from Figure 7 that the training time of model #1 is relatively long, which is a price that must be paid to improve accuracy. In some applications, the time overhead is an "acceptable metric" and the accuracy is a "satisficing metric." That is, the classifier is required to achieve a certain accuracy within the acceptable range of time overhead. Our model can be applied to these occasions.

6.4. Performance Testing

6.4.1. Test Results at Different Lengths. In the experiment, nine different length speech samples with 20 % and 40 % embedding rates were selected to test the validity of F3SNet under different conditions, especially for short samples. The results are listed in Table 2.

Clearly, for 0.1 s samples, our algorithm still achieves 70.12 % and 83.98 % detection rates when the embedding rates are 20 % and 40 %, respectively, which is significantly better than the state-of-art algorithms. In addition, for each fixed embedding rate, the detection accuracy is proportional to the sample length. This means that the longer the sample, the higher the detection accuracy. When the sample length is increased to 5 , the detection accuracy of the proposed algorithm corresponding to the above two embedding rates reaches 95.46 % and 99.9 %, respectively. Furthermore, it can be seen that, as the speech length gradually increases to 5 s, the detection accuracy of the algorithm under each candidate length fluctuates within a relatively small range. However, when the sample length changes from 1 s to 5 s, the detection accuracy increases more clearly. Taking the embedding rate of 40 % as an example, the sample length was increased from 0.1 s to 1 s, and the detection accuracy increased by 10.65 %. However, the sample increased from 1 s to 5 s, and the detection accuracy only increased by 5.27%.

From another angle, we can make some observations about FNR and FPR. Regardless of the embedding rate, the FNR of different lengths is significantly greater than the FPR. This shows that the missed detection rate is higher than the false alarm rate in our detection algorithm. Therefore, the algorithm is suitable for some application environments that do not require high missing detection rates, such as online real-time detection.

6.4.2. Test Results at Different Embedding Rates. This experiment evaluates the performance of F3SNet with fixed length and different embedding rates. The results are shown in Table 3.

From the experimental results above, we can find that there is a positive relationship between the detection accuracy rate and embedding rate (ER in Table 3). For samples with a length of 0.2 s, when the embedding rate is 10 %, the detection accuracy is 62.3 %, and as the embedding rate rises to 40 %, the detection accuracy is up to 87.11 %. Finally, the detection accuracy ends up at 98.88 % under 100 % embedding rate.

At the same time, for fixed-length samples, when the embedding rate is low, the embedding rate increases by a certain percentage, and the accuracy rate increases accordingly. However, when the embedding increases to a certain value, the increase in accuracy is not significant. Similarly, for a 0.2-second sample, the embedding rate ranges from 20% to 100%, each time increasing by 20 %, and the ratios of the increase in detection accuracy are12.4%, 7.27%, 2.84%, and 1.66%, respectively. In addition, two conclusions can be drawn from the horizontal comparison of different sample lengths. First, the longer the sample, the higher the detection rate. Second, when the embedding rate is lower, the sample length increases by a certain value and the detection accuracy increases more significantly.

6.4.3. Comparison with Existing Algorithms. We focus on comparing the detection accuracy of various algorithms for different sample lengths (0.2 s, 0.4 s, 0.6 s, 0.8 s, 1 s, and 2 s) with embedding rates 20%, 40%, and 60%, respectively. The results are shown in Figures 8–10 . Comparing, we conclude that, as the sample length increases, the detection accuracy of all algorithms participating in the comparison keeps increasing, and FNR and FPR keep decreasing, despite occasional fluctuations. In addition, according to the

Table 1: Experiment with different types of network models.

Number	Network model	Hyperparameters
Model #1	F3SNet	The dimension of embedding layer = 100, the number of word LSTM hidden unit = 100, the number of sentence LSTM hidden unit = 50, dropout = 0.5, dropout_recurrent = 0.5, batch size = 128, and epoch = 50
Model #2	Embedding + LSTM + Self_Attention + Dense	The dimension of embedding layer = 100, the number of LSTM hidden unit = 100, dropout = 0.5, dropout_recurrent = 0.5, batch size = 128, and epoch = 50
Model #3	Embedding + Self_Attention + Self_Attention + Dense	The dimension of embedding layer = 100, dropout = 0.5, batchsize = 128, epoch = 50.
Model #4	LSTM + Self_Attention + BiLSTM + Self_Attention + Dense	The number of word LSTM hidden unit = 100, the number of sentence LSTM hidden unit = 100, dropout = 0.5, dropout_recurrent = 0.5, batch size = 128, and epoch = 50
Model #5	Embedding + Multi-head Attention + Dense ([22])	The dimension of embedding layer = 100, heads = 8, head_size = 32, dropout = 0.5, batchsize = 128, epoch = 50.
Model #6	LSTM + LSTM + Dense ([20])	The number of the first LSTM hidden unit = 50, the number of the second LSTM hidden unit = 50, batch size = 128, and epoch = 50

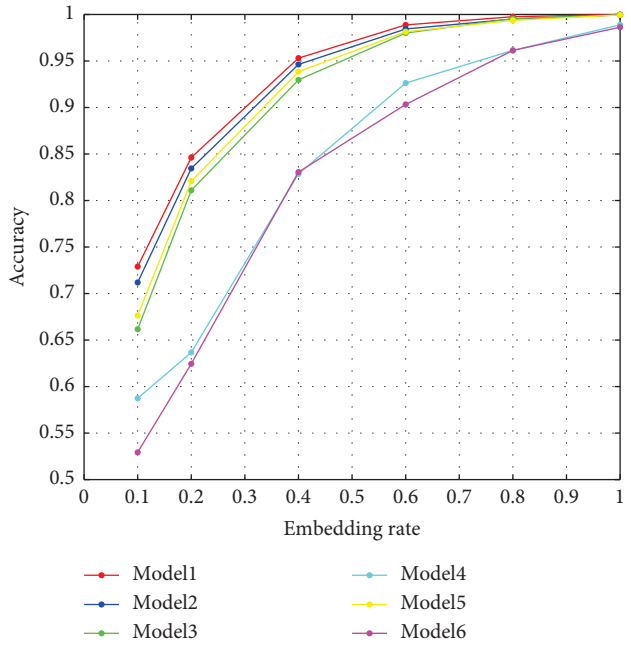


Figure 6: The accuracy of different models.

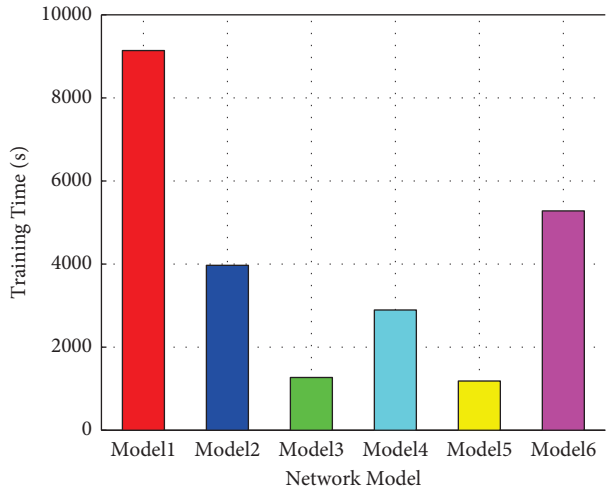


Figure 7: The time cost of different models.

Table 2: Detection results for different length samples (embedding rate: 20 % and 40 %).

Embedding rate (%)	Sample length (s)	ACC (%)	FNR (%)	FPR (%)
20	0.1	70.12	47.328	12.266
	0.2	74.71	37.451	13.417
	0.4	76.46	32.393	14.608
	0.6	80.18	25.911	15.306
	0.8	81.59	21.816	14.694
	1.0	83.45	16.488	16.618
	2.0	90.58	11.868	6.961
	4.0	94.63	4.485	6.3
	5.0	95.46	5.769	3.274
40	0.1	83.98	25.882	6.225
	0.2	87.11	17.083	8.549
	0.4	90.53	11.874	7.094
	0.6	92.48	8.847	6.238
	0.8	95.07	5.058	4.804
	1.0	94.63	4.762	5.962
	2.0	98.14	2.649	1.069
	4.0	99.66	0.294	0.390
	5.0	99.90	0	0.194

Table 3: Detection results under different embedding rates (sample length: 0.2 s and 1 s).

Sample length (s)	Embedding rate (%)	ACC (%)	FNR (%)	FPR (%)
0.2	10	62.30	58.763	13.666
	20	74.71	37.451	13.417
	40	87.11	17.083	8.549
	60	94.38	7.892	3.271
	80	97.22	3.783	1.770
	100	98.88	1.130	1.116
1	10	71.19	33.845	23.582
	20	83.45	16.488	16.618
	40	94.63	4.762	5.962
	60	98.44	1.760	1.366
	80	99.51	0.869	0.098
	100	99.95	0.095	0

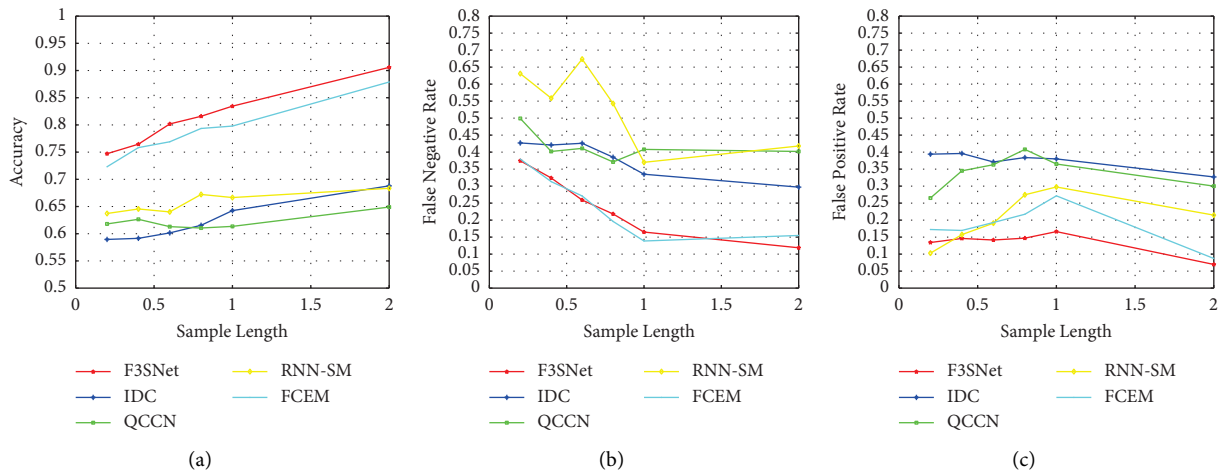


Figure 8: Performance comparison under 20% embedding rate. (a) ACC under different sample lengths. (b) FPR under different sample lengths. (c) FNR under different sample lengths.

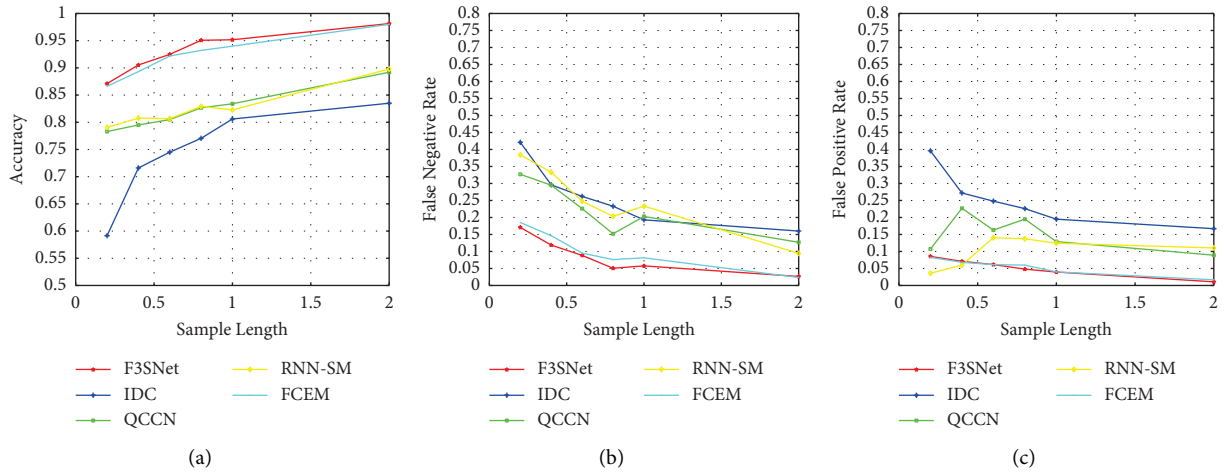


Figure 9: Performance comparison under 40% embedding rate. (a) ACC under different sample lengths. (b) FPR under different sample lengths. (c) FNR under different sample lengths.

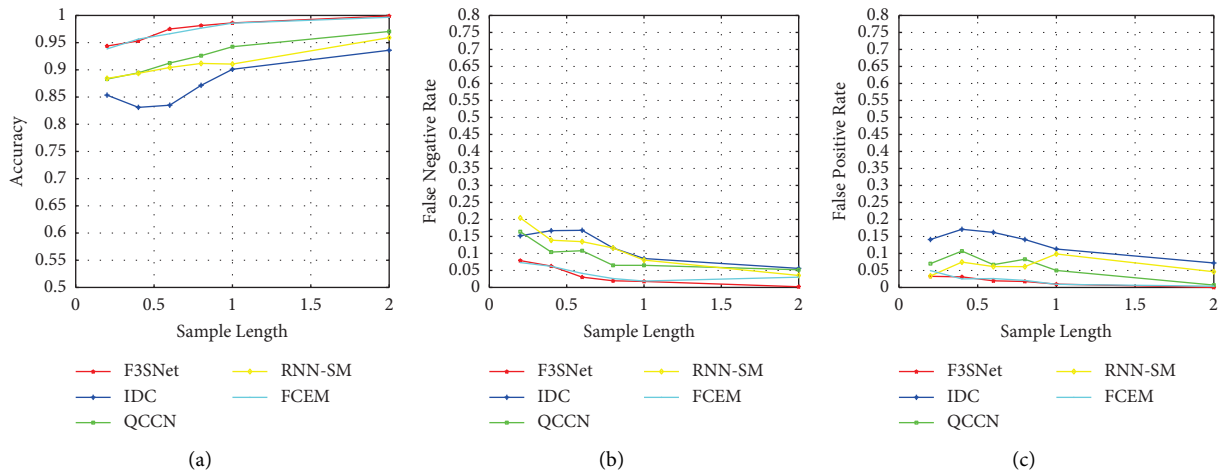


Figure 10: Performance comparison under 60% embedding rate. (a) ACC under different sample lengths. (b) FPR under different sample lengths. (c) FNR under different sample lengths.

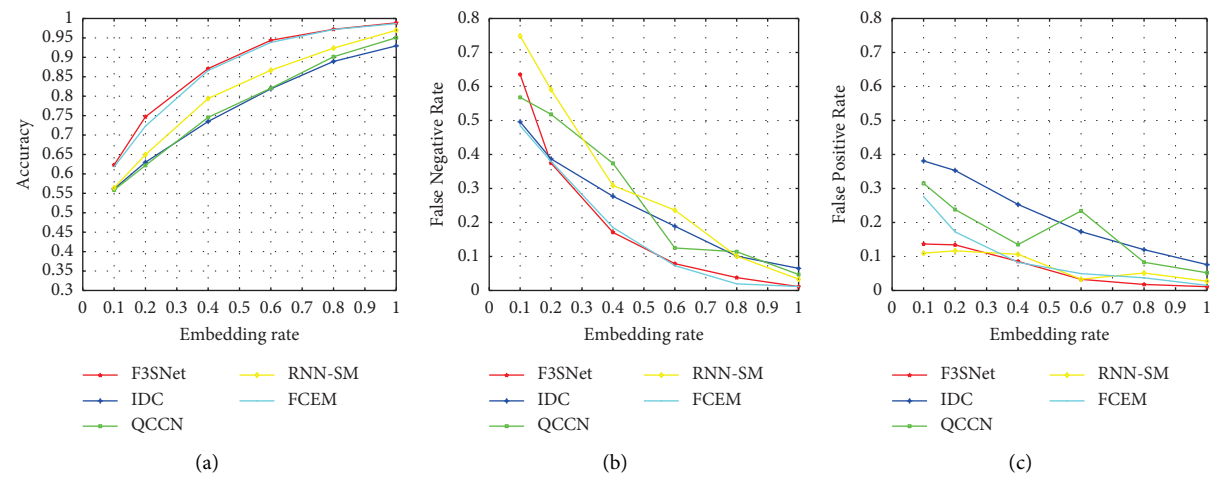


Figure 11: Performance comparison for 0.2 s samples. (a) ACC under different embedding rates. (b) FPR under different embedding rates. (c) FNR under different embedding rates.

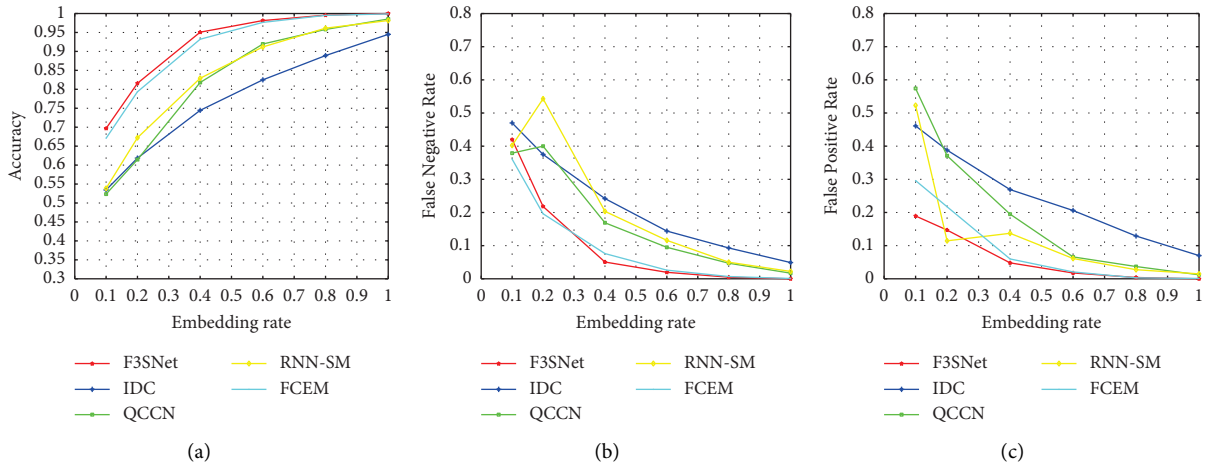


Figure 12: Performance comparison for 0.8 s samples. (a) ACC under different embedding rates. (b) FPR under different embedding rates. (c) FNR under different embedding rates.

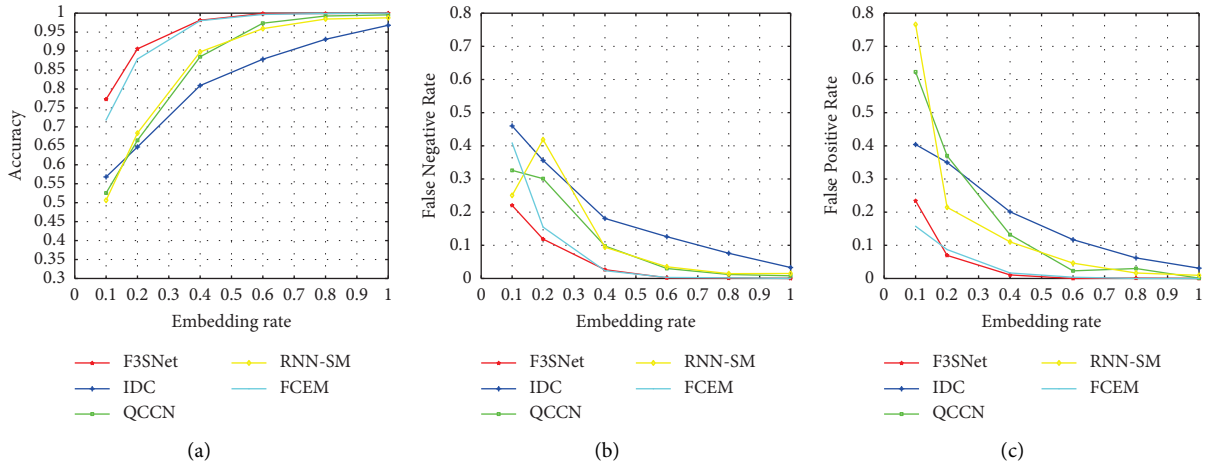


Figure 13: Performance comparison for 2 s samples. (a) ACC under different embedding rates. (b) FPR under different embedding rates. (c) FNR under different embedding rates.

performance distribution curve in Figure 8, the five types of algorithms can be divided into three different performance ranges. The detection algorithms IDC and QCCN based on traditional machine learning have poor performance, RNN-SM is in the middle, and FCEM and F3SNet have the best performance. And, among all the algorithms, the performance of F3SNet has obvious advantages. On average, F3SNet leads RNN-SM by about 15.41 % and FCEM by about 2.48%. Furthermore, from the longitudinal comparison of the three graphs, two conclusions can be drawn. Firstly, in the case of 20% embedding rate, ACC, FPR, and FPR fluctuate significantly, indicating that the detection efficiency is low at this time and it is susceptible to noise. Secondly, when the sample length is fixed, the higher the embedding rate, the higher the detection accuracy and the lower the FPR and FNR. For example, with a fixed length of 0.2 s, when the embedding rate is 20%, the accuracy of F3SNet is about 74%. If the embedding rate is increased to 40%, the detection accuracy will increase to 87%.

In addition, to further evaluate the performance of F3SNet, the detection accuracy of different algorithms under different embedding rates (10%, 20%, 40%, 60%, 80%, and 100%) is tested. Here, we select three samples with lengths of 0.2 s, 0.8 s, and 2 sseparately for the experiment. The results are presented in Figures 11–13. We can see that, as the embedding rate increases, the detection accuracy of all algorithms is increasing, and F3SNet has the best performance among all algorithms. Taking 2 s as an example, when the embedding rate is 20 %, the detection accuracy of F3SNet can reach 90.58%. In contrast, the other algorithms are 64.7%, 66.45%, 68.35%, and 87.89%, respectively. Besides, IDC, QCCN, and RNN-SM can hardly obtain effective detection.

7. Conclusion and Future Work

In this paper, we mainly focus on how to use the hierarchical attention network to detect the disparities in the correlation of LPC coefficients before and after steganography. First, to demonstrate the existence and complexity of the correlation, we performed Bayesian network modeling on the quantized codeword index and then calculated the link strength between different nodes as a measure of the strength of the codewords' correlation. Then, we propose a four-step strategy for QIM steganalysis based on HAN, which can automatically extract the features reflecting the correlation.

In the proposed model, the LSTM layer and the attention layer are two core components. The former considers possible dependencies in the codebook structure because of its memory properties in time series, and the latter further determines which vectors have a greater impact on the final classification result, thereby effectively avoiding information overload. Experimental results showed that even for speech with a length of 1 s, F3SNet could effectively detect QIM steganography under an embedding rate of 10% and outperforms FCEM by about 5.27%.

It must be noted that F3SNet currently can only detect QIM steganography. A future research suggestion would be extending the method to detect other steganography with compressed speech.

Data Availability

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

Disclosure

An early version of our paper has been published as a preprint on the arxiv website at https://arxiv.org/abs/2101.05105.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this work.

References

[1] W. Mazurczyk, "Voip steganography and its detection—a survey," *ACM Computing Surveys*, vol. 46, no. 2, p. 20, 2013.

[2] E. Zielinska, W. Mazurczyk, and K. Szczypiorski, "Trends in steganography," *Communications of the ACM*, vol. 57, no. 3, pp. 86–95, 2014.

[3] H. Ghasemzadeh and M. H. Kayvanrad, "Comprehensive review of audio steganalysis methods," *IET Signal Processing*, vol. 12, no. 6, pp. 673–687, 2018.

[4] B. Xiao, Y. Huang, and S. Tang, "An approach to information hiding in low bit-rate speech stream," in *Proceedings of the IEEE Globecom 2008 - 2008 IEEE Global Telecommunications Conference*, pp. 1–5, New Orleans, LA, USA, December 2008.

[5] J. Liu, H. Tian, J. Lu, and Y. Chen, "Neighbor-index-division steganography based on QIM method for G.723.1 speech streams," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 1, pp. 139–147, 2016.

[6] P. Liu, S. Li, and H. Wang, "Steganography integrated into linear predictive coding for low bit-rate speech codec," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2837–2859, 2017.

[7] B. Geiser and P. Vary, "High rate data hiding in acelp speech codecs," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4005–4008, Las Vegas, NV, USA, 4 April 2008.

[8] H. Miao, L. Huang, Z. Chen, W. Yang, and A. Al-Hawbani, "A new scheme for covert communication via 3G encoded speech," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1490–1501, 2012.

[9] W. Zhijun and S. Yongpeng, "An implementation of speech steganography for ILBC by using fixed codebook," in *Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1970–1974, Chengdu, China, 17 Oct. 2016.

[10] Y. Huang, C. Liu, S. Tang, and S. Bai, "Steganography integration into a low-bit rate speech codec," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1865–1875, 2012.

[11] C. Gong, X. Yi, and X. Zhao, "Pitch delay based adaptive steganography for amr speech stream," in *Proceedings of the International Workshop on Digital Watermarking*, pp. 275–289, Springer, Jeju Island, Korea, 24 Jan. 2019.

[12] S.-b. Li, H.-z. Tao, and Y.-f. Huang, "Detection of quantization index modulation steganography in G.723.1 bit stream based on quantization index sequence analysis," *Journal of Zhejiang University - Science C*, vol. 13, no. 8, pp. 624–634, 2012.

[13] S. Li, Y. Jia, and C.-C. J. Kuo, "Steganalysis of qim steganography in low-bit-rate speech signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1011–1022, 2017.

[14] H. Miao, L. Huang, Y. Shen, X. Lu, and Z. Chen, "Steganalysis of compressed speech based on markov and entropy," in *Proceedings of the International Workshop on Digital Watermarking*, pp. 63–76, Springer, Taipei, Taiwan, 09 July 2014.

[15] Y. Yanzhen Ren, T. Tingting Cai, M. Ming Tang, and L. Lina Wang, "Amr steganalysis based on the probability of same pulse position," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1801–1811, 2015.

[16] Q. Qingzhong Liu, A. H. Sung, and M. Mengyu Qiao, "Temporal derivative-based spectrum and mel-cepstrum audio steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 359–368, 2009.

[17] S.-B. Li, Y.-Z. Jia, J. Y. Fu, and Q.-X. Dai, "Detection of pitch modulation information hiding based on codebook correlation network," *Chinese Journal of Computers*, vol. 37, no. 10, pp. 2107–2116, 2014.

[18] H. Zhou, K. Chen, W. Zhang, Y. Yao, and N. Yu, "Distortion design for secure adaptive 3-d mesh steganography," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1384–1398, 2019.

[19] H. Tian, Y. Wu, C.-C. Chang et al., "Steganalysis of adaptive multi-rate speech using statistical characteristics of pulse pairs," *Signal Processing*, vol. 134, pp. 9–22, 2017.

[20] Z. Lin, Y. Huang, and J. Wang, "Rnn-sm: Fast steganalysis of voip streams using recurrent neural network," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1854–1868, 2018.

[21] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

Information Processing Systems 2017, pp. 5998–6008, Long Beach, CA, USA, December 4-9 2017.

[22] H. Yang, Z. Yang, Y. Bao, S. Liu, and Y. Huang, "FCEM: A novel fast correlation extract model for real time steganalysis of voip stream via multi-head attention," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2822–2826, ICASSP, Barcelona, Spain, May 4-8, 2020.

[23] Q. Ding and X. Ping, "Steganalysis of compressed speech based on histogram features," in *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, IEEE*, pp. 1–4, Chengdu, China, 25 Sept. 2010.

[24] Y. F. Huang, Y. Zhang, and S. Tang, "Detection of covert voice-over internet protocol communications using sliding window-based steganalysis," *IET Communications*, vol. 5, no. 7, pp. 929–936, 2011.

[25] 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*, p. 1, 2020.

[26] W. Wen-Nung Lie and G. Guo-Shiang Lin, "A feature-based classification technique for blind image steganalysis," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1007–1020, 2005.

[27] S. Wu, S.-h. Zhong, and Y. Liu, "A novel convolutional neural network for image steganalysis with shared normalization," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 256–270, 2020.

[28] S. Li, Y. Huang, and J. Lu, "Detection of qim steganography in low bit-rate speech codec based on statistical models and svm," *Chinese Journal of Computers*, vol. 36, no. 6, pp. 1168–1176, 2013.

[29] B. Xiao, J. Luo, X. Bi, W. Li, and B. Chen, "Fractional discrete tchebyshev moments and their applications in image encryption and watermarking," *Information Sciences*, vol. 516, pp. 545–559, 2020.

[30] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Media Watermarking, Security, and Forensics 2015*, vol. 9409, Article ID 94090J, 2015.

[31] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[32] G. Xu, H.-Z. Wu, and Y. Q. Shi, "Ensemble of CNNs for Steganalysis," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2016*, pp. 103–107, Vigo, Spain, June 20-22, 2016.

[33] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[34] C. Paulin, S.-A. Selouani, and É. Hervet, "Audio steganalysis using deep belief networks," *International Journal of Speech Technology*, vol. 19, no. 3, pp. 585–591, 2016.

[35] B. Chen, W. Luo, and H. Li, "Audio steganalysis with convolutional neural network," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, ACM*, pp. 85–90, New York, NY USA, 20 June 2017.

[36] C. Gong, X. Yi, X. Zhao, and Y. Ma, "Recurrent convolutional neural networks for AMR steganalysis based on pulse position," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec*, pp. 2–13, Paris, France, July 3-5, 2019.

[37] H. Yang, Z. Yang, Y. Bao, and Y. Huang, "Hierarchical representation network for steganalysis of QIM steganography in low-bit-rate speech signals," in *Proceedings of the Information and Communications Security - 21st International Conference, ICICS*, pp. 783–798, Beijing, China, December 15-17.

[38] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[39] I. Ebert-Uphoff, "Measuring connection strengths and link strengths in discrete bayesian networks," Tech. rep., Georgia Institute of Technology, Atlanta, Georgia, 2007.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5524930, 13 pages
https://doi.org/10.1155/2021/5524930

Research Article

Channel-Wise Spatiotemporal Aggregation Technology for Face Video Forensics

Yujiang Lu,[1] **Yaju Liu**,[2] **Jianwei Fei**,[2] **and Zhihua Xia**[3]

[1]*Changwang School of Honors, Nanjing University of Information Science Technology, Nanjing 210044, China*
[2]*School of Computer and Software, Nanjing University of Information Science Technology, Nanjing 210044, China*
[3]*Engineering Research Center of Digital Forensics, Ministry of Education, School of Computer and Software, Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science Technology, Nanjing 210044, China*

Correspondence should be addressed to Zhihua Xia; xia_zhihua@163.com

Received 27 February 2021; Revised 30 June 2021; Accepted 14 August 2021; Published 29 August 2021

Academic Editor: Guoying Zhao

Recent progress in deep learning, in particular the generative models, makes it easier to synthesize sophisticated forged faces in videos, leading to severe threats on social media about personal privacy and reputation. It is therefore highly necessary to develop forensics approaches to distinguish those forged videos from the authentic. Existing works are absorbed in exploring frame-level cues but insufficient in leveraging affluent temporal information. Although some approaches identify forgeries from the perspective of motion inconsistency, there is so far not a promising spatiotemporal feature fusion strategy. Towards this end, we propose the Channel-Wise Spatiotemporal Aggregation (CWSA) module to fuse deep features of continuous video frames without any recurrent units. Our approach starts by cropping the face region with some background remained, which transforms the learning objective from manipulations to the difference between pristine and manipulated pixels. A deep convolutional neural network (CNN) with *skip connections* that are conducive to the preservation of detection-helpful low-level features is then utilized to extract frame-level features. The CWSA module finally makes the real or fake decision by aggregating deep features of the frame sequence. Evaluation against a list of large facial video manipulation benchmarks has illustrated its effectiveness. On all three datasets, FaceForensics++, Celeb-DF, and DeepFake Detection Challenge Preview, the proposed approach outperforms the state-of-the-art methods with significant advantages.

1. Introduction

The rapid development of social networks and the emergence of various mobile applications have promoted the creation and dissemination of digital videos. These videos generally contain rich contents of individuals with regard to face and voice, which are very significant biological information for identity authentication. However, manipulation of these videos will seriously undermine their authenticity. Due to the ever-developing artificial intelligence technologies, existing tools make manipulation easier than ever and more imperceptible. Meantime, the convenient creation and spread of multimedia contents make it uncomplicated for an attacker to obtain their desired material and carry out malicious purposes by these tools. This has become a potential threat to ethics, law, and personal privacy and raised a great alarm. It is therefore of great practical significance to study effective forensics technologies to distinguish these fake videos. However, facial manipulation did not attract too much attention before because the conventional digital image editing methods are easy to spot by naked eyes, and the forensics technologies have been at an advantage until the appearance of deep learning based forgery technologies.

However, in recent years, deep learning based face synthesis, manipulation, and swap technologies which are generally referred to as the term DeepFakes have brought new challenges to face forensics. The original DeepFakes can only swap two faces using a pair of autoencoders that share

the same encoder but is composed of different decoders. They are trained to reconstruct the source and target face images, respectively. Once trained, the target decoder can generate a realistic face image of target identity with the expressions of the source face by being fed with the source face representation output from the source encoder.

Original DeepFakes always produces obvious artifacts when warping faces back to the target images, and this defect has been utilized by the existing approach [1]. In recent years, the continuous development of generative networks can generate very photo-realistic fake faces or completely synthesize videos from a single image and even from portrait paintings [2]. This puts forward higher requirements for forensics approaches in terms of detection accuracy and generalization ability. The forensics approaches have also been developing with the help of deep learning and previous work in digital forensics. According to the clues used, the detection approaches of face video manipulation can be mainly divided into two: intraframe information based and interframe information based. The former focuses on spatial artifacts and realizes video manipulation detection by processing independent frames. The latter captures the dynamic flaws in videos through temporal models like Recurrent Neural Network (RNN) [3] or optical flow [4].

In this paper, we adopt a novel approach to capture the interframe cues by aggregating deep feature sequences channel wisely. It achieves better performance with relatively few parameters. The main contributions of this paper are summarized as follows:

A novel module CWSA is proposed to exploit temporal information by aggregating deep features of consecutive frames but different channels. With a powerful feature extraction backbone EfficientNet B0 [5], our approach reaches the state-of-the-art level on three large datasets.

It is revealed that by keeping the moderate background in face cropping preprocessing, models can learn the difference between pristine and manipulated pixels to obtain gains in detection accuracy.

We demonstrate that *skip connection* preserves the detection-helpful low-level features well. Thus it plays a central role when deep models are used for extracting frame-level features.

This paper is organized as follows. In Section 2, we briefly introduce the existing forensics approaches. In Section 3, we give a detailed description of our approach. The experimental results and analysis are presented in Section 4, and we make a conclusion and prospects for future work in Section 5.

2. Related Work

2.1. Manipulation Forensics. Before the emergence of deep learning based forgery technologies, conventional multimedia contents manipulation such as removal, copy-move, and splicing were realized with image editing technologies. The research of multimedia forensics has been committed to solving the problems of detecting this kind of manipulation for long. These manipulations tend to leave obvious clues, particularly in statistical characteristics caused by editing or compression. Considering this, Cozzolino et al. have proposed a feature-based splicing detection method. Their algorithm computes local features from the cooccurrence matrix of the image residuals, and the parameters extracted from different images were proved to be efficacious on both detection and localization [6]. Similarly, the study in [7] discovered the influence of times of JPEG compression that images go through. With the help of the Nonnegative Matrix Factorization model and histograms of Discrete Cosine Transform, multiple JPEG compression can be successfully detected and indirectly, the authenticity of images.

Another kind of popular approach is to discover clues that are related to the camera itself. In 2006, Lukas et al. proposed to identity camera models through photoresponse nonuniformity, a pattern that reveals the different sensitivity of pixels to light caused by the inhomogeneity of silicon wafers [8]. Researchers also found out camera-related patterns left in out-camera processing history. In [9], Cozzolino et al. have researched to detect and localize forgeries by a camera-based noise pattern. This noise pattern is produced during the compression or gamma correction and can be seen as unique fingerprints of specific camera models. However, the estimation of this noise requires a considerable number of samples, and when encountered with an unknown camera model, detection approaches based on noise pattern would show weakness.

2.2. GAN Forensics. Using the Adversarial Generative Network (GAN), many fake images or videos are completely generated instead of manipulated. This somehow reduces the performance of earlier detection approaches. Inspired by the camera fingerprints, recent researches try to analyze the fingerprints in generated images and explore the feasibility of attributing fake images to a GAN with certain architecture. Moreover, Zhang et al. proposed an AutoGAN to simulate artifacts produced by common GANs and detect GAN-generated images using spectrum features [10]. In [11], Cozzolino et al. attempted to spoof a smart pretrained embedder which is originally used to distinguish camera traces in capturing images. Their work revealed the vulnerabilities of current approaches. Durall et al. also investigated the artifacts left out of visual content. They analyzed the differences in the classical frequency domain and constructed 1D power spectrum statistics. Using this feature, a simple binary classifier trained with few annotated samples can achieve good performance [12]. Color abnormality is also a strong hint for GAN-generated content. In [13], McCloskey et al. demonstrated that GAN generators may leak some clues when converting feature representations to red, green, and blue pixels. Li et al. analyzed the difference between pristine and generated images in HSV, YCbCr, and RGB color spaces, and a statistical feature set was proposed to characterize the difference [14]. More directly, Nataraj et al. trained their CNN detector on cooccurrence matrices extracted from the RGB channels in the space domain and achieved competitive performance as well [15].

2.3. DeepFakes Forensics. Recently, many novel deep learning based technologies have also shown astonishing performance in face synthesis, among which the most famous is DeepFakes. Along with the continuous development of DeepFakes, the corresponding forensics technologies are also being researched. Similar to previous studies, early work mainly focused on detecting visual artifacts. Li et al. simulated the DeepFakes artifacts by Gaussian blur and affine warpage, and their evaluations indicated the simulated artifacts can make CNN detectors more robust [1]. Some other work focuses on dynamic defects in the temporal domain. In the pipeline of [3], a CNN is used as a spatial feature extraction backbone, and an RNN is connected to the backbone, aggregating the CNN outputs over time and makes final classifications. Zhou et al. aggregated short-term, long-term, and global statistics to characterize the relations among different face regions. Their evaluations indicated these relations, especially the temporal order within the tracklet, are informative for recognizing temporal inconsistency in manipulated face sequences [16]. Actually, most dynamic artifacts based detection approaches utilize a CNN backbone to firstly extract features of every single frame.

Facial expression habits are unique from person to person and are extremely hard to simulate. Therefore, DeepFakes may leave traces in respect to personality behavior habits and sometimes even the physical law of motions or illuminations. For example, by modeling the face and head movements as the unique speaking pattern of a specific individual, the high prediction error can be a strong hint of fake. Biological signals such as eye blinking and pulse are also discriminating cues to expose DeepFakes. Li et al. observed that the regular eye blinking cannot be realized in the synthesized videos, and they proposed a CNN and Long Short-Term Memory (LSTM) joint architecture to expose DeepFakes by predicting the eye blinking [17]. By the noncontact heart rate detection technology, it is easy to detect whether there is a regular heart rate in videos and identify the video authenticity. Similarly, Fernandes et al. proposed to estimate the heart rates in DeepFakes videos by Neural-ODE trained with normalized heart rate [18]. Due to insufficient datasets, the research on DeepFakes detection was seriously hindered in early time. To promote the research of DeepFakes detection, many large-scale datasets are made and open-sourced. Rossler et al. introduced a large facial manipulation dataset with 4k forged videos named Faceforensics++ created by four different approaches [19]. Recently, Facebook released a database containing 19154 pristine and 100K forged videos for the DeepFakes Detection Challenge (DFDC). There are various background conditions and manipulation approaches that are great challenges for detection approaches [20]. Li et al. proposed a new large scale benchmark named Celeb-DF that contains 5639 sophisticated DeepFakes videos [21]. Though some existing methods can expose fake videos, they generally make the video-level real/fake classification by fusing the predictions of several frames. This does not actually leverage the features of consecutive frames and leaves some room for fake video detection. To end this, we propose the CWSA module to accurately capture the temporal cues by fusing deep features of consecutive frames.

3. The Channel-Wise Spatiotemporal Aggregation

This section presents details of our proposed approach. Given a face patch sequence, the weights-sharing backbone extracts deep features of each patch. The proposed CWSA module then recombines the feature maps into a new feature sequence which is then compressed to a vector and connected to a single neural unit for real or fake classification. The complete pipeline of our approach is shown in Figure 1.

We propose a simple but effective module CWSA as in Figure 2. The proposed module is easy to cooperate with a backbone and serves as CWSA Net for face video forensics. Specifically, given a deep feature sequence of successive frames produced by the backbone, although it is unknown to us about the semantics of a specific channel, we hypothesis that feature maps of the same channels but different frames contain dynamic information as successive frames do. By stacking the feature maps of different frames with the same channel and carrying out further feature extraction separately, we can capture both frame-level artifacts and more refined interframe defects.

For every input frame, the backbone produces a feature map of size $F \in \mathbb{R}^{H\times W\times C}$, where H and W denote the resolutions and C denotes the channels. For a video clip that contains N successive frames, the weights-sharing backbone generates a set of feature maps of size $F' \in \mathbb{R}^{N\times H\times W\times C}$. Our approach firstly decomposes F' into base feature map $f_{n=1,c=1}^{N,C} \in \mathbb{R}^{H\times W}$ and recombines them by going through N and stacking f that has the equal c:

$$F_{new} = \left[f_{n=1,*}, f_{n=2,*}, \ldots, f_{n=N,*}\right]. \quad (1)$$

where $[\cdot]$ denotes channel-wise stacking. As in Figure 2, we finally get a new feature set with size $C \times F_{new} \in \mathbb{R}^{H\times W\times N}$, and in this paper, C equals to 1280 as we use EfficientNet B0 and H and W are both equal to 7.

The following up layers that deal with F_{new} are all weights-sharing, i.e., repeated C times to reduce the number of parameters. Batch Normalization is the first layer to avoid internal covariate shift that may seriously hindrance the training. Next, are convolution and LeakyReLU blocks with 128, 64, and 1 kernels with no downsampling and padding. A single feature map will happen to be converted to a single element, and regardless of the length of the input sequence, we will get a feature vector of size $i^{C\times 1}$. A single neural with sigmoid activation is connected to it and makes the classification fake or real. The pipeline of the proposed CWSA is summarized in Algorithm 1.

4. Evaluation and Discussion

4.1. Experimental Settings

4.1.1. Datasets and Preprocessing. In this work, we have carried out evaluations on a list of large scale fake face video

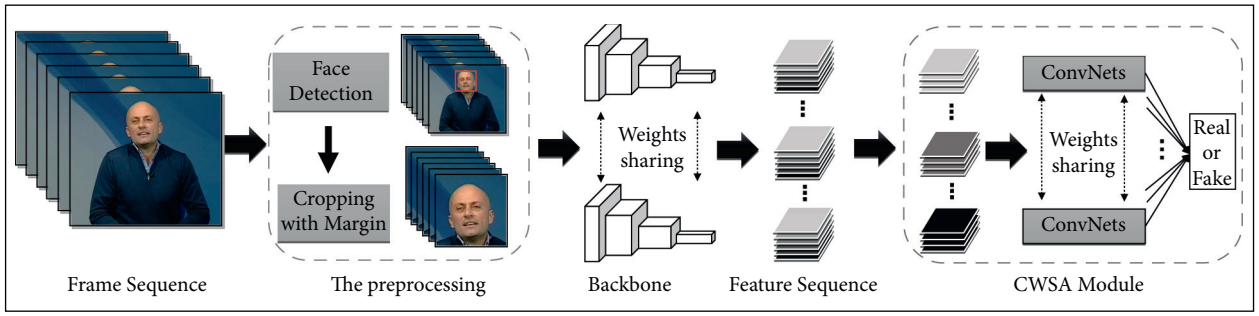


Figure 1: The proposed video forensics approach.

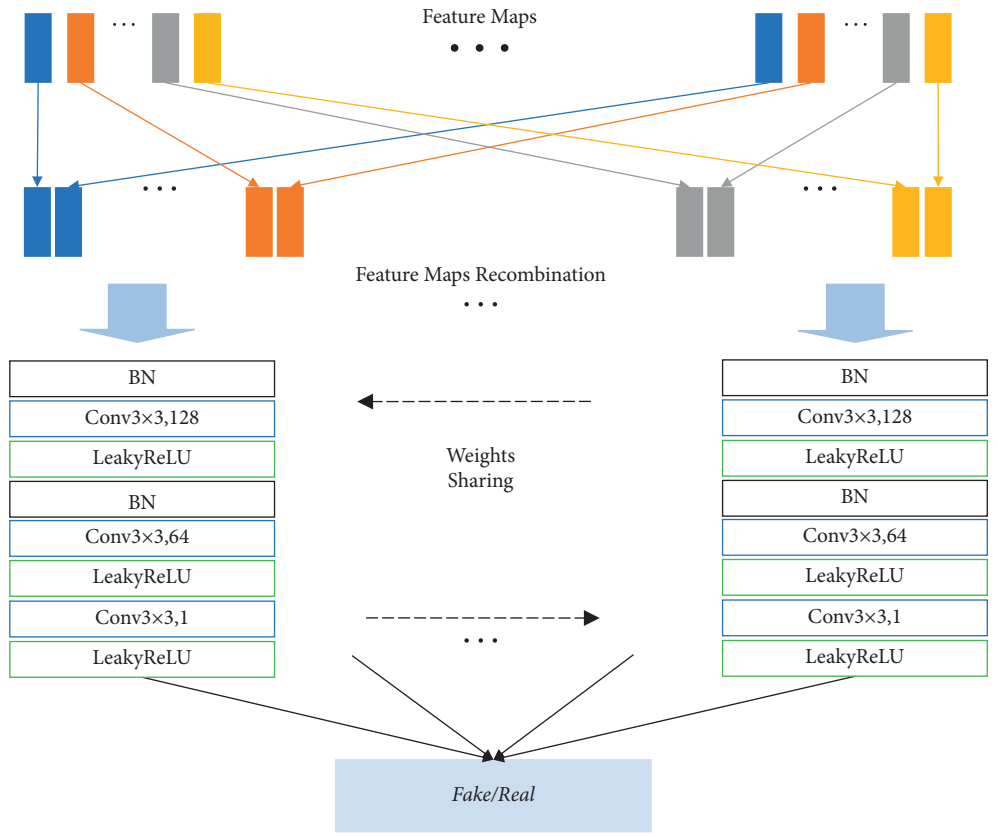


Figure 2: The architecture of the CWSA module; a colorful rectangle denotes a feature map output from the backbone.

Require: k Training face video clips $V_1, V_2, \ldots, V_k$; Corresponding label $y_1, y_2, \ldots, y_N$.
1: **for** each $i \in k$ **do**
2: Decompose V_i into the sequence of n frames $V_i^1, V_i^2, \ldots, V_i^n$;
3: Detect and crop faces frames from $V_i^1, V_i^2, \ldots, V_i^n$, then denote them as $V'^1_i, V'^2_i, \ldots, V'^n_i$;
4: **end for**
5: Feed $V'^1_i, V'^2_i, \ldots, V'^n_i$ into the backbone, producing a set of feature maps $F' \in \mathbb{R}^{N\times H\times W\times C}$;
6: Decompose F' into $f^{N,C}_{n=1,c=1} \in \mathbb{R}^{H\times W}$;
7: Combine f by going through N and stacking f that has the equal c, producing $C \times F_{new} \in \mathbb{R}^{H\times W\times N}$;
8: Feed F_{new} into weights-sharing classifier, producing y^{pred};
9: Calculating binary classification error between y and y^{pred};
10: Update the parameters of the model by back propagation;
Ensure: Optimal model for fake face detection

Algorithm 1: The CWSA algorithm.

datasets: FaceForensics++ [19], Celeb-DF [21], and DFDC Preview [20].

FaceForensics++ consists of 1000 pristine and 4000 forged videos evenly created by four different forgery methods: DeepFakes, Face2Face, FaceSwap, and NeuralTextures. In the following parts, we refer FaceForensics++ as FF++ and its subsets as DF, F2F, FS, and NT for simplicity.

Celeb-DF includes 590 pristine and 5639 forged videos created by advance DeepFakes technology. The source videos are publicly available YouTube video clips, including 59 celebrities of different genders, ages, and races. In this work, we use the second version of this dataset which contains another 300 pristine videos from YouTube.

Facebook DeepFake Detection Challenge Preview (DFDC-P) is the early release for this competition, which composes 1131 pristine of 66 actors, and 4113 forged videos created by two face synthesis algorithms.

Table 1 lists some more basic information about total frame numbers and video sizes.

Because the face region only makes up a tiny proportion in videos, it is necessary to crop off the face patches to reduce interference of redundant backgrounds and computation cost. Thus, we design a novel face cropping strategy which is proved to be beneficial for fake detection.

In the stage of preprocessing, we first detect faces in videos and then carry out face cropping, which raises a question about the optimal cropping strategy. In our earlier work, we naturally held that the characteristic of forged pixels is what a CNN mainly learns. In this case, we only have to crop off faces according to the results of face detection, and no more operations are required.

However, the evaluations reveal that by feeding inputs that include both pristine and forged pixels, deep CNNs can learn more about their difference. That is, networks may be benefited from this kind of input by detecting its global consistency. To validate if remaining some pristine pixels could help us with detection, we further evaluate two additional face cropping strategies for comparison:

(1) Crop off a convey hull of the detected face along with the key points of facial contour. The pixel density of other regions is set as 0.

(2) Crop off a minimal square that encloses the detected face with no extra margin.

(3) Crop off a minimal square that encloses the detected face and expand it by a factor of 1.4.

Samples of all three face cropping strategies are shown in Figure 3. Table 2 presents the performance between different face cropping strategies of image-level classification accuracy on EfficientNet B0. Obviously, by retaining more pristine pixels, the network is able to make some gain in accuracy. This is consistent on different datasets.

In order to verify the effect of the extended clipping factor on feature extraction, we add a simple ablation experiment based on EfficientNet B0. The experimental results on NT, which is a subset of FF++ and is a highly compressed version, are shown in Table 3. Obviously, the larger the margin is, the more the detection accuracy is, but the gain stops when expanding the original margin by a factor of 1.3. The detection accuracy decreases gradually with the increase of the factor, when it is greater than 1.3.

For this result, we think the reason is that the square of 1.3 is 1.69, whose half is about 0.85. When the square enclosing the detected face has no extra margin, the face region accounts for about 0.85 of the entire image region. Therefore, the factor of 1.3 makes the ratio between the number of face pixels and background pixels close to 1 : 1. That is, the ratio between the number of true and forged pixels is close to 1 : 1. We consider that preserving an appropriate proportion of true and forged pixels in the detection data is helpful to improve detection accuracy. We compare the accuracy convergence of different extended clipping factors in the training process of the whole model and display it in Figure 4. Weighing the accuracy and stability, we finally choose the factor of 1.4, which performs better and generalizes well to different datasets overall.

Specifically, given $F(x, y, w, h)$ that represent the coordinates of the upper left, width, and height of the detected face respectively α denotes the expanding coefficient that controls the size of the margin. We first compute the center position of this rectangle as shown in equation (2), and then generate new $F(x, y, w, h)$ as in equation (3), and accordingly cut off a square. We then resize the expanded faces to a uniform size regardless of the resolutions of the original videos, we set size = 224 and $\alpha = 1.4$ in this work.

$$P_c = \left(x + \frac{w}{2}, y + \frac{h}{2} \right), \tag{2}$$

$$\begin{aligned} \mathrm{Rect}_{\mathrm{new}} = F\Bigg(& P_c - \alpha \cdot \frac{\max(w,h)}{2}, \\ & P_c - \alpha \cdot \frac{\max(w,h)}{2}, \\ & \alpha \cdot \max(w,h), \\ & \alpha \cdot \max(w,h)). \end{aligned} \tag{3}$$

Considering that the head movements in videos are in a limited region in the short term, it is unwise to detect faces for every frame. Therefore, we only detect a portion of frames at regular intervals. For the undetected frames, we crop off faces by the detection results of the previously detected frame. In this paper, we detect faces for every 20 frames since the videos commonly contain 30 or 24 frames per second.

4.1.2. Hyperparameters. The performance is reported differently: frame-level accuracy is used to evaluate the performance of backbones that can only take single frames as input; video clip level accuracy is used to evaluate the models that take short sequences of consecutive frames. As in Table 4, the training of backbones and CWSA Net both consists of 40 epochs without early stopping. We use minibatch stochastic gradient descent as the optimizer and set *learning rate* = 0.01, *momentum* = 0.9, and learning rate decays by

Table 1: Basic information of datasets used in this work.

Dataset	Real/fake	Frames (k)	Size
FF++	1000/4000	509.9/2039.6	480p,720p,1080p
DFDC-P	1131/4113	88.4/1783.3	180p–2160p
Celeb-DF	890/5639	358.8/2116.8	Multiple

Figure 3: Results of three different face cropping strategies.

Table 2: Binary classification accuracy (%) of different face cropping strategies.

Cropping type	DF	NT	Celeb-DF
Convey hull	99.03	95.78	92.59
No-margin	99.09	97.34	91.88
1.4 × margin	**99.31**	**99.13**	**93.97**

2.375e-4 per epoch. For models trained on face images, *batch size* = 32 and *iterations* = 50. For CWSA Net, *batch size* = 16 due to memory limit and *iterations* = 100 for adequate training samples in each epoch. All performances are reported on 3200 random test samples.

In terms of evaluation metrics, we consider video forensics a binary classification task and adopt the metric binary classification accuracy that represents how many samples are correctly classified. Although pristine and forged videos in DFDC-P and Celeb-DF are unbalanced, we deliberately pick up samples of each class with a 50% probability to make it balanced in both training and testing. We also report the AUC (area under the curve) for comprehensive assessments.

Note that there is not any data augmentation used in this work. However, it is highly possible to achieve better results with appropriate augmentation, training hyperparameters, and other tricks. We choose not to do so because the aim of this work is to study the characteristics of deep models used for face forgery detection and the effectiveness of our approach.

4.2. Backbone Selection. The backbone is a key component that extracts deep features preliminarily. Thus, we systematically investigate the performance of different deep CNNs in fake face detection to determine the most task-orient one.

Table 3: Binary classification accuracy (%) of different face cropping strategies on highly compressed NT.

Cropping type	Compressed NT
1.1 × margin	75.32
1.2 × margin	75.64
1.3 × margin	**76.12**
1.4 × margin	75.48
1.5 × margin	74.30

EfficientNet B0 [5] shows its remarkable potential, and we consider it the backbone of our approach.

It is hard to design a face forensics task-oriented model from scratch. Although neural architecture search technology may help, it could lead to overfitting on specific datasets. The existing research on computer vision, especially general image classification on ImageNet, has provided some off-the-shelf deep models that perform preeminently on image feature extraction. However, their performance on ImageNet cannot be the only point of reference due to the difference between general image classification and forgery detection. It is not clear enough about how model architectures, internal modules, and layer combinations affect the detection performance. To this end, we systematically investigate the difference between various deep models.

As in Table 5, we evaluate a list of models and there is indeed some consistency when deep models are applied in forensics detection. Empirically, we chose models that can be divided by different standards. Considering *skip connections* and *inception modules* are the two most popular and effective components to construct modern deep models, the first standard classifies the chosen models by whether it contains *skip connections* (EfficientNet B0 [5] & Xception [22]) or not (Inception V3 [23] & MobileNet V1 [24]), and the second classifies by if the model contains *inception modules* (Inception V3 & Xception) or not (EfficientNet B0 & MobileNet V1). To classify real/forgery face patches, the output of the last convolution layer in all these models is compressed by a global average pooling to produce a feature vector, and a single neural unit with sigmoid activation is connected to it for classification.

It can be seen from Table 5 that *skip connection* is the key factor affecting detection accuracy. This is not evident enough on FF++ since the accuracies are saturated. On DFDC-P, it becomes more obvious, and the gap expands to 27% on Celeb-DF. A reasonable explanation is that low-level features help to expose facial manipulations, and *skip connection* can directly deliver these features to the downstream of models. To validate this, we remove the *skip connections* in EfficientNet B0 and Xception. On FF++, the performance of both models without *skip connections* seriously decreases and is even much worse than the other two. This is also can be seen on two other datasets, and their performance degrades to the same level as those models without *skip connections*. Overall, EfficientNet B0 performs best and generalizes well to different datasets and is an ideal backbone for image level feature extraction.

To further verify this, we define the variance distribution of the last dense layer as the *Neural Activity* of deep models:

$$\text{Neural Activity} = \left[\sum_{i=1}^{N}\frac{\left(O_i^1-\mu^1\right)^2}{N},\ldots,\sum_{i=1}^{N}\frac{\left(O_i^L-\mu^L\right)^2}{N}\right]^{\mathrm{T}}. \tag{4}$$

where for the dense layer with L units, O_i^l denotes the output of l-*th* neural unit of i-*th* sample, and μ^l denotes the mean output value of l-*th* neural over N samples. Because a neural unit of the last dense layer with a larger variance will contribute more to the final classification. Accordingly, an intensive *Neural Activity* indicates that most units are active at the same level and approximately equally contribute to final classification. We calculate the variances of every unit in the last dense layer of four models over $N = 3200$ test samples. Because performance on Celeb-DF is the most variable, thus we display the *Neural Activity* on four models by box plots in Figure 5. For Xception and EfficientNet B0, the first and the third quartiles are very close, which represents that their *Neural Activity* is very intensive, and most units contribute to detection. For the rest two, this range is relatively larger, which means there are many lazy units that contribute less to detection.

4.3. Performance of the Proposed Approach. On DFDC-P and Celeb-DF, we carry out sufficient experiments of different video clip lengths, and the results are shown in Tables 6 and 7, and the best results are shown in bold. The first column shows the performance of the EfficientNet B0 backbone on frames. Obviously, CWSA Net effectively improves the detection accuracy, and the longer the input sequence length is, the more the detection accuracy is. But this gain stops when the length is about 12 and is not continuing when the length further raises. As for parameters, the EfficientNet B0 backbone is the major part and contains about 4.05 M parameters. For the following layers, an input with 3 frames only needs extra 79234 parameters, and then for each additional frame, only additional 1152 parameters are required. We also evaluate the commonly used CNN-LSTM architecture with the same experimental hyperparameters. For the CNN-LSTM, it also leverages EfficientNet B0 as the backbone, and a 2048-unit LSTM takes the global average pooled outputs of the backbone which is a 1280-d vector. The LSTM is followed by a 512-d dense layer and a single neural to make the prediction. On DFDC-P, the CNN-LSTM even performs worse than the backbone. Although it indeed makes some gains on Celeb-DF that increases with sequence length, our CWSA Net still outperforms the CNN-LSTM.

In order to verify the superiority of the CWSA module, we compare it with a method that is a simple average fusion of each frame. In addition, we also compare the CWSA module with traditional RNN and LSTM based on the same feature extraction backbone (EfficientNet B0). Table 8 presents a comparison of the accuracy of these methods on NT, which is the subset of FF++. It should be noted that the NT used here is highly compressed, which is of lower quality. Obviously, the performance of RNN and LSTM is not very good, even not as effective as the simple average fusion of each frame. We consider that this is because RNN

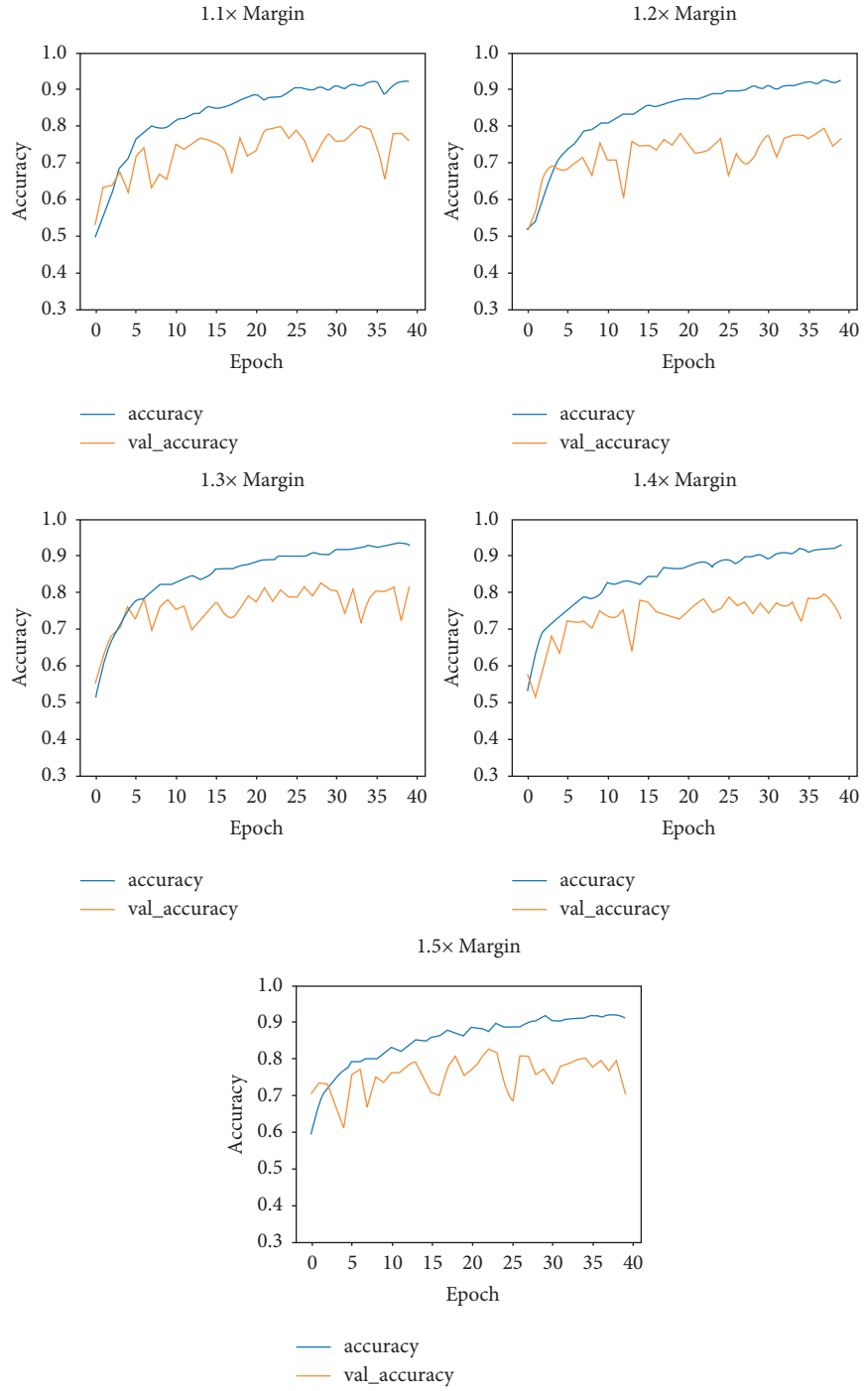


Figure 4: The comparison of accuracy convergence of different extended clipping factors.

Table 4: Training settings of different parts.

Settings	Training backbones	Training CWSA net
Batch size	32	16
Iteration	50	100
Epoch	40	40
Optimizer	SGD	SGD
Learning rate	0.01	0.01
Momentum	0.9	0.9
Decay	$2.375e-4$	$2.375e-4$

Table 5: Binary classification accuracy (%) (higher is better) of different backbones on frames.

Model	DF	F2F	FS	NT	DFDC-P	Celeb-DF
EfficientNet B0 [5]	**99.31**	99.69	99.53	**99.13**	**81.97**	93.97
Xception [22]	99.22	99.62	**99.56**	99.00	80.75	**94.84**
Inception V3 [23]	98.84	**99.78**	99.47	98.24	79.72	66.19
MobileNet V1 [24]	99.16	98.75	99.53	98.47	79.09	66.69
EfficientNet B0(w/o skip)	83.56	58.62	58.84	60.94	76.31	66.66
Xception (w/o skip)	94.91	58.80	64.62	53.91	65.44	67.50

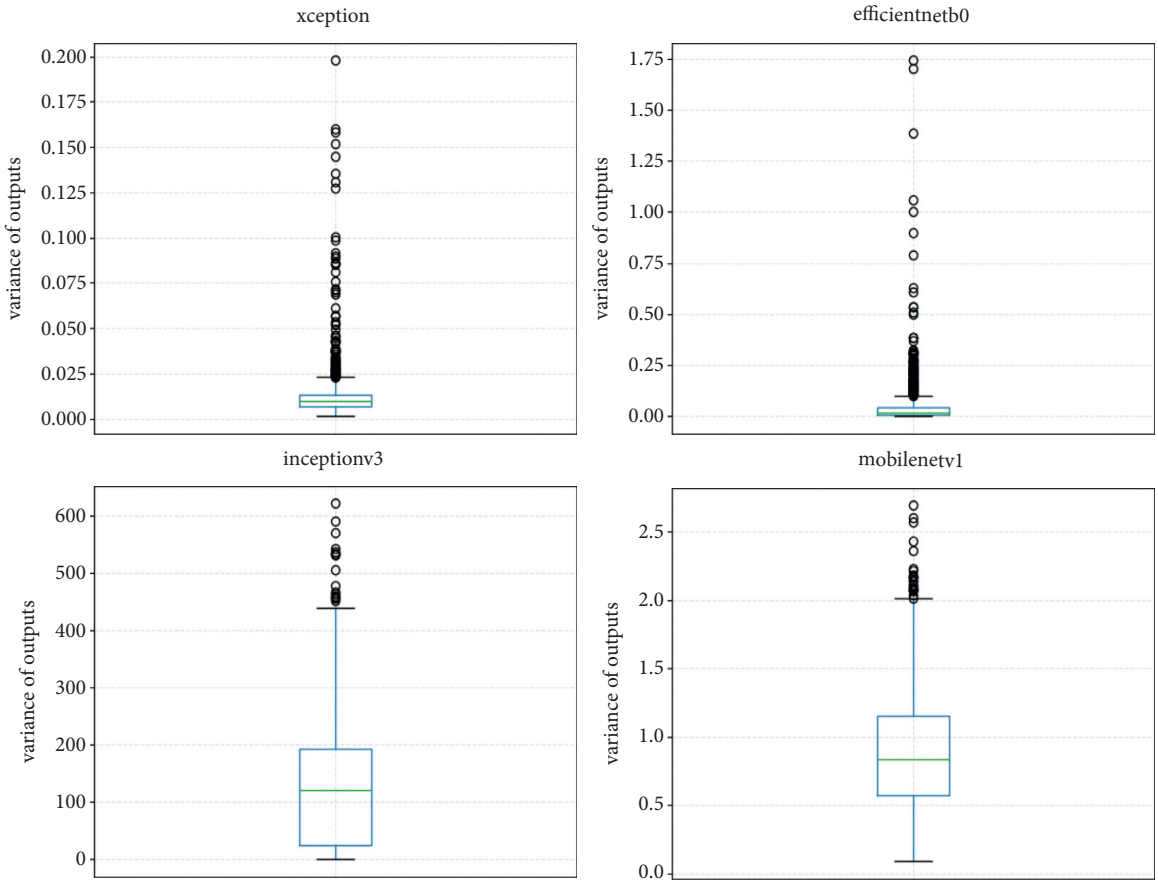


Figure 5: Neural activity of four models on Celeb-DF.

Table 6: Binary classification accuracy (%) (higher is better) on video clips on the DFDC-P dataset.

Model	1	3	6	9	12	15	18
CWSA net	81.97	84.76	83.14	82.75	**85.28**	84.81	83.19
CNN-LSTM	—	79.08	80.50	80.28	80.78	81.91	79.75

Table 7: Binary classification accuracy (%) (higher is better) on video clips on the Celeb-DF dataset.

Model	1	3	6	9	12	15	18
CWSA net	93.97	95.86	96.27	96.17	**97.12**	96.91	95.28
CNN-LSTM	—	95.22	95.06	95.13	96.53	96.38	95.28

and LSTM lose the spatial feature information within each frame when aggregating temporal features, resulting in a negative impact. From the experimental results, compared with the traditional RNN and LSTM, the proposed CWSA module performs better.

Table 8: Binary classification accuracy (%) (higher is better) comparison based on the same backbone on highly compressed NT datasets.

Approach	Compressed NT
CWSA	**80.6**
Simple fusion	79.4
LSTM	76.0
RNN	75.6

Table 9 presents a comparison of accuracy between our approach and the state-of-the-art on all three datasets. Although the CWSA Net is a little weaker on Celeb-DF, it is, on average, better than the state-of-the-art approach. Even on FF++, the accuracy almost saturates, CWSA Net still has a

Table 9: Binary classification accuracy (%) (higher is better) comparison of ours and the state-of-the-art approach on three datasets.

Approach	Celeb-DF	DFDC-P	FF++	Average
Ours	97.1	**85.3**	**99.4**	**93.9**
Biometric [25]	**98.5**	82.4	98.9	93.3

significant advantage. Also, it is worth noting that DFDC-P is obviously the most challenging dataset. Both methods are not very ideal in detection accuracy. However, CWSA Net still surpasses the state-of-the-art by 2.9%, which is a significant improvement.

A whole comparison based on AUC is in Table 10. There are various methods that derive from different perspectives on the list. CWSA Net achieves the highest AUC scores on both Celeb-DF and DFDC-P, demonstrating its efficiency. And obviously, compared to most of the other methods, CWSA Net improves the detection performance by a great gap.

We also compare the accuracy and AUC on all four subsets of FF++ with multiple detection methods in Tables 11 and 12. In this part, the results are provided by the EfficientNet B0 backbone only, and nothing expect the specially designed face cropping strategy is used. Apparently, our approach achieves the state-of-the-art level on average, and it goes beyond other methods on 3 out of 4 subsets. Although FF++ is rather easy to be exposed, and some of the previous methods perform nearly 100% in terms of both AUC and binary classification accuracy, our approach still shows obvious advantages on this dataset. These excellent results are not only because of the Efficient B0 but also because of the face cropping scheme presented in this work.

4.4. Analysis. We present some failure cases of forged face detection with our proposed approach on highly compressed NT, as shown in Figure 6. For the first type of failure case shown in Figures 6(a) and 6(b), it is obvious that the face detector fails to correctly extract the face from the highly compressed image, which directly degrades the detection accuracy of the forged face. Therefore, improving the robustness of the face detector can effectively solve such failures. For the second type of failure case, we adjust the color contrast of the images for a better display of the details and show them in Figures 6(c) and 6(d). Actually, color contrast is also one of the main factors affecting the detection of fake faces. In order to deal with such failures, digital image processing methods can be used to preprocess the samples with low color contrast. For the last type of failure case shown in Figures 6(e) and 6(f), the samples wrongly detected are video frames with different face poses. Due to the relatively few video frames of the side face in the datasets, the detection model is not sensitive to such samples, resulting in detection accuracy not being good enough. Therefore, the model can perform better with appropriate data augmentation.

Our work is one of the few existing methods in the field of fake face video detection that utilizes both airspace features and time domain features, and in Section 4.3, the experimental results have demonstrated its effectiveness. Differently, previous methods have focused on searching for clues of forgery at the image level [9, 19, 29]. Although these methods have had some success, they still leave room for improvement in terms of detection accuracy since they do not take advantage of significant temporal differences between real and fake as well. And our method attempts to further exploit temporal features to improve detection accuracy while maintaining the use of spatial features. In fact, we are not the first to consider this [3], but previous work destroyed the spatial structure of spatial features before extracting temporal features. This inevitably leads to a degradation of accuracy feature representation. Thus, the difference is that the proposed method extracts spatial-temporal features at the same stage, which mitigates the deterioration of spatial features, leading to advantages across multiple datasets.

Table 10: AUC (higher is better) comparison on Celeb-DF and DFDC-P datasets.

Approach	Celeb-DF	DFDC-P
Ours	**0.997**	**0.925**
Metric learning [26]	0.992	—
Face X-ray [27]	0.748	—
Fakespotter [28]	0.668	—
Mesonet [29]	—	0.753

Table 11: Binary classification accuracy (%) (higher is better) comparison on FF++ datasets.

Approach	DF	F2F	FS	NT	Average
Ours	**99.31**	**99.69**	**99.53**	**99.13**	**99.42**
Xception [19]	—	—	—	—	99.26
Fakespotter [28]	—	—	—	—	98.50
CNN-RNN [3]	96.90	94.35	96.30	—	95.85

Table 12: AUC (higher is better) comparison on FF++ datasets.

Approach	DF	F2F	FS	NT	Average
Ours	**0.997**	**1.0**	**0.999**	0.988	**0.996**
CNN-RNN [3]	0.996	0.984	0.994	—	0.991
Face X-ray [28]	0.992	0.991	0.992	**0.989**	0.991
Camera noise [9]	0.963	0.939	0.978	—	0.960

4.5. Industrial Applications. Currently, the negative effects of the fake face videos mainly remain on the network, as presented in Figure 7, and due to the constraints of laws and policies, they are not too excessive to bring serious adverse effects. However, these face manipulation technologies are nonnegligible threats to the systems that rely on face recognition in real word, not just in cyber world. A normal face recognition system without a strong face antispoofing module often requires the user to make corresponding facial movements as instructed to verify his legitimacy. If this step is passed, the system will retrieve the captured faces from a local or cloud-based database to further determine whether he/she is authorized or not. But this kind of face recognition system without forgery algorithms or modules usually cannot resist face-swap attacks.

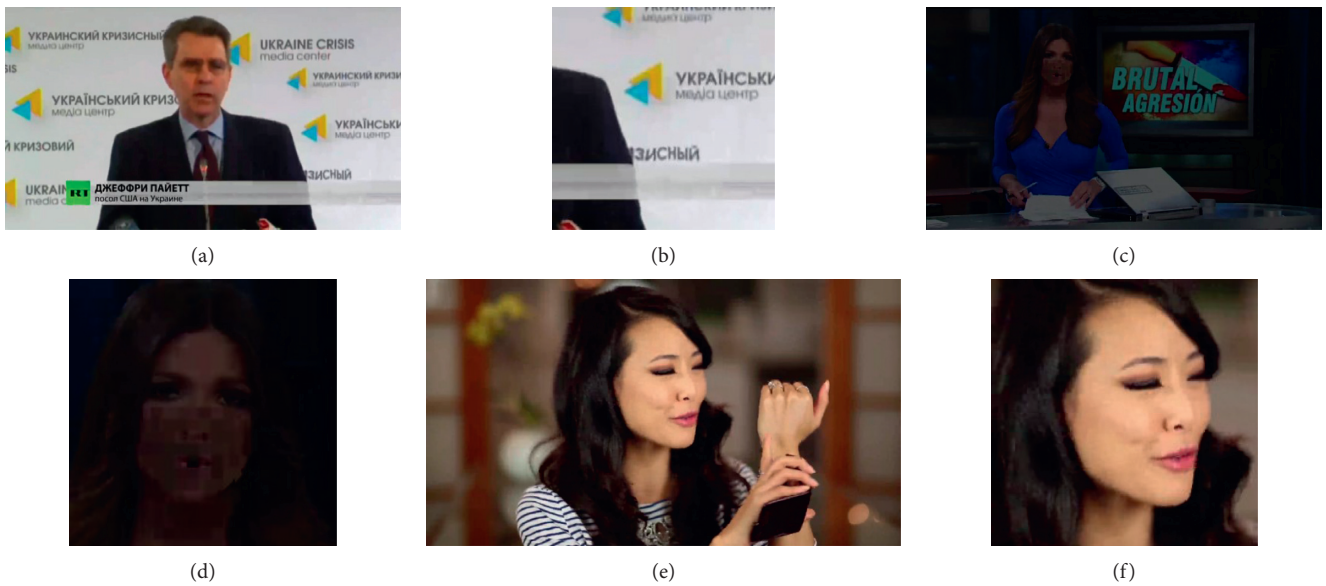


Figure 6: Failure cases on highly compressed NT. (a, c, e) Video frame. (b, d, f) Extracted face.

Figure 7: Fake faces videos circulating on the Internet [30].

Face recognition technology, due to its convenience and remarkable, has been applied in a few interactive intelligent applications. In those scenarios with high security requirements, these easily exposed face recognition systems have a number of security implications. Exiting face recognition systems are vulnerable to presentation attacks ranging from makeup, print, 3D-mask, etc. In recent years, in order to ensure the security of face recognition systems, face antispoofing (FAS) technology is also highly concerned [31]. Yu et al. proposed the first FAS method based on neural architecture search to discover the well-suited task-aware networks [32]. However, the forged face can also indirectly attack the face recognition system in these ways, which can hardly be ignored. The hacker may leverage the face-swap algorithms to simulate the facial movements following the instruction and print or display the forged face on some medium like paper or electronic screen in order to deceive the system. This calls the requirement of an additional fake face detection module in the first phase of the face recognition system to eliminate the safety hazards, as shown in Figure 8. More importantly, the forgery algorithm in the real-life scenario is unknown, and the detection algorithm needs to be highly robust to multiple forgery types. The CSWA tested on benchmark containing different types of face swapping and reenactment, which are both capable of assaulting face recognition systems, can assist these systems in defending these attacks. To ensure user experience, face recognition systems usually require the entire pipeline to be relatively fast, so the face forgery detection module can only acquire a short video of the face, but our method only requires a limited number of frames to achieve high precision detection.

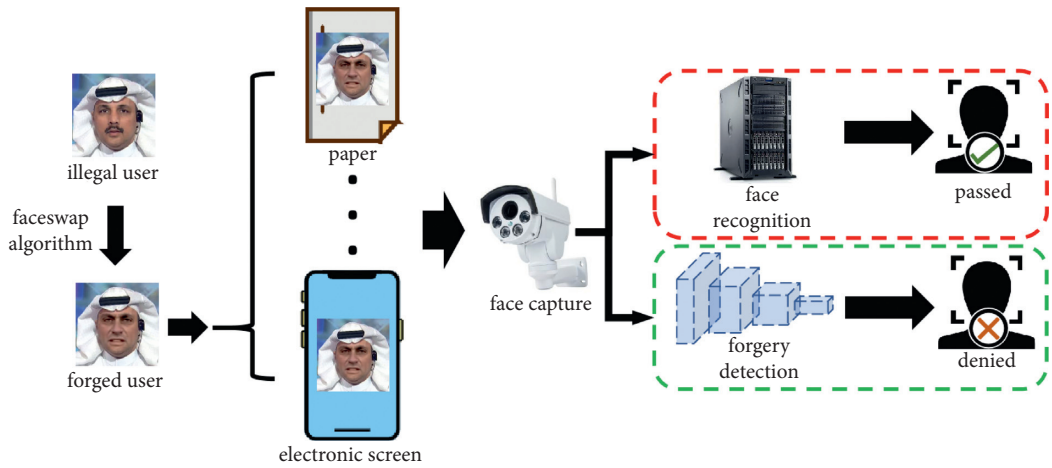


Figure 8: Detecting fake face attack with the forgery detection module.

5. Conclusion and Future Work

In this paper, we describe a novel forensic module named CWSA to detect face video manipulations. To take a close look at the problem of manipulation detection using deep CNNs, we first study the influence of face cropping strategies and architectures of different networks. We find that in face cropping, a suitable margin helps models perform better. And *skip connections* that pass low-level features downstream are also very beneficial in this task. On these bases, we propose our simple but smart CWSA Net that recombines feature maps belonging to the same channel from consecutive frames and fuses them by separately convoluting the new feature map sets. Our approach is demonstrated to be very competitive by the evaluations on three large-scale face video manipulation benchmarks. It achieves the state-of-the-art level on average and goes beyond other methods on most of the datasets. On the most challenging dataset DFDC-P, the performance of both our and the state-of-the-art approaches is not very ideal but the CWSA Net still surpasses it by 2.9%, which is a significant improvement.

Our work indicates some opportunities for future research, as it proves the feasibility of detecting forged faces from spatial and temporal perspectives. Firstly, although the CWSA module aggregates interframe features without destructing their spatial structure, there is no further constraint on the interesting regions. That is, the module treats different locations of features equally. Intuitively, the amount of information about forgery flaws exposed in the time domain varies across regions, and the more informed regions are supposed to be more focused. Thus, we can turn to the attention mechanism, but due to the lack of ground truth of interested regions, we have to design an attention module in an unsupervised or semisupervised manner. Another opportunity for future work is domain generalization. Existing detection approaches, including ours, are not robust enough to unknown types of fakes, but facing unknown attacks is common in real-life scenarios, and the ability to generalize to an unknown domain is essential if we want our approach to be more pragmatic. To this end, in future work, we expect our approach to be more than just a binary classifier, but to aggregate real faces through metric learning while making the fake face as separate from the real face as possible, in this case, the fake face detection tasks is viewed as anomaly detection tasks.

Data Availability

The data that support the findings of this study are openly available in Yujiang-Lu/CWSA-tensorflow at https://github.com/Yujiang-Lu/CWSA-tensorflow.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Jiangsu Basic Research Programs-Natural Science Foundation under grant no. BK20181407, in part by the National Natural Science Foundation of China under grant nos. U1936118, 61672294, U1836208, 61702276, 61772283, 61602253, and 61601236, in part by Six Peak Talent Project of Jiangsu Province (R2016L13), Qinglan Project of Jiangsu Province, and "333" Project of Jiangsu Province, in part by National Key R&D Program of China under grant 2018YFB1003205, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, and in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

References

[1] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *Proceedings Of the IEEE Conference On Computer Vision And Pattern Recognition Workshops*, pp. 46–52, Long Beach, California, 2019.

[2] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings Of the IEEE International*

Conference On Computer Vision, pp. 9459–9468, Seoul, korea, October 2019.

[3] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *Interfaces*, vol. 3, p. 1, 2019, https://arxiv.org/abs/1905.00582.

[4] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based cnn," in *Proceedings Of the IEEE International Conference On Computer Vision Workshops*, Seoul, Korea (South), October 2019.

[5] M. Tan and Q. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," in *Proceedings International Conference On Machine Learning*, pp. 6105–6114, PMLR, Long Beach, California, June 2019.

[6] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proceedings of IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, Rome, Italy, November 2015.

[7] S. Mandelli, N. Bonettini, P. Bestagini, V. Lipari, and S. Tubaro, "Multiple jpeg compression detection through task-driven non-negative matrix factorization," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2106–2110, IEEE, Calgary, Alberta, Canada, April 2018.

[8] J. Luka, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[9] D. Cozzolino, "Extracting camera-based fingerprints for video forensics," in *Proceedings Of the IEEE Conference On Computer Vision And Pattern Recognition Workshops*, pp. 130–137, Long Beach, CA, USA, June 2019.

[10] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images,," in *Proceedings of IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, Hong Kong, China, December 2019.

[11] D. Cozzolino, J. Thies, A. Rössler, M. Niezner, and L. Verdoliva, "Spoc: spoofing camera fingerprints," 2019, https://arxiv.org/abs/1911.12069.

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

[13] S. McCloskey and M. Albright, "Detecting gan-generated Imagery using color cues," 2018, https://arxiv.org/abs/1812.08247.

[14] H. Li, B. Li, S. Tan, and J. Huang, "Detection of deep network generated images using disparities in color components," 2018, https://arxiv.org/abs/1808.07276.

[15] L. Nataraj, T. M. Mohammed, B. S. Manjunath et al., "Detecting gan generated fake images using cooccurrence matrices," *Electronic Imaging*, vol. 2019, no. 5, pp. 532–1–532–7, 2019.

[16] T. Zhou, W. Wang, Z. Liang, and J. Shen, "Face forensics in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5778–5788, New Orleans, US, November 2021.

[17] Y. Li, M.-C. Chang, and S. Lyu, "In Ictu Oculi: Exposing Ai Created Fake Videos by Detecting Eye Blinking," in *Proceedings of 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, Hong Kong, June 2018.

[18] S. Fernandes, S. Raj, E. Ortiz et al., "Predicting heart rate variations of deepfake videos using neural ode," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Seoul, South Korea, October 2019.

[19] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nie, "Faceforensics++: learning to detect manipulated facial images," in *Proceedings Of the IEEE International Conference on Computer Vision*, pp. 1–11, Seoul, South Korea, November 2019.

[20] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," 2019, https://arxiv.org/abs/1910.08854.

[21] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: a new dataset for deepfake forensics," 2019, https://arxiv.org/abs/1909.12962.

[22] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings Of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[24] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[25] S. Agarwal, T. El-Gaaly, and H. Farid, "Detecting deep-fake videos from appearance and behavior," 2020, https://arxiv.org/abs/2004.14491.

[26] A. Kumar, A. Bhavsar, and R. Verma, "Detecting deepfakes with metric learning," in *2020 8th International Workshop On Biometrics And Forensics (IWBF)*, pp. 1–6, IEEE, Porto, Portugal, April 2020.

[27] L. Li, J. Bao, T. Zhang et al., "Face x-ray for more general face forgery detection," in *Proceedings Of the IEEE/CVF Conference On Computer Vision And Pattern Recognition*, pp. 5001–5010, Seattle, WA, USA, June 2020.

[28] R. Wang, F. Juefei-Xu, L. Ma et al., "Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces," in *Proceedings of International Joint Conference On Artificial Intelligence (IJCAI)*, Yokohama, Japan, 2020.

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

[30] Fake faces videos circulating on the internet. [Online]. Available: https://www.theguardian.com/technology/2020/jan/13/what-are-deepfakes-and-how-can-you-spot-them.

[31] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep learning for face anti-spoofing: a survey," 2021.

[32] Z. Yu, J. Wan, Y. Qin, X. Li, and G. Zhao, "Nas-fas: static-dynamic central difference network search for face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3005–3023, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6854586, 12 pages
https://doi.org/10.1155/2021/6854586

Research Article

A Saliency Detection and Gram Matrix Transform-Based Convolutional Neural Network for Image Emotion Classification

Zelin Deng,[1] **Qiran Zhu**,[1,2] **Pei He**,[3] **Dengyong Zhang**,[1] **and Yuansheng Luo**[1]

[1]*School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2]*School of Big Data and Artificial Intelligence, Xinyang University, Xinyang 464000, China*
[3]*School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Dengyong Zhang; zhdy@csust.edu.cn

Received 28 May 2021; Accepted 2 August 2021; Published 10 August 2021

Academic Editor: Beijing Chen

Using the convolutional neural network (CNN) method for image emotion recognition is a research hotspot of deep learning. Previous studies tend to use visual features obtained from a global perspective and ignore the role of local visual features in emotional arousal. Moreover, the CNN shallow feature maps contain image content information; such maps obtained from shallow layers directly to describe low-level visual features may lead to redundancy. In order to enhance image emotion recognition performance, an improved CNN is proposed in this work. Firstly, the saliency detection algorithm is used to locate the emotional region of the image, which is served as the supplementary information to conduct emotion recognition better. Secondly, the Gram matrix transform is performed on the CNN shallow feature maps to decrease the redundancy of image content information. Finally, a new loss function is designed by using hard labels and probability labels of image emotion category to reduce the influence of image emotion subjectivity. Extensive experiments have been conducted on benchmark datasets, including FI (Flickr and Instagram), IAPSsubset, ArtPhoto, and Abstract. The experimental results show that compared with the existing approaches, our method has a good application prospect.

1. Introduction

Image sentiment analysis is becoming a research hotspot in the field of computer vision [1–6]. It is more difficult to analyze images at the emotional level compared with the recognition of objects in images [7–13] mainly because of the complexity and subjectivity of emotions [4]. First of all, due to the complexity of emotion, image emotion recognition work is to analyze the image at the emotional level, and the expression of emotion is also affected by numerous feature information [14], so it is difficult to design a discriminative representation feature to cover enough feature information, such as color, texture, and semantic information. Secondly, due to the subjectivity of image emotion, people with different lives and cultural backgrounds may have different emotional responses to the same image which makes it difficult to collect hard emotion labels of the image and lead to the uncertainty of the image's category label.

In previous studies, many researchers have proposed methods to solve the complexity and subjectivity of image emotion. For instance, Borth et al. [14] developed a visual sentiment ontology, which consisted of 1200 concepts and associated classifiers, and each concept was composed of an adjective expressing emotion and a noun related to object or scene. In the work of image emotion analysis, manual features, including color, texture, composition, balance, and harmony [2, 15, 16], are first used to analyze the emotion of the image. However, handmade features are unable to fully express the relationship between visual information and emotional arousal because handmade features cannot cover the important features related to image emotion [17].

Recently, researchers began using CNNs to solve difficult problems in image sentiment classification to further improve classification performance [1]. Different from the manual features, CNN can learn image representation in an end-to-end manner. Research results have proved that deep

CNN features are better than manual features in image emotion recognition [17]. However, due to the complexity and subjectivity of emotions, analyzing images at the emotional level is a more challenging task compared with traditional visual tasks, such as object classification and detection in the image. For the complexity of image emotion, most images can cause different emotional reactions, rather than a unique emotion. Previous studies mainly used visual features extracted from the global view of the image for emotion recognition, while ignored the fact that expression of image emotion mainly depends on the local regions of an image. Figure 1 shows the image samples and the main regions in them to evoke emotion. Obviously, some local regions of the image contain more emotional information than others. Besides, Alameda-Pineda et al. [18] pointed out that CNNs were unable to effectively extract emotional information from abstract paintings, which means emotions not only are induced by image semantics but also are conveyed through low-level visual features, such as texture, color, and shape.

In order to understand how CNNs designed for object recognition task works in image emotion recognition task, many studies on deep feature representation on convolutional neural network processing level have been conducted. Research shows that emotion recognition of the deep model is mainly based on semantic features of images, which can explain the successful application of CNN in image emotion recognition [2]. On the other hand, when the image is processed by the deeper CNN layers, the low-level visual features are gradually reduced. In some cases, people pay more attention to the background of the image than to the object in the image, that is, nonobject components may be more emotional than image contents [18]. This requires us to introduce the low-level visual features of the image when designing the classification features, but if we directly use the feature map obtained from the shallow network to describe the low-level visual features, there will be a problem of redundancy because the feature map also contains the image content information. Inspired by the work of image style transformation [19–21], we apply Gram matrix transformation on the feature maps from the shallow layers of the network to reduce the redundancy of image content.

In order to enhance the image emotion recognition performance, the CNN is proposed to improve with the following. Firstly, use the saliency detection method to extract the features of the local emotional regions to better invoke the emotions. Secondly, introduce multiple side branch structures in network to obtain the feature maps of the shallow layers and use the Gram matrix to transform the feature maps to decrease redundancy. Finally, design a new loss function by using the hard labels and probability labels of image emotion categories to reduce the impact of image emotion subjectivity on classification.

In summary, the contributions of our paper are summarized as follows:

(1) Use saliency detection algorithm to locate the emotional region in the image and extract the features of the emotional region in the image, which can avoid the noise information in the nonemotional region and give more attention to the local emotional regions.

(2) Design a method to calculate the Gram matrix of the feature map. After Gram matrix transformation, the redundancy of the image content information in the feature map is reduced, and new low-level visual features are obtained.

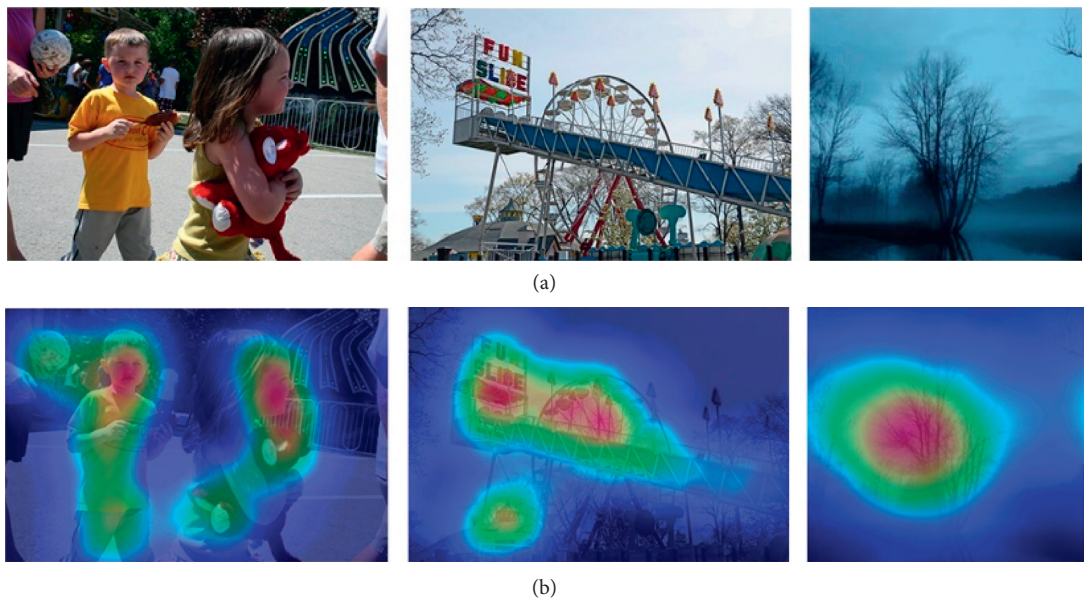
(3) Propose a new loss function by using the hard labels and probability labels of image emotion categories to reduce the impact of image emotion subjectivity on classification.

The remainder of this paper is as follows. In Section 2, we summarized and reviewed the related work of image emotion recognition and image saliency detection. Section 3 introduced our model and improvement work. Section 4 introduced the datasets used in the experiment and presented the experimental results and analysis of this work. In Section 5, our main work and future research keys are summarized.

2. Related Works

The analysis of images and videos on the emotional level has attracted the attention of more and more researchers [22–25], and a lot of research works have been carried out. In this section, we focus on reviewing the related work of image emotion analysis and image saliency detection.

2.1. Image Emotion Analysis. In the work of image sentiment classification, the method of designing multilevel visual features of images and applying them to image sentiment analysis has been widely tracked. Yanulevskaya et al. [15] first proposed low-level visual features, including Gabor and Wiccest features, to classify the emotions of artworks. Soli and Lenz [26] introduced an image descriptor based on color and emotion. This method is derived from psychophysical experiments for image classification and uses SIFT features for emotion prediction. Machajdik and Hanbury [2], based on art and psychological theories, defined a rich handcrafted middle-level feature from the aspects of composition, color change, and texture. Zhao et al. [16] introduced the middle-level visual features designed based on the concept of principle-of-art to extract emotion features (PAEF) to classify image emotion. However, compared with the features extracted from the CNN model, these manual features are mainly concentrated on low-level visual features. Due to the limited feature types and lack of exploration of high-level semantic information in images, it is difficult to cover all important factors related to image emotions.



(a)

(b)

Figure 1: Examples of emotion images and emotion arouse regions. (a) Images from the image emotion datasets. (b) Visualization of the main regions to invoke emotions.

In recent years, due to the excellent performance of CNN methods, researchers have applied the CNN method in image emotion analysis. Peng et al. [27] first applied the pretrained CNN model on ImageNet [28] for image sentiment analysis and achieved excellent classification results. You et al. [29] introduced a progressive strategy training to train the CNN model on a large-scale web image dataset to detect the emotion of the image. Rao et al. [17] proposed a multi-instance learning framework in order to obtain the multilevel deep representations of an image and obtained an exciting recognition result. You et al. [30] used the attention model to extract local emotional region features for emotional analysis. Yang et al. [31] proposed coupled CNN with two branches, which used both global and local information of an image. However, most of the studies did not fully use the local emotional regions of image, which limited the classification performance of the model.

2.2. Saliency Detection. Due to the powerful representation ability of deep features, the saliency detection method based on deep learning gradually surpasses the traditional method based on manual features [32–34]. Inspired by fully convolutional networks [35], more and more researches paid attention to predict the saliency map at the pixel level. Liu et al. [36] introduced an attention mechanism to guide the feature integration process by a U-shape model. Liu et al. [37] proposed a two-stage network algorithm. The algorithm generates a rough saliency map and combines local context information to refine the saliency map recursively and hierarchically. Hou et al. [38] introduced short connections in the multiscale side output to capture fine details. Zhang et al. [39] used a bidirectional structure to pass messages between the multilevel features extracted by the convolutional neural network to better predict the saliency map. Xiao et al. [40] first used a distracted detection network D-Net to crop the interference region in the image and then used the saliency detection network S-Net for saliency detection.

3. The Proposed Method

In order to improve image emotion recognition performance, an improved CNN is proposed, and the framework of our method is shown in Figure 2. The model includes the following improved components. (1) Two input branches: one is the original image input branch, and the other is the saliency image input branch. In the first branch, the network structure is modified based on Inception-v4 [41]. Firstly, the fully connected layer after the last convolutional layer in the Inception-v4 network is removed. Secondly, the side branch structure is introduced at three different depths of the network, and each side branch structure is composed of a convolutional layer and the convolution kernel size is 1×1. In the second branch, the network structure is also modified based on Inception-v4, and the fully connected layer after the last convolutional layer is removed. (2) Three fully connected layers work after the two branch inputs are completed. (3) A softmax layer generates the probability of each category and works after the fully connected layers.

In the input branch of the original image, the image semantic features on the global view are obtained from the last fully connected layer, and the feature maps from the multiple layers of the network are obtained from the side branches, and these feature maps are used as the input to calculate the Gram matrix. In the input branch of saliency map, the feature of local emotion region is extracted from the last convolution layer. Semantic features, local emotional features, and low-level visual features of the image are integrated into the hybrid representation features of image emotion classification. Finally, the hybrid representation

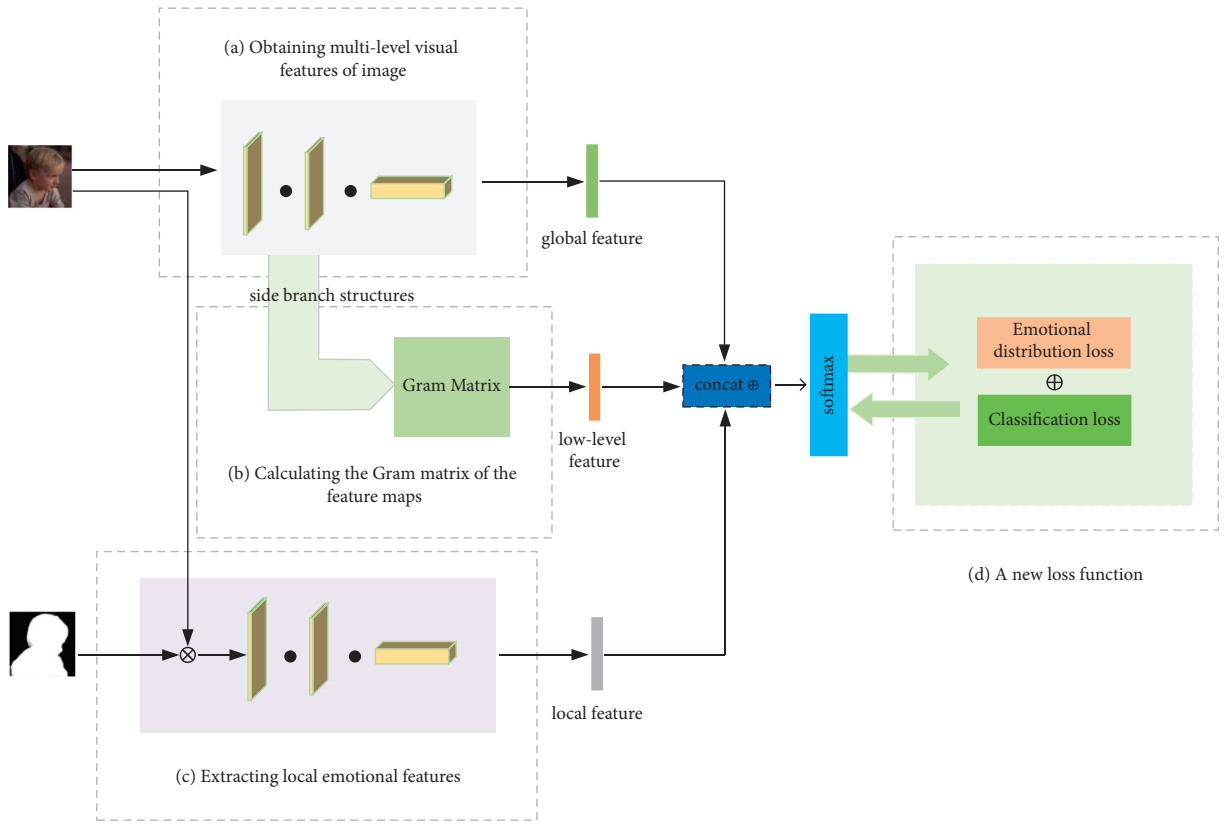


FIGURE 2: An overview of the proposed model. (a) The multilevel visual features are extracted by multiple side branch structures. (b) The Gram matrix of feature maps is calculated to reduce redundancy. (c) The saliency detection algorithm is used to locate the local emotion region of the image. (d) The hard label and probability label of image emotion category is used to design a new loss function.

features are input into the final fully connection layer and Softmax layer to predict the emotion category.

3.1. Saliency Detection and Local Emotional Features' Extraction. The human visual system only processes the vital part of image and meanwhile pays little attention to other parts, which prove that the human visual system has a certain mechanism to choose possible object positions when observing objects. So, the researchers consider that the object regions in the image are an emotional region with more emotions. In fact, the local regions covered by objects are more likely to attract people's attention and arouse their emotion. The saliency of the image highlights the degree of human attention to information-rich region and represents the different visual perceptions presented by different regions in the image. Based on the image saliency features, the saliency detection is used to locate the local region covered by objects in the image and extract the local emotional features of the image.

Firstly, image saliency detection algorithm is used to generate saliency image $Y = \{y_1, y_2, \ldots, y_n\}$, $y_i \in R^{w \times h}$, from corresponding original images $X = \{x_1, x_2, \ldots, x_n\}$, $x_i \in R^{3 \times w \times h}$, where w and h represent the width and height of the image, respectively. The saliency image is a binary image, and the size of the saliency image is the same as that of the original image. The element value of the object region of the original image is 1, while the element value of the nonobject region is 0. Thus, the local emotion region T can be calculated according to

$$T = X \bullet Y, \tag{1}$$

where $\bullet$ is the operator to multiply the elements of matrix X and the matrix Y. Then, input T into the saliency image input branch of the Siamese network to extract the local emotional features of the image.

3.2. Gram Matrix and Low-Level Visual Feature Extraction. The low-level visual features of the image are mainly concentrated in the shallow layers of the neural network [17]. There exists a problem of redundancy if we directly use the feature map obtained from the shallow layer of the network to describe the low-level visual features because the feature map also contains the image content information (e.g., objects and general scenery) [18].

In this paper, the low-level visual features are transformed by Gram matrix operation to reduce the redundancy. For each layer, use the feature maps to calculate the Gram matrix with the following steps. Firstly, vectorize each feature map F_i of size $w \times w$ in the convolutional layer to obtain a one-dimensional vector of length $L = w \times w$. Secondly, combine one-dimensional vectors in the order of the feature maps to obtain a matrix $F \in R^{N \times L}$, where N represents the number of feature maps in the convolutional layer.

Finally, calculate the Gram matrix $M \in R^{N\times N}$ of this convolutional layer according to

$$M = FF^T. \tag{2}$$

Each element M_{ij} in the Gram matrix is the inner product between the F_i and F_j, which can be obtained by

$$M_{ij} = \sum_k F_{ik}F_{jk}. \tag{3}$$

The procedure is summarized in Algorithm 1.

3.3. Loss Function of Emotional Subjectivity Constraint. In the collection of affective image data, the majority voting strategy is widely used to obtain the emotional label of the image. We calculate the distribution of image emotion based on the label probability to reduce the subjective influence of image emotion. The emotion theory research shows that the relationship between two emotions determines their similarity, and the two emotions from similar to completely opposite can be represented by Mikels' wheel [42]. As shown in Figure 3, a distance equation $\text{dist}(e_i, e_{i-1} = ''\text{feart}'')$ is defined in Mikels' wheel to quantify two emotional relationships. For example, the distance between the emotion fear and the emotion sadness is $\text{dist}(\text{fear}, \text{sadness}) = 1$, and the distance between the emotion fear and the emotion disgust is $\text{dist}(\text{fear}, \text{disgust}) = 2$, which indicates that the similarity between the emotion sadness and the emotion fear is higher.

Based on the definition of distance in Mikels' Wheel, the probability distribution of dominant emotion and other emotion can be calculated according to

$$f(i) = \begin{cases} \dfrac{\left(1/\text{dist}_{ij}\right)}{\sum_{i\neq j}\left(1/\text{dist}_{ij}\right)}\left(1 - p_j^*\right), & i \in V, \\ 0, & i \notin V, \end{cases} \tag{4}$$

where j is the dominant emotion category of the image, V denotes all the sentiment of the same polarity with the dominant emotion j, p_j^* is the probability of dominant emotion, and $f(i)$ is the probability of other emotions except the dominant emotion j. So, the probability distribution label of image emotion $d(i) = \{d_1, d_2, \ldots, d_n | n = 8\}$ can be obtained, and the sum of probabilities distribution $\sum d_i$ is normalized to 1.

Through using the hard label and probability distribution label, a new loss function can be designed according to

$$L_{\text{sub}j} = (1 - \lambda)L_{\text{cls}} + \lambda L_{kl}, \tag{5}$$

where L_{cls} is the cross-entropy classification loss, and it can be calculated by

$$L_{\text{cls}} = -\sum_i y_i \log(p_i), \tag{6}$$

where y_i is the ground truth label and p_i represents the probability that the image belongs to the i emotion category. Then, the Kullback–Leibler divergence [43] is used to measure the loss between probability distribution label $d(i)$ and predict emotion distribution p_i. Here, λ controls the weight of L_{kl}, and L_{kl} can be calculated by

$$L_{kl} = \sum_i d(i)\log(p_i). \tag{7}$$

4. Experiments and Results

In this section, our method is compared with other methods on FI, IAPSsubset, ArtPhoto, and Abstract datasets to evaluate our model.

4.1. Datasets. In the work of image emotion analysis, the widely used datasets mainly include FI, IAPSsubset, ArtPhoto, and Abstract, and the number of image samples in these datasets is shown in Table 1.

Flickr and Instagram (FI) [1]: this emotional dataset consists of about 23308 affective images. These pictures are collected by using 8 emotions as search keywords on Flickr and Instagram social networking sites. Then, these images were further labeled by Amazon Mechanical Turk, and the label of each image was done by five people voting.

In fact, the actual number of images that can be acquired in this dataset is 22,598 because the network connection for some images has failed. Table 2 shows the statistics of the number of available images.

IAPSsubset [2]: international affective image system (IAPS) is an international general emotion image dataset, which is widely used in image emotion classification. The dataset contains 1182 documentary-style natural images. Mikels et al. [42] selected 395 images from IAPs dataset and mapped them to eight emotion categories.

ArtPhoto [2]: in this dataset, photos are selected from the art photo-sharing website with emotion category as the search keyword, with a total of 806 photos. The emotional category of a photo is determined by the artist who uploaded it.

Abstract [2]: this dataset contains 228 abstract paintings. The emotional category of each abstract painting is decided by 14 different people. The emotion that gets the most votes is the emotion category of each image.

4.2. Implementation Details. The experiment was conducted on a computer based on the Pytorch environment. The computer used Intel(R) Xeon(R) CPU E5-2640 2.40 GHz CPU and NVIDA GeForce GTX TITAN GPU (12G memory). Our classification model is a Siamese network, and the backbone networks of the two branches are Inception-v4. The images in the dataset are randomly divided into training set (80%) and test set (20%): the training set totally has 18,078 images, and the test set totally has 4519 images. The image first scales the image in the range of [320, 480] based on the shortest side, then flips the image horizontally to obtain a mirror image, and then randomly crops 299×299 image blocks from the original image and the mirror image as the input of the model. We use the parameters pretrained on ImageNet to initialize the backbone

Input: feature map F_i of size $w \times w$
Output: Gram matrix $M \in R^{N \times N}$
Step 1: for each feature map F_i,

$$F_i = \begin{pmatrix} f_{11}^i, & f_{12}^i, & f_{1w}^i, \\ \vdots & \vdots & \vdots \\ f_{w1}^i, & f_{w2}^i, & f_{ww}^i \end{pmatrix}$$

vectorize F_i in convolution layer into a one-dimensional vector
$L_i = \left(f_{11}^i, f_{12}^i, \cdots f_{1w}^i, \cdots f_{w1}^i, f_{w2}^i, \cdots f_{ww}^i, \right)$, denoted as $L_1, L_2, \ldots, L_i$ $i = 1, 2, 3, \ldots N$;
Step 2: combine N one-dimensional vectors L_i into a matrix $\boldsymbol{F}$ in the order of the feature maps, denoted as $F \in R^{N \times L}$, $L = w \times w$.

$$F = \begin{pmatrix} f_{11}^1, & \cdots & f_{12}^1, & f_{1w}^1, & \cdots & f_{w1}^1, & f_{w2}^1, & \cdots & f_{ww}^1, \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{11}^N, & f_{12}^N, & \cdots & f_{1w}^N, & \cdots & f_{w1}^N, & f_{w2}^N, & \cdots & f_{ww}^N, \end{pmatrix}$$

Step 3: get the transposed matrix $F^T \in R^{N \times L}$ of matrix $F \in R^{N \times L}$, and compute the Gram matrix $M \in R^{N \times N}$ according to equation (3).

Algorithm 1: Procedure for applying the Gram matrix to convert the feature map.

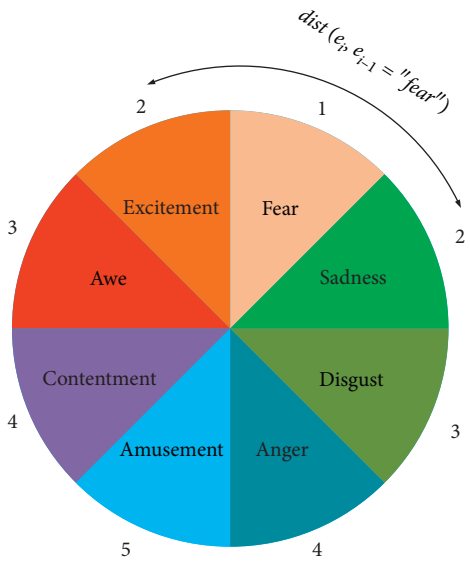


Figure 3: Mikels' emotion wheel and example of the emotion distance for emotion fear and other emotion [42].

Table 1: Statistics of the number of images in each image emotion datasets

Dataset	IAPSsubset	Artphoto	Abstract	FI
Amusement	37	101	25	4942
Anger	8	77	3	1266
Awe	54	102	15	3151
Contentment	63	70	63	5374
Disgust	74	70	18	1658
Excitement	55	105	36	2963
Fear	42	115	36	1032
Sadness	62	166	32	2922
Sum	**395**	**806**	**228**	**23308**

network of the model and use the stochastic gradient descent method to optimize the model. The parameters of our model are set as follows: the learning rate of model is set to 0.001,

Table 2: Statistics of the number of available images in FI emotion dataset.

Categories	Number of samples	Categories	Number of samples	Sum
Awe	3036	Disgust	1631	
Contentment	5268	Fear	1009	
Excitement	2808	Sadness	2771	
Amusement	4847	Anger	1228	
Sum	**15959**		**6639**	**22598**

and the weight decay is set to 0.0001. In particular, the learning rate is divided by 10 after every 5 epochs. The model is trained for up 20 epochs. The specific parameter settings are shown Table 3. Since the backbone network is a pre-trained model, the learning rate of the backbone network is set to 1/10 of the global learning rate for fine tuning.

4.3. Baseline

4.3.1. Handcrafted Features. In terms of handcrafted design features, GCH/LCH/GCH + BoW [44] used SIFT features based on bag-of-words to establish a 64-bit color histogram model for global color histogram (GCH) and local color histogram (LCH). Zhao et al. [16] introduced the middle-level visual features designed based on the concept of principles-of-art to extracted emotion features (PAEF) to classify image emotion. Rao et al. [45] proposed an emotion classification method based on multiscale blocks. Pyramid segmentation and simple linear iterative clustering (SLIC) method are used to segment the image into multiscale blocks. SentiBank [14] developed a visual sentiment ontology, which consist of 1200 concepts and associated classifiers, and each concept is composed of an adjective expressing emotion and a noun related to the object or scene.

4.3.2. Deep Features. In terms of deep features, AlexNet [8], VGG-16 [9], and Inception-v4 [41] all fine tune the pre-trained weights on the ImageNet dataset and complete the

Table 3: Initial parameters of our model.

Parameters	Value
Learning rate	0.001
Weight decay	0.0001
Momentum	0.9
Batch size	32
Epoch	20

emotion classification with the help of transfer learning. Deep SentiBank [46] proposed 2089-dim adjective-noun pair features based on CNN. PCNN [29] proposed a progressive strategy training to train the CNN model on the large-scale web image dataset to detect the emotion of the image. On the basis of AlexNet, Rao [17] obtained multilevel deep features by constructing multiple side branches in the network. Yang [47] proposed a learning method based on label distribution, which aims to solve the subjective problem of image emotion. WSCNet [31] proposed a weakly supervised coupled convolutional network with two branches.

4.4. Experimental Validation. In this paper, the classification model for large-scale emotional image dataset (FI) is initialized by using the parameters pretrained on the ImageNet dataset and then fine tuning the model on the FI dataset to complete the classification task. For small-scale datasets (IAPSsubset, Artphoto, and Abstract), the classification model is initialized by using the parameters pretrained on the FI dataset and then further fine tuning the model to complete the classification tasks.

4.4.1. The Effectiveness of Local Emotional Feature. To validate the effectiveness of the local emotional features, we designed a comparative experiment on the FI dataset. (1) Our model only uses the global feature from the last convolutional layer of the original image input branch of our model and low-level visual features. (2) Our model only uses the local emotional feature extracted from the local emotional region of the image. (3) Our model uses hybrid classification features composed of global semantic features, local emotional features, and low-level visual features. Table 4 shows the classification performance of our model with the three configurations on the FI dataset. Specifically, the global view only means that the model uses the global semantic feature and the low-level visual features, the emotional region only means that the model only uses the local emotional feature extracted from the local emotional region of the image, and the global view + emotional region means that the model uses hybrid classification features composed of global semantic features, local emotional features, and low-level visual features. As shown in Table 4, the model in (1) only uses global semantic features and low-level visual features, while the model in (3) uses local emotional features as supplementary information, and the classification accuracy of the model is improved about 4%, which shows that combining emotional features from local emotional regions can effectively improve emotional classification performance

Table 4: The classification accuracy on the FI dataset.

Method	Accuracy (%)
Global view only	66.14
Emotional region only	59.82
Global view + emotional region (ours)	**70.23**

than using global features only. In (2), when the model only uses the features from the local emotional region, the classification performance of the model is severely reduced, which illustrates the importance of extracting semantic features from the global view of the image.

In Figure 4, the classification confusion matrixes of our model are shown in the two configurations of whether or not to use image local emotional features. It can be seen that applying local emotional features can enhance the classification performance of model and produce a more balanced recognition result for each emotion category.

4.4.2. The Effectiveness of Gram Matrix Transform. In order to get more low-level visual features, we introduce multiple side branches into the network. Each side branch is composed of a convolution layer. We apply Algorithm 1 to each side branch, respectively, and transform the feature map to obtain the low-level visual feature of the image $\{G_i | i = 1, 2, 3, \ldots\}$. As shown in Table 5, C represents a hybrid feature composed of global semantic features and local emotional features, L represents the low-level visual features described by the feature map directly, and G represents the low-level visual features captured from the feature map by using the Gram matrix. In Table 5, the best classification result can be obtained by combining the feature C and feature $\{G_1, G_2, G_3\}$. The low-level visual features captured from the feature map can get better classification results. It also can be seen that when L_4, L_5 or G_4, G_5 from the high layers of the network are added, the classification accuracies decreased. Adding feature G_4, G_5 has less effect on classification performance compared with adding feature L_4 and L_5. This shows that the Gram matrix transform can effectively reduce the redundancy of image content information in the feature map.

4.4.3. The Effectiveness of Loss Function. Our new loss function $L_{\text{sub}j}$ is designed by using the hard labels and probability labels of image emotion categories, trying to reduce the impactive of image emotion subjectivity. Different from the cross-entropy loss function L_{cls}, $L_{\text{sub}j}$ maximizes the difference between emotion classes and emphasizes the relationship between emotion categories by comprehensively constraining the classification loss and emotion distribution loss. The two loss functions mentioned above were used to conduct comparative experimental on the FI dataset, and the results are shown in Table 6. As can be seen, the classification performance of the model has been improved after applying the $L_{\text{sub}j}$ loss function. In particular, the classification accuracy of our model is improved by about 1.4% after applying the $L_{\text{sub}j}$ loss function, which shows the effectiveness of our loss function.

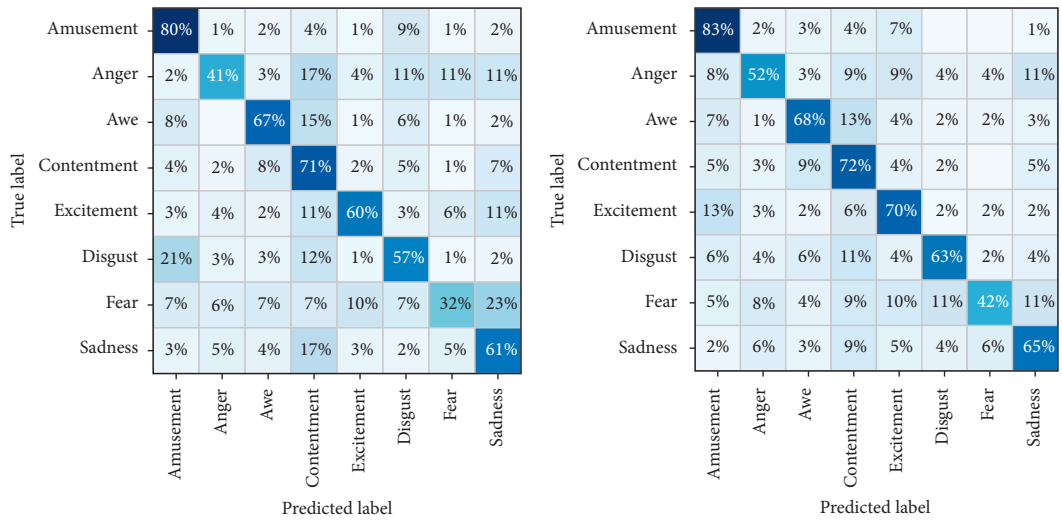


Figure 4: Classification confusion matrix on the FI dataset. (a) Our model without local emotional feature. (b) Our model with local emotional feature.

Table 5: Comparison of classification results on the FI dataset using different feature combinations.

Method	Accuracy (%)
C	67.8
$C + \{L_1\}$	68.20
$C + \{L_1, L_2\}$	68.67
$C + \{L_1, L_2, L_3\}$	69.12
$C + \{L_1, L_2, L_3, L_4\}$	67.52
$C + \{L_1, L_2, L_3, L_4, L_5\}$	67.13
C	67.8
$C + \{G_1\}$	68.54
$C + \{G_1, G_2\}$	69.30
$C + \{G_1, G_2, G_3\}$	70.23
$C + \{G_1, G_2, G_3, G_4\}$	68.74
$C + \{G_1, G_2, G_3, G_4, G_5\}$	68.31

Table 6: Comparison of classification results on the FI dataset using different loss functions.

Method	Accuracy (%)
AlexNet + L_{cls}	58.61
ResNet101 + L_{cls}	60.82
Inception-v4 + L_{cls}	60.75
Ours + L_{cls}	70.23
AlexNet + L_{subj}	60.32
ResNet101 + L_{subj}	62.77
Inception-v4 + L_{subj}	62.66
Ours + L_{subj}	**71.65**

4.4.4. The Choice of Parameter λ. In this work, parameter λ is used to control the weight of classification loss and sentiment distribution loss. When λ is set to 0, the proposed loss function is the cross-entropy loss, and λ is set to 1 and indicates that the proposed loss function is equal to KL loss essentially. Figure 5 shows the accuracy change under different values of parameterλ. When λ increases from 0 to 0.4, the classification performance has a significant improvement. However, when it further increases to more than 0.5, the classification accuracy begins to decrease. Figure 5 shows that when the weight of L_{ed} is set too large, it may lead too much ambiguity.

4.5. Compare with the Other Methods

4.5.1. Compare on Large-Scale Datasets. To further indicate the effectiveness of the proposed model, we compare it with the methods shown in Table 7. Our model has obviously achieved better results compared with the method based on manual features of SentiBank [14] through using hybrid representation features, which consist of global sematic, local visual, and low-level visual features. We can see that the performance of our model is better than those of CNN networks specifically proposed for object recognition tasks in Table 7, such as AlexNet [8], VGG-19 [9], and Inception-v4 [41]. Moreover, our model achieves better classification performance compared with the deep learning model proposed for image emotion classification, such as Yang et al. [47], MldrNet [17], and WSCNet [31], which shows the

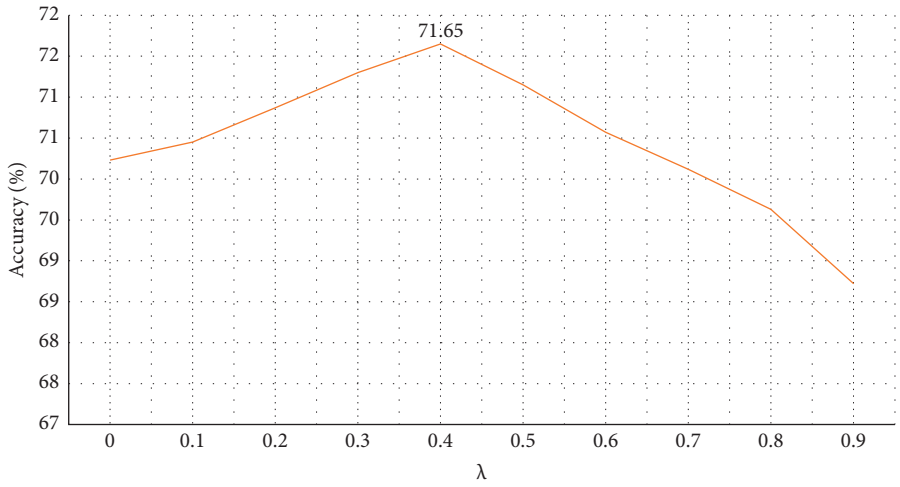


Figure 5: Impact of different λ on the FI dataset.

Table 7: Comparison of classification performance on the FI dataset.

Method	Accuracy (%)
AlexNet [8]	58.61
VGG-16 [9]	59.75
Inception-v4 [41]	60.75
MldrNet [17]	67.75
Yang [47]	67.64
WSCNet [31]	70.07
Ours	**71.65**

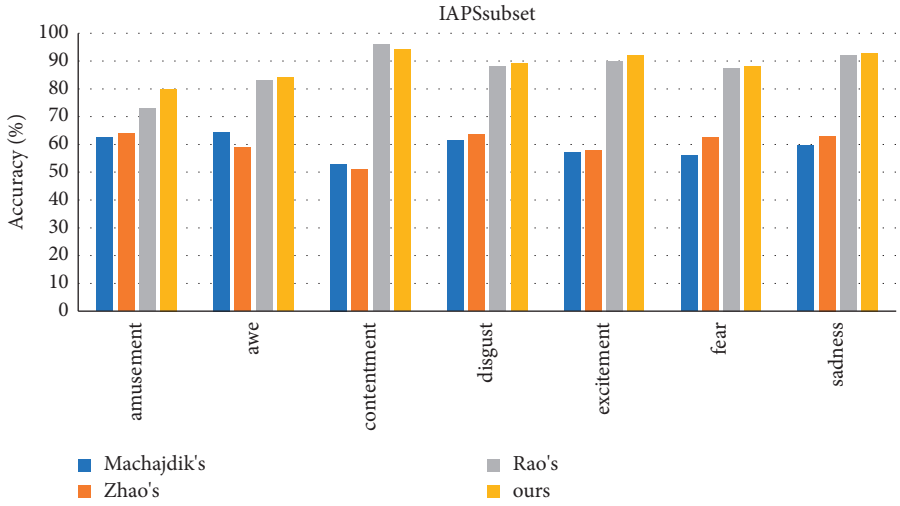


Figure 6: Performance evaluation on the IAPSsubset dataset.

effectiveness of our global and local hybrid representation features, as well as the effectiveness of our loss function.

4.5.2. Compare on Small-Scale Datasets. In order to verify the performance of the model more comprehensively, we also designed a comparative experiment on a small dataset, including IAPSsubset, Abstract, and ArtPhoto. Before the experiment, we randomly divided the image samples of each category in the dataset into 5 batches. Then, 5-fold cross validation is performed to obtain results. Especially, the emotion category anger has only 8 and 3 samples in the Abstract and IAPSsubset datasets, respectively, performing 5-fold cross validation is not enough. Therefore, the classification result of emotion anger on these two datasets is not reported. The experiment results are shown in Figures 6–8.

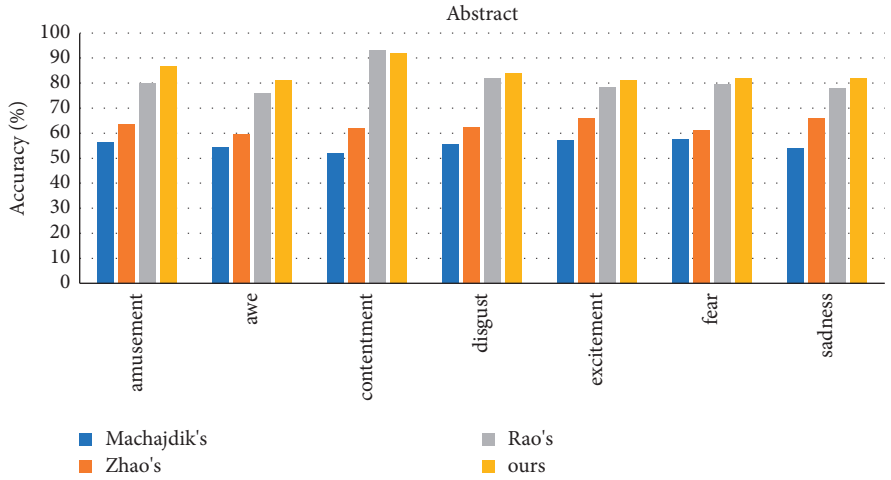


Figure 7: Performance evaluation on the Abstract dataset.

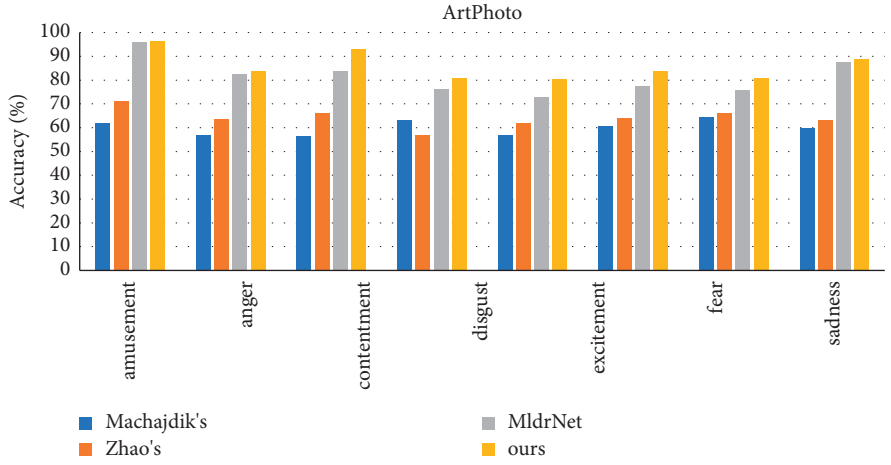


Figure 8: Performance evaluation on the ArtPhoto dataset.

Our method outperforms to Machajdik et al. [2], Zhao et al. [16], and MldrNet [17] in IAPSsubset, Abstract, and Artphoto.

5. Conclusions

In this paper, a CNN framework based on saliency detection and Gram matrix is proposed to improve image emotion recognition performance, and our method have been applied on many famous problems, including FI (Flickr and Instagram), IAPSsubset, ArtPhoto, and Abstract. The classification accuracies have been compared with those of other competing methods in the literatures, and the results show that our method has improved the image emotion recognition performance. Through experimental analyzing, it can be drawn that saliency detection, Gram matrix transformation, and new loss function are effective in increasing recognition accuracy, which indicates that the proposed method has potential application ability. In the future work, our main task is to integrate this improved CNN into the actual applications and conduct emotion recognition for video data automatically to better serve the society.

Data Availability

The datasets used in this study are Flickr and Instagram (FI) (https://onedrive.live.com/?authkey=%21AH57YMUbsP%2DqNls&cid=AB6522E29F6ED9A0&id=AB6522E29F6ED9A0%21101730&parId=AB6522E29F6ED9A0%21101729&action=defaultclick), Abstract (https://www.imageemotion.org/testImages_abstract.zip), IAPSsubset (https://www.csea.phhp.ufl.edu/media.html), and ArtPhoto(https://www.imageemotion.org/testImages_artphoto.zip).

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China, under Grant no.61977018, the Research Foundation of Education Bureau of Hunan Province of China, under Grant no. 16B006, the Hunan Provincial Natural Science Foundation of China, under Grant no.

2020JJ4626, and the Scientific Research Fund of Hunan Provincial Education Department of China, under Grant no. 19B004.

References

[1] Q. You, J. Luo, H. Jin, and J. Yang, "Building a large scale dataset for image emotion recognition: the fine print and the benchmark," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 308–314, Palo Alto, CA, USA, May 2016.

[2] J. Machajdik and A. Hanbury, "Affective image classification using features inspired by psychology and art theory," in *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 83–92, Firenze, Italy, October 2010.

[3] Y.-H. Chew, L.-K. Wong, J. See, H.-Q. Khor, and B. Abivishaq, "LiteEmo: lightweight deep neural networks for image emotion recognition," in *Proceedings of the 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, Kuala Lumpur, Malaysia, September 2019.

[4] W. Wang, Y. Yu, and J. Zhang, "Image emotional classification: static vs. dynamic," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, pp. 6407–6411, Hague, Netherlands, October 2004.

[5] D. T. Priya and J. D. Udayan, "Affective emotion classification using feature vector of image based on visual concepts," *The International Journal of Electrical Engineering & Education*, vol. 60, 2020.

[6] X. He and W. Zhang, "Emotion recognition by assisted learning with convolutional neural networks," *Neurocomputing*, vol. 291, pp. 187–194, 2018.

[7] 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. 20, p. 1, 2020.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[10] Y. Qiao, Y. Tian, Y. Liu, and J. Jiao, "Genetic feature fusion for object skeleton detection," *Security and Communication Networks*, vol. 2021, Article ID 6621760, 9 pages, 2021.

[11] Y. Wang, X. Cui, Z. Gao, and B. Gan, "Fed-scnn: a federated shallow-cnn recognition framework for distracted driving," *Security and Communication Networks*, vol. 2020, Article ID 6626471, 10 pages, 2020.

[12] B. Chen, X. Ju, B. Xiao, W. Ding, Y. Zheng, and V. H. C. De Albuquerque, "Locally GAN-generated face detection based on an improved Xception," *Information Sciences*, vol. 572, pp. 16–28, 2021.

[13] F. Cen, X. Zhao, W. Li, and G. Wang, "Deep feature augmentation for occluded image classification," *Pattern Recognition*, vol. 111, Article ID 107737, 2021.

[14] D. Borth, T. Chen, R. Ji, and S. F. Chang, "Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content," in *Proceedings of the 21st ACM International Conference on Multimedia*, pp. 459-460, Barcelona, Spain, October 2013.

[15] V. Yanulevskaya, J. C. Gemert, K. Roth, A. K. Herbold, N. Sebe et al., "Emotional valence categorization using holistic image features," in *Proceedings of the 2008 15th IEEE International Conference on Image Processing*, pp. 101–104, San Diego, CA, USA, October 2008.

[16] S. Zhao, Y. Gao, X. Jiang, H. Yao, T.-S. Chua, and X. Sun, "Exploring principles-of-art features for image emotion recognition," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 47–56, Orlando, FL, USA, November 2014.

[17] T. Rao, X. Li, and M. Xu, "Learning multi-level deep representations for image emotion classification," *Neural Processing Letters*, vol. 51, no. 3, pp. 2043–2061, 2020.

[18] X. Alameda-Pineda, E. Ricci, Y. Yan, and N. Sebe, "Recognizing emotions from abstract paintings using non-linear matrix completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5240–5248, San Francisco, CA, USA, August 2016.

[19] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, San Francisco, CA, USA, August 2016.

[20] M. Elad and P. Milanfar, "Style transfer via texture synthesis," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2338–2351, 2017.

[21] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," 2017, https://arxiv.org/abs/1701.01036.

[22] T. Rao, X. Li, H. Zhang, and M. Xu, "Multi-level region-based convolutional neural network for image emotion classification," *Neurocomputing*, vol. 333, pp. 429–439, 2019.

[23] F. Zhou, C. Cao, T. Zhong, and J. Geng, "Learning meta-knowledge for few-shot image emotion recognition," *Expert Systems with Applications*, vol. 168, Article ID 114274, 2021.

[24] D. T. Priya and J. D. Udayan, "Transfer learning techniques for emotion classification on visual features of images in the deep learning network," *International Journal of Speech Technology*, vol. 23, no. 2, pp. 361–372, 2020.

[25] X. Liu, N. Li, and Y. Xia, "Affective image classification by jointly using interpretable art features and semantic annotations," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 576–588, 2019.

[26] M. Solli and R. Lenz, "Color based bags-of-emotions," in *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, pp. 573–580, Münster, Germany, September 2009.

[27] KC. Peng, T. Chen, A. Sadovnik, and A. Gallagher, "A mixed bag of emotions: model, predict, and transfer emotion distributions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 860–868, Boston, MA, USA, June 2015.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the I2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, June 2009.

[29] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, January 2015.

[30] Q. You, H. Jin, and J. Luo, "Visual sentiment analysis by attending on local image regions," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 231–237, San Francisco, CA, USA, February 2017.

[31] J. Yang, D. She, Y.-K. Lai, P. L. Rosin, and M.-H. Yang, "Weakly supervised coupled networks for visual sentiment

analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7584–7592, San Francisco, CA, USA, August 2018.

[32] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: a discriminative regional feature integration approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2083–2090, San Francisco, CA, USA, August 2013.

[33] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5012–5024, 2016.

[34] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: contrast based filtering for salient region detection," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–740, Providence, RI, USA, June 2012.

[35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, San Francisco, CA, USA, August 2015.

[36] N. Liu, J. Han, and M. H. Yang, "Picanet: learning pixel-wise contextual attention for saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3089–3098, San Francisco, CA, USA, August 2018.

[37] N. Liu and J. Han, "Dhsnet: deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 678–686, San Francisco, CA, USA, August 2016.

[38] Q. Hou, M. -M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3203–3212, San Francisco, CA, USA, August 2017.

[39] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1741–1750, San Francisco, CA, USA, August 2018.

[40] H. Xiao, J. Feng, Y. Wei, M. Zhang, and S. Yan, "Deep salient object detection with dense connections and distraction diagnosis," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3239–3251, 2018.

[41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[42] S. Zhao, H. Yao, Y. Gao, R. Ji, and G. Ding, "Continuous probability distribution prediction of image emotions via multitask shared sparse regression," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 632–645, 2016.

[43] F. Pérez-Cruz, "Kullback-Leibler divergence estimation of continuous distributions," in *Proceedings of the 2008 IEEE International Symposium on Information Theory*, pp. 1666–1670, Toronto, Canada, July 2008.

[44] S. Siersdorfer, E. Minack, F. Deng, and J. Hare, "Analyzing and predicting sentiment of images on the social web," in *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 715–718, Firenze, Italy, October 2010.

[45] T. Rao, M. Xu, H. Liu, J. Wang, and I. Burnett, "Multi-scale blocks based image emotion classification using multiple instance learning," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 634–638, Phoenix, AZ, USA, August 2016.

[46] T. Chen, D. Borth, T. Darrell, and S. F. Chang, "Deepsentibank: visual sentiment concept classification with deep convolutional neural networks," 2014, https://arxiv.org/abs/1410.8586.

[47] J. Yang, D. She, and M. Sun, "Joint image emotion classification and distribution learning via deep convolutional neural network," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 3266–3272, Melbourne, Australia, August 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5511435, 10 pages
https://doi.org/10.1155/2021/5511435

Research Article

Detection of GAN-Synthesized Image Based on Discrete Wavelet Transform

Guihua Tang, Lei Sun, Xiuqing Mao, Song Guo, Hongmeng Zhang, and Xiaoqin Wang

Information Engineering University, Zhengzhou 450001, China

Correspondence should be addressed to Lei Sun; sl0221@sina.com

Received 25 January 2021; Revised 13 April 2021; Accepted 3 June 2021; Published 16 June 2021

Academic Editor: Beijing Chen

Recently, generative adversarial networks (GANs) and its variants have shown impressive ability in image synthesis. The synthesized fake images spread widely on the Internet, and it is challenging for Internet users to identify the authenticity, which poses huge security risk to the society. However, compared with the powerful image synthesis technology, the detection of GAN-synthesized images is still in its infancy and face a variety of challenges. In this study, a method named fake images discriminator (FID) is proposed, which detects that GAN-synthesized fake images use the strong spectral correlation in the imaging process of natural color images. The proposed method first converts the color image into three color components of R, G, and B. Discrete wavelet transform (DWT) is then applied to RGB components separately. Finally, the correlation coefficient between the subband images is used as a feature vector for authenticity classification. Experimental results show that the proposed FID method achieves impressive effectiveness on the StyleGAN2-synthesized faces and multitype fake images synthesized with the state-of-the-art GANs. Also, the FID method exhibits good robustness against the four common perturbation attacks.

1. Introduction

With the remarkable development of artificial intelligence (AI) and progress of high-performance computing hardware, image synthesis technology has evolved dramatically. The Internet users share a large number of multimedia contents on social media every day. It is challenging to identify authenticity of these contents, posing huge security risk to social. In particular, the generative adversarial networks (GANs) proposed in 2014 [1] have spawned a new type of image synthesis method. The images synthesized by four typical GANs are shown in Figure 1, which are really hard for humans to distinguish at the first glance. Besides, GAN's powerful image synthesis and editing capabilities bring new industrial value. For example, it can be used to create virtual characters, perform video rendering and sound simulation in film production, and create a new way of communication. However, security and privacy concerns are also raised. If these fake contents are disseminated as news materials, they will damage the reputation of news organizations and the public's confidence in the media and even mislead the public opinion and disturb the social order. The increasingly open network environment creates an ideal space for the spread of fake information. In the countries such as Britain and France, there have been cases of using deep-learning forgery technology to produce fake images, deceive the public to even conduct espionage. The hazard and impact of synthesized images has spread throughout the world, resulting in ethical, legal, and security problems. It is extremely urgent to find effective techniques for detection of fake images.

GAN-synthesized images show impressively high quality. Accordingly, the detection of GAN-synthesized images has become a hot research field. Various detection methods for GAN-synthesized images have been proposed successively [2–5] and achieved good results. However, with the increasing variety and quality of GAN-synthesized images, as well as the various perturbation attacks, these methods begin to expose their limitations.

To overcome the limitation in existing methods for detecting GAN-synthesized images, a method named fake images discriminator (FID) is proposed in this study. The FID method relies on both the discrete wavelet transform (DWT) and the standard correlation coefficient to extract the spectral correlation

Figure 1: Fake images synthesized with various GANs.

of natural color images. Besides, the support vector machine (SVM) was used for classification. Experimental results show that the FID method outperforms prior works of AutoGAN [6] and FakeSpotter [7] on StyleGAN2-synthesized faces and maintains robustness in tackling four common perturbation attacks. An additional experiment is conducted on images forged by other state-of-the-art (SOTA) GANs, and the FID method also achieves good effectiveness on multiple types of fake images.

The main contributions of this study are as follows:

(1) FID method: the fake images discriminator (FID) method employs the DWT and the standard correlation coefficient to detect fake images. Through the analysis of the imaging process of natural color images, it is found that the spectral correlation between RGBs can be utilized to distinguish GAN-synthesized images, which is also robust against the four common perturbation attacks at various intensities.

(2) The first comprehensive evaluation is on typical GAN-synthesized images. Experiments are conducted on high-quality fake images synthesized with SOTA GANs. These fake images include faces, buildings, animals, natural scenes, and so on. Experimental results indicate good effectiveness and robustness of the proposed FID method.

(3) Extensibility: the FID method is based on the imaging process of natural color images and the analysis on the difference between real and GAN-synthesized images. This difference may be widespread in fake images, and it could be extended to other AI-synthesized images and DeepFake.

The rest of the study is organized as follows. Section 2 reviews the related literature of GAN-synthesized images and detection methods. Section 3 describes the imaging process of digital images, followed by the presentation of the proposed FID method in Section 4. The experimental results and analysis are illustrated in Section 5. Section 6 concludes the study.

2. Related Work

Digital image forensics is a technology that distinguishes the authenticity, completeness, and source of image content. It mainly includes active forensics technology and passive (blind) forensics technology [8]. Active forensics is suitable for an image authentication scenario where digital signatures, digital watermarks, or digital fingerprinting have been embedded in digital images in advance. But in the actual environment, most images do not have embedded prior information, which limits the application of active forensics technology. Passive forensics does not require any prior information, and the images are identified based on the changes of image characteristics caused by the forgery operation. Currently, most of the detection methods for GAN-synthesized images conforms to the passive forensics. In the following sections, the latest developments in GAN-synthesized images and image forgery detection methods will be discussed.

2.1. GAN-Based Images Synthesis Methods. Generally, the GAN contains a generator and a discriminator. The generator synthesizes images and the discriminator differentiates between the fake and real images. The generator and discriminator play game mutually and finally achieve a dynamic balance. Since it is first proposed in 2014, the GAN has shown an impressive ability in image synthesis, the most studied area of GAN applications.

Entire face synthesis means that a facial image can be wholly synthesized with GANs, and the synthesized faces do not exist in the world. In entire face synthesis, the progressive growing of GANs (PGGAN) [9] and style-based generator architecture for GANs (StyleGAN) [10, 11], released by NVIDIA, produce an unprecedented high-quality and high-resolution entire synthesis face. As one of the models that can generate images with highest quality, StyleGAN has a new generator architecture proposed by NVIDIA. Without affecting other layers, the input of each layer is modified separately to control the visual features represented by each layer. CycleGAN [12] has achieved remarkable success in image-to-image conversion in two domains. Since each pair of image domains requires independent modeling, the scalability and robustness of CycleGAN are limited for processing of more than two domains. STGAN [13] and StarGAN [14] focus on face editing through manipulating the attributes and expressions of humans' faces, such as changing the color of hair, facial decorations, and expressions. StarGAN designed a generator of star structure to perform image-to-image conversion for multiple domains. The unified model architecture of StarGAN allows training datasets from multiple domains simultaneously in a single network. STGAN aims to improve the accuracy and quality of attribute manipulation. FaceApp, ZAO, and FaceSwap employ GANs to produce DeepFake which involves the swap of person's face [15, 16].

GANs can be applied in numerous aspects of image synthesis and swapping personal identities. In many cases, the fake images synthesized with SOTA GANs are nearly indistinguishable to humans. We cannot believe our eyes anymore in the media.

2.2. Detection of GAN-Synthesized Images. Traditional forensics-based techniques [17–19] usually analyze the traces inducted in image synthesis and inspect the pixel-level

disparities in real and fake images. Compared with traditional fake images, GAN-synthesized images have better quality, and no traces are inducted in image mosaic. Therefore, the effectiveness of these detection methods is greatly reduced. Also, these methods are sensitive to perturbation attacks like blur that is common in media images.

Nataraj et al. [3] built a pixel-level image detection model based on the deep neural network (DNN) and detected GAN-synthesized images by extract co-occurrence matrices on three color channels in the pixel domain. McCloskey et al. [2] found that the frequency of saturated pixels in GAN-synthesized images is limited due to the normalization operation in the generator. Also, the statistical relationship of color component of GAN-synthesized images is different from natural images. Though corresponding detection strategies are designed using these two clues, it is vulnerable to noise and adversarial examples attacks.

Another way to detect GAN-synthesized images is to learn the difference between real and fake images with DNN. Stehouwer et al. [20] introduced an attention mechanism to improve facial forgery detection and manipulated region localization. Wang et al. [21] used ResNet-50 to design a binary classifier to detect images synthesized by the convolutional neural network (CNN). Zhang et al. [6] explored the fingerprint of GAN [22] and proposed a classifier model named AutoGAN based on the input of frequency spectrum. AutoGAN identifies the artifacts inducted in the upsampling component of the GAN so as to realize the detection of GAN-synthesized images. The DNN-based methods [21, 23, 24] achieve better performance than the methods based on traditional image forensics and pixel-level differences. Other work explores various special features to study the disparities between real and synthesized facial images. For example, the uncoordinated facial features of the fake faces is exposed through the facial landmarks [4]. Lyu et al. [5] used the difference in head pose as the classification characteristic. However, GAN technology progresses rapidly, making the GAN features extracted by the above detection methods hard to keep good durability and universality. Besides, these works are vulnerable to common perturbation attacks, and robustness is essential for detecting fake images in the wild. The FakeSpotter proposed by Wang et al. [7] depends on monitoring neuron behaviors to spot AI-synthesized fake faces. This approach exhibited effectiveness on SOTA GANs and robustness against perturbation attacks.

3. Study on Spectral Correlation of Digital Imaging

Spectral correlation means the correlation existing between the three color components in finite neighboring pixels of color images. In the color imaging system, most consumer-grade digital cameras use one CCD or CMOS, and the imaging process of natural color image is shown in Figure 2.

The single-sensor camera obtains the color information of the image through a color filter array (CFA). The Bayer CFA is the most widely used array, using an alternate sampling mode, the RGB components are shown in Figure 3. The number of sampling in the G channel is twice of that in

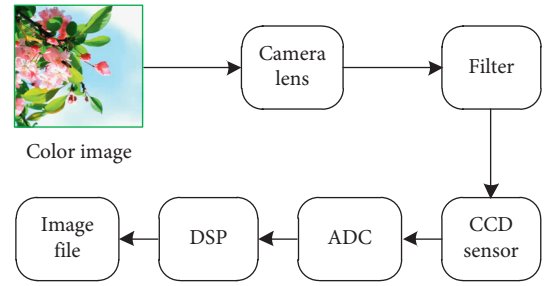


Figure 2: The single-CCD imaging process.

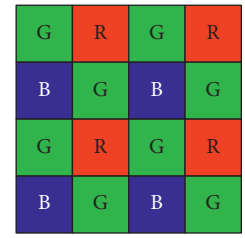


Figure 3: The Bayer CFA.

the R and B channels, which conforms to the spatial sensitivity of the human visual system to different spectral wavelengths. Since only one color component is captured per pixel, the CFA interpolation algorithm is needed to calculate the missing two color values at the pixel.

The main task of the CFA interpolation algorithm is the reconstruction of RGB images, specifically, to estimate the missing two color values from the neighborhood pixels. There are many CFA interpolation algorithms, such as the nearest neighbor, bilinear, bicubic, and convolution interpolation algorithms. These algorithms perform interpolation mainly in the neighborhood of a one-color channel. Taking bilinear algorithm as an example, each color component of R_1 is estimated as follows:

$$R_1 = R_1, \tag{1}$$

$$G(R_1) = \frac{G_1 + G_2 + G_3 + G_4}{4}, \tag{2}$$

$$B(R_1) = \frac{B_1 + B_2 + B_3 + B_4}{4}. \tag{3}$$

This example illustrates that the estimated color component is directly related to the value of the color pixels captured in the neighborhood, so there must be a strong spectral correlation between all RGB pixels of a real image. No matter which CFA interpolation algorithm is used to reconstruct the digital color image, all involve the neighborhood sampling values of 3 color components when estimating the missing color component, which leads to a strong spectral correlation existing in the R, G, and B channels.

Unlike the generation process of natural color images, the GAN trains the network with a large amount of data to synthesize images, which inevitably lead to the differences in some features, especially the spectral correlation between

RGB components of color images. To further prove the differences between GAN-synthesized images and real images, four types of GAN-synthesized images and real images, respectively, performed DWT in RGB channel, and the kernel density curve of transformed RGB components is shown in Figure 4.

Each figure includes three curves, representing the kernel density curve of the R, G, and B. The first row shows the RGB component distribution of the GAN-synthesized images; the RGB component of the second row is from the real images. It can be seen that the real image has similar kernel density curve on the three color channels, and the peaks and valleys appearing areas are highly coincident. The RGB components of the GAN-synthesized images are relatively independent, and the correlation cannot be clearly seen.

In conclusion, strong spectral correlation between RGB is caused by the interpolation operation in the color imaging process, while GAN-synthesized images do not have this characteristic. Therefore, GAN-synthesized images can be recognized based on this difference.

4. Our Method

The imaging process of natural color image causes high spectral correlation. In contrast, synthesizing fake images with the GAN can weaken or even eliminate this correlation. Consequently, the proposed method for detecting GAN-synthesized image employs wavelet multiscale decomposition to extract the correlation characteristics between the spectra of RGB channels. The FID method includes two stages of feature extraction and classification. The block diagram of this method is shown in Figure 5.

4.1. Features Extraction. DWT can decompose an image into subband coefficients that represent different direction information in same scale. Decomposing the two-dimensional image $f(x, y)$ with DWT, it can obtain

$$f(x, y) = W_j^A + \sum_{k \geq j} \left(W_k^H + W_k^V + W_k^D \right), \tag{4}$$

where W^A is the low-frequency approximation under scale j, and W_k^i, $i = \{H, V, D\}$, $k \geq j$ is the detailed component in the horizontal, vertical, and diagonal directions under different scales of the image. The multiresolution decomposition capability of pyramid wavelet transform can decompose the image information layer by layer, so it is widely used to extract image features, especially the statistical features in the spatial domain.

DWT is utilized to construct the correlation between the frequency spectrums of images in the three color spaces. Also, the correlation coefficient is used to measure the constructed correlation. The specific feature extraction process is described as follows:

(1) RGB channels separation: since a stronger statistical correlation of the three color components exists in the RGB color space. The color image is first converted into the three independent color components of R, G, and B.

(2) DWT: each color component is decomposed by level-1 DWT and divided into four subband images (plus the low-frequency approximation itself). Therefore, 12 subband images can be obtained from a color image.

(3) Calculate the correlation coefficient matrix F_{NCC}. The co-correlation coefficient is a basic measure of correlation. The standard correlation function is used to measure the correlation between the subband images of the three color components. The detailed calculation process is shown in Figure 6.

The correlation coefficient $NCC(I_1, I_2)$ corresponds subband image of two color components, and its calculation is shown in equation (2). After calculating all wavelet subband images, 3 correlation coefficient matrix F_{NCC} can be obtained.

$$NCC(I_1, I_2) = \frac{\sum_{I_1} \sum_{I_2} (I_1 - E(I_1))(I_2 - E(I_2))}{\sqrt{\sum_{I_1} (I_1 - E(I_1))^2} \sqrt{\sum_{I_2} (I_2 - E(I_2))^2}}. \tag{5}$$

$E(I_1)$ and $E(I_2)$ in equation (2) are the means of gray images I_1 and I_2, respectively. The calculation is shown in equation (3). $M \times N$ is the image size.

$$E(I) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} I(i, j). \tag{6}$$

(4) Extracting matrix feature: by calculating the four matrix features (kurtosis, mean, skewness, and standard deviation) of real and GAN-synthesized images separately, it is found that the real and GAN-synthesized images have the largest difference in kurtosis feature, which can better distinguish the real and GAN-synthesized images. The experimental results are shown in Figure 7.

The experimental results show that the difference between real and GAN-synthesized images in kurtosis is the largest. Therefore, the kurtosis ku of F_{NCC} is chosen as the final measurement for spectrum correlation of the color image, and its calculation is shown as follows:

$$ku = \frac{E\left[\left(f(i, j) - (1/M \times N) \sum_{i=1}^{M} \sum_{j=1}^{N} f(i, j) \right)^4 \right]}{\left(E[(f(i, j) - \mu)2] \right)^2}, \tag{7}$$

where $f(i, j)$ represents the element of F_{NCC}, and the size of F_{NCC} is $M \times N$. Three kurtosis values can be obtained by calculating the kurtosis of the correlation matrix F_{NCC} (RG), F_{NCC} (RB), and F_{NCC} (GB), respectively.

4.2. Classification. SVM is commonly used for pattern recognition, classification, and regression analysis. LibSVM [25] is a tool library for SVM developed by Professor Chih-Jen Lin in 2001, which can be used for data classification or

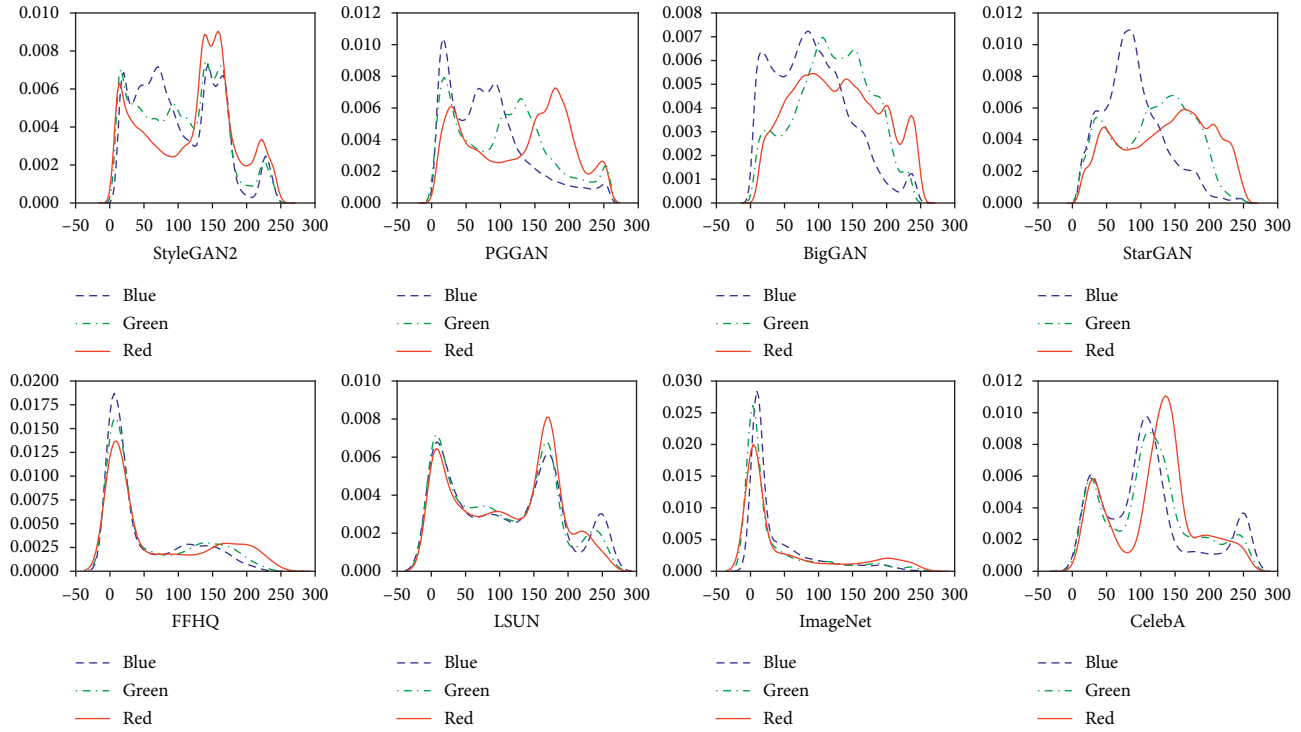


Figure 4: The kernel density curve for different color components.

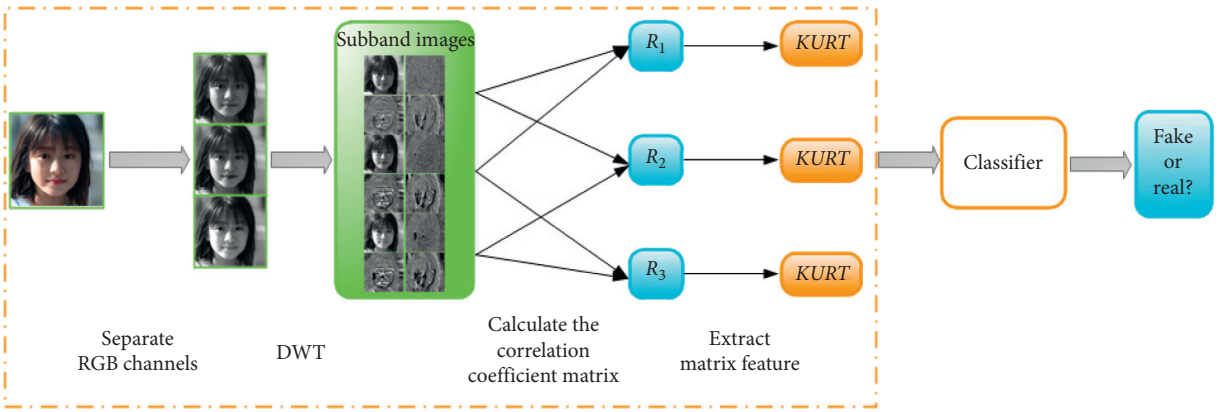


Figure 5: Block diagram of the proposed method.

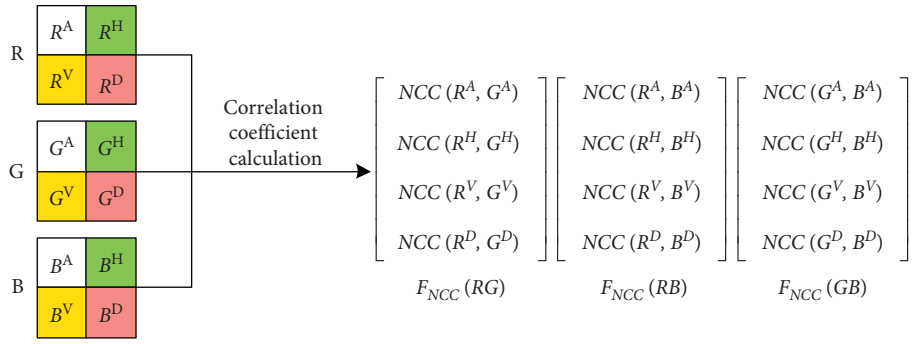


Figure 6: Calculation of the correlation coefficient matrix.

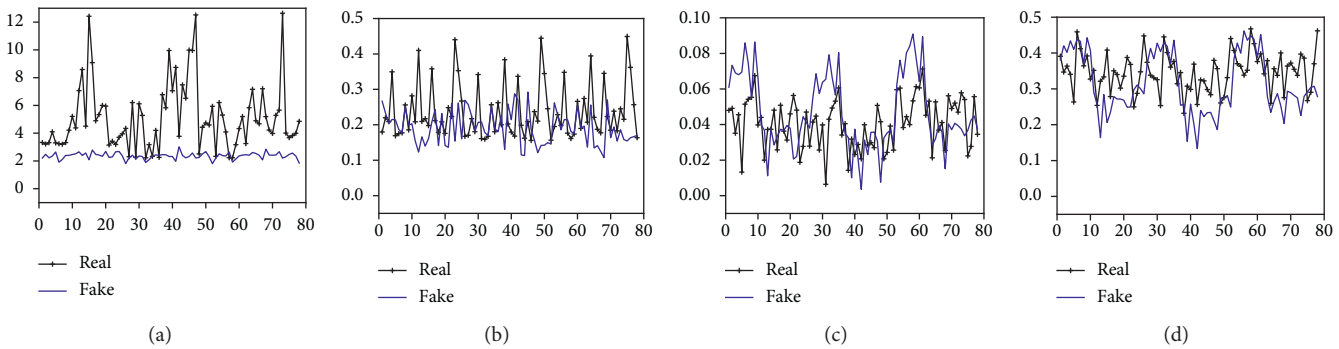


Figure 7: Matrix features of real and GAN-synthesized images. (a) Kurtosis. (b) Mean. (c) Skewness. (d) Standard deviation.

regression conveniently. Since the focus of this study is to employ DWT for feature extraction, there is no special requirement for the classification, and the final feature used for classification is a set of three-dimensional vectors in a simple form. Therefore, LibSVM is used in this study to implement a simple binary classifier, and the radial basis function (RBF) kernel is used to train the SVM for classification.

$$\kappa\left(x_i, x_j\right) = e^{-g\left\|x_i - x_j\right\|^2}, \quad g > 0, \tag{8}$$

where x_i, x_j is a vector; g is the only hyperparameter of RBF; $\|x_i - x_j\|$ indicates the vector norm. The grid-search method is used to optimize the parameters.

5. Result and Analysis

In this section, experiments are conducted to evaluate the effectiveness of the proposed FID method in detecting GAN-synthesized images and its robustness against the four common perturbation attacks. First, experiments are conducted on StyleGAN2-synthesized faces, and the results are compared with that of recently published work, i.e., AutoGAN and FakeSpotter.

5.1. Experimental Setup

5.1.1. Data Collection. For the experiment, real faces are collected from CelebFaces Attributes Dataset (CelebA) [26] due to its good diversity. StyleGAN2 is used to synthesize fake faces. To ensure the diversity and high-quality of the fake image dataset, the various images produced by other newest GANs (e.g., StarGAN and PGGAN) are used. Table 1 presents statistics of the collected fake image dataset from [21]. The first column shows the data type, where variety means that there are more than ten different types of fake images (e.g., building, animals, airplane, and so on). The second column denotes the source of real faces for synthesizing fake images. The last column indicates the source of synthesized fake images, released by official, collected from online, or synthesized by ourselves.

5.1.2. Implementation Details. Binary classifier is implemented by LibSVM for detecting fake images, and the kernel function is RBF. The training dataset includes 5,000 real and 5,000 StyleGAN2-synthesized faces and 1,000 real and 1,000 StyleGAN2-synthesized faces for test. The training dataset and the test dataset are employed for evaluating the effectiveness and robustness of the FID method. Four common perturbation attacks are selected to evaluate the robustness, namely, compression, blur, resizing, and adding noise.

Table 1: Dataset description.

Fake images	GAN type	Collection	Total
Entire synthesis faces	StyleGAN2	Officially released	6 k
Bedroom Cat Car	StyleGAN	[21]	12.0 k
Variety Apple Horse Orange Summer Winter Zebra	BigGAN [27]	[21]	4.0 k
	CycleGAN	[21]	2.6 k
Variety	GauGAN [28]	[21]	10.0 k
Variety	PGGAN	[21]	8.0 k
Variety	PixelRNN	Self-synthesis	3 k
Edited faces	StarGAN	[21]	4.0 k
Edited faces	DiscoGAN	Self-synthesis	1.2 k

5.1.3. Evaluation Metrics. In detecting StyleGAN2-synthesized faces, eight popular metrics are adopted to obtain a comprehensive performance evaluation of the FID method. Also, the performance is compared with prior works, i.e., AutoGAN and FakeSpotter. Specifically, the precision, recall, F1-score, accuracy, AP (average precision), AUC (area under curve of receiver operating characteristics), FPR (false-positive rate), and FNR (false-negative rate) are reported. The AUC is also used as a metric to evaluate the performance of the FID method in tackling the four perturbation attacks and detecting other GANs-synthesized images.

5.2. Detection Performance. In the section, the influence of DWT levels for detecting StyleGAN2-synthesized face is first explored. In the feature extraction stage, 1000 real and 1000 StyleGAN2-synthesized faces are subjected to multilevel

DWT, and the AUC score is adopted to evaluate the performance. The experimental results are shown in Figure 8. The overall value of AUC fluctuates with the increase of the DWT level. The AUC score is the highest when the DWT level equals 1, so the level-1 DWT is selected to extract the spectral correlation.

The performance of the three methods, i.e., the FID, AutoGAN, and FakeSpotter, in detecting StyleGAN2-synthesized faces is measured, and the result is given in Table 2. AutoGAN is an open-source work published in 2019 that exploits the artifacts in GAN-synthesized images and detects the fake images with a classifier based on the deep neural network. FakeSpotter spots AI-synthesized fake faces through monitoring the neuron behaviors. Experimental results demonstrate that the FID method outperforms AutoGAN and FakeSpotter for all eight metrics, achieving competitive performance with a high detection rate and low false alarm rate in detecting the StyleGAN2-synthesized faces.

To illustrate the performance of the FID method in balancing the precision and recall, the precision and recall curves are presented in Figure 9 as well. The proposed method achieves a good balance between precision and recall on StyleGAN2-synthesized faces.

5.3. Robustness Analysis. Since image transformations are common, especially in the social media, the objective of robustness analysis is to evaluate the capabilities of the FID method against perturbation attacks. Four different perturbation attacks (compression, blur, resizing, and adding noise) under different intensities are used for evaluation, and the AUC is taken as a metric for the performance evaluation.

As for the four perturbation attacks, the compression quality measures the intensity of compression. 0 and 100, respectively, are the maximum and minimum values. Blur indicates that the Gaussian blur is employed to faces. The value of Gaussian kernel standard deviation is adjusted to control the intensity of blur, and the Gaussian kernel size is (3, 3). In resizing, the scale factor is applied to control the size of an image in horizontal and vertical axes. The Gaussian additive noise is added to produce noisy images, and the variance is used for controlling the intensity of the noise.

The experimental results of the FID method against the four common perturbation attacks are shown in Figure 10. As the intensity of perturbation attacks increases, the AUC score of the FID method fluctuates within a small range. Due to the interpolation and quantization operations in the resizing and compression, the pixel relationship in finite neighborhood changes, making a relatively obvious variation. The FID method achieves an AUC score of about 80% and more than 85% for tackling the compression and resizing attacks, respectively. Besides, the AUC score of the FID method is more than 95% for tackling the blur and noise attacks under different intensities.

Similarly, the proposed FID method is evaluated on other GANs-synthesized image datasets, which contain rich image types, and the results are compared with AutoGAN; the training datasets and the test datasets were divided in 5 to 1.

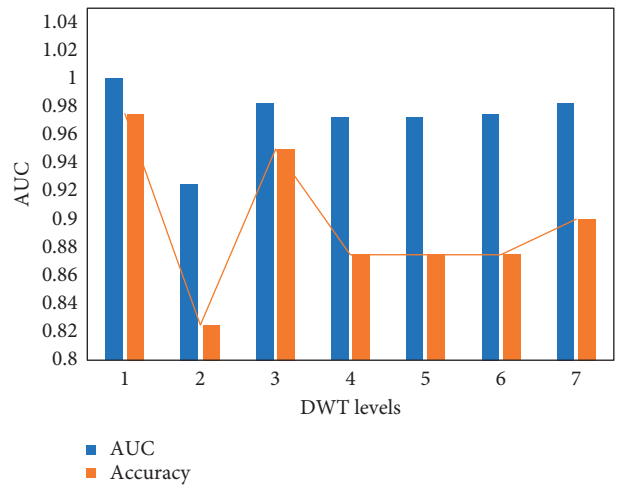


Figure 8: AUC score under multilevel DWT.

The AP score is also taken as a metric for performance evaluation, and the experimental results are given in Table 3. It can be seen that the FID method always maintains a good performance for different types of images synthesized with SOTA GANs. Because the pretrained model was trained on CycleGAN and StarGAN, AutoGAN obtained 100% AP on CycleGAN and StarGAN. DiscoGAN and CycleGAN have a similar architecture, so AutoGAN also achieved a good performance on DiscoGAN. While on other GANs, except BigGAN, FID has achieved better performance compared to AutoGAN. The performance of the FID method in detecting images synthesized by BigGAN, PGGAN, and StyleGAN are not as high as other types of fake images. The reason for the inferior performance could be that the fake images synthesized by BigGAN and PGGAN involve more image types and more complicated image content; thus, the feature vector for classification is more scattered in the hyperplane. FID got a relatively low AP on StyleGAN, because StyleGAN-synthesized image has high quality and contains three types, more difficult to detect. Although GauGAN also contains a variety of images, the quality of the images is not good, and AP arrives at 91.22%. The AP of detecting other types of fake images is also above 90%. According to the experimental results, the detection of fake images with complex types is still challenging.

5.4. Discussion. The proposed FID method achieves impressive effectiveness in detecting SOTA GANs-synthesized images. Also, the method exhibits satisfactory robustness against the four common perturbation attacks. Since the compression attack changes the pixel relationship in the finite neighborhood and affects the spectral correlation of color images, the performance degradation of the FID method under compression attack is relatively obvious.

However, the FID method also has some limitations. For example, the performance of detecting fake images of multiple types is inferior than that of a single type. The content in fake images of multiple types is quite different, making the distribution of the extracted feature vectors in

Table 2: Performance of FakeSpotter, AutoGAN, and FID.

	Precision	Recall	F1	Accuracy	AP	AUC	FPR	FNR
FID	0.9845	0.9845	0.9845	0.9845	0.9889	0.9901	0.021	0.015
FakeSpotter	0.912	0.924	0.918	0.919	0.881	0.919	0.076	0.087
AutoGAN	0.757	0.663	0.707	0.725	0.67	0.725	0.033	0.213

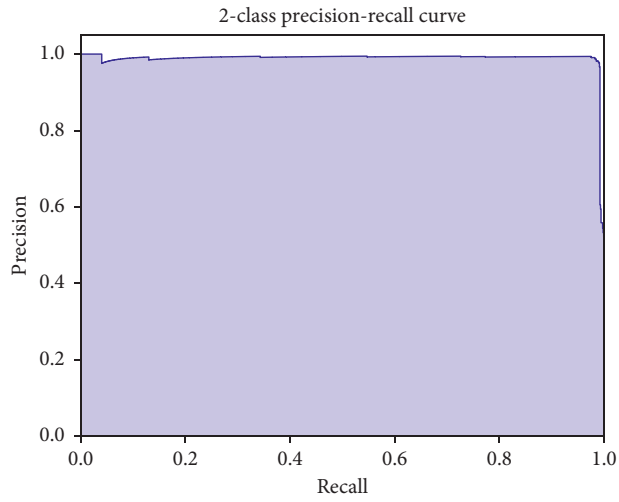


Figure 9: Precision-recall curves of StyleGAN2-synthesized faces.

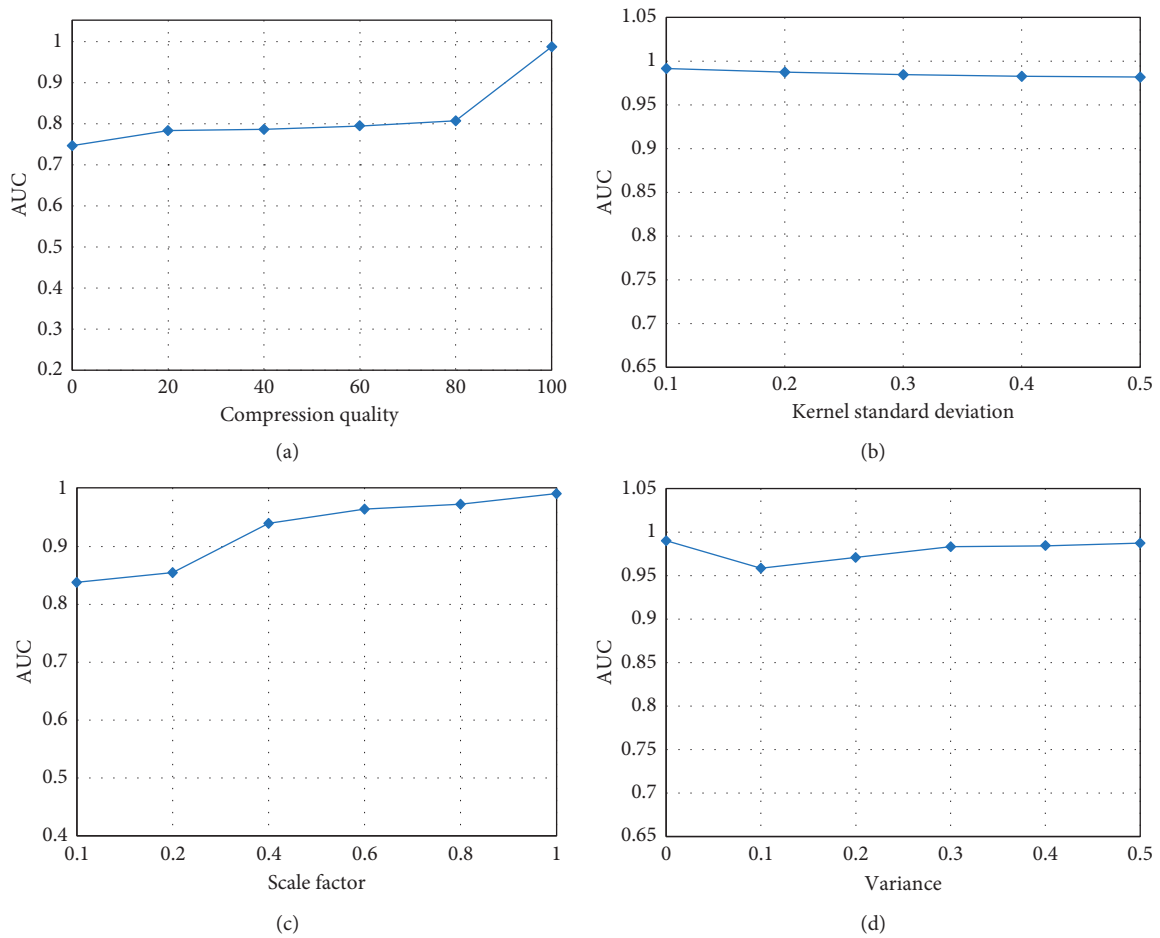


Figure 10: Four perturbation attacks under different intensities. (a) Compression. (b) Blur. (c) Resizing. (d) Noise.

Table 3: Performance of AutoGAN and FID.

	StyleGAN	PGGAN	BigGAN	CycleGAN	StarGAN	GauGAN	PixelRNN	DiscoGAN
FID	85.33	76.32	75.10	96.19	99.71	91.22	90.33	95.23
AutoGAN	68.60	75.60	84.90	100.00	100.00	61.00	71.01	95.10

the hyperplane more scattered. This brings challenges to the classification and inevitably leads to a declined detection effect. The detection of multitype fake images may be a trend in the future, which poses a challenge and calls for effective approaches.

6. Conclusion and Future Research Directions

The rapid development of AI technology makes it possible to produce fake content (e.g., fake audio, fake video, and fake image) that can deceive humans, posing potential challenges to the society and people. This study proposes a method for detecting GAN-synthesized fake images based on DWT and the standard correlation coefficient. Also, the RGB correlation introduced in the imaging process of natural color images are studied. Besides, an extensive evaluation of the FID method on detecting fake images synthesized by StyleGAN2 and several typical SOTA fake images is performed. Experimental results show that the proposed method achieves effectiveness in detecting GAN-synthesized fake images and exhibits robustness against common perturbation attacks. Furthermore, the analysis on the difference between real and fake images in the image imaging process could be extended to other AI-synthesized images.

The research on forgery and fake detection is fundamental, and it is necessary to establish a powerful defense mechanism to avoid AI risks. Currently, the face swap is common with DeepFake, and application of the FID method to DeepFake could be our future work.

Data Availability

The related images used to support the findings of the study are at https://github.com/NVlabs/stylegan2 and https://github.com/peterwang512/CNNDetection. The source codes will be uploaded to GitHub and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2016YFB0501900).

References

[1] J. P.-A. Goodfellow and M. Mirza, "Generative adversarial nets," in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, June 2014.

[2] S. McCloskey and M. Albright, "Detecting GAN-generated imagery using saturation cues," in *Proceedings of the 26th IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, China, September 2019.

[3] L. Nataraj, "Detecting GAN generated fake images using Co-occurrence matrices," *Journal of Electronic Imaging*, vol. 5, pp. 1–7, 2019.

[4] X. Yang, Y. Li, H. Qi et al., "Exposing GAN-synthesized faces using landmark locations," in *Proceedings of the 7th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, Paris, France, July 2019.

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

[6] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," in *Proceedings of the 11th IEEE International Workshop on Information Forensics and Security (WIFS)*, Delft, The Netherlands, December 2019.

[7] R. Wang, L. Ma, F. Juefei-Xu et al., "Fakespotter: a simple baseline for spotting ai-synthesized fake faces," in *Proceedings of the 29th International Joint Conference on Artifical Intelligence (IJCAI)*, Yokohama, Japan, July 2020.

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

[9] T. Karras, T. Aila, S. Laine et al., "Progressive Growing of GANs for improved quality, stability, and variation," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, May 2018.

[10] T. Karras, S. Laine, T. Aila et al., "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Angeles, USA, June 2019.

[11] T. KarrasS. Laine et al., "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.

[12] J.-Y. Zhu, T. Park, P. Isola et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[13] M. Liu, Y. Ding, M. Xia et al., "STGAN: a unified selective transfer network for arbitrary image attribute editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Angeles, USA, June 2019.

[14] Y. Choi, M. Choi, M. Kim et al., "StarGAN: unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, June 2018.

[15] S. Agarwal, H. Farid, Y. Gu et al., "Protecting world leaders against deep fakes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Los Angeles, USA, June 2019.

[16] S. Cole, "We are truly F—ed: everyone is making AI-generated fake porn now," 2018, https://www.vice.com/en us/article/bjye8a/reddit-fake-porn-app-daisy-ridley/.

[17] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, 2005.

[18] H. Wang and H. Wang, "Perceptual hashing-based image copy-move forgery detection," *Security and Communication Networks*, vol. 2018, pp. 1–11, Article ID 6853696, 2018.

[19] E. Gürbüz, G. Ulutas, and M. Ulutas, "Detection of free-form copy-move forgery on digital images," *Security and Communication Networks*, vol. 2019, pp. 1–14, Article ID 8124521, 2019.

[20] H. Dang, F. Liu, J. Stehouwer et al., "On the detection of digit manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.

[21] S.-Y. Wang, O. Wang, R. Zhang et al., "CNN-Generated images are surprisingly easy to spot for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.

[22] F. Marra, D. Gragnaniello, L. Verdoliva et al., "Do GANs leave artificial fingerprints?" in *Proceedings of the 2th IEEE International Conference on Multimedia Information Processing and Retrieval (MIPR)*, San Jose, USA, March 2019.

[23] H. Mo, B. Chen, and W. Luo, "Fake faces identification via convolutional neural network," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, New York, USA, June 2018.

[24] T. Do Nhu, N. In Seop, H.-J. Yang et al., "Forensics face detection from GANs using convolutional neural network," in *Proceedings of the International Symposium on Information Technology Convergence (ISITC)*, Chonbuk National University, South Korea, October 2018.

[25] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[26] Z. Liu, P. Luo, X. Wang et al., "Deep Learning face attributes in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, June 2015.

[27] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, USA, April 2019.

[28] T. Park, M. Y. Liu, T. C. Wang et al., "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Angeles, USA, June 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5383573, 9 pages
https://doi.org/10.1155/2021/5383573

Research Article

Driver Fatigue Detection Based on Facial Key Points and LSTM

Long Chen,[1] Guojiang Xin,[2] Yuling Liu,[1] and Junwei Huang[3]

[1]*College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China*
[2]*School of Informatics, Hunan University of Chinese Medicine, Changsha 410208, China*
[3]*HERE North American LLC, Burlington 01803, VT, USA*

Correspondence should be addressed to Guojiang Xin; lovesin_guojiang@126.com

Received 14 April 2021; Accepted 5 June 2021; Published 14 June 2021

Academic Editor: Beijing Chen

In recent years, fatigue driving has been a serious threat to the traffic safety, which makes the research of fatigue detection a hotspot field. Research on fatigue recognition has a great significance to improve the traffic safety. However, the existing fatigue detection methods still have room for improvement in detection accuracy and efficiency. In order to detect whether the driver has fatigue driving, this paper proposes a fatigue state recognition algorithm. The method first uses MTCNN (multitask convolutional neural network) to detect human face, and then DLIB (an open-source software library) is used to locate facial key points to extract the fatigue feature vector of each frame. The fatigue feature vectors of multiple frames are spliced into a temporal feature sequence and sent to the LSTM (long short-term memory) network to obtain a final fatigue feature value. Experiments show that compared with other methods, the fatigue state recognition algorithm proposed in this paper has achieved better results in accuracy. The average accuracy of the proposed method in detecting key points of the face is as high as 93%, and the running time is less than half of the ordinary DLIB method.

1. Introduction

Automobiles have become the most popular tools of transportation. As the frequency of automobile use continues to increase, traffic accidents are also increasing. In many traffic accidents, fatigue driving is one of the main reasons. Fatigue driving has caused many major traffic accidents, which caused huge losses to people's lives and properties.

Relevant Chinese traffic laws stipulate that driving for 4 hours without a break is fatigue driving. In a survey in the United States, more than half of the drivers admitted that they had fatigue driven [1]. When a driver is fatigued, his concentration, judgment ability, and reaction sensitivity are reduced [2]. These factors will make traffic accidents more likely to occur. Long-distance driving is the most prone to fatigue driving and often causes the safety accidents. Therefore, fatigue driving detection technology has become a research hotspot in the field of the traffic safety.

At present, fatigue detection methods are divided into the following categories: methods based on the physiological information, methods based on the vehicle status, methods based on the computer vision, and methods based on the information fusion models [3].

Physiological information mainly refers to the driver's breathing rate, pulse, blood pressure, and heart rate. These parameters can quickly and accurately reflect a person's physical and mental state. The detection methods based on the physiological information not only have strong real-time performance but also have high accuracy [4]. However, the driver needs to wear related equipment during the detection process, which will affect the normal operation of the driver, so that the practical applications are limited. The status of the vehicle refers to the vehicle's trajectory, steering wheel manipulation, and lane deviation. These detection methods indirectly analyze the driver fatigue state by analyzing vehicle information [5]. The main disadvantage of these methods is low accuracy. The detection methods based on the computer vision can quickly and accurately detect the driver fatigue state by capturing and analyzing the driver's face video in real-time. These methods do not need the driver to wear the related equipment and have good

performance in the terms of detection rate and reliability. The main difficulty of these methods is face image processing. Information fusion methods are the comprehensive use of the physiological information, vehicle information, and computer vision algorithms to detect the driver's fatigue state. The advantage is that it can improve the accuracy of the detection, but the disadvantage is that it is difficult to establish an information fusion model and obtain various information.

The main contribution of this paper is to propose a new, high-precision, real-time fatigue detection method based on the computer vision. We combine MTCNN and DLIB together, which allows us to extract the facial features fast and accurately and then combine the facial features of multiple frames to make our fatigue judgment results more accurate. This method first divides the video into image frames and cuts out the facial area through MTCNN and then uses the DLIB library to extract the fatigue features of the eye and mouth for each image frame. Finally, multiple frames of the fatigue feature are input into the recognition network based on LSTM to obtain fatigue judgment results.

2. Related Work

In recent years, many scholars and institutions have conducted a lot of researches on the fatigue driving detection based on the computer vision.

D'orazio et al. proposed an algorithm by eye detection. The algorithm used iris geometric information to determine the entire image [6]. Sun et al. studied the relationship between the closed eyes and the fatigue, and they used PERCLOS to detect the driver's fatigue and obtained better test results [7]. Ma et al. designed a system to detect the fatigue driving state at night. They used a deep framework based on ConNN and verified it on their own dataset [8]. Zhang et al. created a model to solve the influence of the sunglasses on the fatigue detection, which used the IRF dataset [9]. Gupta et al. observed the facial features of the driver through a camera and classified the fatigue levels through principal component analysis and support vector machine (SVM) classifier [10]. Junaedi and Akbar calculated PERCLOS by detecting the eyes and used it to judge the fatigue. They used the YawDD dataset [11]. Savas and Becerikli tried to use the SVM algorithm to detect driver fatigue. In their study, they used the number of yawns, the internal area of the mouth, and the number of blinks to determine the driver fatigue level on the dataset [12]. Amodio et al. designed a driver state detection system based on pupil light reflection. They used the pupil size contour and SVM classifier to judge the driver's state [13]. Li et al. designed a human behavior recognition classification system based on ConNN. They proposed a face recognition algorithm based on LBP-EHMM [14]. Liu et al. proposed a driver fatigue detection algorithm using a two-stream network model with multiple facial features. They applied gamma correction to enhance the image contrast to obtain better results [15]. Savaş and Becerikli proposed a multitask convective neural network model to detect driver drowsiness/fatigue. The features of the eyes and mouth were used to model the behavior of the driver. The changes in these characteristics were used to monitor the driver's fatigue [16]. Liu et al. proposed a fatigue detection algorithm based on the deep learning facial expression analysis. They trained a facial key point detection model through multiple local binary patterns and AdaBoost classifiers [17]. Ed-Doughmi et al. proposed a method to analyze and predict driver drowsiness by applying a recurrent neural network on the driver's face in sequence frames. They used a 3D convolutional network based on a repetitive neural network architecture of a multilayer model to detect the driver's drowsiness [18].

Yawning and frequent blinking are the most obvious signs of driver fatigue. Therefore, the first task is to determine the human eyes' state and mouth's state. There are generally two ways to detect the eyes and mouth. One is to directly detect the positions of the eyes and mouth. The other is to firstly find the facial area and then detect the positions of the eyes and mouth. The human face has more information, and the features are more stable than the human eyes. Cutting out the face area can reduce the test range of the eye position and avoid the interference of the background.

The existing face detection algorithms can be divided into two categories: one is a multilevel detection algorithm based on the proposed region. The other is the target detection algorithm based on anchor frame [19]. The representative algorithms of the former are Faster-RCNN [20] and MTCNN [21]. The representative algorithms of the latter are S3FD [22] and SSH [23]. Compared with traditional learning methods, detection methods based on deep learning do not require manual feature extraction. With the support of a large amount of training data, the detection performance will be greatly improved.

Fatigue driving is a continuous behavior. Therefore, the fatigue detection method based on continuous multiple frames will definitely be better than the single frame method. Donahue proposed the LRCN framework [24], which can process continuous multiple frames of relevant information to perform behavior recognition and classification.

3. Methodology

The framework proposed in this paper is shown in Figure 1. We will introduce the implementation details of each part in detail.

3.1. Face Detection. In this task, we use MTCNN for face detection, which is based on deep learning and can quickly and efficiently complete face detection and face alignment [25]. MTCNN can detect five key points of the face: left and right corners of the mouth, nose, and left and right eyes. However, the five key points are not enough to extract facial fatigue information, so we use MTCNN just for face detection. MTCNN includes three subnets: proposal network (P-Net), refine network (R-Net), and output network (O-Net). MTCNN is composed of cascades of them [26].

3.1.1. P-Net. The main task of this network is to obtain the bounding box and regression vector of the candidate window. After the candidate window is calibrated, nonmaximum

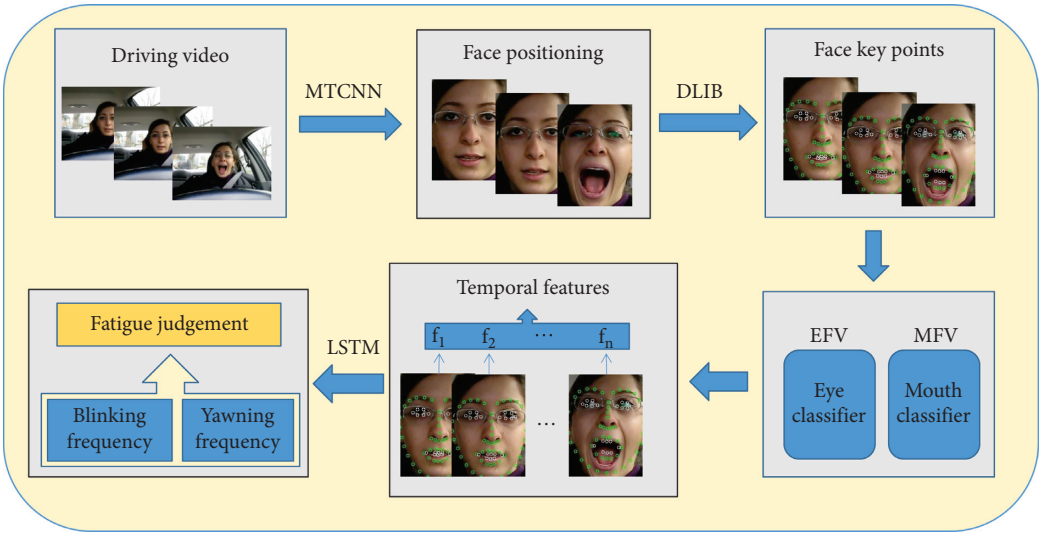


Figure 1: Framework of fatigue detection.

suppression is used to eliminate highly overlapping windows. P-Net is a regional proposal network for face regions. The network uses a face classifier to determine whether there is a face in the area and uses border regression and a locator of facial key points to make a preliminary proposal of the face area. This part will output many candidate windows and use these windows as the input of R-Net.

3.1.2. R-Net. The main task of this network is to eliminate false samples and continue to obtain bounding boxes and regression vectors. Unlike the previous network, R-Net has a more complete connection layer. When the test sample passes the P-Net layer, many candidate windows are gotten. The network will filter out a large number of wrong candidate windows. Finally, bounding-box regression and non-maximum suppression (NMS) were performed on the selected candidate boxes to further optimize the prediction results.

3.1.3. O-Net. This network is more complicated than the first two networks. O-Net has a 256 fully connected layer. After further filtering the candidate window of R-Net, this layer of network will also calculate the position of the facial feature points. In addition, this operation can eliminate the influence of some obstructions, such as sunglasses, hats, and ordinary glasses.

3.2. Facial Key Point Detection. In this phase of the task, we use DLIB to label the key points of the face. DLIB can be regarded as a machine learning toolbox, which is designed to solve the extraction of key points of human faces. DLIB has received widespread attention once it is launched, and it can be applied to mobile devices or large-scale high-performance computing environments. Like many open-source libraries, DLIB can be used by researchers for free. We choose the DLIB library because it can provide training and extraction tools for 68 facial key points. We can use it to obtain 68 facial key points and use these key points to extract fatigue features [27].

3.2.1. Closed-Eye Detection. Obviously, when people's eyes are open, the distance between the upper and lower feature points of the eyes will be relatively large. When the eyes are closed, the distance becomes smaller. The EYE value is calculated by using the distance of the eye feature points. Among the 68 feature points on the face, the eye points correspond to 37–42 and 43–48, respectively. Figures 2 and 3, respectively, show the state of open and closed eyes.

The calculation formula of EYE values is as follows:

$$\mathrm{EYE} = \frac{\|P_{38} - P_{42}\| + \|P_{39} - P_{41}\|}{2\|P_{37} - P_{40}\|}. \tag{1}$$

The numerator represents the Euclidean distance between the vertical feature points of the eyes, and the denominator is the Euclidean distance between the horizontal feature points of the eyes. The Euclidean distance between two points is calculated as follows:

$$\mathrm{Dis}(a,b) = \sqrt{(P_a \cdot x - P_b \cdot x)^2 + (P_a \cdot y - P_b \cdot y)^2}, \tag{2}$$

where $P_a \cdot x$ and $P_a \cdot y$ represent the coordinates x and y of point a, respectively, and the horizontal and vertical Euclidean distances of the eye can be expressed as follows:

$$\mathrm{Eye}_h = \mathrm{Dis}(P_{37}, P_{40}), \tag{3}$$

$$\mathrm{Eye}_v = \mathrm{Mean}(\mathrm{Dis}(P_{38}, P_{42}), \mathrm{Dis}(P_{39}, P_{41})), \tag{4}$$

where $\mathrm{Mean}(A, B)$ means the average of A and B, and then the aspect ratio of the eye can be expressed as follows:

$$\mathrm{EYE}_{\mathrm{left}} = \frac{\mathrm{Eye}_v}{\mathrm{Eye}_h}. \tag{5}$$

Since the value calculation process of the left and right eyes is the same, the calculation process of the right eye will not be repeated. The eye feature vector (EFV) is composed of $\mathrm{EYE}_{\mathrm{left}}$ and $\mathrm{EYE}_{\mathrm{right}}$.

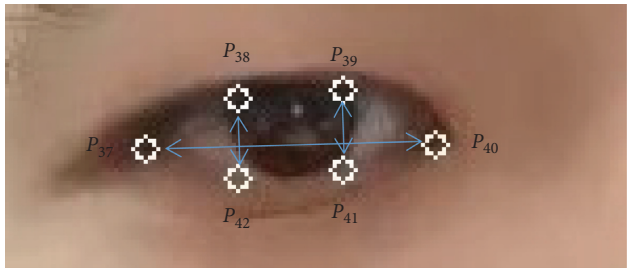


Figure 2: Eye open state diagram.

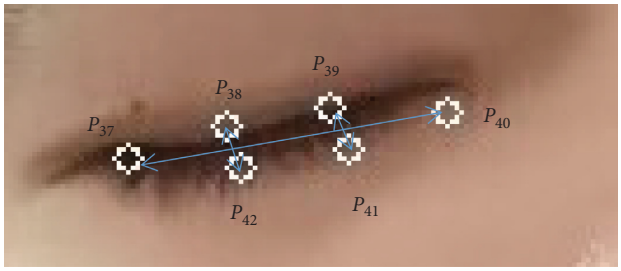


Figure 3: Eye-closed state diagram.

3.2.2. Yawn Detection. Yawn detection is similar to closed eye detection. The key points of the mouth are 61–68. These points make up the key points of our inner lips. Some scholars use the key points of the outer lips. However, due to the individual lip differences, the calculated value will not be accurate enough. The MOUTH value is calculated by using the distance of the mouth feature points, which can judge the state of the mouth. The mouth feature vector (MFV) only consists of MOUTH. Figures 4 and 5 show the open and closed states of the mouth, respectively.

The horizontal and vertical Euclidean distances of the mouth can be expressed as follows:

$$\text{Mouth}_v = \text{Mean}\left(\text{Dis}\left(P_{62}, P_{68}\right), \text{Dis}\left(P_{63}, P_{67}\right), \text{Dis}\left(P_{64}, P_{66}\right)\right), \tag{6}$$

$$\text{Mouth}_h = \text{Dis}\left(P_{61}, P_{65}\right), \tag{7}$$

where $\text{Mean}(A, B, C)$ means the average of A, B, and C.

Then, the aspect ratio of the mouth can be expressed as follows:

$$\text{MOUTH} = \frac{\text{Mouth}_v}{\text{Mouth}_h}. \tag{8}$$

3.3. Fatigue Recognition Network. Many existing fatigue identification methods only use a single fatigue feature, which will lead to many misjudgments. Assuming that only the mouth information is used to determine whether you are tired, it is likely to misjudge your speech as fatigue [28]. Therefore, the fatigue detection results obtained by analyzing one single frame are not accurate. Inspired by LRCN [24], a two-stage fatigue identification method is designed in this

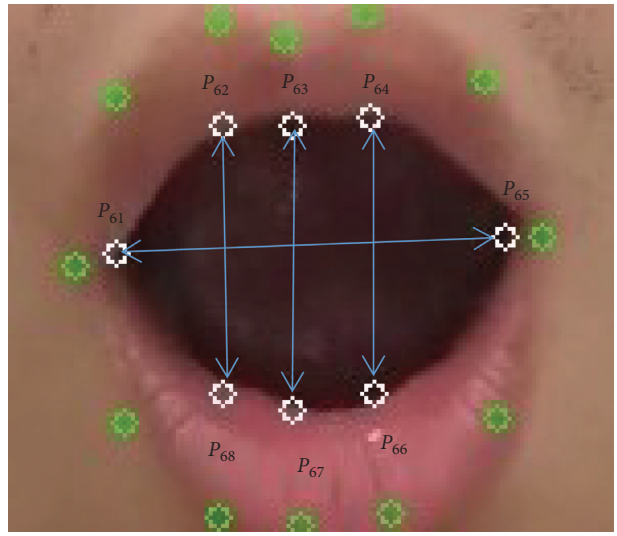


Figure 4: Mouth open-state diagram.

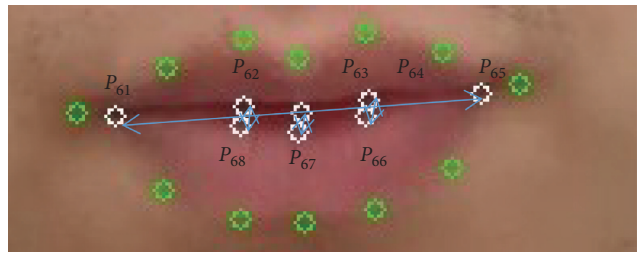


Figure 5: Mouth closed-state diagram.

paper. The first stage is splitting the input video into frames of pictures. The fatigue vector of a single frame is extracted through MTCNN and DLIB, and the information of multiple consecutive frames is combined to form a temporal feature vector. The second stage is as follows: these fatigue feature sequences are input into the LSTM-based network to identify the fatigue state.

3.3.1. Temporal Fatigue Characteristic Sequence. The feature extraction task needs to extract the eyes and mouth state values of each frame. Therefore, we set the single frame feature vector length to 3. The fatigue feature vector of a single frame image is as follows:

$$f_n = \left\{\text{Value}_{\text{leye}}, \text{Value}_{\text{reye}}, \text{Value}_{\text{mouth}}\right\}, \tag{9}$$

where $\text{Value}_{\text{leye}}$ and $\text{Value}_{\text{reye}}$ represent the state of the left eye and the right eye and $\text{Value}_{\text{mouth}}$ represents the state of the mouth.

The feature vector of each frame is 1×3. So, we splice the feature vectors of multiple frames, and a temporal feature sequence of $n \times 3$ will be formed. The vector length is 3, and the number of spliced frames is n. The splicing process is shown in Figure 6.

As shown in Figure 6, the length of the time window is a key parameter to construct the temporal fatigue characteristic sequence. If the length is too short, the obtained sequence may not be able to completely cover the fatigue state, and the excessively long time window will cause the sequence to contain too much redundant information. Another key parameter is the number of the skipped frames. Since the information of adjacent frames will be almost the same, it is not necessary to extract the information of each frame, which will cause a lot of waste of calculation and greatly reduce the efficiency. We split each video sample at a rate of two frames per second. Since the fatigue process usually does not exceed three seconds, we chose a time window length of 6 and skipped frames number of 2.

3.3.2. Fatigue Recognition Network Based on LSTM. LSTM is carefully designed to avoid the problem of long dependencies. Remembering long historical information is actually a default behavior. LSTM works very well on various problems and is now widely used in pattern recognition. Based on this idea, a fatigue identification network based on LSTM is applied in this paper. Its structure is shown in Figure 7.

As shown in Figure 7, the input of the LSTM network is a sequence of time features. The time feature sequence is composed of six single frame feature vectors. Therefore, the length of LSTM is also 6. LSTM will return a probability value, which represents the probability of driver fatigue in the current time window. When the probability value is more than 0.5 or equal to 0.5, we set this value to 1 and indicate that the driver is in a state of fatigue during the current period. When the probability value is less than 0.5, we set this value to 0, which indicates that the driver is awake during the current period. As long as a period is judged to be fatigued, we will treat the video as a fatigue sample.

4. Experiment

4.1. Dataset. In the experimental part, we selected the YawDD dataset and self-built dataset to verify the performance of the method.

4.1.1. YawDD Video Dataset. The dataset is collected by Abtahi et al. [29], which was captured in a static environment. The collectors gathered a large number of volunteers. The volunteer group was composed of drivers of different skin colors, sexes, and ages. They did different actions according to the instructions as normal driving, talking, and yawning. Each volunteer was shot multiple videos. When the driver wears pure black sunglasses, the human eye cannot recognize the eye condition of driver. Therefore, we selected 100 videos where volunteers were not wearing pure black sunglasses, including 50 men and 50 women for testing. A part of the dataset is shown in Figure 8.

4.1.2. Self-Built Dataset. In the YawDD dataset, some drivers in the video do not yawn naturally, but just open their mouths to make a yawning action. In order to capture the most natural fatigue state as much as possible, our fatigue video samples are all taken after the volunteers get off work. After working for a long time, most people are more prone to fatigue. We cannot guarantee that every sample captures the natural yawning action, but we filmed the behavior that fits the most natural fatigue. Our algorithm was tested on behaviors often associated with fatigue versus actual fatigue. Second, the proportion of yellow people in the YawDD dataset is low, mostly whites and Indians. Adding a self-built dataset can help reduce the difference in experimental results caused by races of different skin colors.

Self-built dataset was collected by our experimental team. We gathered 10 volunteers and each was shot two videos: one is a normal video, and the other is a fatigue video; they included closing eyes, talking, laughing, and yawning. These videos had slightly different face orientations, mouth shapes, and whether they wear glasses, and they were collected under different lighting conditions. Part of the dataset is shown in Figure 9.

4.2. Experimental Results and Analysis. The platform of this experiment is Windows 10, the processor is Inter(R) CoreTM i7-9700k, the main frequency of the CPU is 3.6 GHZ, and the memory is 8 GB. The programming language is Python. In the experiment, we split the video dataset into images and use MTCNN to detect and crop the face images. After cropping the face image, the DLIB library is used to mark the key points of the face to calculate the state value of the eye and mouth. By calculating the aspect ratio of the eyes and the mouth, we can perform closed eye detection and yawn detection. In order to verify the performance of the proposed algorithm, we compare our algorithm with the key point detection algorithms proposed in recent years. The experimental results are shown in Tables 1 and 2.

Tables 1 and 2, respectively, show the detection accuracy of our model and other methods in the YawDD dataset and the self-built dataset. It can be seen that our model is significantly better than other algorithms. The method proposed in this paper has a higher eye-mouth marking rate than other methods. Compared with the Viola–Jones algorithm, our method has significantly better results in the detection of faces, eyes, and mouths. Second, the detection results on the YawDD dataset are slightly lower than the self-built dataset. This may be due to the small number of videos in our self-built dataset. There is not much difference in actual detection results. Next, we compare the detection time between different methods.

Tables 3 and 4, respectively, show the detection time of our model and other methods in the YawDD dataset and self-built dataset. The Viola–Jones algorithm uses integral images to calculate its Haar-like features, which greatly reduces the amount of calculation. However, this algorithm was originally used to detect frontal face images, and it is not very robust to the detection of side face images. Therefore, its detection accuracy is low. The head pose estimation

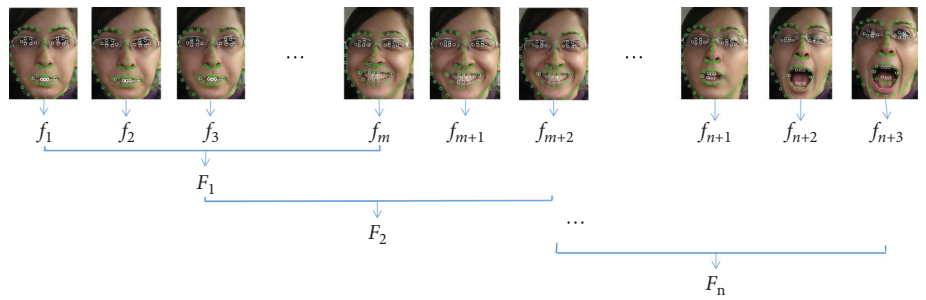


Figure 6: Temporal feature sequence composition diagram.

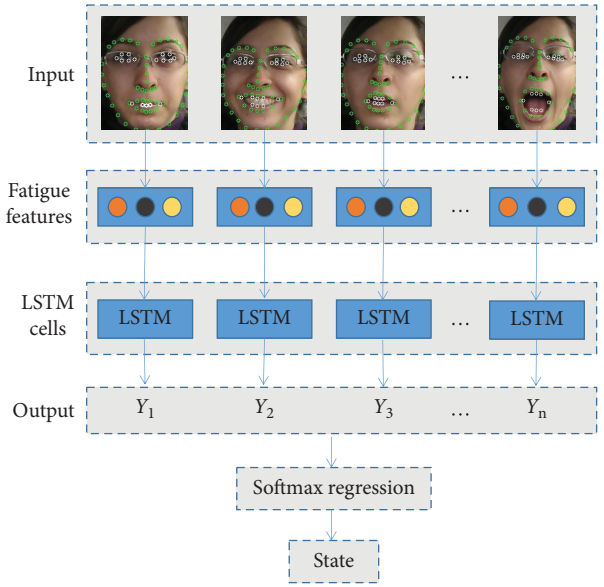
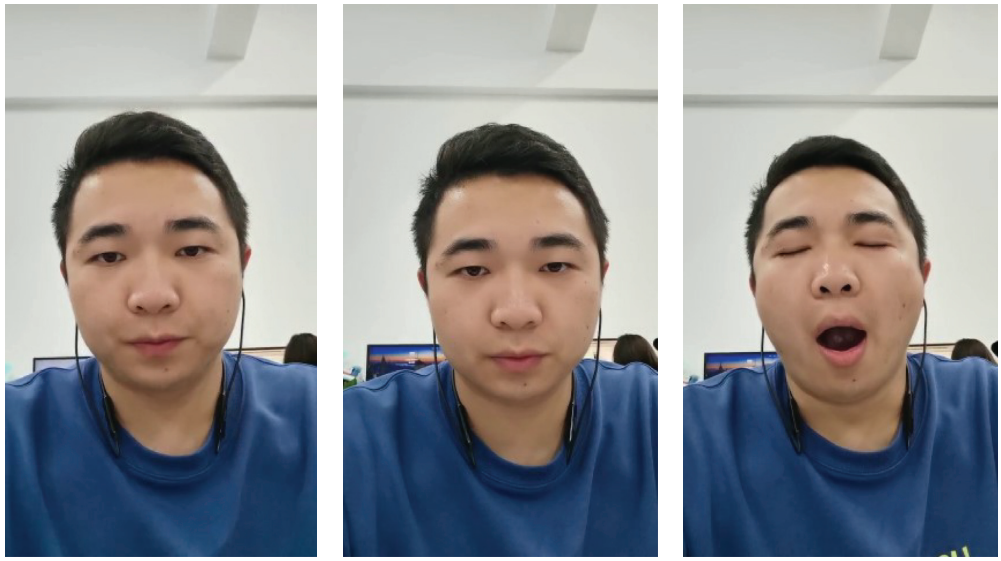


Figure 7: Fatigue recognition network diagram.

Figure 8: YawDD video dataset.



(a) (b) (c)

Figure 9: Self-built dataset.

Table 1: Detection accuracy of different areas in the YawDD dataset.

Algorithm	Face detection accuracy (%)	Eye detection accuracy (%)	Mouth detection accuracy (%)	Average detection accuracy (%)
Head pose estimation [27]	83	82	83	82.7
Viola–Jones [30]	73	79	81	77.7
Proposed	98	91	89	92.7

Table 2: Detection accuracy of different areas in the self-built dataset.

Algorithm	Face detection accuracy (%)	Eye detection accuracy (%)	Mouth detection accuracy (%)	Average detection accuracy (%)
Head pose estimation [27]	85	84	85	84.7
Viola–Jones [30]	77	83	84	81.3
Proposed	97	90	92	93

algorithm mainly uses the DLIB library to detect facial key points. The method proposed in this paper first uses MTCNN to extract the face and then uses the DLIB library to detect the key points of the face. In the process of detecting the key points of the eyes and mouth, the head pose estimation algorithm uses DLIB to detect the entire picture, which increases the amount of calculation and the detection rate is low. It can be seen from the data in the two tables that our method has a longer detection time than the Viola–Jones algorithm, but our average detection accuracy is 11%–15% higher than the Viola–Jones algorithm. Compared with the head pose estimation algorithm, the detection time is reduced by half, and the accuracy is increased by 8%–10%. Finally, we compared the accuracy of fatigue detection.

Tables 5 and 6 , respectively, show the fatigue detection accuracy of our model and other methods in the YawDD dataset and self-built dataset. This study selected videos of drivers driving normally, talking, laughing, and yawning from the dataset and analyzed the results of driver fatigue through the state of the eyes and mouth. We use MTCNN + DLIB, DLIB + LSTM, head pose estimation method, and Viola–Jones method to compare the results with the method in this paper. When using DLIB + LSTM to detect the fatigue state, DLIB directly detects the entire picture, which not only takes a long time to detect but also has lower accuracy. The facial key points' detection accuracy directly affects the judgment of the fatigue state. When we use MTCNN + DLIB to detect the fatigue state, we only rely on the fatigue feature value of a

Table 3: Detection time of different areas in the YawDD dataset.

Algorithm	Face detection time (s)	Eye detection time (s)	Mouth detection time (s)	Average detection time (s)
Head pose estimation [27]	0.1675	0.1273	0.1328	0.1425
Viola–Jones [30]	0.033	0.0237	0.0319	0.0295
Proposed	0.1064	0.0447	0.0415	0.0642

Table 4: Detection time of different areas in the self-built dataset.

Algorithm	Face detection time (s)	Eye detection time (s)	Mouth detection time (s)	Average detection time (s)
Head pose estimation [27]	0.1655	0.1250	0.1323	0.1409
Viola–Jones [30]	0.0319	0.0245	0.0295	0.0286
Proposed	0.1082	0.0455	0.0413	0.065

Table 5: Fatigue detection accuracy in the YawDD dataset.

Method	Fatigue detection accuracy (%)
MTCNN + DLIB	79
Head pose estimation [27]	77
Viola–Jones [30]	82
DLIB + LSTM	74
Proposed	88

Table 6: Fatigue detection accuracy in the self-built dataset.

Method	Fatigue detection accuracy (%)
MTCNN + DLIB	85
Head pose estimation [27]	80
Viola–Jones [30]	85
DLIB + LSTM	75
Proposed	90

single frame to determine the fatigue state, but fatigue is a continuous time behavior. So, the accuracy of this detection method is significantly lower than our method. In addition to these two methods, we also select two methods with superior performance to compare with our method. It can be seen from the result in Tables 5 and 6 that the accuracy rate of our method has reached 88%–90%.

5. Conclusion

We proposed a fatigue detection algorithm based on facial key points and long short-term memory. Since the face contains more features than the eyes and mouth, it is easier to be detected. So, we first obtained the face image and marked the key points of the eyes and mouth in the face image. This can reduce the scope of the eyes and mouth test and also avoid the interference of the background area in the image. Fatigue is a continuous behavior. It is easy to make misjudgments if the result only relies on the eye and mouth features of a single frame, so we split the fatigue feature values of a single frame into a temporal fatigue feature sequence and sent it to LSTM network. Although our method is superior to other methods in the extraction accuracy of facial key points and the final fatigue determination accuracy, the detection performance under insufficient light still needs to be improved. Our next step is to study fatigue driving detection in complex lighting environments and focus on the challenge of fatigue testing under poor light conditions, such as strong light and weak light. These application scenes are more practical and more difficult. When an automobile enters a tunnel or runs at night, how can we recognize the driver's fatigue driving behavior in time? This direction is also one of the current researches focuses in the field of fatigue driving detection.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 61872134 and

61672222, in part by Science and Technology Project of Transport Department of Hunan Province under grant no. 201935, in part by Science and Technology Program of Changsha City under grant nos. kh200519 and kq2004021, in part by National Key Research & Development Plan under grant no. 2017YFC1703306, and in part by School Level Project of Hunan University of Chinese Medicine under grant no. 2018GL01.

References

[1] S. Nordbakke, "Driver fatigue and falling asleep-experience, knowledge and action among private drivers and professional drivers," *Fatigue*, 2004.

[2] S. Nordbakke and F. Sagberg, "Sleepy at the wheel: Knowledge, symptoms and behaviour among car drivers," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 10, no. 1, pp. 1–10, 2007.

[3] P. Chen, "Research on driver fatigue detection strategy based on human eye state," in *Proceedings of the CAC*, Jinan, China, October 2017.

[4] G. Sikander and S. Anwar, "Driver fatigue detection systems: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2019.

[5] D. Ma, X. Luo, S. Jin, W. Guo, and D. Wang, "Estimating maximum queue length for traffic lane groups using travel times from video-imaging data," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 3, pp. 123–134, 2018.

[6] T. D'orazio, M. Leo, and A. Distante, "Eye detection in face images for a driver vigilance system," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 95–98, Parma, Italy, June 2004.

[7] X. Sun, C. Lan, and X. Mao, "Eye locating arithmetic in fatigue detection based on image processing," in *Proceedings of the CISP-BMEI*, pp. 1–5, Shanghai, China, October 2017.

[8] X. Ma, L. P. Chau, and K. H. Yap, "Depth video-based two-stream convolutional neural networks for driver fatigue detection," in *Proceedings of the ICOT*, pp. 155–158, Singapore, December 2017.

[9] F. Zhang, J. Su, L. Geng, and Z. Xiao, "Driver fatigue detection based on eye state recognition," in *Proceedings of the CMVIT*, pp. 105–110, Singapore, February 2017.

[10] R. Gupta, K. Aman, N. Shiva, and Y. Singh, "An improved fatigue detection system based on behavioral characteristics of driver," in *Proceedings of the ICITE*, Singapore, September 2017.

[11] S. Junaedi and H. Akbar, "Driver drowsiness detection based on face feature and PERCLOS," in *Proceedings of the Journal of Physics: Conference Series 1090*, pp. 1–6, Ancona, Italy, June 2018.

[12] B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection based on SVM algorithm," in *Proceedings of the CEIT*, pp. 1–4, Bergen, Norway, October 2018.

[13] A. Amodio, M. Ermidoro, D. Maggi, S. Formentin, and SM. Savaresi, "Automatic detection of driver impairment based on pupillary light reflex," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 3038–3048, 2018.

[14] T. Li, L. Wang, Y. Chen, Y. Ren, L. Wang, and J. Xia, "A face recognition algorithm based on LBP-EHMM," *Journal on Artificial Intelligence*, vol. 1, no. 2, pp. 61–68, 2019.

[15] W. Liu, J. Qian, Z. Yao, X. Jiao, and J. Pan, "Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection," *Future Internet*, vol. 11, no. 5, p. 115, 2019.

[16] B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection system based on multi-task ConNN," *IEEE Access*, vol. 8, pp. 12491–12498, 2020.

[17] Z. Liu, Y. Peng, and W. Hu, "Driver fatigue detection based on deeply-learned facial expression representation," *Journal of Visual Communication and Image Representation*, vol. 29, no. 2, pp. 87–91, 2020.

[18] Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, "Real-time system for driver fatigue detection based on a recurrent neuronal network," *Journal of Imaging*, vol. 6, no. 3, pp. 1–14, 2020.

[19] S. Jida, B. Aksasse, and M. Ouanan, "Face segmentation and detection using Voronoi diagram and 2D histogram," in *Proceedings of the ISCV*, pp. 1–5, Fez, Morocco, April 2017.

[20] J. Zou and R. Song, "Microarray camera image segmentation with faster-RCNN," in *Proceedings of the ICASI*, pp. 86–89, Taiwan, China, April 2018.

[21] X. Chen, X. Luo, X. Liu, and J. Fang, "Eyes localization algorithm based on prior MTCNN face detection," in *Proceedings of the ITAIC*, pp. 1763–1767, Chongqing, China, May 2019.

[22] N. L. Arifin, H. Widiastuti, and A. Wibowo, "Study on effect of source to film distance (SFD) on the radiographic images," in *Proceedings of the ICAE*, pp. 1–4, Hongkong, China, August 2018.

[23] P. Samangouei, R. Chellappa, M. Najibi, and R. Chellappa, "Face-magnet: magnifying feature maps to detect small faces," in *Proceedings of WACV*, pp. 122–130, Lake Tahoe, NV, USA, March 2018.

[24] J. Donahue, LA. Hendricks, M. Rohrbach, S. Venugopalan, and S. Guadarrama, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the CVPR*, pp. 2625–2634, Boston, MA, USA, June 2015.

[25] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[26] Y. Ji, S. Wang, Y. Zhao, J. Wei, and Y. Lu, "Fatigue state detection based on multi-index fusion and state recognition network," *IEEE Access*, vol. 7, pp. 64136–64147, 2019.

[27] N. Zhang, H. Zhang, and J. Huang, "Driver fatigue state detection based on facial key points," in *Proceedings of the ICSAI*, pp. 144–149, Shanghai, China, September 2019.

[28] C. Zhang, X. Lu, and Z. Huang, "A driver fatigue recognition algorithm based on spatio- temporal feature sequence," in *Proceedings of the CISP-BMEI*, pp. 1–6, Shanghai, China, October 2019.

[29] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proceedings of the 5th ACM Multimedia Systems Conference*, pp. 24–28, Singapore, March 2014.

[30] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi et al., "Yawning detection using embedded smart cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 570–582, 2016.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9987792, 9 pages
https://doi.org/10.1155/2021/9987792

Research Article

Craniofacial Reconstruction via Face Elevation Map Estimation Based on the Deep Convolution Neutral Network

Yining Hu,[1,2] **Zhe Wang**,[1] **Yueli Pan**,[1] **Lizhe Xie**,[3,4,5] **and Zheng Wang**[1,2]

[1]*School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China*
[2]*Jiangsu Provincial Key Laboratory of Computer Network Technology, Southeast University, Nanjing 211189, China*
[3]*Institute of Stomatology, Nanjing Medical University, Nanjing 210029, China*
[4]*Jiangsu Key Laboratory of Oral Diseases, Nanjing Medical University, Nanjing 210029, China*
[5]*Affiliated Hospital of Stomatology, Nanjing Medical University, Nanjing 210029, China*

Correspondence should be addressed to Yining Hu; hyn.list@seu.edu.cn and Lizhe Xie; xielizhe@njmu.edu.cn

Received 4 March 2021; Accepted 19 May 2021; Published 8 June 2021

Academic Editor: Beijing Chen

In this study, to achieve the possibility of predicting face by skull automatically, we propose a craniofacial reconstruction method based on the end-to-end deep convolutional neural network. Three-dimensional volume data are obtained from 1447 head CT scans of Chinese people of different ages. The facial and skull surface data are projected onto two-dimensional space to generate a two-dimensional elevation map, and then, use the deep convolution neural network to realize the prediction of skull to face shape in two-dimensional space. The encoder and decoder are composed of first feature extraction through the encoder and then as the input of the decoder to generate the craniofacial restoration image. In order to accurately describe the features of different scales, we adopt an U-shaped codec structure with cross-layer connections. Therefore, the output features are decomposed with the features of the corresponding scales in the encoding stage to achieve the integration of different scales while restoring the feature scales in the compression and decoding stage. Meanwhile, the U-net structures help to avoid the problem of loss of detail features in the downsampling process. We use supervised learning to obtain the prediction model from skull to facial elevation map. Back-projection operation is performed afterwards to generate facial surface data in 3D space. Experiments show that the proposed method in this study can effectively achieve craniofacial reconstruction, and for most part of the face, restoration error is controlled within 2 mm.

1. Introduction

Craniofacial reconstruction is a technique producing a reconstructed face from a human skull. Based on the relationship between the skull and face in forensic medicine, anthropology, and anatomy, this technique has been widely used in criminal investigation and archaeology. The traditional craniofacial reconstruction is mainly implemented manually by experts, based on the anatomical law of the human head and face on the plaster model of the victim's skull and according to the relationship between the soft tissue of the human head and face and the morphological characteristics of the face and skull. The facial appearance of the victim is gradually reproduced with adding rubber clay and other materials. This method usually requires a complicated process, high cost, and time-consuming. In addition, the result largely depends on the practitioner's experience, so its application in criminal investigations based on timeliness and truthfulness is greatly restricted.

With the development of computer visualization and virtual three-dimensional technology, computer-aided craniofacial reconstruction technology has greatly reduced the repair time and work difficulty and reduced the subjective deviation factors, which has attracted widespread attention. The current reconstruction methods are based on either template [1] or feature points [2, 3]. For the template-based methods, a face template set in advance is required. In the reconstruction process, the template is deformed according

to the shape of the skull until the feature points on the face template match with the feature points estimated from the skull. Reconstruction can be based on fixed templates [4–7] or dynamic templates [8–14]. The feature points-based methods first estimate the soft tissue thickness of the facial key points and then restores the facial surface. Although feature points based methods have been practically applied in the field of forensics, there are still limitations, which are mainly reflected in two aspects. First, in the process of recovering complete face surface from sparse feature point information, the loss of facial details will be inevitable. Second, human interaction is often required to ensure the accuracy of feature points positioning, which result in an extra anthropic factor.

Craniofacial reconstruction is essentially a problem of sample generation based on reference data. With the rapid development of deep learning technology, data generation based on the convolutional neural network shows significant advantages, among which the representative technologies are the variational autoencoder (VAE) [15, 16] and generative adversarial network (GAN) [17]. Both VAE and GAN attempt to learn the mapping of hidden space variables to real data distribution through training samples. The difference is that VAE calculates the mean and variance of samples through the neural network, constrains them to obey standard normal distribution, and then samples out hidden variables for reconstruction [18]; while, the GAN adopts the idea of game theory and directly measures the distance between real distribution and generated distribution through the discriminator, forcing the generator to generate a more realistic distribution. In recent years, the GAN has received extensive attention from the industry, and many variants have been derived, such as the WGAN [19], CGAN [20], Pix2Pix [21], and BEGAN [22].

The convolution neural networks have also been introduced into the field of craniofacial reconstruction. Li et al. [23] proposed to use a convolutional neural network based on a codec structure, which can well predict the distribution of skeleton soft tissue. The method is with high computation cost, and high performance hardware requirements are also needed, but the generated results are not satisfying. Yuan et al. [24] used the GAN to reconstruct 3D face images. Limited by the data amount and computing power, the author use sparse representation of 3D data to reduce the computation cost and improve the recovery ability; Liu and Xin [25] proposed a prediction method based on the autoencoder and GAN. Candidate faces are generated through the autoencoder. The human face and skull are superimposed to determine the best face. The GAN is used afterwards to optimize the results. Such scheme is essentially a deep learning version of the template-based method. Although the reconstruction accuracy is relatively high, the common problem of the template method is inevitable, that is, the generation process is cumbersome, and the network structure is complex.

Based on the above research, we propose an end-to-end facial morphology prediction method based on the deep convolutional neural network to automatically estimate face information from skull data. For the proposed method, named cylindrical facial projection residual net (CFPRN), it needs neither preset face template nor feature point detection. In order to ignore unnecessary calculations, we do not reconstruct the face data directly in 3D space but try to estimate the face elevation map in 2D cylindrical projection space, and back-projection operation is performed afterwards to get the 3D face surface. We use U-shape network structure so as to adapt with features of different scales. The CFPRN is easy to implement, and experiments have verified the robustness and accuracy of the proposed method.

2. Data Preprocessing

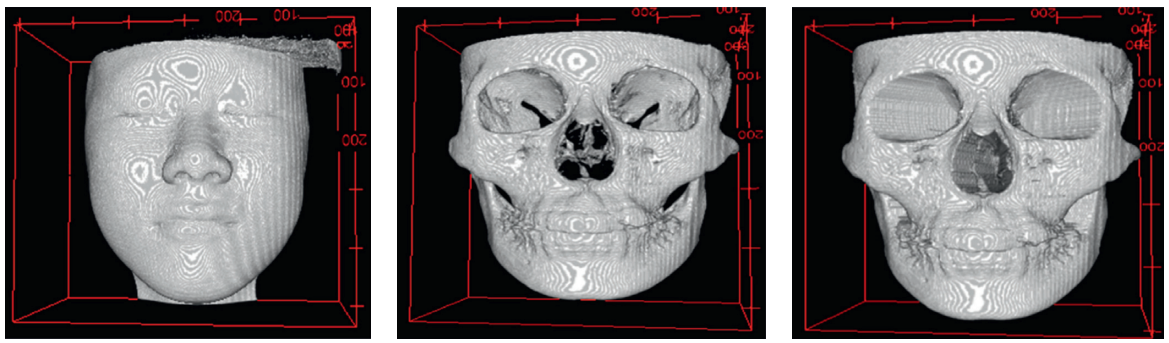
2.1. Data Segmentation. The objective of craniofacial reconstruction is to recover the 3D face surface data from 3D skull data. Both data are obtained from 3D head CT scan. The face surface can be simply retrieved via threshold segmentation, as shown in Figure 1(a); however, due to the complexity distribution of soft tissue and cartilage, threshold segmentation is not suitable for the skull. In order to obtain a clean skull structure, we choose to use adaptive threshold segmentation with a sliding window. The size of the sliding window is set to be 7*7*7. The comparison of global and adaptive thresholding is shown in Figures 1(b) and 1(c).

2.2. Projection and Back-Projection. For craniofacial reconstruction task based on convolutional neural networks, the 3D volume data obtained via head CT scans are usually with excessive data volume [26]. The existing hardware conditions are difficult to meet the problem of constructing a feature network directly for 3D data under the original resolution. In fact, during the reconstruction, only the surface of the skull and the face needs to be considered. Therefore, we use projection operations to map the 3D data to the 2D space for calculation. Considering that the human head is close to a circle in the cross-section and in order to avoid the inconsistency of the resolution in the vertical axis, we use a cylindrical projection surface. The plane projection and sphere projection are not considered because the former leads to inconsistent resolution in vertical direction, and the latter results in inconsistent resolution in different horizontal slices. As shown in Figure 2(a), the cross-section of the CT scan is the XOY plane, and the Z-axis is perpendicular to the cross-section. The coronal plane and the sagittal plane are the XOZ plane and the YOZ plane, respectively. Figure 2(b) shows the projected plane coordinate system.

The coordinate transform between 3D space and the cylindrical projection plane is defined as follows. For projection,

$$\begin{aligned} u &= \arctan\left(\frac{x'}{y'}\right) * \frac{n}{\pi} = \alpha * \frac{n}{\pi}, \\ v &= z', \\ r &= \sqrt{\left(x'^2 + y'^2\right)}. \end{aligned} \tag{1}$$

For back-projection,



(a) (b) (c)

Figure 1: Result of adaptive threshold segmentation. (a) 3D face image. (b) 3D skull via thresholding. (c) 3D skull via adaptive thresholding.

$$x' = r\ \sin\left(\frac{u\pi}{2n}\right),$$
$$y' = r\ \cos\left(\frac{u\pi}{2n}\right), \quad (2)$$
$$z' = v,$$

where x', y', z' are the coordinates from the 3D space, and u, v are the coordinates from the projection plane. r is the pixel value of the 2D projected altitude map which represents the distance from the point to the projection axis in 3D space. Thus, the depth information in 3D facial and skull surface is preserved in the projection and back-projection steps. $2n$ is the total sample number in U axis.

3. Network Architecture

The network structure refers to the encoder-decoder structure of U-net [27] and draws on the relevant ideas of the CGAN [20], Pix2Pix [21], and other networks to realize an end-to-end network.

In the encoder-decoder structure, the first half of the network acts as an encoder, which successively is downsampling through pooling, convolution with strides, to extract deep features from the input image. The second half of the network acts as a decoder, which successively is upsampling through deconvolution, interpolation, to map the feature output by the encoder back to the size of the previous level. In the meantime, cross-layer connection is considered, so that the high-level feature map after being upsampled by the decoder and the low-level feature map of the same scale in the encoder are connected in the channel dimension, and feature information of different scales are merged to make the prediction result more accurate and stable. Figure 3 shows the specific structure of the proposed network.

The network is generally divided into two parts: encoder module and decoder module.

The encoder module is mainly composed of a convolutional layer and five convBlocks; each convBlock, as shown in the bottom right of Figure 3, contains a leaky Relu activation layer, a 3*3 convolutional layer, and a group normalization layer. The encoder module performs 6 downsampling in total, and the pooling operation is replaced by a convolution operation with a step size of 2 so as to retain more feature information.

The decoder module is composed of five deconvBlocks and a convolutional layer. Each deconvBlock, as shown in the bottom right of Figure 3, contains a leaky Relu activation layer, an upsampling layer, a 3*3 convolutional layer, and a group normalization layer. The decoder module performs upsampling 6 times in total; bilinear interpolation is considered for upsampling, expanding the height and width of the feature map by 2 each time. The feature map after each upsampling is connected in the channel dimension with the feature map of the corresponding scale in the encoder. Through such a cross-layer connection, the deep and shallow features can be effectively merged.

In the meantime, we use some tricks to improve the performance of the entire network. (i) Replace deconvolution with a structure of upsampling using bilinear interpolation and convolution, which can effectively avoid the checkerboard effect [28]. (ii) Replace Relu with leaky Relu, which can effectively reduce the dead neurons. Replace pooling operation with convolution operation with a step size of 2 to retain more features. (iii) Use group normalization [29] instead of batch normalization which can effectively avoid the impact of batch size on the training results.

We use normalized skull elevation map as network input. The data range is limited to (−1, 1). The normalization can speed up the convergence of the network and increase the generalization ability of the model. For the supervised data, we have 2 options: one is to use the face elevation map directly and the other is to use the residual between the face and skull surface (mentioned as "face" and "res," respectively, in the experiment section).

The loss is defined as the distance between predicted and real face elevation map. We use mean square error

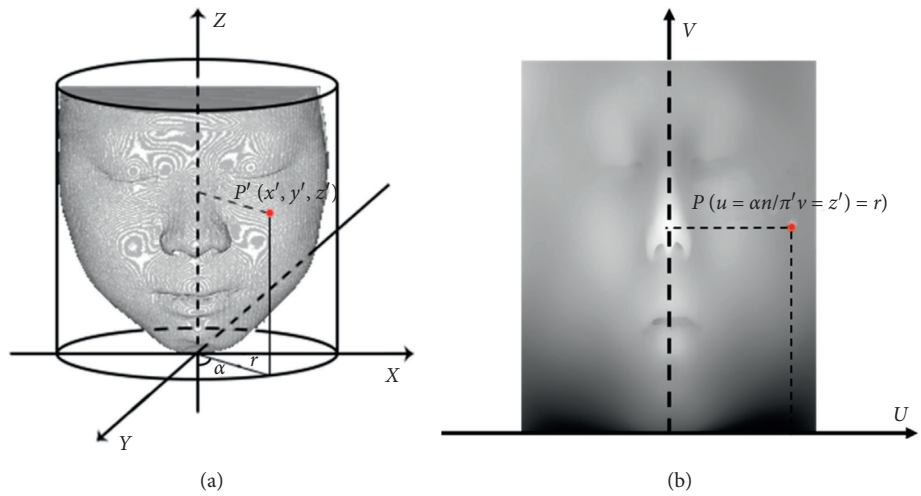


Figure 2: Cylindrical projection. (a) 3D face image before projection. (b) 2D face image after projection.

(MSE) to define the loss function, which represents the average value of the square of the difference between the predicted and the real elevation map. The expression is as follows:

$$\mathrm{MSE}(x, y) = \frac{1}{m}\sum_{i=1}^{m}(x_i - y_i)^2, \tag{3}$$

where m denotes the number of pixels, and the terms x, y denote the predicted and the label value, respectively.

4. Experiment

4.1. Data Description. The dataset used for experiment is acquired from the head cone-beam CT scan from NewTom 5G. The dataset contains CT data of 1447 participants from Affiliated Hospital of Stomatology, Nanjing Medical University. Each sample has 540 CT slices, the resolution for each slice is 610×610, and the pixel size is 0.3 mm*0.3 mm. 1310 samples were randomly selected as training set, and the validation set is composed of the rest 137 samples.

4.2. Evaluation Indices. Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [30] is chosen as evaluation indices for the experiments. The peak signal-to-noise ratio (PSNR) measures the ratio between the energy of the peak signal and the average energy of the noise, which is commonly used for signal recovery quality. The PSNR is defined as

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{MAX}^2}{\mathrm{MSE}}, \tag{4}$$

where MAX denotes the maximum pixel value in the data, and MSE is the mean square error. Besides PSNR, SSIM is considered for the similarity measure between the ground truth and the predictions. The SSIM measures image similarity from three aspects of brightness, contrast, and structure, with value range (0, 1), and larger value stands for smaller image distortion.

5. Results and Discussion

5.1. Result. We intuitively visualized the experimental results. Figure 4 shows the input elevation map of the skull. Figure 5 shows that the prediction results of the face elevation map correspond to skulls in Figure 4 and the corresponding ground truth. We can see the predicted result is very close to the ground truth. Pseudocolour maps shown in Figure 6 visualize the difference between the output and the ground truth (in percentage), from which we may see that the error mainly occurs in the eyes, nose, and mouth area. Obviously, because of the cavity in the skull, it is impossible to accurately predict the eyes and nose.

We use the predicted elevation map to generate 3D facial data through back-projection. The generated 3D face is compared with ground truth. The difference map is shown in Figure 7, from which we may see that for most part, the error is limited to 1 mm.

5.2. Comparison. We have repeated experiments on different network architectures and different image sizes. The specific results are given in the table, and the bold line is our proposed one. Table 1 indicates that the proposed CFPRN is with high accuracy and shows best performance among all the candidates. The abbreviation "Res" means the network output is the residual of the face and skull, and the abbreviation "Face" means the network output is the face surface directly. Table 2 indicates that the CFPRN works well under different resolution settings.

5.3. Error Analysis

(1) In order to simplify the network and improve efficiency, we reduce the dimension of the input, which causes a partial loss of data accuracy. After the prediction is

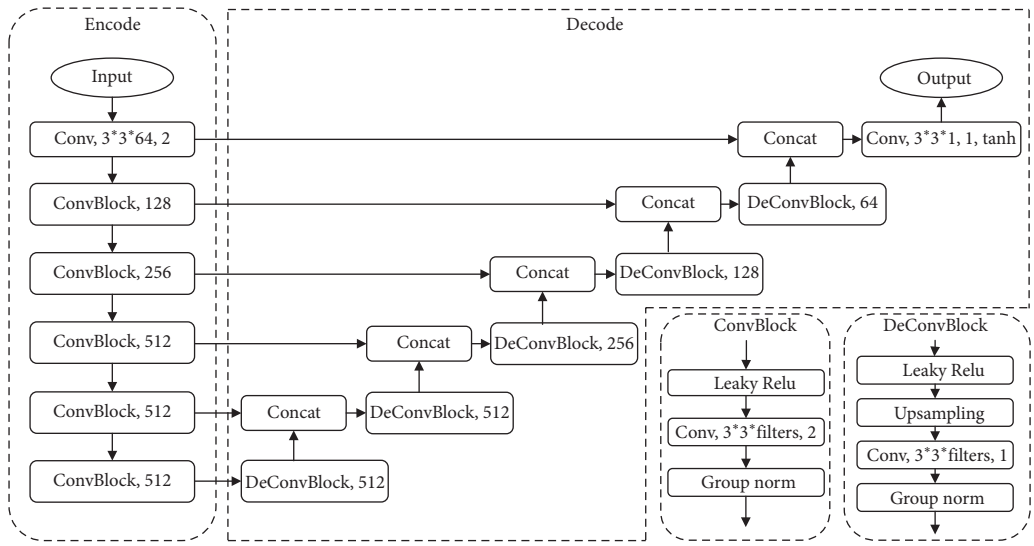
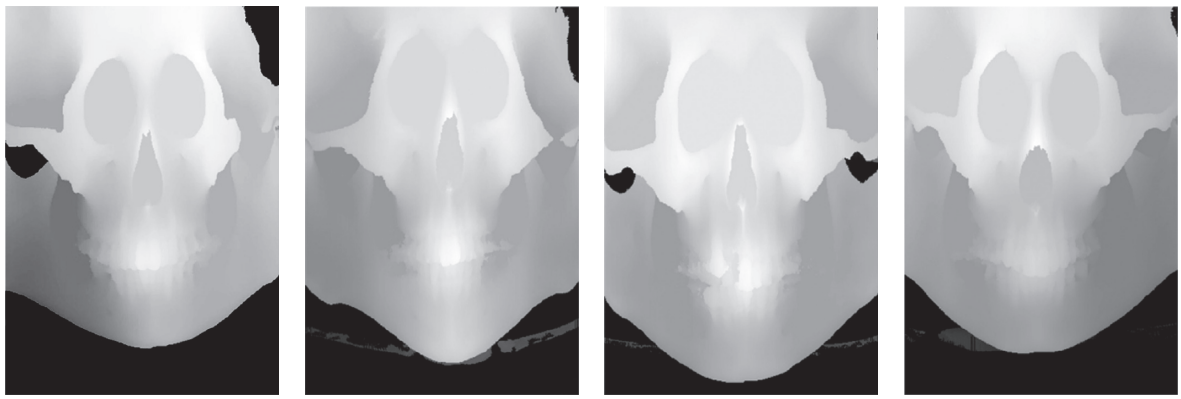
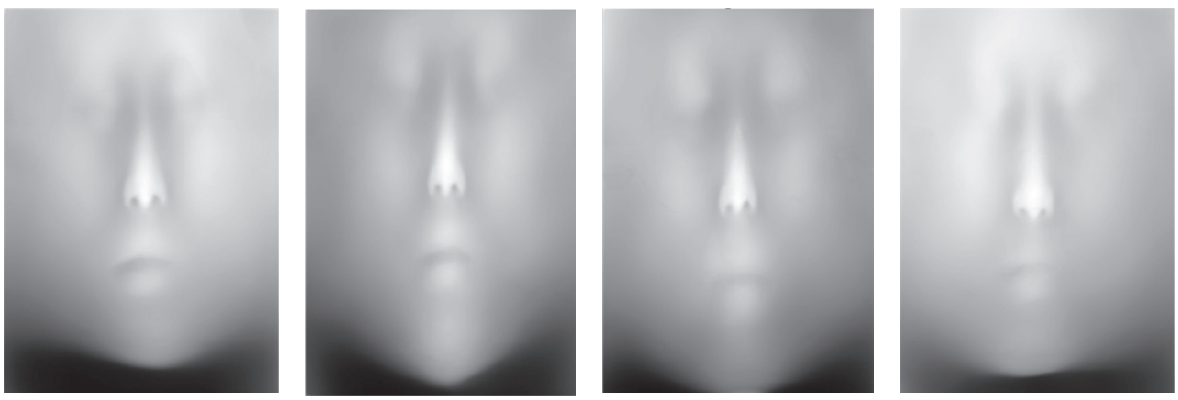


Figure 3: Network structure.



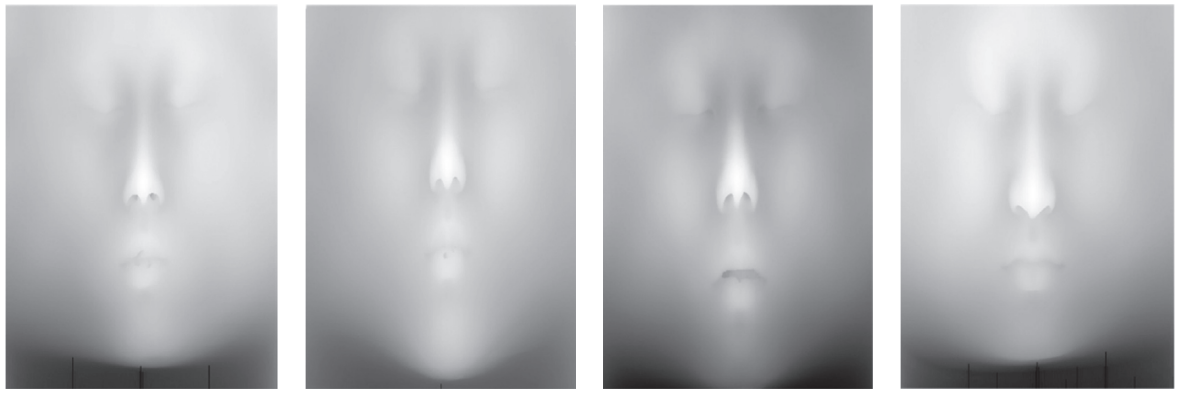
(a) (b) (c) (d)

Figure 4: Input skull images. (a) Skull 1. (b) Skull 2. (c) Skull 3. (d) Skull 4.



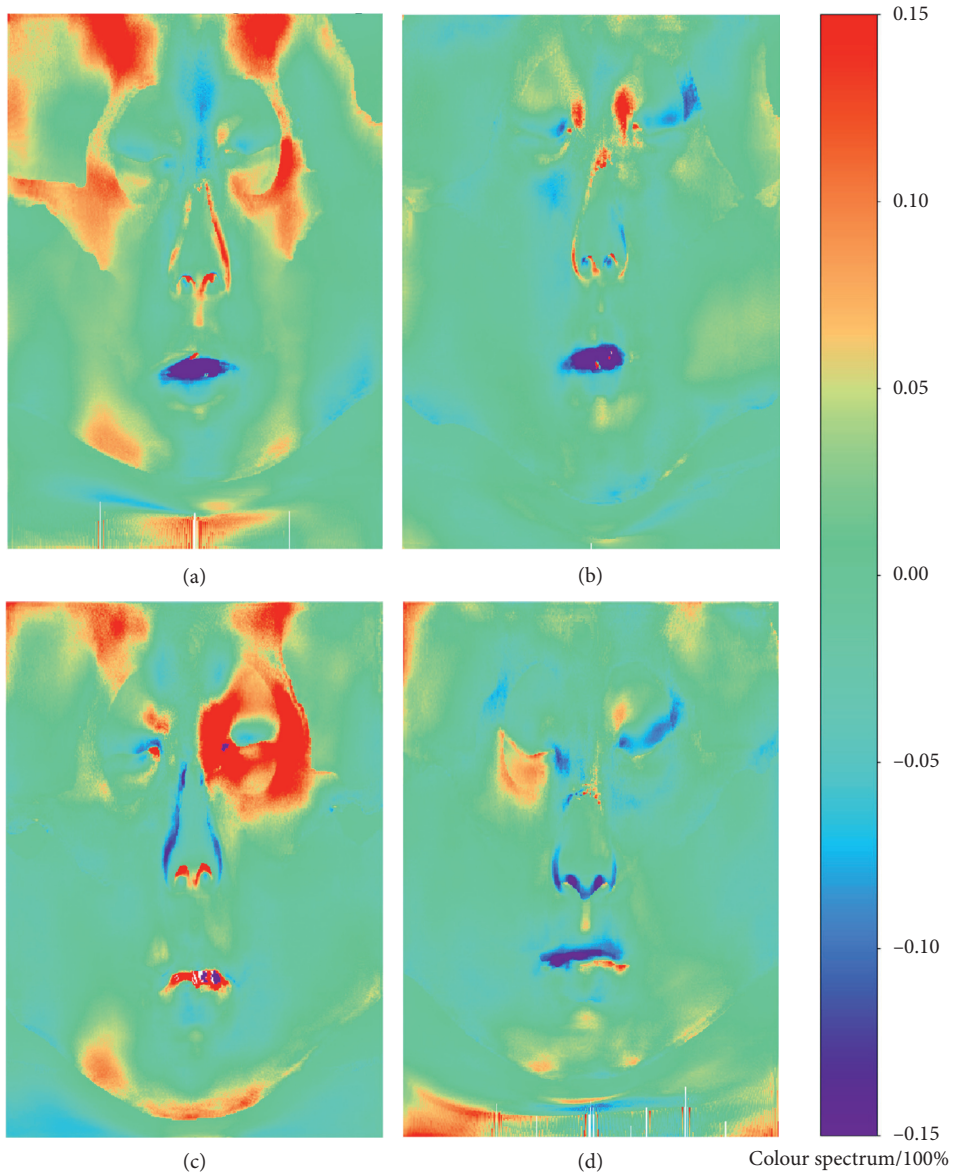
(a) (b) (c) (d)

Figure 5: Continued.



(e) (f) (g) (h)

Figure 5: Comparison between label face and predicted face. (a) Prediction face 1. (b) Prediction face 2. (c) Prediction face 3. (d) Prediction face 4. (e) Label face 1. (f) Label face 2. (g) Label face 3. (h) Label face 4.



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

Figure 6: 2D difference maps. (a) Difference map of face 1. (b) Difference map of face 2. (c) Difference map of face 3. (d) Difference map of face 4 colour spectrum/100%.

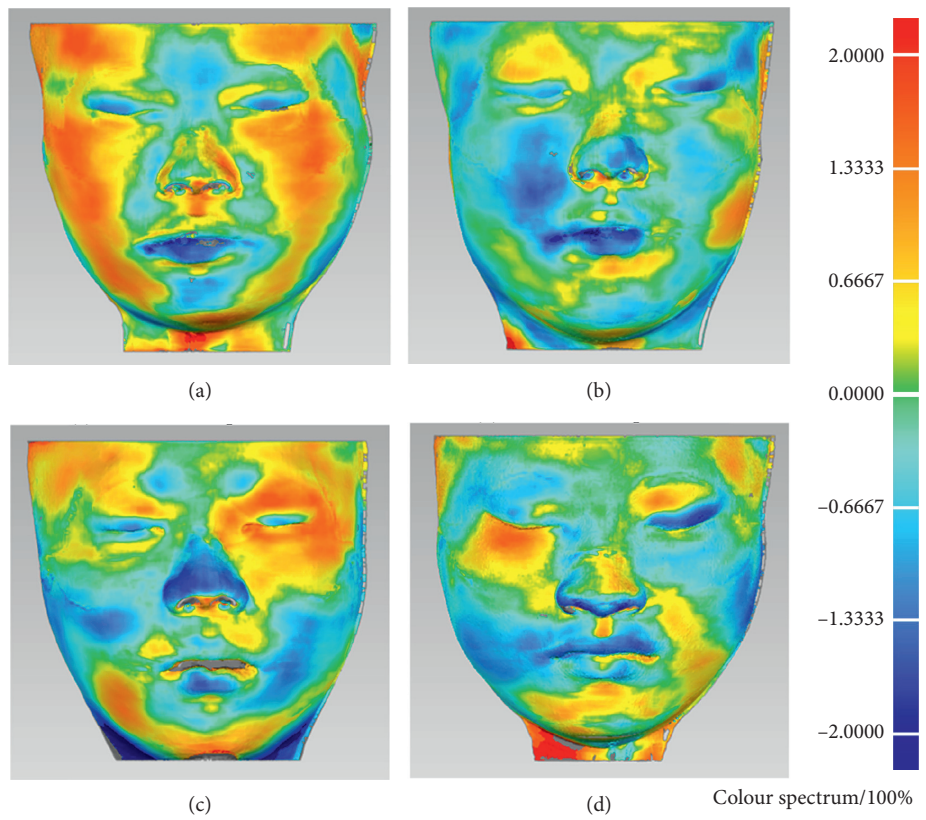


Figure 7: 3D difference maps. (a) Difference map of face 1. (b) Difference map of face 2. (c) Difference map of face 3. (d) Difference map of face 4 colour spectrum/mm.

Table 1: Results of different network architectures.

Network	RMSE	PSNR	SSIM
Inception ResNet U, Res	11.245456	31.477464	0.979727
Res, Pix2Pix, group norm	10.770702	31.938354	0.968041
Face, Pix2Pix, group norm	**10.16162**	**32.424038**	**0.987716**
Face, Pix2Pix, batch norm	10.724989	31.925444	0.986937

Table 2: Results of different network generation sizes.

Size	RMSE	PSNR	SSIM
Pix2Pix, 128*90	10.258215	32.379723	0.990736
Pix2Pix, 256*180	10.348463	32.392857	0.990038
Pix2Pix, 512*360	**10.16162**	**32.424038**	**0.987716**

completed, back-projection is performed, which may also cause extra error transfer.

(2) From the error map, we may see that basically, all samples have large errors in the part of the nose and eyes. This is because the skull has holes in the eyes and nose, which cannot be accurately predicted, and this might be overcome by introducing much more samples.

6. Conclusion and Prospects

In this study, we propose an end-to-end deep learning method for craniofacial reconstruction. The main contribution of the proposed method can be summarized as follows:

(1) We use projection and back projection for the transfer between 3D skull and face data into 2D elevation map. Instead of performing craniofacial

reconstruction in 3D space, the recovery runs in 2D space. The face elevation map is estimated according to the skull elevation map. Such design largely reduces the data size and computation cost, so that the proposed method is available on consumer graphics cards.

(2) We design an U-shaped end-to-end network to fit for the features in different scales. The accuracy and robustness of the prediction are guaranteed according to the experiment results.

According to our experiment results, we can also make further prospects:

(1) We should expand the amount of samples. Divide samples according to gender and age to balance the distribution of sample data.

(2) The eyes and nose of the skull should be hollowed out or filled. Because the specific shape of the face in these parts cannot be inferred from the skull, it is helpful to reduce the impact on the experimental results by hollowing out or filling these parts.

(3) We will try other network architectures, such as the conditional GAN. By introducing more conditions, we may provide subdivided predictions with higher accuracy.

Data Availability

All the experiment data are obtained from Affiliated Hospital of Stomatology, Nanjing Medical University. The access to the original data is restricted due to the patient privacy. However, the data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] P. Vanezis, M. Vanezis, G. McCombe, and T. Niblett, "Facial reconstruction using 3-D computer graphics," *Forensic Science International*, vol. 108, no. 2, pp. 81–95, 2000.

[2] Q. H. Dinh, T. C. Ma, and T. D. Bui, *Facial Soft Tissue Thicknesses Prediction Using Anthropometric Distances*, Springer, Berlin, Heidelberg, 2011.

[3] P. Guyomarc'H, B. Dutailly, C. Couture et al., "Anatomical placement of the human eyeball in the orbit--validation using CT scans of living adults and prediction for facial approximation," *Journal of Forensic Sciences*, vol. 57, no. 5, pp. 1271–1275, 2012.

[4] A. J. Tyrrell, M. P. Evison, A. T. Chamberlain et al., "Forensic three-dimensional facial reconstruction: historical review and contemporary developments," *Journal of Forensic Sciences*, vol. 42, no. 4, p. 653, 1997.

[5] M. W. Jones, *Facial Reconstruction Using Volumetric Data*, Aka GmbH, Augsburg, Germany, 2001.

[6] G. Quatrehomme, S. Cotin, G. Subsol et al., "A fully three-dimensional method for facial reconstruction based on deformable models," *Journal of Forensic Sciences*, vol. 42, no. 4, pp. 649–652, 1997.

[7] L. A. Nelson and S. D. Michael, "The application of volume deformation to three-dimensional facial reconstruction: a comparison with previous techniques," *Forensic Science International*, vol. 94, no. 3, pp. 167–181, 1998.

[8] M. Berar, M. Desvignes, G. Bailly et al., "Statistical skull models from 3D X-ray images," 2006, http://arxiv.org/abs/0610182.

[9] M. Desvignes, G. Bailly, Y. Payan et al., "3D semi-landmarks based statistical face reconstruction," *Journal of Computing & Information Technology*, vol. 14, 2006.

[10] P. Claes, D. Vandermeulen, S. De Greef, G. Willems, and P. Suetens, "Craniofacial reconstruction using a combined statistical model of face shape and soft tissue depths: methodology and validation," *Forensic Science International*, vol. 159, pp. S147–S158, 2006.

[11] P. Claes, D. Vandermeulen, S. D. Greef et al., "Statistically deformable face models for cranio-facial reconstruction," in *Proceedings of the Ispa International Symposium on Image & Signal Processing & Analysis*, IEEE, Zagreb, Croatia, September 2006.

[12] P. Claes, D. Vandermeulen, S. D. Greef et al., "Bayesian estimation of optimal craniofacial reconstructions," *Forensic Science International*, vol. 201, no. 1–3, pp. 146–152, 2010.

[13] P. Paysan, M. Lüthi, T. Albrecht et al., "Face reconstruction from skull shapes and physical attributes," in *Proceedings of the Symposium of the German Association for Pattern Recognition (DAGM 2009)*, pp. 232–241, Springer, Jena, Germany, September 2009.

[14] Y. Hu, F. Duan, B. Yin et al., "A hierarchical dense deformable model for 3D face reconstruction from skull," *Multimedia Tools and Applications*, vol. 64, no. 2, pp. 345–364, 2013.

[15] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," 2014, http://arxiv.org/abs/1401.4082.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014, http://arxiv.org/abs/1312.6114.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[18] C. Doersch, "Tutorial on variational autoencoders," 2016, http://arxiv.org/abs/1606.05908.

[19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, http://arxiv.org/abs/1701.07875.

[20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computer Science*, vol. 6, pp. 2672–2680, 2014.

[21] P. Isola, J. Y. Zhu, T. Zhou et al., "Image-to-Image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, IEEE, Seattle, WA, USA, June 2016.

[22] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," 2017, http://arxiv.org/abs/1703.10717.

[23] X. Li, B. Sheng, L. Ping et al., "Voxelized facial reconstruction using deep neural network," in *Proceedings of the Computer Graphics International*, New York, NY, USA, June 2018.

[24] Y. Yuan, Y. Zhang, S. Wang et al., "Sparse representation-based face object generative via deep adversarial network," in *Proceedings of the 2018 7th International Conference on Digital Home (ICDH)*, Guilin, China, December 2018.

[25] C. Liu and L. Xin, "Superimposition-guided facial reconstruction from skull," 2018, http://arxiv.org/abs/1810.00107.

[26] F. Tilotta, F. Richard, J. Glaunès et al., "Construction and analysis of a head CT-scan database for craniofacial reconstruction," *Forensic Science International*, vol. 191, no. 1–3, 2009.
[27] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional Networks for Biomedical Image Segmentation*, Springer, Berlin, Germany, 2015.
[28] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, 2016.
[29] Y. Wu and K. He, "Group normalization," *International Journal of Computer Vision*, vol. 14, 2018.
[30] Z. Wang, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 41, 2004.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9942754, 8 pages
https://doi.org/10.1155/2021/9942754

Research Article

FeatureTransfer: Unsupervised Domain Adaptation for Cross-Domain Deepfake Detection

Baoying Chen [1,2,3,4] **and Shunquan Tan** [1,2,3,4]

[1]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*
[2]*Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen, China*
[3]*Shenzhen Key Laboratory of Media Security, Shenzhen, China*
[4]*Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China*

Correspondence should be addressed to Shunquan Tan; tansq@szu.edu.cn

Received 16 March 2021; Revised 1 May 2021; Accepted 19 May 2021; Published 7 June 2021

Academic Editor: Mamoun Alazab

Recently, various Deepfake detection methods have been proposed, and most of them are based on convolutional neural networks (CNNs). These detection methods suffer from overfitting on the source dataset and do not perform well on cross-domain datasets which have different distributions from the source dataset. To address these limitations, a new method named FeatureTransfer is proposed in this paper, which is a two-stage Deepfake detection method combining with transfer learning. Firstly, The CNN model pretrained on a third-party large-scale Deepfake dataset can be used to extract the more transferable feature vectors of Deepfake videos in the source and target domains. Secondly, these feature vectors are fed into the domain-adversarial neural network based on backpropagation (BP-DANN) for unsupervised domain adaptive training, where the videos in the source domain have real or fake labels, while the videos in the target domain are unlabelled. The experimental results indicate that the proposed method FeatureTransfer can effectively solve the overfitting problem in Deepfake detection and greatly improve the performance of cross-dataset evaluation.

1. Introduction

Recently, the Deepfake video generation technology has attracted much attention, especially the popular Deepfake application called "ZAO". The application requires the user to provide a clear personal face image and complete facial feature verification, but the image collection protocol is not user-friendly. The majority of users express anxiety about the security of face information. In addition, the Deepfake technology could also be used to create fake news, posing threats to user privacy and social security [1–6]. Thus, it is critical to detect the Deepfake images or videos for face forensics. As we know, Deepfake detection, a branch of face forensics, is a binary classification task. The goal of face forensics is to detect whether a face in image or video has been created or manipulated.

The Deepfake video detection method mainly uses deep learning technology, which is usually composed of two parts: face detection and classification. As for face detection [7–9], MTCNN (multitask convolutional neural network) [7] and dlib [8] are mostly used as face detectors. As for the classification part, some researchers detect the Deepfake videos with the visible artifacts in the videos. For example, Matern et al. [10] found the inconsistent color of the left and right eyes and the geometric deformations of teeth in Deepfake videos. Li et al. [11] found that the people in Deepfake videos blink less frequently. Yang et al. [12] detected videos Deepfake through the cue of inconsistent head poses. Li et al. [13] exposed Deepfake videos by detecting face warping artifacts. These methods are effective for detecting some early Deepfake videos. However, with the development of Deepfake video generation technology, the visible artifacts used by these methods can be significantly reduced, degrading the performance of some artifacts-based methods. Therefore, some other cues in Deepfake videos need to be found for detection. Zhang et al. [14] found that the upsample or transposed

convolution used by the Deepfake technology inevitably results in a checkerboard effect on the generated face. Based on this, they proposed that CNN can be used to learn the checkerboard effect characteristics to detect Deepfake videos by directly inputting the face images extracted from video frames, such as MesoNet [15] and XceptionNet [16]. Unlike the spatial cues mentioned above, the temporal flickering, i.e., inconsistent temporal changes in videos, can be taken as the temporal cues in Deepfake videos. To make full use of both spatial and temporal cues in Deepfake videos, Guera et al. [17] and Chen et al. [18] combined CNN and recurrent neural networks (RNNs) to detect Deepfake videos. Unfortunately, Li et al. [19] found that most of the Deepfake detection methods trained and tested on specific datasets can achieve satisfactory performance, but their performances are significantly reduced when the methods are tested on cross-domain datasets, indicating that these methods are overfitting on a specific dataset. To improve the generalization ability of the methods on cross-domain datasets, multitask learning approaches [20–22] were introduced for Deepfake detection. Specifically, Nguyen et al. [20] developed a multitask learning approach to simultaneously perform classification, reconstruction, and segmentation of manipulated facial images. Cozzolino et al. [21] proposed the "ForensicTransfer" by combining classification and reconstruction, while Li et al. [22] proposed the "Face X-Ray" to detect Deepfake videos based on blending boundaries by combining classification and segmentation. However, those methods still need to improve the performance of the cross-dataset evaluation because they tend to train the classifier on a single small-scale dataset (i.e., FaceForensics++ [16] dataset), which is difficult to be generalized to other unseen datasets generated by using unseen Deepfake manipulation methods.

To make the Deepfake video detection method more robust on cross-domain datasets, this paper proposes a new method called FeatureTransfer, which is based on unsupervised domain adaptation. Extensive experiments demonstrate that the proposed method FeatureTransfer can improve the Deepfake detection performance of cross-dataset evaluation. The contributions of this work are summarized as follows:

(1) The unsupervised domain adaptation is first used to detect Deepfake videos in this work. A two-stage training pipeline called FeatureTransfer is designed for Deepfake detection.

(2) The feature extractor in preprocessing stage is pretrained on a large-scale Deepfake dataset DFDC-P [23] to extract more transferable feature vectors.

(3) Based on BP (backpropagation) and DANN (domain-adversarial neural network), an unsupervised domain adaptive network called BP-DANN is proposed.

The remainder of this paper is organized as follows. In Section 2, the related works are presented. In Section 3, our proposed method is described in detail. In Section 4, we provide comprehensive experimental results and analysis, as well as ablation studies. Finally, concluding remarks are drawn in Section 5.

2. Related Work

While the main focus of our work lies in the field of Deepfake detection, FeatureTransfer also intersects with the field of transfer learning, especially unsupervised domain adaptation. In the section, we clearly review previous Deepfake detection methods and transfer learning methods.

2.1. Deepfake Detection. To detect the Deepfake images or videos, most of the previous works are based on deep learning methods, which can be categorized into two detection methods: CNN-based methods [10, 13, 15, 16, 20–22] and RCNN-based methods [11, 17, 18]. The CNN-based methods extract face images from video frames and input them into the CNN for training and prediction to obtain the image-level result. These methods only use spatial information of a single frame in Deepfake videos. In addition, Qian et al. [24] detected Deepfake videos by mining clues in the frequency domain instead of the RGB domain. By contrast, the RCNN-based methods need a sequence of video frames for training and prediction to obtain the video-level result. These methods use both CNN and RNN, and they are called RCNN. Therefore, the RCNN-based methods can make full use of spatial and temporal information of Deepfake videos. Moreover, some Deepfake detection methods [12, 25] are based on traditional machine learning methods, Yang et al. [12] and Ciftci et al. [25] used SVM (support vector machine) as a classifier by extracting handcrafted features, such as biological signals. Finally, the methods mentioned above are summarized in Table 1.

2.2. Transfer Learning and Domain Adaptation. Transfer learning is an important branch of deep learning, which uses the knowledge of the source domain to assist the model in learning the knowledge of the target domain faster and better. Recently, transfer learning has been widely used in the field of forensics [21, 26, 27]. For example, loading the pretrained weight of ImageNet to the model before the model is trained is a simple transfer learning. Cozzolino et al. [21] trained the ForensicTransfer on the samples from the source domain and then performed fine-tuning with a small number of samples from the target domain to improve the performance of the ForensicTransfer on the target domain.

As a key field in transfer learning, domain adaptation aims to make the distribution of the source domain and the target domain in the feature space as close as possible. Meanwhile, the target model trained in the source domain can be transferred to the target domain to obtain good performance. Most works exploiting deep domain adaptation are based on discrepancy measurement. For instance, correlation alignment (CORAL) [28] and maximum mean discrepancy (MMD) [29] are used to reduce the distribution divergence between domains. Some works are based on discrepancy measurement domain-adversarial learning, such as domain-adversarial neural network (DANN) [30], multiadversarial domain adaptation (MADA) [31], and

Table 1: A summary of Deepfake detection methods.

Method	Classifier	Description
Matern et al. [10]	CNN	Handcrafted
Li et al. [13]	CNN	Self-supervised
MesoNet [15]	CNN	RGB
XceptionNet [16]	CNN	RGB
Nguyen et al. [20]	CNN	Multitask
ForensicTransfer [21]	CNN	Multitask
Face X-Ray [22]	CNN	Multitask
Qian et al. [24]	CNN	Frequency
Li et al. [11]	CNN + LSTM	Handcrafted
Guera et al. [17]	CNN + LSTM	RGB
Chen et al. [18]	CNN + LSTM	RGB
Yang et al. [12]	SVM	Handcrafted
FakeCatcher [25]	SVM	Handcrafted

transfer learning with dynamic adversarial adaptation network (DAAN) [32].

FeatureTransfer is a CNN-based method. In this work, a third-party Deepfake dataset is first used to train the CNN to extract the feature vectors of the face images. Then, the domain-adversarial neural network based on backpropagation (BP-DANN) is exploited for feature transfer training, which can improve the performance of Deepfake on cross-domain datasets.

3. Proposed Method

In this section, we introduce the details of the proposed method FeatureTransfer. Unlike the end-to-end adversarial training method NANN, FeatureTransfer exploits a two-stage adversarial training pipeline. As shown in Figure 1, the FeatureTransfer is composed of two parts: (a) the preprocessing stage, including face detection and feature vector extraction, and (b) BP-DANN unsupervised domain adaptive module.

3.1. Motivation. Most of the methods studying cross-dataset evaluation mainly trained the model on the FaceForensics++ [16] dataset or other small-scale datasets and then tested it on other datasets. Unfortunately, the methods used to generate Deepfake videos on different datasets are often different, which may lead to great gaps in the generated videos. As a result, it is difficult to train a model with good detection ability for all or most of the Deepfake datasets on a specific small-scale Deepfake dataset. In addition, many forensics methods are data-driven, so it is important to find a large-scale training model of the Deepfake dataset which contains a variety of Deepfake generation methods. Fortunately, a large-scale Deepfake dataset DFDC-F [23], including 23654 real videos and 104500 fake videos, meets our data-driven requirements. The fake videos in the DFDC-F dataset were created by different methods, including Deepfake Autoencoder (DFAE) [33], MM/NN face swap [34], NTH [35], and FSGAN [36]. Thus, the feature extractor CNN pretrained on the DFDC-F dataset can be used to extract more transferable feature vectors, which will be fed into BP-DANN for unsupervised domain adaptive training.

3.2. Problem Definition. In the unsupervised domain adaptation for Deepfake detection, it is assumed that the source distribution is $D_s = \{(x, y)|x \in X^s, y \in Y^s\}$, where X^s and Y^s are the input and label space of the source domain, respectively. Meanwhile, the target distribution is $D_t = \{(x, y)|x \in X^t, y \in Y^t\}$, where X^t and Y^t are the input and label space of the target domain. However, the input samples in the source domain are labelled but unlabelled in the target domain. D_s and D_t have the same label space so that $Y^s = Y^t = \{0, 1\}$, where "0" represents the real image or video and "1" represents the fake image or video. Moreover, each input x, the feature vector extracted from CNN in the preprocessing stage, has a domain label $d = 0$ if $x \in X^s$ while $d = 1$ if $x \in X^t$. The distributions between the two domains are similar, i.e., $D_s \cap D_t \neq \varnothing$ and $D_s \neq D_t$. This work aims to extract the more generalized feature vectors from the pretrained CNN in the preprocessing stage and design a deep neural network that enables learning of transferable features $f = G_f(x)$ and adaptive classifier $y = G_y(f)$ to reduce the gap between the two domains, such that the target risk $E_{(x,y)\sim D_t}[G_y(G_f(x)) \neq y]$ can be bounded by minimizing the source risk and the cross-domain discrepancy.

3.3. Preprocessing Stage. In the preprocessing stage, the face detection network MTCNN is first used to obtain the face region of the video frame, and the region is expanded by 1.2 times to crop the face image and save it. Then, the CNN (i.e., se_resnext101_32 × 4 d [37]) is pretrained on the third-party large-scale Deepfake dataset (i.e., DFDC-F [23]). Finally, the face images are fed into the CNN to extract the feature vectors with 2048 dimensions. The extracted feature vectors are saved so that they can be quickly loaded to the BP-DANN for unsupervised domain adaptive training.

3.4. Domain-Adversarial Network. The DANN can learn domain-invariant features through end-to-end adversarial training. The learning procedure is a two-player game: the first player is the domain discriminator G_d that is trained to distinguish the source domain from the target domain; the second player is the feature extractor G_f which extracts domain-invariant features that can confuse the domain discriminator. In the adversarial training for the two players, the parameter θ_f of feature extractor G_f is learned by maximizing the loss of the domain discriminator G_d, while the parameter θ_d of domain discriminator G_d is learned by minimizing the loss of the domain discriminator. In addition, the loss of label classifier G_y is also minimized. The overall loss function of DANN can be formalized as

$$L(\theta_f, \theta_y, \theta_d) = \frac{1}{n_s} \sum_{x_i \in D_s} L_y\left(G_y\left(G_f\left(x_i; \theta_f\right); \theta_y\right), y_i\right) - \frac{\lambda}{n_s + n_t} \sum_{x_i \in \left(D_s \cup D_t\right)} L_d\left(G_d\left(G_f\left(x_i; \theta_f\right); \theta_d\right), d_i\right), \tag{1}$$

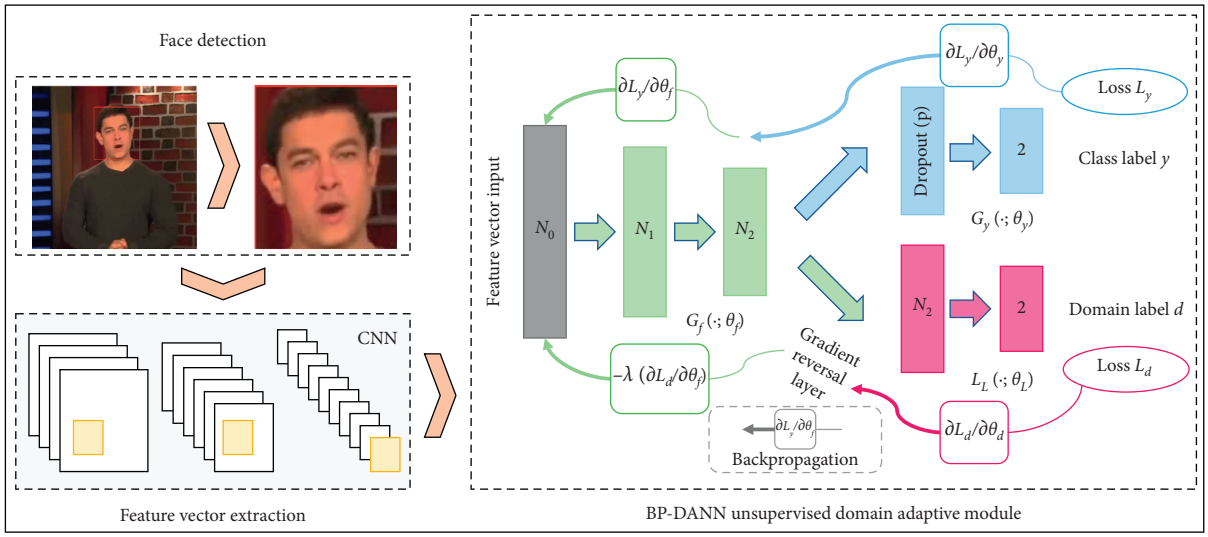


Figure 1: The pipeline of the proposed method FeatureTransfer. In the preprocessing stage, we obtain the face images of the video frame from the source and target domain and then feed them into CNN to extract the feature vectors. In the unsupervised domain adaptation stage, the BP-DANN consists of a feature extractor G_f (green), a label classifier G_y (blue), and a domain discriminator G_d (red). The gradient reversal layer connects G_f and G_d to realize unsupervised domain adaptation, and it multiplies the gradient by a certain negative constant during the backpropagation-based training.

where n_s and n_t are the number of samples in the source domain and the target domain, respectively, $d_i \in \{0, 1\}$ is the domain label of x_i, L_y is the loss for label prediction while L_d is the loss for domain discriminator, and λ is a hyper-parameter to trade-off the label classifier and the domain discriminator in the optimization problem. Based on equation (2) and equation (3), the optimization problem is to find the optimal parameters $\widehat{\theta}_f$, $\widehat{\theta}_y$, and $\widehat{\theta}_d$ that deliver a saddle point of equation (1) after the training converges.

$$\left(\widehat{\theta}_f, \widehat{\theta}_y\right) = \arg\min_{\theta_f, \theta_y} L\left(\theta_f, \theta_y, \widehat{\theta}_d\right), \tag{2}$$

$$\left(\widehat{\theta}_d\right) = \arg\min_{\theta_d} L\left(\theta_f, \theta_y, \widehat{\theta}_d\right). \tag{3}$$

3.5. BP-DANN Network Architecture. As shown in Figure 1, the network architecture of the proposed BP-DANN consists of three parts: feature extractor G_f, label classifier G_y, and domain discriminator G_d. These three parts are built by BP structure. G_f is composed of two fully connected layers, i.e., $L_f(N_0, N_1)$ and $L_f(N_1, N_2)$. The input and output dimensions of $L_f(N_0, N_1)$ are N_0 and N_1, where N_0 is 2048 and N_1 is 512. N_2 in $L_f(N_1, N_2)$ is set as 64. G_y is composed of a dropout layer with probability (p) of 0.5 and a fully connected layer $L_y(N_2, 2)$. G_d is composed of two fully connected layers, i.e., $L_d(N_2, N_2)$ and $L_d(N_2, 2)$. To obtain the more appropriate values of N_1, N_2, and p, the grid search is used for traversal search in this work.

4. Experiment

4.1. Dataset. In this section, the datasets related to the experiment are first introduced. Then, the details of the experiment implementation are given, and the experimental results are finally analyzed.

The DeepfakeTIMIT (DF-TIMIT) [38] dataset contains 640 Deepfake videos generated with a GAN-based method [39] and based on VidTIMIT [40] dataset. The videos are divided into two equal subsets: lower quality (LQ) and higher quality (HQ). In our experiment, we add 320 real videos of 32 related subjects in VidTIMIT, and the LQ subset is used for test.

The FaceForensics++ (FF) [16] dataset contains 1000 pristine (P) videos and 4000 fake videos generated by using the four most advanced facial manipulation methods, including DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT). This dataset covers three versions of compression qualities: Raw, c23, and c40. In our experiment, the FF-DF and FF-FS subsets with a compression quality of c23 are taken.

The DeepFakeDetection (DFD) [41] contains 363 real videos and 3068 Deepfake videos released by Google. Similar to FF, this dataset also covers three versions of compression qualities, including Raw, c23, and c40. In our experiment, c23 is taken.

The Celeb-DF [19] includes 408 real videos and 795 synthesized videos generated by using an improved version of the Deepfake algorithm.

The DFDC [23] dataset contains two versions: DFDC-Preview (DFDC-P) [42] and DFDC-Final (DFDC-F) [23]. The DFDC-P includes 1131 real videos and 4113 fake videos. The DFDC-F was released for the Deepfake Detection Challenge, and it includes 23654 real videos and 104500 fake videos. In our experiment, DFDC-F is taken to pretrain the CNN (i.e., se_resnext101_32 × 4 d), and DFDC-P is used for test.

As mentioned above, 30 frames are extracted from each video at equal intervals. Then, the face region of each frame is detected and saved as a face image. To balance the real and fake face images in DFDC-F, 30 frames from each fake video are extracted, but 150 frames from each real video are extracted. The numbers of face images in each dataset are listed in Table 2.

4.2. Implementation Details. Unlike the end-to-end adversarial learning training in DANN, a two-stage training strategy is adopted for FeatureTransfer.

In the first stage, a large-scale Deepfake dataset DFDC-F is used to train the CNN (i.e., se_resnext101_32 × 4 d). The CNN was initialized with pretrained weights on ImageNet, such that it can be used to extract more transferable feature vectors. The batch size is set to 128, and the total training epoch is 10. The Adam optimizer is used, where the initial learning rate is set to 2×10^{-3} and weight decay of 4×10^{-5}. After training, the CNN is used to extract the feature vectors of images, and the feature vectors are saved according to different datasets.

In the second stage, the feature vectors are loaded, and the BP-DANN is then trained. During the unsupervised domain adaptive adversarial training, the feature vectors of FF-DF (train set) are selected as the source domain, while the feature vectors of other test datasets are selected as the target domain. It should be noted that, due to a large number of images in the DFD, DFDC-P, and Celeb-DF datasets, only 10% of the images (the number of real and fake images is the same) in each dataset are used as the target domain for unsupervised adversarial training, and all images in each dataset are then tested after training. As for FF-FS and DF-TIMIT datasets, all images in the datasets are used as the target domain for unsupervised adversarial training, where the batch size is set to 128 and the total training epoch is 50. Instead of SGD used in DANN, the Adam optimizer with an initial learning rate of 1×10^{-4} is used. To suppress noisy signals from the domain classifier at the early stages of the training procedure, the hyperparameter λ in equation (1) is changed from 0 to 1 gradually based on the following equation:

$$\lambda = \frac{2}{1 + \exp(-\gamma \times p)} - 1, \tag{4}$$

where p is the training progress linearly changing from 0 to 1 and γ is set to 10.

4.3. Results and Analysis. The proposed method is compared with previous Deepfake detection methods, including Xception [16], FSSpotter [18], Face X-Ray [22], and se_resnext101_32 × 4 d [37]. The cross-domain Deepfake detection results are exhibited in terms of AUC (area under the curve) and ERR (equal error rate) on recently released datasets, such as DF-TIMIT, FF-FS (test set), DFD, DFDC-P, and Celeb-DF. The pretrained weight (all c23. p) provided by the author is loaded into Xception, and the model is then directly used to test on other datasets without retraining. Similarly, the se_resnext101_32 × 4 d is trained on DFDC-F, and the trained model is then directly used to test on other datasets without retraining. Due to the lack of open-source code for FSSpotter and Face X-Ray, the experimental results in the corresponding papers are directly used for comparison. The result with the clip length (T) of 1 in FSSpotter trained on FF-DF dataset is chosen as the image-level result. The Face X-Ray in the paper is trained on FF and BI [22] datasets.

Table 3 listed the cross-domain performance of all compared methods on different datasets. It can be seen that FeatureTransfer achieves the best performance on DFDC-P (seen dataset) and Celeb-DF (unseen dataset) compared to other methods in terms of AUC and ERR. Also, FeatureTransfer obtains a comparable result in FF-FS (unseen facial manipulations), DFD (unseen dataset), and DF-TIMIT (unseen dataset). In addition, Xception obtains the best performance on DF-TIMIT (unseen dataset) and FF-FS (seen dataset), while Face X-Ray obtains the best performance on DFD (unseen dataset) in terms of AUC and ERR. The performance of FSSpotter is relatively general, which could be caused by the fact that FSSpotter was only trained on the FF-DF dataset. However, the AUC result of the proposed method is only 2.24% lower than that of Xception on DF-TIMIT and 2.24% lower than that of Face X-Ray on DFD. Compared with se_resnext101_32 × 4 d, FeatureTransfer achieves a performance improvement ranged from 1% to 8% in terms of AUC on different datasets, especially 8% on the Celeb-DF. Compared with Xception, se_resnext101_32 × 4 d obtains better performance on more datasets, and this is why se_resnext101_32 × 4 d is used as the feature extractor of FeatureTransfer. In general, the results indicate that FeatureTransfer achieves better or comparable performance on cross-dataset evaluation, which mainly benefits from the more transferable feature vectors extracted from the deeper CNN called se_resnext101_32 × 4 d that was pretrained on a large-scale dataset DFDC-F. Moreover, using unsupervised domain adaptation can also improve the performance of the unlabelled Deepfake datasets in target domain.

4.4. Ablation Studies. To confirm the effectiveness of the proposed method, we explore the effect of different level evaluation and the effect of different training strategies in this section.

4.5. Effect of Different Level Evaluation. To verify the effectiveness and better generalization of the proposed method on different levels of evaluation, the results of image level and video level are compared. To get the video-level result, the prediction score for video is the predicted probability that the video is fake, which is calculated by averaging the scores of face images extracted from frames of a video. It can be seen from the image-level and video-level results shown

Table 2: The numbers of face images from each dataset.

	FF-DF		FF-FS		DF-TIMIT	DFD	Celeb-DF	DFDC-P	DFDC-F
	Train	Test	Valid	Test	LQ				
Real	21600	4200	4200	4200	9600	10890	12240	33897	2839521
Fake	21600	4200	4200	4200	9600	91740	23850	123412	2885045

Note. "Valid" is the short form of validation.

Table 3: The image-level results of all compared methods in terms of AUC (%) and EER (%) on each dataset.

Method	Test set									
	DF-TIMIT		FF-FS		DFD		DFDC-P		Celeb-DF	
	AUC	ERR	AUC	ERR	AUC	ERR	AUC	ERR	AUC	ERR
Xception [16]	98.80	5.95	99.56	2.74	83.06	25.92	82.10	27.23	72.54	34.71
FSS [18]	97.33	—	—	—	—	—	—	—	76.26	—
X-Ray [22]	—	—	98.00	-	95.40	8.37	80.92	27.54	80.58	26.70
Se_Res [37]	90.61	16.22	84.52	22.83	89.02	21.06	97.99	6.25	78.21	29.80
FT (ours)	96.56	8.05	88.62	19.52	91.00	16.21	98.77	5.75	86.21	22.42

Note. The "FSS," "X-Ray," "Se_Res," and "FT" are the short forms of "FSSpotter," "Face X-Ray," "se_resnext101_32 × 4 d," and "FeatureTransfer," respectively.

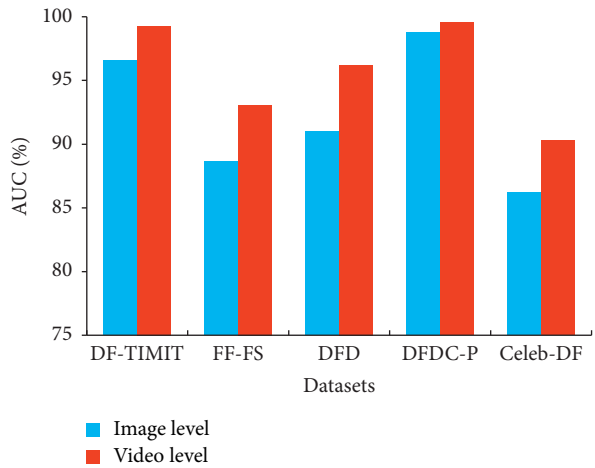


Figure 2: The results of different levels in terms of AUC (%) on each dataset.

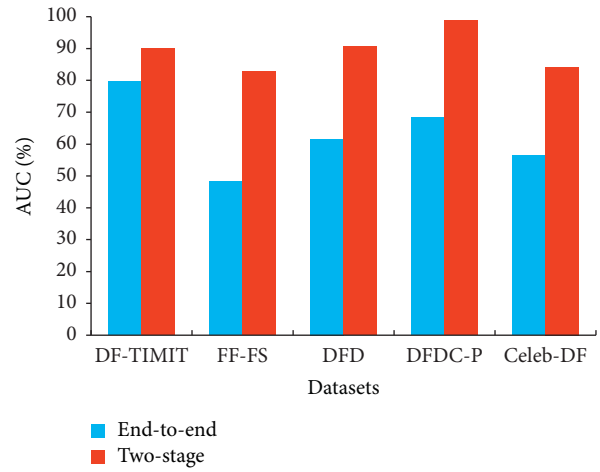


Figure 3: The image-level results of different training strategies in terms of AUC (%) on each dataset.

in Figure 2 that the video-level results are significantly improved on each dataset in terms of AUC (%).

4.6. Effect of Different Training Strategies. To demonstrate the benefits of the two-stage training strategy used in the proposed method, the experiments are conducted with the proposed FeatureTransfer and DANN having the same training epoch of 20. It should be noted that only the feature vectors of the source domain FF-DF (train set) and the target domain FF-FS (validation set) are used for unsupervised adversarial learning in our proposed method FeatureTransfer. The trained model is then directly evaluated on other datasets without additional adversarial learning. The backbone of DANN is se_resnext101_32 × 4 d, and DANN is trained by using an end-to-end training strategy with FF-DF (train set) as the source dataset and FF-FS (validation set) as the target dataset. As shown in Figure 3, in terms of AUC (%), the image-level results of FeatureTransfer using the two-stage training strategy are significantly improved on each dataset compared with DANN using the end-to-end training strategy.

5. Conclusions

In this work, FeatureTransfer, a two-stage Deepfake detection method based on unsupervised domain adaptation, is proposed. The feature vectors extracted from CNN are used for adversarial transfer learning in BP-DANN, which contributes to better performance than the end-to-end adversarial learning. Moreover, the feature extractor CNN pretrained on a large-scale Deepfake dataset can be used to extract more transferable feature vectors, which greatly reduce the gap between the source domain and the target domain during unsupervised domain adaptive training. The experimental results indicate that the proposed method achieves better and comparable performance for cross-domain Deepfake detection compared with previous methods.

However, there are still some limitations in our work. It is not an end-to-end detection method, and it needs a large-scale Deepfake dataset to pretrain the CNN to extract more transferable features, which takes a lot of time. Thus, in future work, we will devote ourselves to studying an end-to-end domain adaptive Deepfake detection method that does not require pretrained feature extractors.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province (2019B010139003), NSFC (61772349, U19B2022, and 61872244), Guangdong Basic and Applied Basic Research Foundation (2019B151502001), and Shenzhen R&D Program (JCYJ20180305124325555). This work was also supported by Alibaba Group through Alibaba Innovative Research (AIR) Program.

References

[1] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021.

[2] S. Hakak, W. Z. Khan, S. Bhattacharya, G. T. Reddy, and K.-K. R. Choo, "Propagation of fake news on social media: challenges and opportunities," in *Proceedings of the International Conference On Computational Data And Social Networks*, pp. 345–353, Dallas, TX, USA, December 2020.

[3] M. A. Azad, M. Alazab, F. Riaz, J. Arshad, and T. Abullah, "Socioscope: I know who you are, a robo, human caller or service number," *Future Generation Computer Systems*, vol. 105, pp. 297–307, 2020.

[4] R. Sagar, R. Jhaveri, and C. Borrego, "Applications in security and evasions in machine learning: a survey," *Electronics*, vol. 9, no. 1, p. 97, 2020.

[5] A. R. Javed and Z. Jalil, "Byte-level object identification for forensic investigation of digital images," in *Proceedings of the 2020 International Conference On Cyber Warfare And Security (ICCWS)*, pp. 1–4, Norfolk, VA, USA, March 2020.

[6] A. Rehman, S. U. Rehman, M. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering*, vol. 2021, Article ID 3059881, 1 page, 2021.

[7] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 964–975, 2017.

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

[9] H. Zhang, A. Jolfaei, and M. Alazab, "A face emotion recognition method using convolutional neural network and image edge computing," *IEEE Access*, vol. 7, pp. 159081–159089, 2019.

[10] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proceedings of the 2009 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92, IEEE, Snowbird, UT, USA, December 2009.

[11] Y. Li, M. 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, IEEE, Hong Kong, China, December 2018.

[12] 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, IEEE, Brighton, UK, May 2019.

[13] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," 2018, https://arxiv.org/abs/1811.00656.

[14] X. Zhang, S. Karaman, and S. Chang, "Detecting and simulating artifacts in GAN fake images," in *Proceedings of the 2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, Delft, Netherlands, December 2019.

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

[16] A. Rossler, D. Cozzolino, L. Verdoliva et al., "Faceforensics++: learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–11, Seoul, South Korea, October 2019.

[17] D. Güera, E. J. 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, IEEE, Auckland, New Zealand, November 2018.

[18] P. Chen, J. Liu, T. Liang et al., "Fsspotter: spotting face-swapped video by spatial and temporal clues," in *Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, London, UK, July 2020.

[19] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3207–3216, Salt Lake City, UT, USA, July 2020.

[20] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," 2019, https://arxiv.org/abs/1906.06876.

[21] D. Cozzolino, J. Thies, A. Rössler et al., "Forensictransfer: weakly-supervised domain adaptation for forgery detection," 2018, https://arxiv.org/abs/1812.02510.

[22] L. Li, J. Bao, T. Zhang et al., "Face x-ray for more general face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5001–5010, Seattle, WA, USA, June 2020.

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

[24] 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, Glasgow, UK, August 2020.

[25] U. A. Ciftci, I. Demir, and L. Yin, "Fakecatcher: detection of synthetic portrait videos using biological signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2020, Article ID 3009287, 1 page, 2020.

[26] H. Lin, J. Hu, W. Xiaoding, M. F. Alhamid, and M. J. Piran, "Towards secure data fusion in industrial IoT using transfer learning," *IEEE Transactions on Industrial Informatics*, vol. 2020, Article ID 3038780, 1 page, 2020.

[27] R. Abbasi, A. Kashif Bashir, J. Chen et al., "Author classification using transfer learning and predicting stars in co-author networks," *Software: Practice and Experience*, vol. 51, no. 3, pp. 645–669, 2020.

[28] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation, Lecture Notes in Computer Science," in *Proceedings of the European Conference on Computer Vision*, pp. 443–450, Springer, Glasgow, UK, August 2016.

[29] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the International Conference on Machine Learning*, pp. 97–105, PMLR, Long Beach, CA, USA, June 2015.

[30] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," ", PMLR, in *Proceedings of the International Conference on Machine Learning*, pp. 1180–1189, PMLR, Lille, France, July 2015.

[31] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," 2018, https://arxiv.org/abs/1809.02176.

[32] C. Yu, J. Wang, Y. Chen, and M. Huang, "Transfer learning with dynamic adversarial adaptation network," in *Proceedings of the 2019 IEEE International Conference On Data Mining (ICDM)*, pp. 778–786, IEEE, Beijing, China, November 2009.

[33] I. Petrov, D. Gao, N. Chervoniy et al., "Deepfacelab: a simple, flexible and extensible face swapping framework," 2020, https://arxiv.org/abs/2005.05535.

[34] D. Huang and F. De La Torre, "Facial action transfer with personalized bilinear regression," in *Proceedings of the European Conference on Computer Vision*, pp. 144–158, Florence, Italy, October 2012.

[35] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9459–9468, Seoul, Korea, October 2019.

[36] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7184–7193, Seoul, Korea, October 2019.

[37] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

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

[39] Shaoanlu, "faceswap-gan github," 2020, https://github.com/shaoanlu/faceswap-GAN.

[40] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference, Advances in Biometrics," in *Proceedings of the International Conference on Biometrics*, pp. 199–208, Springer, Alghero, Italy, June 2009.

[41] N. Dufour, Google Research, and J. A. Gully, "Contributing data to deepfake detection research," 2020, https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

[42] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, https://arxiv.org/abs/1910.08854.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9939232, 11 pages
https://doi.org/10.1155/2021/9939232

Research Article

Face Antispoofing Method Using Color Texture Segmentation on FPGA

Youngjun Moon[1], **Intae Ryoo**[1] **and Seokhoon Kim**[2,3]

[1]*Department of Computer Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, Republic of Korea*
[2]*Department of Software Convergence, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea*
[3]*Department of Computer Software Engineering, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea*

Correspondence should be addressed to Intae Ryoo; itryoo@khu.ac.kr and Seokhoon Kim; seokhoon@sch.ac.kr

Received 4 March 2021; Revised 5 April 2021; Accepted 29 April 2021; Published 10 May 2021

Academic Editor: Beijing Chen

User authentication for accurate biometric systems is becoming necessary in modern real-world applications. Authentication systems based on biometric identifiers such as faces and fingerprints are being applied in a variety of fields in preference over existing password input methods. Face imaging is the most widely used biometric identifier because the registration and authentication process is noncontact and concise. However, it is comparatively easy to acquire face images using SNS, etc., and there is a problem of forgery via photos and videos. To solve this problem, much research on face spoofing detection has been conducted. In this paper, we propose a method for face spoofing detection based on convolution neural networks using the color and texture information of face images. The color-texture information combined with luminance and color difference channels is analyzed using a local binary pattern descriptor. Color-texture information is analyzed using the Cb, S, and V bands in the color spaces. The CASIA-FASD dataset was used to verify the proposed scheme. The proposed scheme showed better performance than state-of-the-art methods developed in previous studies. Considering the AI FPGA board, the performance of existing methods was evaluated and compared with the method proposed herein. Based on these results, it was confirmed that the proposed method can be effectively implemented in edge environments.

1. Introduction

Recently, authentication systems based on biometric information have been applied to various mobile devices such as smartphones, and many users perform identity authentication using facial or fingerprint information instead of the existing password input methods. In addition, biometric authentication is being applied to bank transactions and mobile payment applications. As a result, researchers are greatly interested in developing high-performance authentication systems.

Among user biometric information, face images are the most widely used biometric identifier because the associated registration and authentication processes are noncontact and concise. However, face images are very easy to acquire using social networks, etc., and are vulnerable against various spoofing techniques, including printed photos and video replay. To solve this problem, research utilizing software solutions have become popular, rather than antispoofing hardware solutions using additional sensors. These software approaches can be classified into motion-based methods and texture-based methods [1].

The motion-based counterfeit face detection method measures eye/head movement, eye blinking, and changes in facial expression [2, 3]. In the case of counterfeit face detection methods utilizing eyes, note that a still face such as in a photograph does not exhibit eye blinking or pupil movement, as opposed to real human faces which exhibit

relatively large amounts of movement over time. This method is very simple and fast. However, this method classifies a spoofing face using only eye movement and thus cannot defend against simple attack variations that focus on and accurately emulate the eye area based on a photo.

The texture-based spoofing face detection method mainly uses lighting characteristics that appear differently between 2D plane and 3D stereoscopic objects or uses a fine texture difference between the spoofing face data and live face data through an external medium such as printing [4–8]. This method mainly uses a local image descriptor such as an LBP (local binary pattern) [9] to express differences in the texture characteristics between live and spoofing face images. Such texture-based methods have been actively researched due to the advantages of easy implementation and short detection times; however, these methods have difficulty classifying liveness faces in nonuniform images or images with large amounts of noise. Recently, researchers have been working on the detection of spoofing faces using convolutional neural networks (CNNs) [10, 11]. Since this method can effectively derive features through learning, its performance is improved over existing texture-based detection methods.

Although the field of spoofing face detection has developed tremendously, the existing methods mainly focus on the brightness information of face images. More specifically, other color information, which is similar to brightness information, is often overlooked in spoofing face detection. Therefore, by considering both color and brightness information of face images, a method was proposed that independently extracts texture features from the brightness space and color space of the face image using an LBP [12].

The difference between a real face and spoofing face is discriminated using a descriptor (such as an LBP) that encodes comparison results with respect to surrounding pixel values in a binary pattern at all pixel locations. However, since it is possible to produce high-resolution images, it is very difficult to distinguish detailed surface differences between real faces and spoofing faces using only pixel brightness.

In this paper, we propose a liveness face detection method based on a convolutional neural network utilizing the color and texture information of a face image. The proposed method analyzes the combined color-texture information in terms of its luminance and color difference channels using an LBP descriptor. For color-texture information analysis, the Cb, S, and H bands are used from the color spaces.

The rest of the paper is organized as follows. In Section 2, the related key technologies are illustrated. The proposed scheme for our color-texture-based antispoofing is presented in Section 3. Section 4 thoroughly presents the results and discussion. Finally, conclusions are presented in Section 5.

2. Related Works

2.1. Face Antispoofing. Conventional face antispoofing methods generally create spoofing patterns by extracting features from face images. Classic local descriptors such as LBP [13], SIFT [14], SURF [15], HOG [16], and DoG [17] are used to extract frame level functions, while methods such as dynamic texture [18], micromotion [19], and eye blinking [20] extract video features.

Recently, several deep learning-based methods have been studied to prevent face spoofing at the frame and video levels. In frame level methods [21–24], the pretrained CNN model is fine-tuned to extract features from the binary classification setup [25–27].

2.2. Color Spaces. RGB is a color space commonly used for sensing and displaying color images. However, its use in image analysis is typically limited because the three colors (red, green, and blue) are not separated according to luminance and color difference information. Thus, it is common to additionally convert the RGB information into YCbCr and HSV information before use. These two latter color spaces are based on luminance and chrominance information [28–31]. In particular, the YCbCr Color space separates RGB into luminance (Y), chrominance blue, and chrominance red. Similarly, the HSV color space uses the hue and saturation dimensions to define the color differences of the image, and the value dimension corresponds to the luminance.

2.3. LBP (Local Binary Pattern). LBPs [32, 33] are a feature developed for classifying image textures. Since then, LBPs have been used for face recognition. LBPs are a simple operation used for image analysis and recognition and are robust to changes in discrimination and lighting. Equation (1) is an LBP equation:

$$\mathrm{LBP}(p,r) = \sum_{p=1}^{p-1} s\left(g_p - g_c\right)2^p, \tag{1}$$

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here, g_p ranges over the pixel values excluding the center pixel and g_c is the center pixel in equation (1). In Figure 1, P is the number of adjacent pixels and R is the radius of the circle. Figure 2 shows an example result of LBP operation applied to a real photo [34].

3. Proposed Scheme for Color-Texture-Based Antispoofing

The RGB color space contains three color components, red, green, and blue; the YCbCr color space contains brightness and saturation information, and the HSV color space contains three components: hue, saturation, and brightness. Each color space contains different information and has its own characteristics. RGB contains rich spatial information that most closely resembles the colors seen by humans, while the YCbCr and HSV color spaces contain information that is more sensitive to brightness. The RGB color space can be converted into HSV and YCbCr, and the specific calculation is as follows:

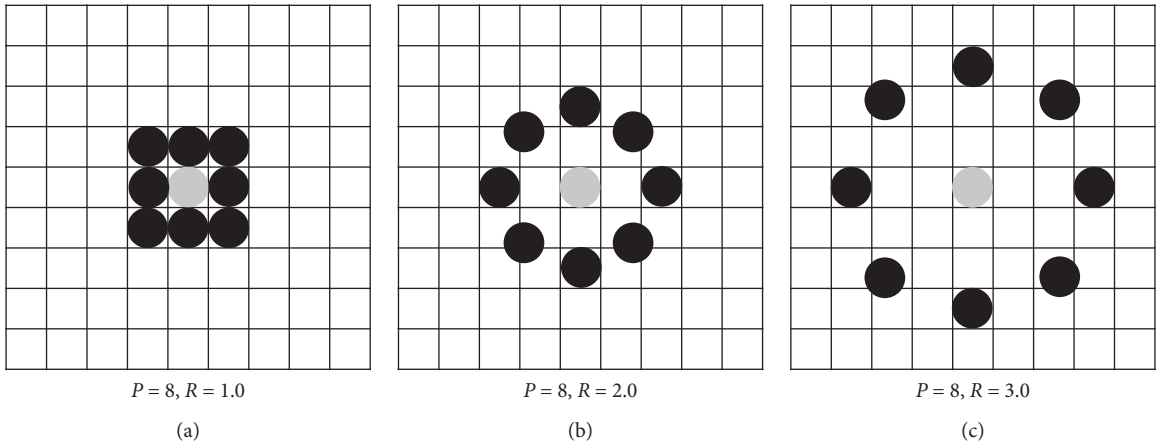


Figure 1: Example of a local binary pattern.

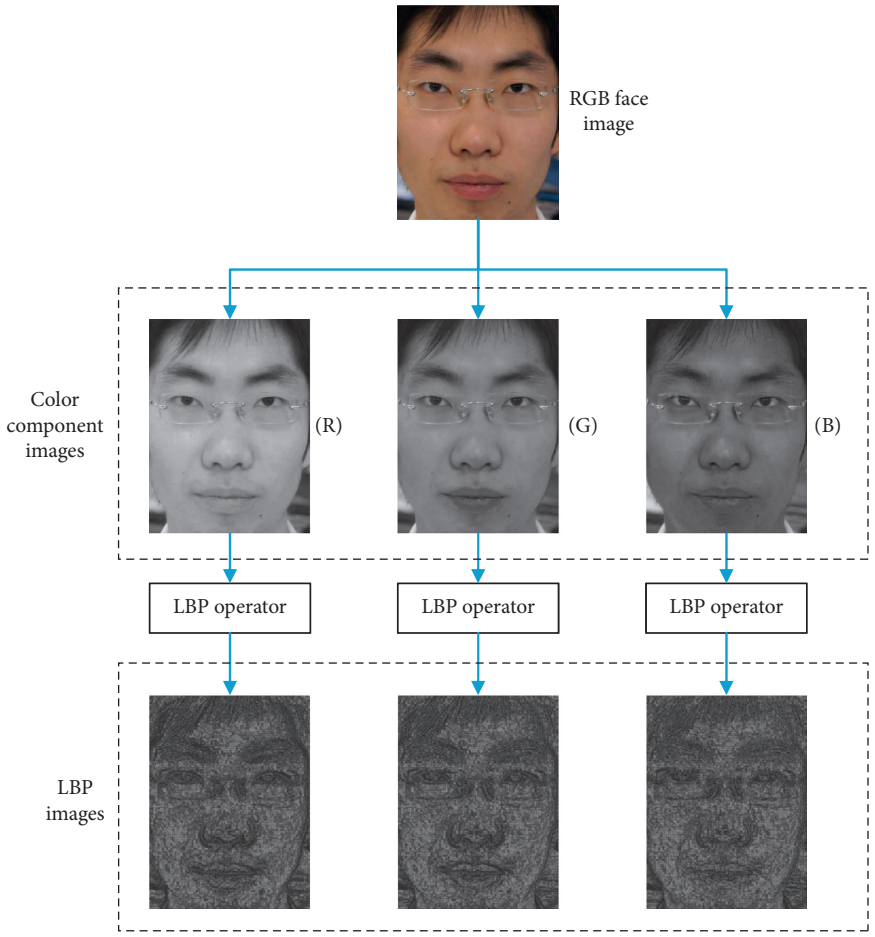


Figure 2: Visualization of LBP operation performed on each color band image.

$$V = \max(R, G, B),$$

$$S = \begin{cases} \dfrac{V - \min(R, G, B)}{V}, & \text{if } V \neq 0, \\ 0, & \text{if } V = 0, \end{cases}$$

$$H = \begin{cases} \dfrac{60(G - B)}{V - \min(R, G, B)}, & \text{if } V = R, \\ 120 + \dfrac{60(B - R)}{V - \min(R, G, B)}, & \text{if } V = G, \\ 240 + \dfrac{60(R - G)}{V - \min(R, G, B)}, & \text{if } V = B, \end{cases} \tag{3}$$

if $H < 0, H = H + 360$.

The YCbCr calculation formula is shown as

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B, \\ \text{Cb} &= 0.564(B - Y), \\ \text{Cr} &= 0.713(R - Y). \end{aligned} \tag{4}$$

In existing methods, RGB face images are converted into the YCbCr and HSV color spaces, and the spoofing images are classified by applying an LBP to each color space. However, this method increases the amount of computation because it uses a 6-channel color space. Figure 3 shows a conceptual diagram of the existing methods.

In this paper, we use a 3-channel color space consisting of Cb, S, and V, from which many facial features can be derived. The proposed method aims toward high-speed processing and robustness against lighting changes in face antispoofing. Figure 4 shows a conceptual diagram of the proposed scheme.

The advantages of this approach are summarized as follows:

(1) This proposed scheme reduces false detection by using a 3-channel color space in which sufficient facial feature information is expressed

(2) This proposed scheme uses less memory with fewer feature dimensions, thus enabling high-speed processing

4. Performance Evaluation

4.1. Train/Test Dataset. In this paper, we performed a spoofing face detection test using the CASIA Face Antispoofing Database (CASIA-FASD) [35] for performance evaluation. CASIA-FASD consists of real face videos and fake face videos acquired from 50 different users. The real face videos consist of three types of videos: low quality, medium quality, and high quality. Similarly, the fake face videos consist of three types of fake attack videos: printed photo attacks, cut photo attacks, and video relay attacks. Videos for 20 people are used for learning, while the remaining videos for 30 people are used for performance evaluation.

We extracted each frame from the CASIA-FASD dataset videos images for performance evaluation. In total, 4,577 live face images, 5,054 printed photo attack images, 2,368 cut photo attack images, and 4,429 video replay attack images were used for learning. In addition, 5,912 live face images, 7,450 printed photo attack images, 4,437 cut photo attack images, and 5,652 video replay attack images were used for evaluation. Table 1 shows detailed information on data partitioning of CASIA-FASD.

4.2. Experimental Setup. In this paper, we used FPGA for performance evaluation. We evaluated the performance of the proposed scheme by using the AI Accelerator of FPGA. The specifications of FPGA and the implemented board are shown in Figure 5.

Zynq® UltraScale+™ MPSoC devices provide 64-bit processor scalability while combining real-time control with soft and hard engines for graphics, video, waveform, and packet processing. Built on a common real-time processor and programmable logic-equipped platform, three distinct variants (dual application processor (CG) devices, quad application processor and GPU (EG) devices, and video codec (EV) devices) are included, creating numerous possibilities for various applications such as 5G wireless, next-generation ADAS, and industrial internet-of-things technologies [36].

Vitis AI is Xilinx's development stack for AI inference on Xilinx hardware platforms, including both edge devices and Alveo cards. It consists of an optimized IP, tools, libraries, models, and example designs. Vitis AI is designed with high efficiency and ease of use in mind, leading to great potential for AI acceleration on Xilinx FPGA and ACAP [37].

Face antispoofing detection uses AlexNet based on CNN. AlexNet is a basic model utilizing a convolutional layer, a pooling layer, and a fully connected layer [38].

AlexNet consists of five convolution layers and three full-connected (FC) layers, where the last FC layer uses softmax as an active function for category classification. Figure 6 shows Alexnet's CNN architecture.

4.3. Experimental Analysis Method. To evaluate the proposed scheme, we measured the HTER (Half Total Error Rate) in the CASIA-FASD dataset. The HTER is calculated using the false acceptance rate (FAR) and false rejection rate (FRR) in the attack dataset, both of which are defined below. The HTER calculation is given as follows [39]:

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2}. \tag{5}$$

The FAR [40] is a measure of how likely the biometric security system will incorrectly accept an access attempt by an unauthorized user. A system's FAR typically is defined as the ratio of the number of false acceptances divided by the number of identification attempts.

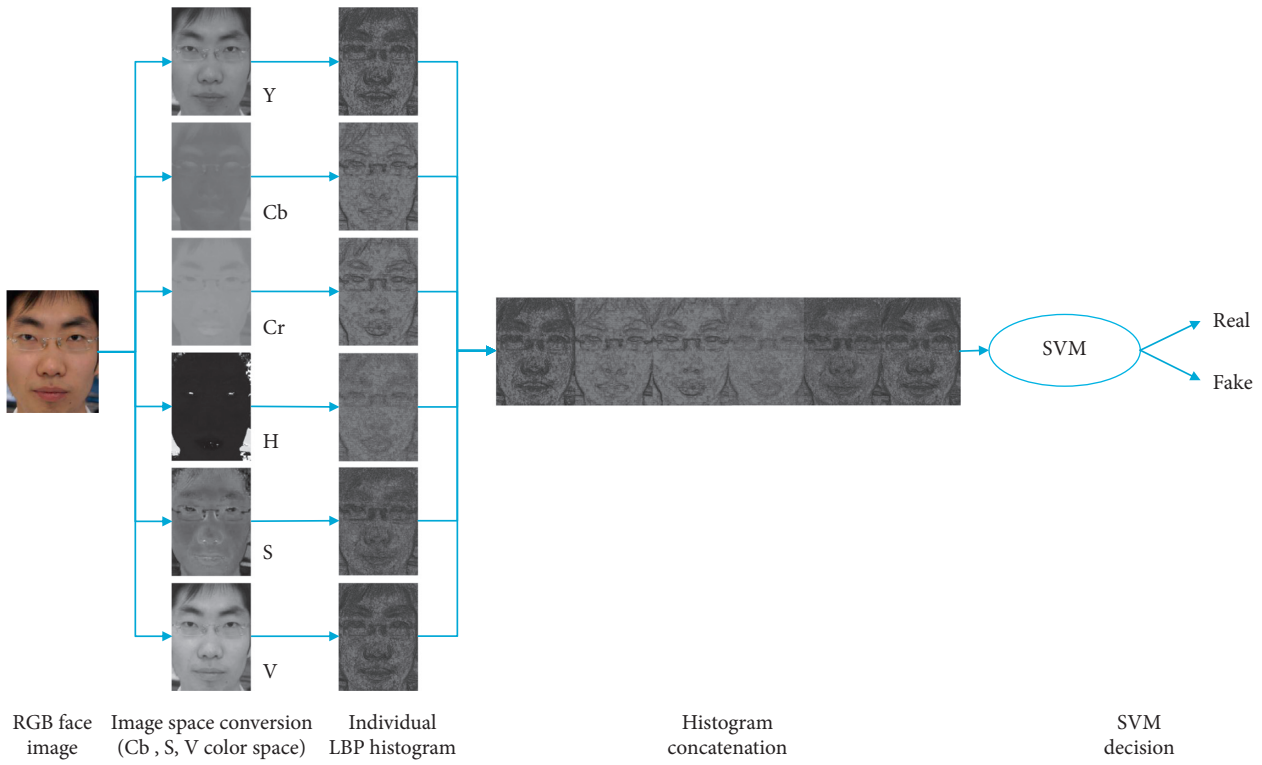


Figure 3: Conceptual diagram of existing methods.

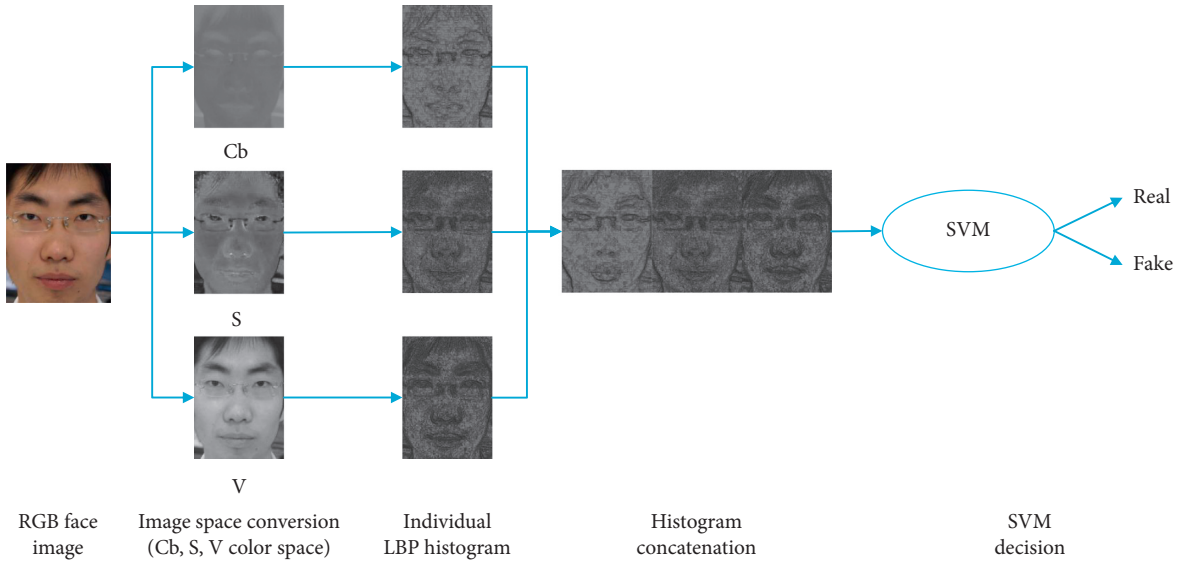


Figure 4: Conceptual diagram of the proposed scheme.

Table 1: Details on data partitioning in CASIA-FASD.

Type	Genuine images (ea)	Spoof images (ea)			
		Printed photo attacks	Cut photo attacks	Video replay attacks	Total
Training set	4,577	5,054	2,368	4,429	11,851
Testing set	5,912	7,450	4,437	5,652	17,539

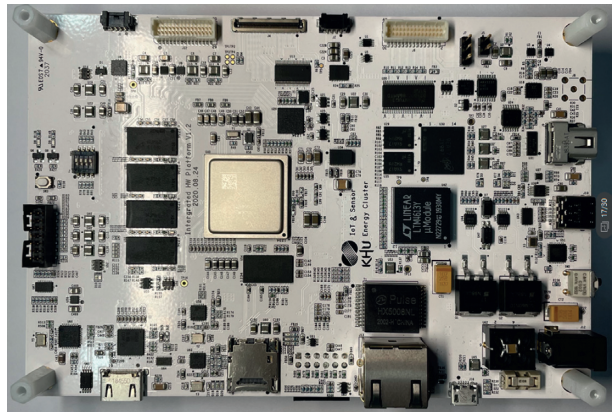


Figure 5: AI FPGA board.

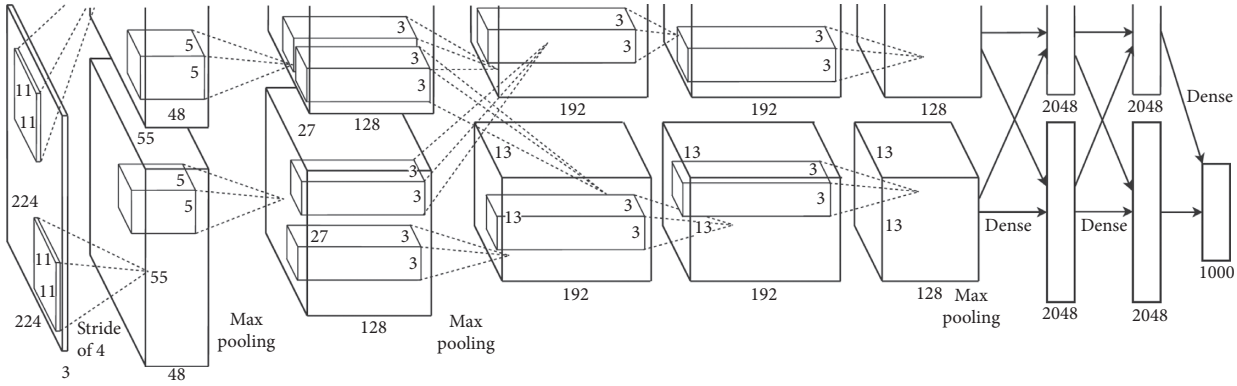


Figure 6: Illustration of the proposed CNN architecture, explicitly showing the delineation of responsibilities between the two GPUs: one GPU runs the layer parts at the top of the figure while the other runs the layer parts at the bottom.

The FRR [41] is a measure of how likely the biometric security system will incorrectly reject an access attempt by an authorized user. A system's FRR typically is defined as the ratio of the number of false rejections divided by the number of identification attempts.

Smaller HTER values indicate good performance, where HTER is defined using only misclassification ratios. Additionally, the EER (equal error rate) refers to the rate at which the FRR and FAR values converge to one another, where a small value also indicates good performance.

The EER [42] is a biometric security system algorithm used to predetermine the threshold values for the FAR and FRR. When the rates are equal, the common value is referred to as the equal error rate. The lower the ERR, the better the accuracy of the biometric system.

ROC (receiver operating characteristic) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

AUC (area under the curve) is the area under the ROC Curve. If the AUC value is high, it means that the model for classifying objects has excellent performance.

4.4. Experimental Results and Discussion. To verify the performance of the proposed scheme, eight scenarios were compared and tested using the CASIA-FASD attack dataset.

Table 2 shows HTERs according to eight different scenarios in the CASIA-FASD dataset. The proposed method showed improved performance for printed photo attacks, cut photo attacks, and video replay attacks. Figure 7 shows the performance comparison for the CASIA-FASD dataset.

Table 3 shows the EER values according to eight different scenarios for the CASIA-FASD dataset. Compared with the proposed scheme, only the "YCbCr_lbp + HSV_lbp" scheme has good EER performance.

The receiver operating characteristic (ROC) curves are presented. These curves show the error of the false positive rates against the true positive rates. ROC curves are best used for comparing the performance of various systems. Figures 8 and 9 show the ROC curves generated for each scenario in the CASIA-FASD dataset.

Table 4 shows the FAR, FRR, and area under the curve (AUC) results according to eight different scenarios in the CASIA-FASD dataset. A high AUC indicates good performance.

Table 5 shows the accuracy for different facial spoofing attacks. The accuracy for YCbCr_lbp + HSV_lbp is the

Table 2: Performance of various scenarios on the CASIA-FASD dataset.

Scenario	HTER (%)			
	Printed photo attacks	Cut photo attacks	Video replay attacks	Total
YCbCr	13.05	12.41	10.28	11.92
HSV	6.34	5.34	5.34	5.67
YCbCr_lbp	2.80	3.05	1.30	2.38
HSV_lbp	9.70	9.16	8.85	9.24
YCbCr + HSV	5.66	4.55	4.55	4.92
YCbCr_lbp + HSV	5.53	4.52	4.50	4.85
YCbCr_lbp + HSV_lbp	2.78	2.53	2.12	2.48
Proposed approach	**2.46**	**1.24**	**0.57**	**1.42**

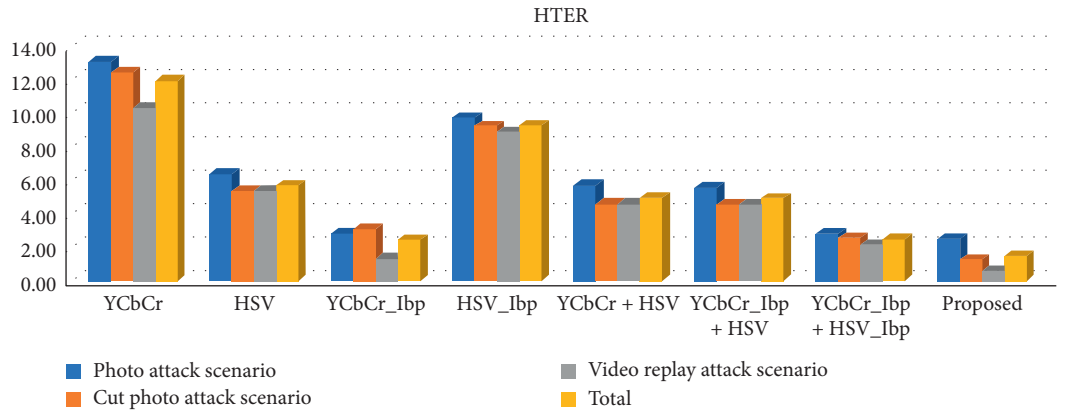


Figure 7: Performance comparison for the CASIA-FASD dataset.

Table 3: Equal error rate values for the CASIA-FASD dataset.

Scenario	EER (%)			
	Printed photo attacks	Cut photo attacks	Video replay attacks	Total
YCbCr	25.22	18.39	**5.39**	16.98
HSV	10.14	**0.00**	12.66	13.23
YCbCr_lbp	14.55	19.35	27.68	23.16
HSV_lbp	11.09	3.57	12.16	10.76
YCbCr + HSV	**6.13**	**0.00**	12.95	11.09
YCbCr_lbp + HSV	7.09	0.02	8.22	**7.58**
YCbCr_lbp + HSV_lbp	7.29	5.56	6.52	9.50
Proposed approach	10.79	12.91	7.76	10.22

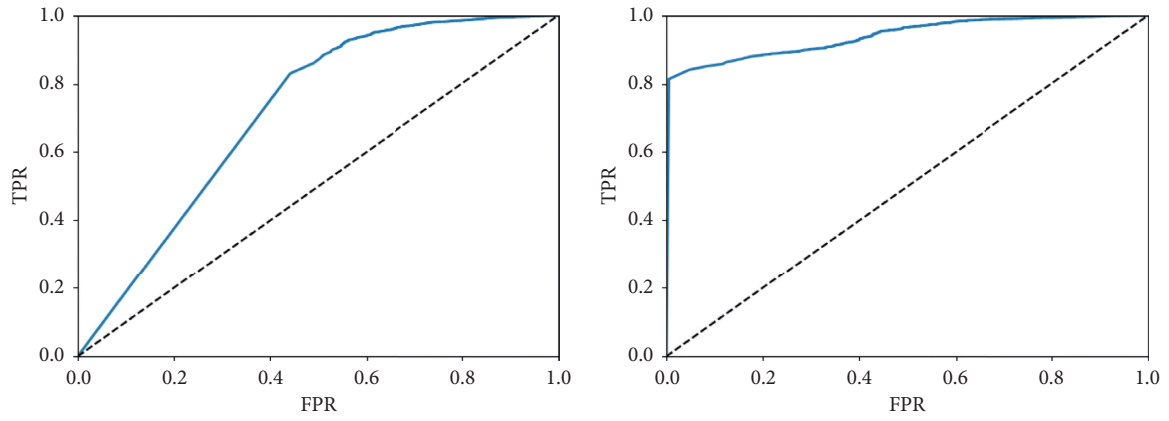


Figure 8: Continued.

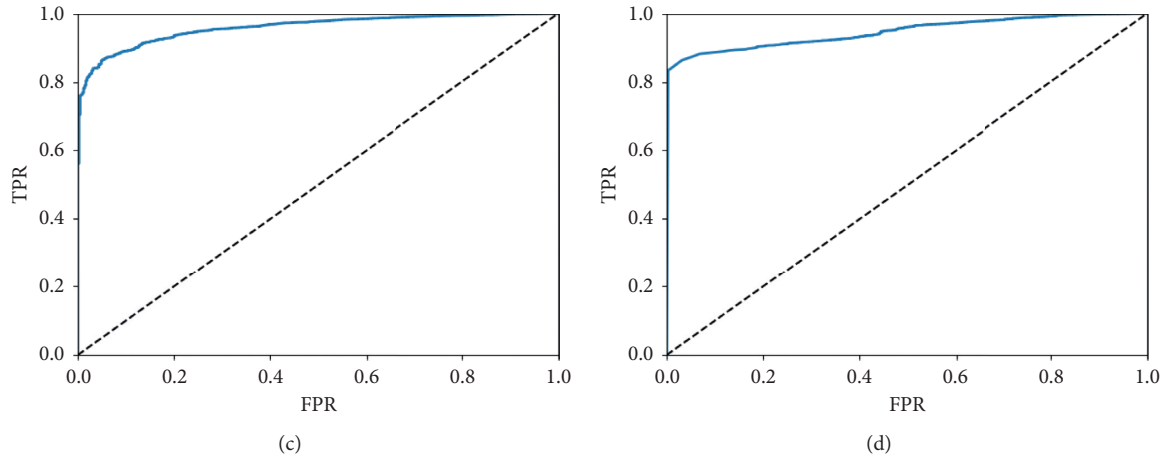


Figure 8: Receiver operating characteristic curves for the (a) YCbCr, (b) HSV, (c) YCbCr_lbp, and (d) HSV_lbp scenarios.

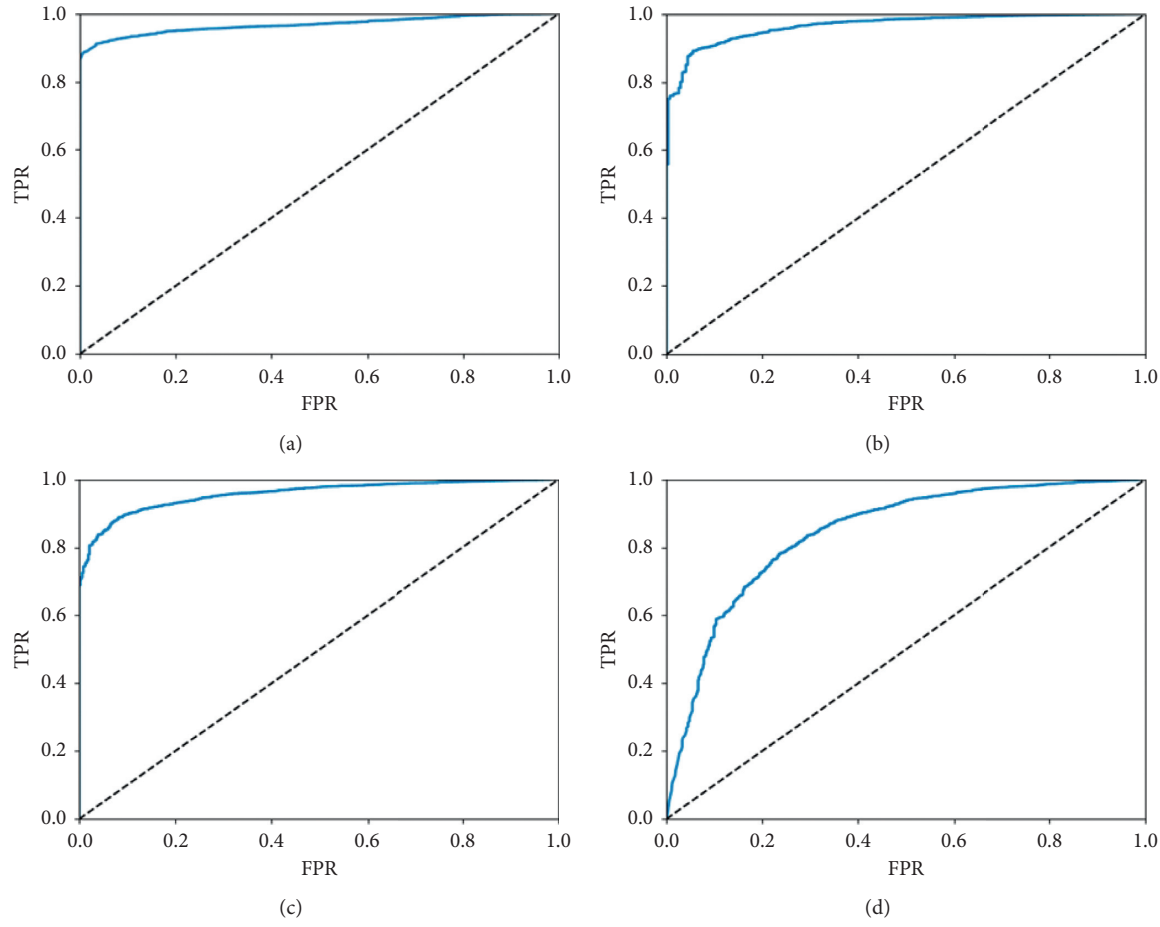


Figure 9: Receiver operating characteristic curves for the (a) YCbCr + HSV, (b) YCbCr_lbp + HSV, (c) YCbCr_lbp + HSV_lbp, and (d) proposed scenarios.

highest, but the proposed method shows similar performance.

The overall test results of this paper are shown in Table 6. Compared to the already existing YCbCr_lbp + HSV_lbp method, the method proposed in this paper has improved performance with respect to printed photo attacks (0.18%), cut photo attacks (0.69%), and video replay attacks (1.52%), with an overall performance improvement of 0.73%. Additionally, the ERR was low, while the accuracy values were similar. Overall, the YCbCr_lbp + HSV_lbp method showed

Table 4: FAR, FRR, and AUC performances for the eight scenarios.

Scenarios	FAR (%)				FRR (%)	AUC
	Printed photo attacks	Cut photo attacks	Video replay attacks	Total		
YCbCr	5.53	4.26	3.52	4.44	20.57	0.72
HSV	2.01	**0.00**	4.05	2.02	10.77	0.94
YCbCr_lbp	2.99	3.49	5.11	3.87	2.65	0.84
HSV_lbp	1.71	0.63	4.51	2.29	17.83	0.96
YCbCr + HSV	2.22	**0.00**	3.26	1.82	9.19	0.95
YCbCr_lbp + HSV	2.05	0.05	2.58	1.56	9.08	**0.97**
YCbCr_lbp + HSV_lbp	**1.30**	0.81	2.44	**1.52**	4.90	**0.97**
Proposed approach	3.78	1.35	**2.25**	2.46	**1.17**	0.96

Table 5: Accuracy comparison.

Scenarios	Accuracy (%)
YCbCr	91.34
HSV	95.66
YCbCr_lbp	96.49
HSV_lbp	93.73
YCbCr + HSV	96.19
YCbCr_lbp + HSV	96.42
YCbCr_lbp + HSV_lbp	**97.76**
Proposed approach	97.54

Table 6: Compare all results.

Scenarios	HTER (%)	EER (%)	FAR (%)	FRR (%)	AUC	Accuracy (%)
YCbCr	11.92	16.98	4.44	20.57	0.72	91.34
HSV	5.67	13.23	2.02	10.77	0.94	95.66
YCbCr_lbp	2.38	23.16	3.87	2.65	0.84	96.49
HSV_lbp	9.24	10.76	2.29	17.83	0.96	93.73
YCbCr + HSV	4.92	11.09	1.82	9.19	0.95	96.19
YCbCr_lbp + HSV	4.85	7.58	1.56	9.08	**0.97**	96.42
YCbCr_lbp + HSV_lbp	2.48	**9.50**	**1.52**	4.90	**0.97**	**97.76**
Proposed approach	**1.42**	10.22	2.46	**1.17**	0.96	97.54

similar performance but uses six color space channels, while the proposed method uses only three-color space channels, leading to a faster calculation speed.

5. Conclusions

In this paper, we proposed a face antispoofing method utilizing CNN learning and inference and constructed important parameters by extracting texture information via an LBP from the face image color space. CASIA-FASD was used as the dataset for performance verification. Images were extracted from videos and divided into printed photo attacks, cut photo attacks, and video replay attacks. These images extracted from the CASIA-FASD dataset were used for both training and evaluation. It was confirmed that the detection performance was improved by separating the color space from the face image in addition to the Cb, S, and V color space, which is useful for antispoofing. In previous studies, a 6-channel (YCbCr + HSV) color space was typically used, leading to large computational costs. On the contrary, the proposed approach reduces the computational load by instead considering only three (Cb, S, V) color space channels. Considering the AI FPGA board, the performances of the existing methods were evaluated with that of the proposed scheme. It was confirmed that the proposed method can be effectively used in edge environments.

As future work, we want to verify the performance against another well-known face spoof dataset. In addition, we plan to conduct performance tests between databases.

Data Availability

The data used to support the finding were included in this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by BK21 FOUR (Fostering Outstanding Universities for Research) (no. 5199990914048), and this research was supported by Basic Science Research Program through the National Research Foundation of

Korea (NRF) funded by the Ministry of Education (NRF-2020R1I1A3066543). In addition, this work was supported by the Soonchunhyang University Research Fund.

References

[1] Z. Akhtar and G. Luca Foresti, "Face spoof attack recognition using discriminative image patches," *Journal of Electrical and Computer Engineering*, vol. 2016, Article ID 4721849, 14 pages, 2016.

[2] H. K. Jee, S. U. Jung, and J. H. Yoo, "Liveness detection for embedded face recognition system," *International Journal of Biological and Medical Sciences*, vol. 1, pp. 235–238, 2006.

[3] W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *Proceedings of the 2009 International Conference on Image Analysis and Signal Processing IASP*, pp. 233–236, IEEE, Linhai, China, April 2009.

[4] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," in *Proceedings of the SPIE - International Society for Optics and Photonics*, pp. 296–303, Choufu, Japan, March 2004.

[5] A. D. S. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Video-based face spoofing detection through visual rhythm analysis," in *Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 221–228, IEEE, Ouro Preto, Brazil, August 2012.

[6] W. R. Schwartz, A. Rocha, and H. P. Edrini, "Face spoofing detection through partial least squares and low-level descriptors," in *Proceedings of the 2011 International Joint Conference on Biometrics (IJCB)*, pp. 1–8, IEEE, Washington, WA, USA, October 2011.

[7] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: a public database and a baseline," in *Proceedings of the 2011 Joint Conference on Biometrics (IJCB)*, pp. 1–7, IEEE, Washington, WA, USA, October 2011.

[8] J. Määttä, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using micro-texture analysis," in *Proceedings of the 2011 international joint conference on Biometrics (IJCB)*, pp. 1–7, IEEE, Washington, WA, USA, October 2011.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[10] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," 2014, https://arxiv.org/abs/1408.5601.

[11] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo, "Transfer learning using convolutional neural networks for face anti-spoofing," in *Lecture Notes in Computer Science*Springer, Berlin, Germany, 2017.

[12] Z. Xu, S. Li, and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in *Proceedings of 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 141–145, IEEE, Kuala Lumpur, Malaysia, November 2015.

[13] T. Pereira, A. Anjos, J. M. De Martino, and S´. Marcel, "Lbp-top based countermeasure against face spoofing attacks," in *Proceedings of Asian Conference on Computer Vision*, pp. 121–132, Springer, Daejeon, Korea, November 2012.

[14] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: spoof detection on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2268–2283, 2016.

[15] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing using speeded-up robust features and Fisher vector encoding," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 141–145, 2017.

[16] J. Komulainen, A. Hadid, and M. Pietikainen, "Context based face anti-spoofing," in *Proceedings of the 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–8, Arlington, VA, USA, September 2013.

[17] P. Bruno, C. Michelassi, and R. Anderson, "Face liveness detection under bad illumination conditions," in *Proceedings of the 2011 18th IEEE International. Conference on Image Processing. (ICIP 2011)*, pp. 3557–3560, IEEE, Brussels, Belgium, September 2011.

[18] J. Komulainen, A. Hadid, and M. Pietik¨ainen, "Face spoofing detection using dynamic texture," in *Asian Conference on Computer Vision*, pp. 146–157, Springer, Daejeon, Korea, November 2012.

[19] T. Ahmad Siddiqui, S. Bharadwaj, T. I Dhamecha et al., "Face anti-spoofing with multifeature videolet aggregation," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 1035–1040, IEEE, Cancun, Mexico, December 2016.

[20] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, Rio de Janeiro, Brazil, October 2007.

[21] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *The sixth International Conference on Image Processing Theory, Tools and Applications (IPTA'16)*, pp. 1–6, Oulu, Finland, December 2016.

[22] K. Patel, H. Han, and A. K. Jain, "Cross-database face anti-spoofing with robust feature representation," in *Proceedings of the Chinese Conference on Biometric Recognition*, pp. 611–619, Springer, Chengdu, China, October 2016.

[23] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," in *Proceedings of the 2019 Inter-national Conference on Biometrics*, Crete, Greece, June 2019.

[24] J. Amin, Y. Liu, and X. Liu, "Face despoofing: anti-spoofing via noise modeling," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 290–306, Munich, Germany, September 2018.

[25] M. Sajid, N. Ali, S. Hanif Dar et al., "Data augmentation-assisted makeup-invariant face recognition," *Mathematical Problems in Engineering*, vol. 2018, Article ID 2850632, 10 pages, 2018.

[26] M. Alghaili, Z. Li, A. Hamdi, and R. Ali, "Face filter: face identification with deep learning and filter algorithm," *Scientific Programming*, vol. 2020, Article ID 7846264, 9 pages, 2020.

[27] Y. Xu, W. Yan, G. Yang et al., "Joint face detection and alignment using face as point," *Scientific Programming*, vol. 2020, Article ID 7845384, 8 pages, 2020.

[28] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, pp. 2636–2640, Quebec, Canada, September 2015.

[29] S. H. Lee, H. Kim, and Y. M. Ro, "A comparative study of color texture features for face analysis," in *Computational Color Imaging. CCIW*, S. Tominaga, R. Schettini, and A. Trémeau, Eds., Berlin, Heidelberg, Springer.

[30] G. Kim, S. Eum, J. Suhr, D. Kim, K. Park, and J. Kim, "Face liveness detection based on texture and frequency analyses,"

in *Proceedings of 2012 5th IAPR International Conference on Biometrics, ICB*, pp. 67–72, New Delhi, India, April 2012.

[31] G. Sang, L. Jing, and Q. Zhao, "Pose-invariant face recognition via RGB-D images," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3563758, 9 pages, 2016.

[32] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition Conference A: Computer Vision & Image Processing,*, pp. 582–585, Jerusalem, Israel, October 1994.

[33] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: a survey in face recognition," *IEEE Access*, vol. 2, pp. 1530–1552, 2014.

[34] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[35] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *Proceedings of the 5th IAPR International Conference on Biometrics (ICB '12)*, pp. 26–31, IEEE, New Delhi, India, April 2012.

[36] https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html.

[37] https://github.com/Xilinx/Vitis-AI.

[38] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 2012.

[39] Md R. Hasan, "Face anti-spoofing using texture-based techniques and filtering methods," in *2019 3rd International Conference on Machine Vision and Information Technology (CMVIT2019)*, Guangzhou, China, February 2019.

[40] https://www.webopedia.com/definitions/false-acceptance.

[41] https://www.webopedia.com/definitions/false-rejection/.

[42] https://www.webopedia.com/definitions/equal-error-rate/.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5554058, 16 pages
https://doi.org/10.1155/2021/5554058

Research Article

Coverless Steganography Based on Motion Analysis of Video

Yun Tan, Jiaohua Qin, Xuyu Xiang, Chunhu Zhang, and Zhangdong Wang

College of Computer Science and Information Technology, Central South University of Forestry & Technology, 410004 Changsha, China

Correspondence should be addressed to Jiaohua Qin; qinjiaohua@163.com

Received 2 March 2021; Revised 25 March 2021; Accepted 30 March 2021; Published 22 April 2021

Academic Editor: Beijing Chen

With the rapid development of interactive multimedia services and camera sensor networks, the number of network videos is exploding, which has formed a natural carrier library for steganography. In this study, a coverless steganography scheme based on motion analysis of video is proposed. For every video in the database, the robust histograms of oriented optical flow (RHOOF) are obtained, and the index database is constructed. The hidden information bits are mapped to the hash sequences of RHOOF, and the corresponding indexes are sent by the sender. At the receiver, through calculating hash sequences of RHOOF from the cover video, the secret information can be extracted successfully. During the whole process, the cover video remains original without any modification and has a strong ability to resist steganalysis. The capacity is investigated and shows good improvement. The robustness performance is prominent against most attacks such as pepper and salt noise, speckle noise, MPEG-4 compression, and motion JPEG 2000 compression. Compared with the existing coverless information hiding schemes based on images, the proposed method not only obtains a good trade-off between hiding information capacity and robustness but also can achieve higher hiding success rate and lower transmission data load, which shows good practicability and feasibility.

1. Introduction

In recent years, the demand for information hiding continues to grow, especially for cloud computing environments. Traditional information hiding technologies usually embed secret information in the carrier [1–5] and lead to variable modification of carrier features. The steganography schemes that hide information by constructing the mapping relationship between cover features and secret information [6, 7] or using autogeneration technology [8–10] have aroused the interest of many researchers, which have a strong ability to resist steganalysis.

Most existing coverless steganography schemes are based on text and images. The text-based methods dug out the text features such as Chinese numeral expression [11], word rank map [12], or word frequency [13, 14] and quantified them. Then, the mapping relationship between text features and secret information was established, and the indexes were constructed. While in the image-based methods, the key problem is how to extract the main features of an image efficiently, which have been extensively studied in previous research studies [15–19]. In the method proposed by Zhou et al. [6], the secret information was converted to bits and divided into several data segments. The image with the same hash sequence as the data segment was selected and transmitted to the receiver as the cover image, from which the receiver could extract the secret information. Zheng et al. [20] used the direction information of scale-invariant feature transform (SIFT) points to design image hash and used the inverted index of quad-tree structure to improve the capacity and retrieval efficiency. An algorithm based on histograms of oriented gradients (HOG) was proposed by Zhou et al. [21], which obtained the hash sequences from the nonoverlapping blocks of the image. After block discrete cosine transformation (DCT) [22] or discrete wavelet transformation (DWT) [23], the relationship between coefficients of adjacent blocks was used to generate robust feature sequences. It can improve the capacity of hiding information by partitioning the image, but the robustness will be reduced by a larger partition number. Zou et al. [24] and Cao et al. [25] used average pixel values of subimages, which achieve a high hiding success rate and

capacity. In [26], LBP feature of the medical image was extracted and mapped to privacy information. Recently, Luo et al. [27] used recognized objects to hide secret information.

At the same time, live network platforms and video social applications are becoming more and more popular [28]. A large number of short videos have been generated and spread on the Internet, which provides sufficient carrier for information hiding. Compared with image, video not only has texture, shape, and color features but also has rich spatial and temporal features, from which some motion characteristics can be mined. Theoretically, motion characteristic is robust and cannot easily be tampered with, which is suitable for steganography. This motivates us to design a novel coverless steganography scheme based on short videos by constructing a mapping function between the motion characteristics and information bits.

Existing research results of coverless steganography based on videos are still rare. Some researchers have proposed some zero-watermarking technologies for copyright protection of video, which constructed watermark information by extracting video features. Li et al. [29] proposed a zero-watermarking algorithm based on logarithmic polar coordinate transformation. After 2D-DWT and 3D-DCT transformation of the original image, the zero-watermarking was realized by transforming the logarithmic polar coordinates. Liu et al. [30] proposed a zero-watermarking scheme for three-dimensional video, which extracted the features of two-dimensional video frames and depth maps to generate copyright information and used secret sharing schemes to achieve copyright protection. Compared with the zero-watermarking algorithms, coverless steganography based on videos has a higher requirement for capacity.

As shown in Figure 1, there is a baby walking in the video. The simplest mapping function is to connect the stepping of different feet with different information bits as the following equation:

$$f_{map} = \begin{cases} 0, & \text{action is "step left,"} \\ 1, & \text{action is "step right."} \end{cases} \quad (1)$$

Then, the continuous stepping of the baby can represent a sequence of information bits. However, this kind of mapping has some shortcomings: first, the stepping characteristic is semantic and can easily be understood and cracked. Second, the calculation complexity of stepping recognition is still relatively high, although the motion analysis and tracking of a video have achieved significant progress recently [31, 32]. Third, the capacity of information hiding is low since only one bit is hidden in every frame. Therefore, how to mine the nonsemantic motion characteristic and construct a correlated mapping function for information bits is the key issue of coverless steganography based on videos.

Video recognition has been studied in depth by a lot of researchers, and many algorithms have been proposed. Optical flow c is the most classical method for video analysis [33–37]. In this work, we mainly study the optical flow characteristics of video and map them with hidden information to realize coverless steganography. The main

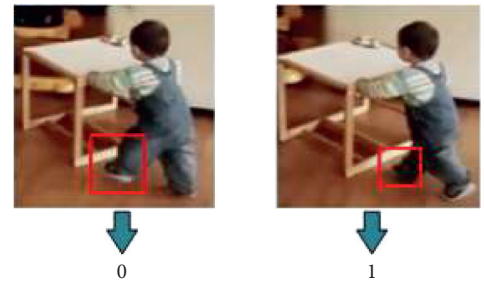


Figure 1: Mapping between motion characteristic and information bit.

contributions of this work are as follows: first, we construct a novel coverless steganography scheme based on motion analysis of video. Second, the mapping algorithm between the robust directional characteristics of video optical flow and secret information is proposed and optimized. Finally, information hiding capacity, robustness performance, efficiency, hiding success rate, and transmission data load of the proposed scheme are analysed and compared with the existing coverless steganography schemes based on images.

The study is organized as follows: preliminaries are introduced in the second section, and the proposed method is described in the third section. Experimental results and comparisons are shown in the fourth section. Finally, we conclude this study in the last section.

2. Preliminaries

2.1. Optical Flow. Optical flow is the instantaneous velocity distribution of the brightness pattern, which is caused by the movements of objects [33, 34] and has been applied widely for motion analysis. In recent studies, the optical flow was used to estimate the traffic flow parameters of moving vehicles in different scenarios and shew effectiveness [35, 36]. Lv et al. realized subpixel image registration based on the optical flow model and feature-point matching [37].

The basic idea of the optical flow method is to find the corresponding relationship between adjacent frames in the image sequence using the change of pixels in time domain. Then, the motion information of objects can be calculated.

Assuming there is a pixel (x, y, t) in one frame, its light intensity is expressed as $I(x, y, t)$. It moves to (x', y', t') in the next frame. According to the assumption of constant brightness, we can get the following equation:

$$I(x, y, t) = I(x', y', t') = I(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t). \quad (2)$$

The Taylor series approximation is applied to equation (2); then, we can get

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x}\mathrm{d}x + \frac{\partial I}{\partial y}\mathrm{d}y + \frac{\partial I}{\partial t}\mathrm{d}t + \varepsilon, \quad (3)$$

where ε is the second-order infinitesimal term and can be neglected. Therefore, equation (3) can be transformed to

$$\frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial I}{\partial t} = 0. \quad (4)$$

Assuming u and v are the velocity vectors of optical flow along the X axis and Y axis, respectively, we have

$$u = \frac{\mathrm{d}x}{\mathrm{d}t}, \quad v = \frac{\mathrm{d}y}{\mathrm{d}t}. \tag{5}$$

Equation (4) can be transformed as

$$I_x u + I_y v + I_t = 0, \tag{6}$$

where $I_x = \partial I/\partial x$, $I_y = \partial I/\partial y$, and $I_t = \partial I/\partial t$. This is the basic constraint equation of optical flow.

In order to solve the above equations and achieve the value of the unknown u and v, there are two classical methods. One is a global differential method, which assumes that the optical flow changes smoothly over the entire image. The other is a local differential method, which assumes that the motion vector remains constant over a small spatial domain. Therefore, it is suitable for small motion detection, but fails for large motion detection. In order to improve this defect, the pyramidal implementation was proposed by Bouguet [38].

The pyramid layering method was used to reduce the size of the image layer by layer, thereby reducing the motion displacement of the object between two frames. The process is shown in Figure 2, and the specific steps are as follows:

Step1: A pyramid is created for every frame, and the resolution is sequentially lowered from the bottom to the top.

Step2: Starting at the top level, the optical flow at every point in the top-level image is obtained by minimizing the minimum matching error sum within the neighbourhood of each point. Assuming d is the optical flow, the residual function is defined as [38]

$$\varepsilon(d,) = \varepsilon\left(d_x, d_y\right) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \cdot \left(I(x,y) - J\left(x + d_x, y + d_y\right)\right)^2, \tag{7}$$

where (u_x, u_y) is the point of the original image I and $(u_x + d_x, u_y + d_y)$ is the point of the target image J.

Supposing there are L layers of the pyramid, the first layer is the original image. If the total displacement is d, then the displacement for each layer is

$$d^L = \frac{d}{2^L}. \tag{8}$$

Step 3: The optical flow of the layer L is propagated to the layer $L-1$ as follows:

$$g^{L-1} = 2\left(g^L + d^L\right). \tag{9}$$

For the layer l, the calculation of optical flow is based on the minimization of the sum of matching error for all points in the neighbourhood area, as the following equation:

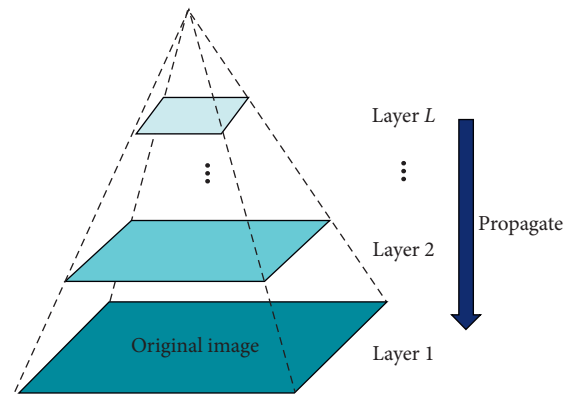


Figure 2: Pyramid layering process.

$$\varepsilon^l\left(d^l\right) = \varepsilon^l\left(d_x^l, d_y^l\right) = \sum_{x=u_x^l-w_x}^{u_x^l+w_x} \sum_{y=u_y^l-w_y}^{u_y^l+w_y} \cdot \left(I^l(x,y) - J^l\left(x + g_x^l + d_x^l, y + g_y^l + d_y^l\right)\right)^2. \tag{10}$$

It is propagated down the pyramid until it reaches the bottom layer. Then, the optical flow is calculated by

$$d = g^1 + d^1. \tag{11}$$

2.2. Robust Histogram of Oriented Optical Flow. Since the size of the moving target usually changes with time in a video, the dimension of the corresponding optical flow descriptor will also change. At the same time, the original optical flow is also sensitive to the background noise, scale change, and the direction of motion. For information hiding, the extracted features are expected to be more stable, which can gain better robustness. Therefore, it is necessary to find a method based on the optical flow that can not only characterize the temporal motion information but also be insensitive to scale. Histogram of oriented optical flow (HOOF) was proposed by Chaudhry et al. [39]. The scale invariance of HOOF feature was achieved by the normalized histogram. In order to further enhance the robustness, the robust histogram of oriented optical flow (RHOOF) is achieved by only counting the number of optical flows located in the directional bins, while the amplitude information is ignored. This means that RHOOF will not be affected by the amplitude variation of optical flow, which is different from the original HOOF.

For every two frames, the optical flow is calculated. And then, the directional angle of the optical flow vector can be achieved by

$$\theta = \operatorname{atan} 2\left(\frac{y}{x}\right), \tag{12}$$

where atan2(·) is a four-quadrant inverse tangent function, x is the horizontal component, and y is the vertical component of optical flow vector. The range of θ is $[-\pi, \pi]$. If the angle range is divided into several groups, then the histogram distribution is statistically obtained. As shown in Figure 3,

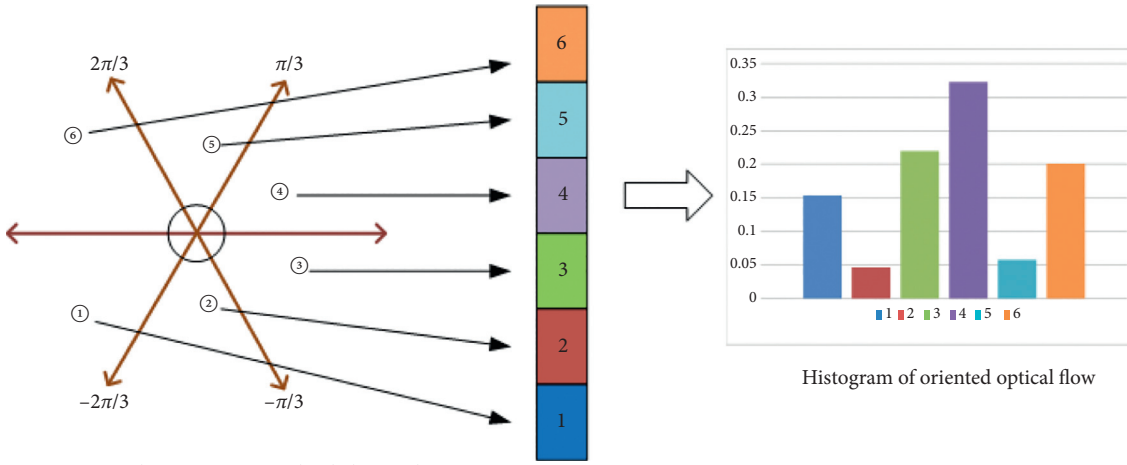


FIGURE 3: The principle of the histogram of oriented optical flow.

the bin number of the histogram is 6 and the bin size is $\pi/3$, which means that the angle range of optical flow is divided to 6 groups. The distribution of the angles is shown in the histogram on the right. The sum of the possibilities of all groups is 1.

3. Proposed Coverless Steganography Scheme

Our proposed coverless steganography scheme based on videos is shown in Figure 4. The framework mainly includes three parts: index construction, secret information hiding, and secret information extraction. First, the video database is composed of multiple videos with different topics. The video database is shared by both secret information sender and receiver, which can be stored on cloud platform to save the storage space of the end user. Second, calculate RHOOF for every video in the video database. After the hash sequences of RHOOF are calculated, the video index database is constructed. The construction of video database and index database is the basis of coverless information hiding.

During the information hiding process, the secret information needs to be preprocessed and divided to binary bit groups. Every bit group can be mapped to a hash sequence of RHOOF. After searching in the video index database, the appropriate one or several videos are selected as cover, and the corresponding mapping indexes are returned to the sender. The mapping indexes will be sent to the receiver. At the receiver, cover video can be found accurately and efficiently according to the received mapping indexes. Through calculating the hash sequences of RHOOF from the cover video, the secret information can be recovered successfully. During the whole process, the cover video remains original without any modification. Therefore, it can resist the detection of steganalysis.

3.1. Generation of Hash Sequence. As described previously, RHOOF can reflect the main movement characteristic of the video. We propose a hash sequence generation method based on RHOOF as shown in Figure 5. For the two adjacent frames (assuming as frame i and frame $i+1$) of a video, they are transformed to gray scale first. Second, the two frames are median filtered in order to suppress the possible noise and protect the edge information. Third, the pixel changes of these two frames are calculated, and the oriented optical flows are achieved as described in the previous section. We can analyse the orientation values of optical flow. The histogram is calculated in several bins for every subblock. In Figure 5, we set the number of subblocks to 4 and the number of bins to 8 as an example.

Assuming the histogram is denoted as

$$H = \{h_1, h_2, \ldots, h_N\}, \tag{13}$$

where N is the number of the bins. We set the threshold as

$$\text{thres} = \sum_{i=1}^{N} \frac{h_i}{N} + \text{th}_0, \tag{14}$$

where th_0 is a correction factor. Then, the hash sequence $b_1 b_2 \ldots b_N$ is achieved by comparing the histogram value with the threshold as the following equation:

$$b_i = \begin{cases} 1, & \text{if } h_i \geq \text{thres}, \\ 0, & \text{if } h_i < \text{thres}, \\ 1 \leq i \leq N. \end{cases} \tag{15}$$

3.2. Construction of Video Index Database. In order to find the cover video efficiently and accurately, the construction of video index database is necessary and important. Therefore, we construct an efficient index database with two levels as shown in Figure 6. The first level index is the hash sequence and the second level index contains the information items of cover video and cover frame.

The index items are sorted by the hash sequence. Here, the bin number is also set to 8 as an example. Therefore, the value of hash sequence varies from "00000000" to "11111111" in the index table. For every index item corresponding to the hash sequence, index ID, video ID, frame

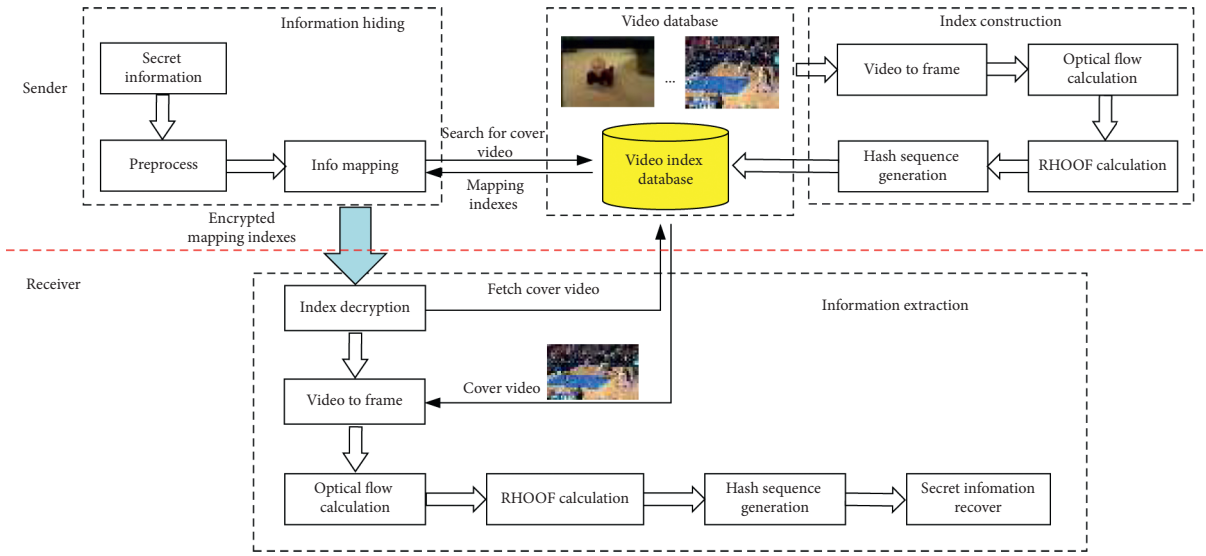


Figure 4: The framework of the proposed coverless steganography scheme.

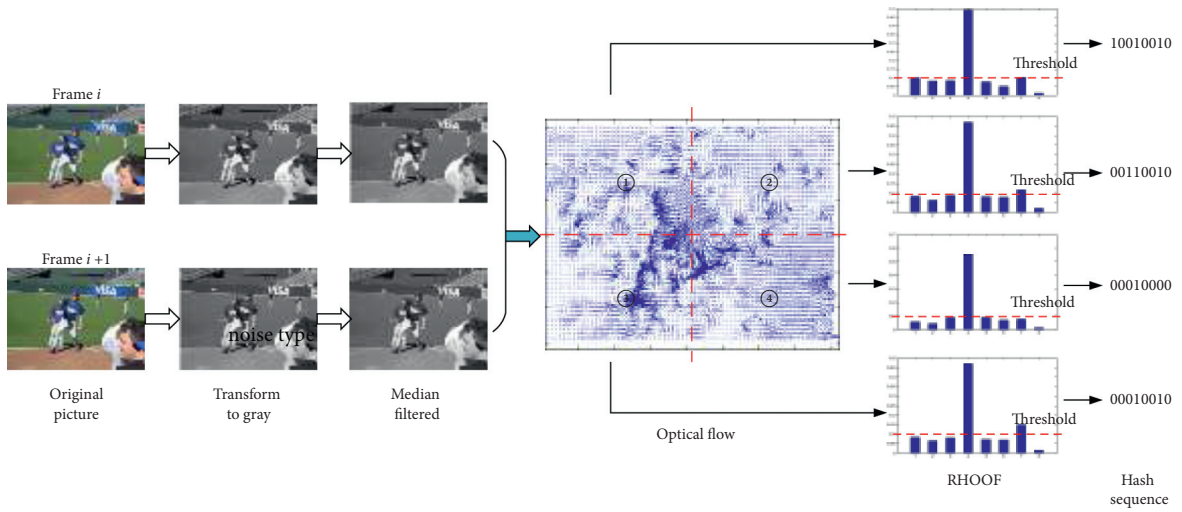


Figure 5: The process of hash sequence generation.

Hash sequence	Index ID	Video ID	Frame ID	Subblock ID
00000000	1	\Video\walking.avi	120	2
00000001	2	\Video\running.avi	10	3
	...	...	...	...
	8	\Video\basketball.avi	145	1
00000010	9	\Video\walking.avi	23	4
	10	\Video\walking.avi	167	2
	...	...	...	...
⋮	13	\Video\crwaling.avi	40	3
11111111	902	\Video\walking.avi	220	2
	903	\Video\boxing.avi	30	1
	...	...	...	...
	1000	\Video\crwaling.avi	78	4

Figure 6: Structure of video index database.

ID, and subblock ID are contained in the index database. Index ID is the serial number of an index, which is incremental. Video ID means the storage path and the name of the cover video. Frame ID means the corresponding frame, and subblock ID means the corresponding subblock. With such information, the cover video can be found accurately, and the hash sequence can be calculated efficiently and conveniently. For one hash sequence, it is possible to contain multiple index items, which means there are multiple cover subblocks with the same hash sequence in the video database. In this situation, any index item can be chosen in principle for subsequent secret information mapping.

3.3. Secret Information Hiding. During secret information hiding, how to map the secret information to the cover video efficiently is the most critical part. The whole process is summarized as follows:

Step 1: construct a video database, which is shared by both the sender and the receiver

Step 2: for each video in the library, the frame optical flows are obtained as described in the previous section

Step 3: the directional angle of the optical flow vector is calculated by equation (12)

Step 4: for every frame, the robust histogram distribution is statistically counted in subblocks based on the oriental information of optical flow. The hash sequences are obtained as described previously.

Step 5: construct the video index database as described previously

Step 6: the secret information needs to be preprocessed before sending. Assuming the length of the secret information is k bits, it will be divided into m segments as

$$m = \lceil \frac{k}{N} \rceil, \tag{16}$$

where N is the bin number of RHOOF statistics. For the last segment, "0" bits are padded to the tail, and the number of the padding bits is

$$p = \lceil \frac{k}{N} \rceil \cdot N - k. \tag{17}$$

Step 7: for every segment, we search the corresponding index item in the index database, which has the hash sequence equal to the information bits. It is possible that there are multiple index items mapping to the same hash sequence. In order to increase the efficiency of information extraction, we should choose the index items with the same video file as much as possible. For the same video file, the mapping index item with smaller index ID will be chosen.

Step 8: the information of mapping indexes corresponding to the secret information segments will be sent to the receiver. In order to enhance security, the index information can be encrypted before transmission.

The detailed algorithm of index database construction is described in Algorithm 1.

The information hiding algorithm at the sender is described in Algorithm 2.

3.4. Extraction of Secret Information. At the receiver, by calculating the hash sequence of RHOOF based on the cover video, the secret information can be extracted successfully. The process of secret information extraction is as follows:

Step 1: after receiving the index information sent by the transmitter, the receiver will decrypt it first if necessary. Then, the index items will be analysed, and video ID, frame ID, and subblock ID can be obtained.

Step 2: the corresponding frame can be found according to video ID and frame ID. The optical flow of the corresponding frame is calculated as described previously.

Step 3: the directional angle of the optical flow vector is calculated by equation (12)

Step 4: the robust histogram distribution is statistically counted in subblocks according to the oriented information of the optical flow. The hash sequence is obtained.

Step 5: repeat steps 1–4 until all the hash sequences corresponding to the mapping index items have been extracted

Step 6: after connecting the hash sequences and removing the padding bits from the tail, the bitstream of secret information is recovered successfully

The detailed algorithm of secret information extraction is described in Algorithm 3.

As shown in Figure 7, there is an example of transmission and extraction of the secret bitstream as "11111111000000." The bin number N is 8. As mentioned before, the length of padded bitstream should be an integral multiple of N. Therefore, the bitstream is padded with one "0" bit at the tail first, which makes the length of bitstream as 16. Then, the padded bitstream is segmented to 2 groups of 8 bits. Next, the hash sequences equal to the segmented bitstream "11111111" and "00000010" are searched in the video index database. From Figure 6, it can be seen that there are multiple index items corresponding to the hash sequences "11111111" and "00000010." The index items with index ID as 902 and 9 are selected since they have the same video ID, which are marked with green as shown in Figure 6. Therefore, the video "walking.avi" is our cover video. At the receiver, RHOOF are calculated based on frame 220 and 221 and frame 23 and 24 of "walking.avi." Then, the hash sequences "11111111" and "00000010" are achieved from the subblock 2 and the subblock 4 separately. After removing the padding bit "0" from the tail, the secret bitstream "111111110000001" can be recovered successfully.

3.5. Algorithm Improvement. One prerequisite of the optical flow method is that the brightness should remain constant.

Input: video database $V = \{V_1, V_2, \ldots, V_d\}$, number of videos d, smoothing window size for optical flow calculation W, number of LK pyramid level L, and bin number N.
Output: video index database $I = \{\text{Ind}_1, \ldots, \text{Ind}_{2^N}\}$
(1) For $i = 1$: d
(2) Decompose video to pictures: $P = \text{VideoToFrame}\ (V_i)$
(3) For $j = 1$: FrameNum-1
(4) Convert RGB to gray: $R_j = \text{Rgb2gray}\ (P_j)$, $R_{j+1} = \text{Rgb2gray}\ (P_{j+1})$
(5) Median filtering: $M_j = \text{medfilt}\ (R_j)$, $M_{j+1} = \text{medfilt}\ (R_{j+1})$,
(6) Calculate the hierarchical optical flow matrixes between frame j and $j+1$: $(u, v) = \text{HierarchicalLK}\ (M_j, M_{j+1}, W, L)$
(7) Calculate RHOOF and hash sequences in subblocks
For $s = 1$: subblock_num
(8) $H_s = \text{RHOOF}\ (u, v, N)$
(9) $\text{Hash}_s = \text{HashCalc}\ (H_s)$
(10) Link MySQL and update index database: index item-> {index ID, video ID, frame ID, subblock ID}
(11) End for
(12) End for
(13) End for

Algorithm 1: Index database construction.

Input: video index database $I = \{\text{Ind}_1, \ldots, \text{Ind}_{2^N}\}$, secret information bitstream $B = \{b_1, b_2, \ldots, b_k\}$
Output: mapping index set $I_m = \{\text{Ind}_1, \ldots, \text{Ind}_m\}$, number of index items m
(1) Padding the secret information bitstream: $B' = \text{pad}(B)$
(2) Divide B' to m segments
(3) For $i = 1 : m$
(4) Search the corresponding index item and update the index set as $I_m = \{I_m, \text{Ind}_i\}$
(5) End for
(6) Send the mapping index set I_m to the receiver

Algorithm 2: Information hiding.

Input: video database $V = \{V_1, V_2, \ldots, V_d\}$, smoothing window size for optical flow calculation W, number of LK pyramid level L, bin number N, mapped index set $I_m = \{\text{Ind}_1, \ldots, \text{Ind}_m\}$, and number of index items m
Output: secret information bitstream $B = \{b_1, b_2, \ldots, b_k\}$
(1) For $i = 1 : m$
(2) Get video ID, frame ID, and subblock ID from the index item i
(3) $P = \text{VideoToFrame}\ (V_{\text{video ID}})$
(4) Convert RGB to gray: $R_1 = \text{Rgb2gray}\ (P_{\text{Frame\_ID}})$, $R_2 = \text{Rgb2gray}\ (P_{\text{Frame\_ID}+1})$,
(5) Median filtering: $M_1 = \text{medfilt}\ (R_1)$, $M_2 = \text{medfilt}\ (R_2)$
(6) Calculate the hierarchical optical flow matrixes between corresponding frames: $(u, v) = \text{HierarchicalLK}\ (M_1, M_2, W, L)$
(7) Calculate the gradient histogram of optical flow of the corresponding subblock: $\text{H}_{\text{Subblock\_ID}} = \text{gradientHist}\ (u, v, N)$
(8) $\text{Hs}_i = \text{HashCalc}\ (H_{\text{Subblock\_ID}})$
(10) End for
(11) Connect all the segments as $\{\text{Hs}_1, \text{Hs}_2, \ldots, \text{Hs}_m\}$
(12) Remove padding bits, and the secret information bitstream is recovered: $B = \{b_1, b_2, \ldots, b_k\}$

Algorithm 3: Information extraction.

In the case of noise or random interference, the optical flow value will be greatly affected, which will lead to wrong extraction of the hidden information. Therefore, the algorithm is further optimized with an averaging window applied. Before the optical flow is calculated, we update the data of every frame by averaging the pixel values of adjacent frames. Through this smoothing operation, the influence of noise and random interference can be reduced. The improved algorithm of index database construction is described in Algorithm 4.

The improved information extraction algorithm is described in Algorithm 5.

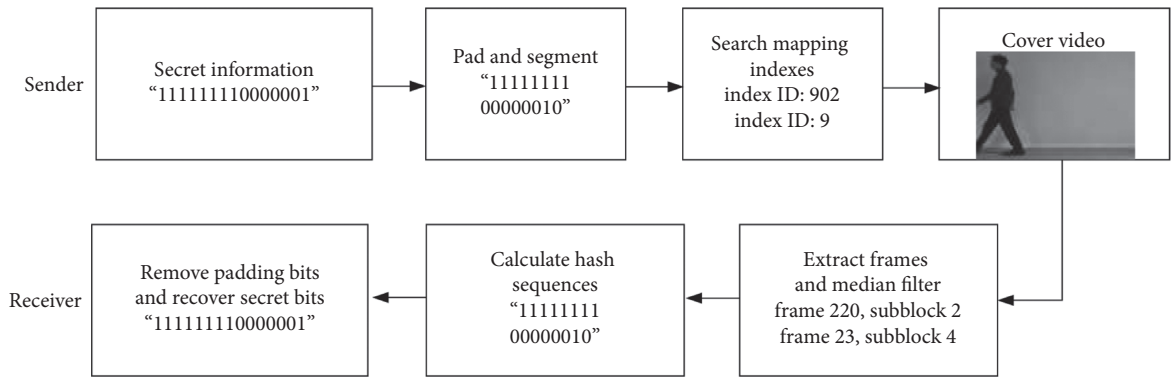


Figure 7: Example of the secret information transmission and extraction.

Input: video database $V = \{V_1, V_2, \ldots, V_d\}$, number of the videos d, frame averaging window length avg_L, smoothing window size for the optical flow calculation W, number of the LK pyramid level L, and bin number N
Output: video index database $I = \{\text{Ind}_1, ..., \text{Ind}_{2^N}\}$
(1) For $i = 1 : d$
(2) Decompose video to pictures: $P = \text{VideoToFrame}\ (V_i)$
(3) For $j = 1$: FrameNum-avg_$L + 1$
(4) Initialize R_sum as all zero
(5) For $k = 0$: avg_$L - 1$
(6) Convert RGB to gray: orig_$R_{j+k} = \text{Rgb2gray}\ (P_{j+k})$
(7) R_sum = R_sum + orig_R_{j+k}
(8) End for
(9) $R_j = R$_sum/avg_L
(10) End for
(11) For $j = 1$: FrameNum − avg_L
(12) Repeat step 1–10 of Algorithm 1.
(13) End for
(14) End for

Algorithm 4: Improved index database construction.

Input: video database $V = \{V_1, V_2, \ldots, V_d\}$, frame averaging window length avg_L, smoothing window size for optical flow calculation W, number of LK pyramid level L, bin number N, mapped index set $I_m = \{\text{Ind}_1, \ldots, \text{Ind}_m\}$, and number of index items m
Output: secret information bitstream $B = \{b_1, b_2, \ldots, b_k\}$
(1) For $i = 1 : m$
(2) Get video ID, frame ID, and subblock ID from the index item i
(3) $P = \text{VideoToFrame}\ (V_{\_\text{video ID}})$
(4) For $j = 1$: FrameNum-avg_$L + 1$
(5) Initialize R_sum as all zero
(6) For $k = 0$: avg_$L - 1$
(7) Convert RGB to gray: orig_$R_{j+k} = \text{Rgb2gray}\ (P_{j+k})$
(8) R_sum = R_sum + orig_R_{j+k}
(9) End for
(10) $R_j = R$_sum/avg_L
(11) End for
(12) Median filtering: $M_1 = \text{medfilt}\ (R_{\text{Frame ID}})$, $M_2 = \text{medfilt}\ (R_{\text{Frame ID}+1})$
(13) Repeat step 6–8 of Algorithm 3.
(14) End for
(15) Connect all the segments as: {Hs_1, Hs_2, . . ., Hs_m}
(16) Remove padding bits and the secret information bitstream is recovered: $B = \{b_1, b_2, \ldots, b_k\}$

Algorithm 5: Improved information extraction.

4. Experimental Results and Analysis

The experiments are conducted with the Intel(R) Core (TM) i7-6500X CPU @ 2.50 GHz and 16.00 GB RAM. Matlab 2018b is used for algorithm simulation, and MySQL workbench 6.3 is used for the index database construction.

A video database is used for the test, which is composed by videos with different movements and scenarios that are randomly chosen from UCF101 and HMDB51 datasets as shown in Figure 8. The file size of the videos is about 200~800 KB, and the duration time is about 2~10 seconds.

4.1. Capacity Analysis. The bit number of the generated hash sequence based on the cover video determines the capacity of information hiding. Assuming the frame number of the video is F, the number of optical flow images should be $F-1$. Every optical flow image will be divided into S subblock, and the oriented histogram are statistically counted in N bins. Therefore, for every optical flow image, the number of mapped bits is:

$$C = S \cdot N. \tag{18}$$

Then for every video, the number of mapped bits is:

$$C' = S \cdot N \cdot (F-1). \tag{19}$$

It can be seen that the capacity of information hiding in our scheme is related to the bin number N, subblock number S, and frame number F. For a specific video, the frame number is fixed, and then, the capacity is determined by the number of bins and subblocks. The larger the N and S are, the larger the capacity is. We compare the capacity of single optical flow image in our scheme with existing coverless information hiding schemes based on single image in Table 1. Here, in the proposed method, the subblock number is set as 4 and the bin number is set as 8.

For a secret message to be hidden, if the capacity of single image is larger, a smaller number of cover images will be needed. Assuming the length of the secret information is K bits, the capacity of single image is C bits; then, the number of required cover images is

$$M_{\text{image}} = \lceil \frac{K}{C} \rceil. \tag{20}$$

With the same hidden information, the number of images needed for different methods is compared in Table 2. It can be seen that our proposed method (set $S=4$, $N=8$) has a larger capacity than other methods. With the increment of S and N, the capacity will be even enlarged more. However, if N is increased, the size of video database needs also to be enlarged in order to ensure the success rate of information mapping. And the robustness will also be affected by the variation of S and N, which will be further investigated in the next tests.

4.2. Robustness Analysis. We investigate the robustness against pepper and salt noise, Gauss noise, and speckle noise with different parameters for performance evaluation. For a video, the compression transformation is used commonly. Therefore, the effect of compressed MPEG-4 transformation (.mp4 file) and compressed motion JPEG 2000 file transformation (.mj2 file) are also investigated. Assuming the original bitstream is $b_1 b_2 \ldots b_m$ and the extracted bit sequence is $b_1' b_2' \ldots b_m'$, the accuracy rate is calculated by

$$ACC = \frac{\sum_{i=1}^{m} f(i)}{m}, \tag{21}$$

where

$$f(i) = \begin{cases} 1, & if\ b_i = b_i', \\ 0, & if\ b_i \neq b_i'. \end{cases} \tag{22}$$

The results of accuracy rate against different types of attacks with different bin number N are shown in Table 3, in which the subblock number S is fixed as 4. It can be seen that the robustness of the proposed scheme is good, especially against salt and pepper noise and MPEG-4 compression. At the same time, the increment of bin number will lead to the decrease of accuracy. It is because the smaller bin number will cause bigger bin size, which is less sensitive to the variation of angle distribution. But according to equation (19), the capacity will be decreased with the lower bin number.

The accuracy results with different types of attacks and different subblock number S are also investigated as given in Table 4. Here, the bin number N is fixed as 8. It can be seen that the effect of subblock number S is relatively small and the variation trend is irregular. The reason is that the spatial distribution of the optical flow is different for various types of videos. Therefore, according to equation (19), we can increase the capacity of information hiding by increasing the subblock number if necessary.

4.3. Improvement Analysis of Frame Averaging. We investigate the performance improvement of introducing the frame averaging window before calculating optical flow. Here, the length of frame averaging window is set as 10, which means that the data of every frame are updated by averaging the pixel values of adjacent 10 frames. The comparison of accuracy rate with different types of noises and compression is shown in Figures 9–12, where the subblock number is set as 4 and the bin number is set as 4, 8, 12, and 16, respectively.

It can be seen that frame averaging operation can improve the robustness significantly. Figure 9 is the accuracy comparison with compression attacks, which shows that the improvement for mj2 compression is much larger than for mp4 compression. For mj2 compression, the accuracy rate is increased by more than 7% with any bin number, while there is only weak improvement for mp4 compression. Figures 10 and 11 show the accuracy comparison with Gaussian noise and speckle noise, respectively, in which both have significant improvement. The increment of accuracy rate is even bigger when the bin number is increased. However, for salt and pepper noise, the accuracy rate will be decreased with frame averaging operation, as shown in Figure 12. And the

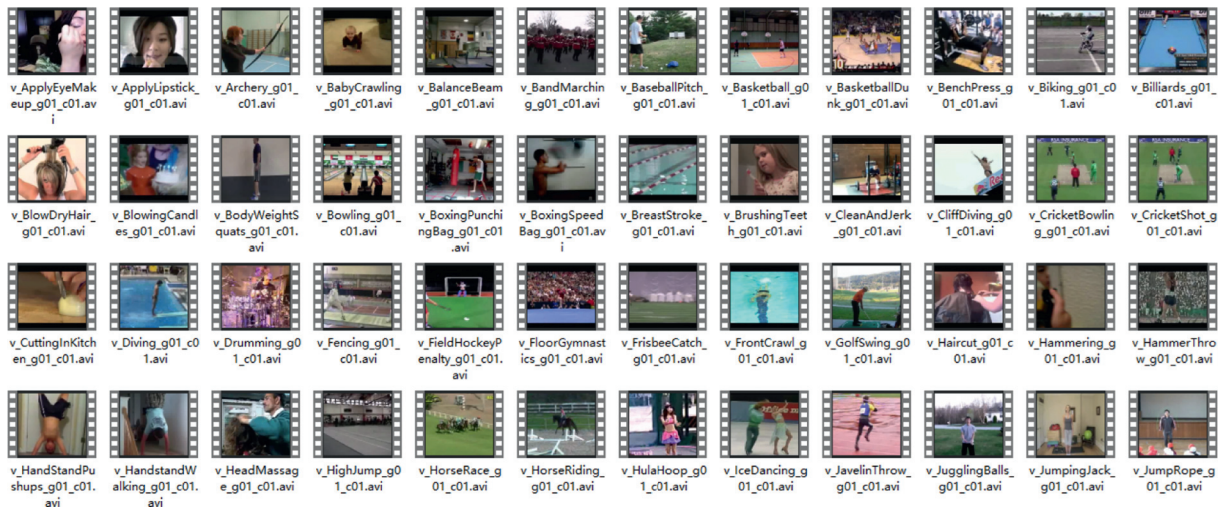


Figure 8: Video database.

Table 1: Capacity comparison.

Method	Pixel method [6]	Hash method [20]	DCT method [22]	DWT method [23]	Proposed method
Capacity	8 bit	18 bit	1~15 bit	1~15 bit	32 bit

Table 2: Number of needed images with the same hidden information.

Size of hidden information	1 B	10 B	100 B	1 kB
Pixel method [6]	1	10	100	1024
Hash method [20]	2	6	46	457
DCT method [22]	2~9	7~81	55~801	548~8193
DWT method [23]	2~9	7~81	55~801	548~8193
Proposed method	1	3	25	256

Table 3: Accuracy with different attacks and bin number N.

Attack	Accuracy			
	$N = 4$	$N = 8$	$N = 12$	$N = 16$
Salt and pepper (σ = 0.001)	0.9991	0.9986	0.9972	0.9950
Salt and pepper (σ = 0.005)	0.9964	0.9923	0.9890	0.9834
Salt and pepper (σ = 0.01)	0.9937	0.9877	0.9785	0.9686
Gauss (σ = 0.001)	0.8337	0.7005	0.6141	0.5133
Gauss (σ = 0.005)	0.8218	0.6485	0.5521	0.4438
Gauss (σ = 0.01)	0.8156	0.6198	0.5211	0.4104
Speckle (σ = 0.001)	0.8835	0.8235	0.7610	0.7037
Speckle (σ = 0.005)	0.8738	0.8098	0.7377	0.6743
Speckle (σ = 0.01)	0.8677	0.8000	0.7278	0.6596
Compressed MPEG-4 file with H.264 (.mp4 file)	0.9806	0.9589	0.9353	0.9126
Compressed motion JPEG 2000 file (.mj2 file)	0.8962	0.8476	0.8028	0.7599

larger the bin number is, the more obvious the impact is. This is because that salt and pepper noise is approximately equal in the amplitude but randomly distributed in different locations. Therefore, frame averaging calculation may possibly cause some clean pixels to be contaminated conversely.

4.4. Robustness Comparison with Different Methods. We use the methods based on images for performance comparison after transferring the videos to frame images. The latest DWT method [22], DCT method [23], and Hash method [20] are considered and tested based on our video database. During tests, the subblock number of the DWT method and the DCT method is set as 8. The subblock number of our method is 4, and the bin number is also set as 4. The accuracy with Gaussian noise, salt and pepper noise, and video compression transformations in the different methods are shown in Figures 13–15. It can be seen that the proposed

Table 4: Accuracy with different attacks and subblock number.

Attack	Accuracy			
	$S = 1$	$S = 2$	$S = 4$	$S = 8$
Salt and pepper ($\sigma = 0.001$)	0.9980	0.9982	0.9986	0.9975
Salt and pepper ($\sigma = 0.005$)	0.9952	0.9952	0.9923	0.9930
Salt and pepper ($\sigma = 0.01$)	0.9896	0.9884	0.9877	0.9867
Gauss ($\sigma = 0.001$)	0.6583	0.6821	0.7005	0.7132
Gauss ($\sigma = 0.005$)	0.5994	0.6310	0.6485	*0.6688*
Gauss ($\sigma = 0.01$)	0.5695	0.6015	0.6198	0.6425
Speckle ($\sigma = 0.001$)	0.8176	0.8059	0.8235	0.8268
Speckle ($\sigma = 0.005$)	0.8074	0.7899	0.8098	0.8149
Speckle ($\sigma = 0.01$)	0.7934	0.7825	0.8000	0.8059
Compressed MPEG-4 file with H.264 (.mp4 file)	*0.9665*	0.9635	0.9589	0.9526
Compressed motion JPEG 2000 file (.mj2 file)	0.8240	0.8331	0.8476	0.8542

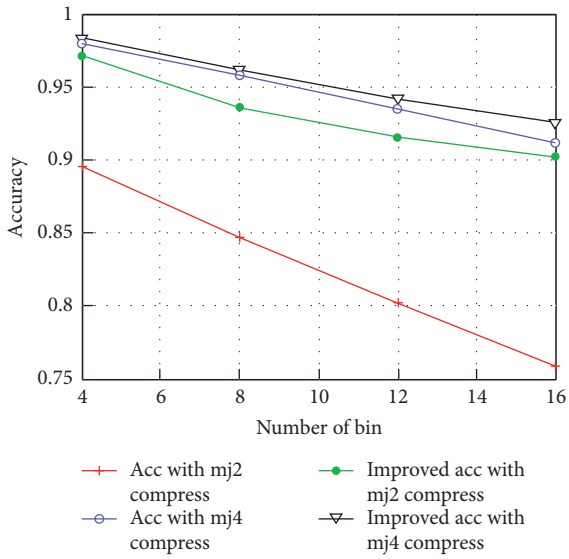


Figure 9: Accuracy improvement with compression.

method has good performance with salt and pepper noise and video compression, while it is more sensitive to Gaussian noise compared with other three methods. Although the overall accuracy performance of the proposed method is slightly worse than the DWT method and the DCT method, the accuracy rate still can arrive at 0.97 for most cases. However, the hiding information capacity of our method is 16 bits for every frame, which is twice that of the DWT method and the DCT method during the tests. Therefore, our proposed method has obtained a good trade-off between steganographic capacity and robustness.

4.5. Hiding Success Rate Analysis. For coverless information hiding based on feature mapping, the extracted feature sequences should not only ensure the robustness but also reflect the differences of features. Therefore, under the premise of a given database, the success rate of information hiding is also an important indicator to measure the feasibility and the practicability of the secret transmission scheme. Assuming that every hidden information segment contains n bits, the number of different mapping sequences that can be generated by the current video library is k, and the hiding success rate is

$$R_{\text{suc}} = \frac{k}{2^N}. \tag{23}$$

In the test, the videos are chosen randomly from UCF101 and HMDB51 datasets. The success rate of our scheme is compared with the DWT method and the DCT method. The subblock number of all the three algorithms is set as 8. The results are shown in Figure 16. It can be seen that the hiding success rate increases with the increment of the number of the videos. With the same number of videos, the hiding success rate of our method is much higher than other two methods. With only about 70 videos, our method can achieve a hiding success rate as more than 90%. The improvement of hiding success rate comes from the consideration of the optical flow features between adjacent frames in our method. However, the DWT method and the DCT method based on images only focus on every separate frame and the adjacent frames usually have similar texture features. Therefore, the generated bit sequences also have high similarity, which will reduce the hiding success rate.

4.6. Complexity and Efficiency Analysis. The complexity of the proposed method mainly lies in the construction of the video index database because the RHOOF of every video frame needs to be calculated. However, the video index database only needs to be constructed once in advance at the sender. During real secret transmission, we only need to consider the time cost of the specific secret information hiding and extraction, in which the main work load lies in the feature analysis of the cover frames. We investigate the efficiency of different methods based on the time cost of hiding information bits with the same length. In Table 5, the time cost of different methods is listed, where "s/B" means the number of seconds that are required for hiding one information byte. It can be seen that the time cost of our method is more than other methods due to the complexity of hierarchical optical flow calculation.

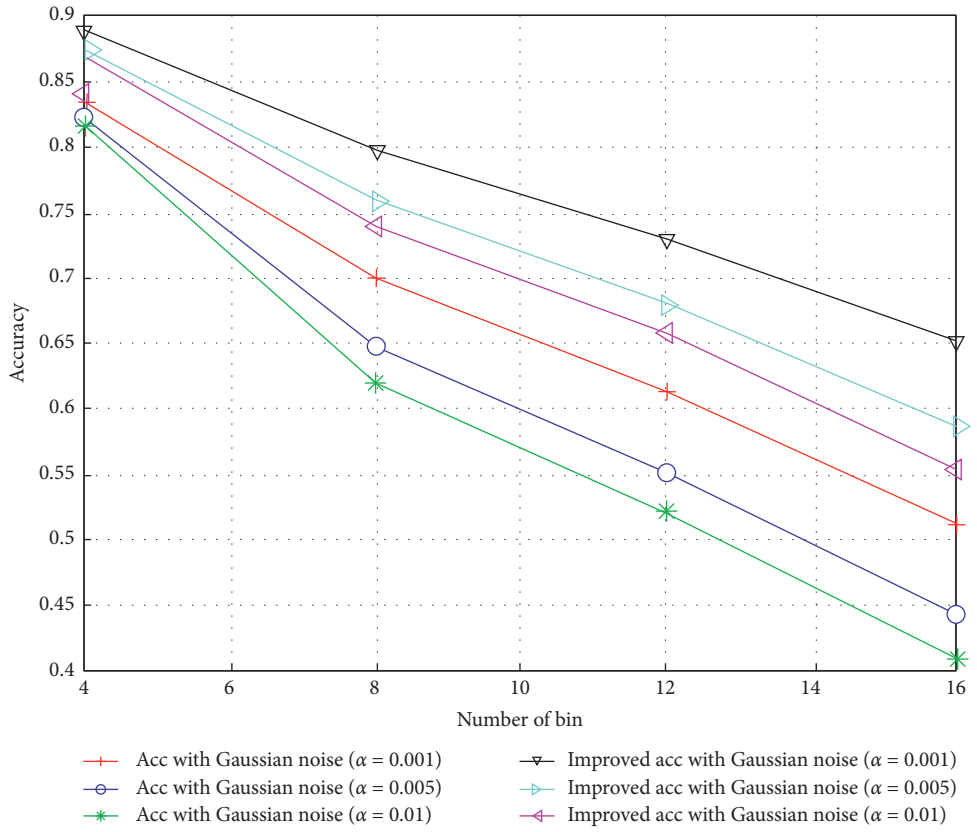


Figure 10: Accuracy improvement with Gaussian noise.

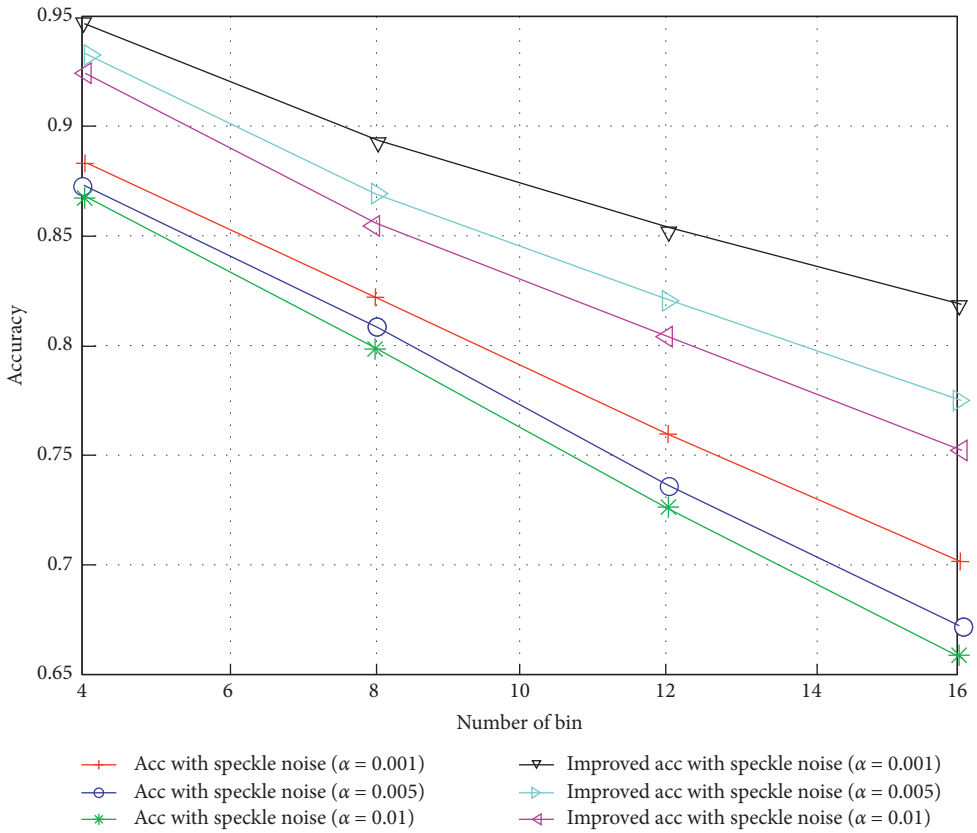


Figure 11: Accuracy improvement with speckle noise.

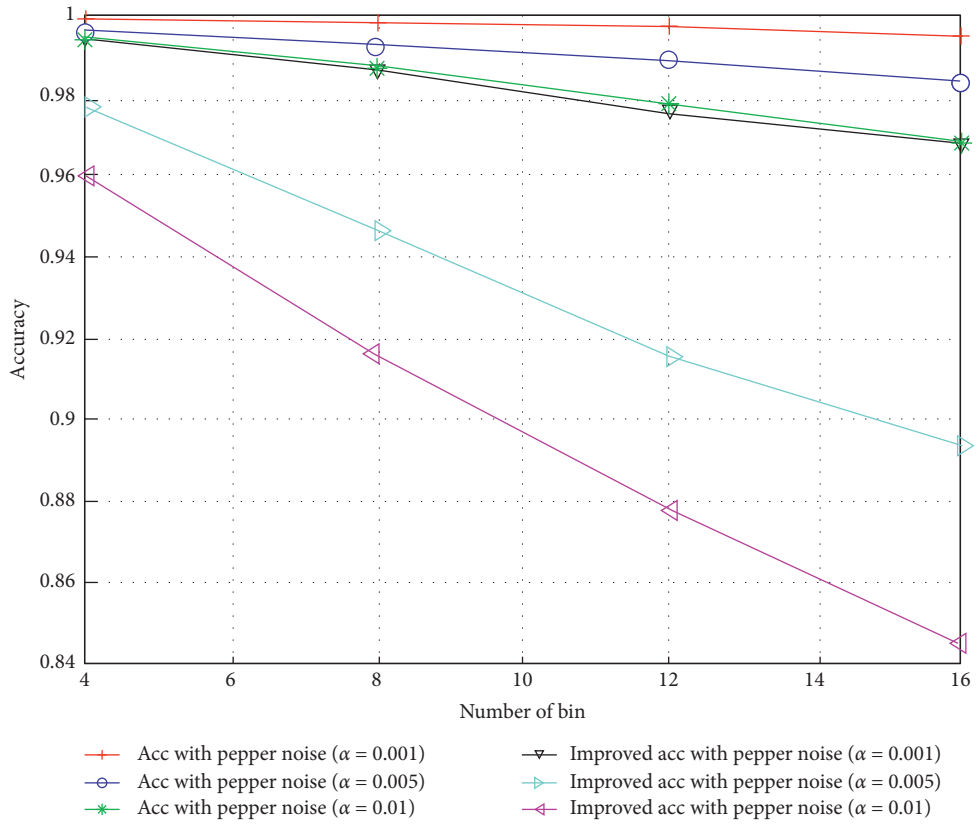


Figure 12: Accuracy improvement with salt and pepper noise.

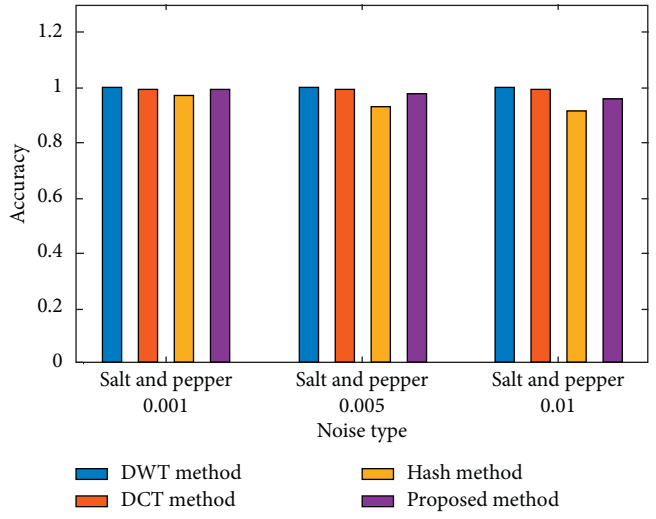


Figure 13: Accuracy comparison with salt and pepper noise.

4.7. Transmission Data Load Analysis. In the proposed steganography scheme, the video database is shared by both the sender and the receiver. During the information hiding process, the secret information will be preprocessed and mapped to the hash sequences of RHOOF. After searching in the video index database, the corresponding mapping indexes will be sent to the receiver. Then, the receiver can find the cover video from the video database. Therefore, the data transmission load only includes the contents of the index item as video ID, frame ID, and subblock ID. Assuming the size of the secret information is n byte, the transmission loads of different methods are analysed in Table 6. Here, the subblock number of the DWT method and the DCT method is 8. It can be seen that the transmission load of our scheme is greatly reduced since the cover video need not to be transmitted. But the sender and the receiver are required to

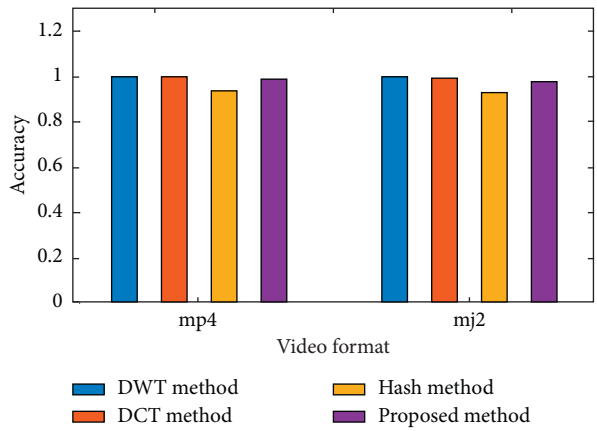


Figure 14: Accuracy comparison with video compression.

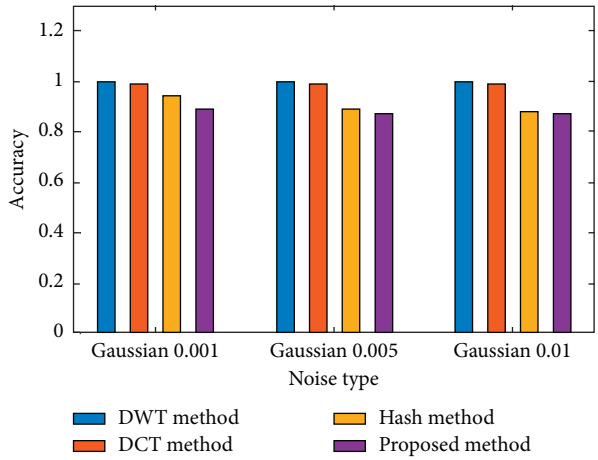


Figure 15: Accuracy comparison with Gaussian noise.

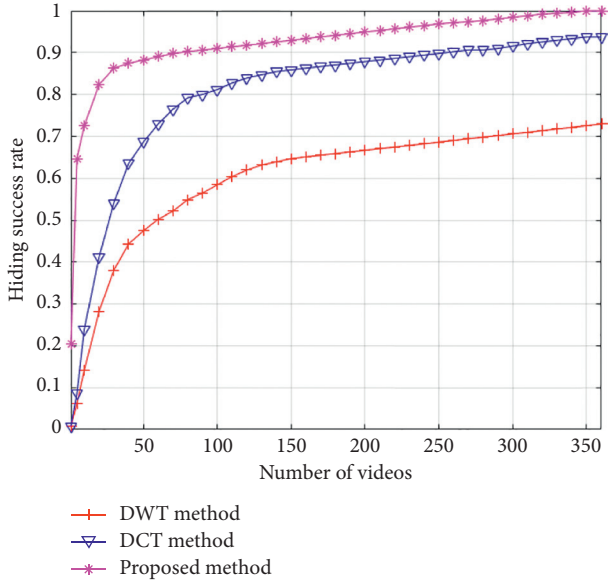


Figure 16: Hiding success rate comparison.

Table 5: Time cost comparison.

Methods	Hash [20]	DCT [22]	DWT [23]	Proposed
Time cost	0.5006 s/B	0.3748 s/B	0.5085 s/B	1.1874 s/B

Table 6: Comparison of transmission data load.

	Transmission data load for n byte secret information
Pixel method [6]	n cover images
Hash method [20]	$\lceil 8n/18 \rceil + 1$ cover images
DCT method [22]	$N + 1$ cover images + side information
DWT method [23]	$n + 1$ cover images + auxiliary information
Proposed method	n indexes

share and update the video database synchronously to ensure the successful information hiding and extraction.

5. Conclusion

In this study, a coverless steganography scheme based on motion analysis of videos is proposed. The capacity, robustness, hiding success rate, time cost, and transmission data load have been investigated and compared with the existing methods. It is shown that the proposed method not only obtains a good trade-off between hiding information capacity and robustness but also achieves higher hiding success rate and lower transmission data load, which shows good practicability and feasibility. However, the time cost of our method is higher due to the complexity of hierarchical optical flow calculation. We will try to improve the efficiency in the future work.

Data Availability

The UCF101 data used to support the findings of this study are available at http://crcv.ucf.edu/data/UCF101/UCF101.rar and the HMDB51 data are available at http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#Downloads.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61772561 and 62002392), the Key Research and Development Plan of Hunan Province (2019SK2022), the Science Research Projects of Hunan Provincial Education Department (18A174 and 19B584), the Degree and Postgraduate Education Reform Project of Hunan Province (2019JGYB154), the National Natural Science Foundation of Hunan (2020JJ4140 and 2020JJ4141),

and the Postgraduate Excellent Teaching Team Project of Hunan Province ([2019] 370–133).

References

[1] B. Wang, W. Kong, W. Li, N. Xiong et al., "A dual-chaining watermark scheme for data integrity protection in Internet of Things," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 679–695, 2019.

[2] W. Wan, J. Wang, J. Li et al., "Pattern complexity-based JND estimation for quantization watermarking," *Pattern Recognition Letters*, vol. 130, no. 1, pp. 157–164, 2020.

[3] J. Wang, W. B. Wan, X. X. Li, J. D. Sun, and H. X. Zhang, "Color image watermarking based on orientation diversity and color complexity," *Expert Systems with Applications*, vol. 140, Article ID 112868, 2020.

[4] Y. Tan, J. Qin, X. Xiang, W. Ma, W. Pan, and N. N. Xiong, "A robust watermarking scheme in YCbCr color space based on channel coding," *IEEE Access*, vol. 7, no. 1, pp. 25026–25036, 2019.

[5] B. Chen, Y. Wu, G. Coatrieux, X. Chen, and Y. Zheng, "JSNet: a simulation network of JPEG lossy compression and restoration for robust image watermarking against JPEG attack," *Computer Vision and Image Understanding*, vol. 197-198, no. 1, pp. 103015–103021, 2020.

[6] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," *Cloud Computing and Security*, vol. 9483, no. 1, pp. 123–132, 2015.

[7] J. Qin, Y. Luo, X. Xiang, Y. Tan, and H. Huang, "Coverless image steganography: a survey," *IEEE Access*, vol. 7, no. 1, pp. 171372–171394, 2019.

[8] Y. Tong, Y. L. Liu, Y. Liu, J. Wang, and G. Xin, "Text steganography on RNN-generated lyrics," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5451–5463, 2019.

[9] Z.-L. Yang, S.-Y. Zhang, Y.-T. Hu, Z.-W. Hu, and Y.-F. Huang, "VAE-stega: linguistic steganography based on variational auto-encoder," *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 1, pp. 880–895, 2021.

[10] L. Y. Xiang, S. H. Yang, Y. H. Liu et al., "Novel linguistic steganography based on character-level text generation," *Mathematics*, vol. 8, no. 1, pp. 1558–1576, 2020.

[11] X. Chen, H. Sun, Y. Tobe, Z. Zhou, and X. Sun, "Coverless information hiding method based on the Chinese mathematical expression," *Cloud Computing and Security*, vol. 9483, pp. 133–143, 2015.

[12] J. Zhang, J. Shen, L. Wang, and H. Lin, "Coverless text information hiding method based on the word rank map," *Cloud Computing and Security*, vol. 10039, pp. 145–155, 2016.

[13] J. Zhang, H. Huang, L. Wang et al., "Coverless text information hiding method using the frequent words hash," *International Journal of Network Security*, vol. 19, no. 6, pp. 1016–1023, 2017.

[14] Z. Zhou, J. Qin, X. Xiang, Y. Tan, Q. Liu, and N. Xiong, "News text topic clustering optimized method based on TF-IDF algorithm on spark," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 217–231, 2020.

[15] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, "Coverless real-time image information hiding based on image block matching and Dense Convolutional Network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2019.

[16] J. Qin, H. Li, X. Xiang et al., "An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, no. 1, pp. 24626–24633, 2019.

[17] 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. 9, pp. 1–12, 2020.

[18] H. Li, J. Qin, X. Xiang, L. Pan, W. Ma, and N. N. Xiong, "An efficient image matching algorithm based on adaptive threshold and RANSAC," *IEEE Access*, vol. 6, no. 1, pp. 66963–66971, 2018.

[19] T. Zhou, B. Xiao, Z. Cai, and M. Xu, "A utility model for photo selection in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 48–62, 2021.

[20] S. Zheng, L. Wang, B. Ling, and D. Hu, "Coverless information hiding based on robust image hashing," *Intelligent Computing Methodologies*, vol. 10363, pp. 536–547, 2017.

[21] Z. Zhou, Q. M. J. Wu, and C. N. Yang, "Coverless image steganography using histograms of oriented gradients-based hashing algorithm," *Journal of Internet Technology*, vol. 18, no. 5, pp. 1177–1184, 2017.

[22] X. Zhang, F. Peng, and M. Long, "Robust coverless image steganography based on DCT and LDA topic classification," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3223–3238, 2018.

[23] Q. Liu, X. Xiang, J. Qin, Y. Tan, J. Tan, and Y. Luo, "Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping," *Knowledge-Based Systems*, vol. 192, no. 1, pp. 105375–105389, 2020.

[24] L. Zou, J. Sun, M. Gao et al., "A novel coverless information hiding method based on the average pixel value of the sub-images," *Multimedia Tools and Applications*, vol. 21, no. 1, pp. 1–16, 2018.

[25] Y. Cao, Z. Zhou, X. Sun et al., "Coverless information hiding based on the molecular structure images of material," *Computers, Materials & Continua*, vol. 54, no. 2, pp. 197–207, 2018.

[26] Y. Tan, J. Qin, H. Tang et al., "Privacy protection for medical images based on DenseNet and coverless steganography," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1797–1817, 2021.

[27] Y. Luo, J. Qin, X. Xiang, and Y. Tan, "Coverless image steganography based on multi-object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1–13, 2020.

[28] T. Xu, M. Zhao, X. Yao, and K. He, "An adjust duty cycle method for optimized congestion avoidance and reducing delay for WSNs," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1605–1624, 2020.

[29] D. Li, L. Qiao, and J. Kim, "A video zero-watermarking algorithm based on LPM," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13093–13106, 2016.

[30] X. Liu, R. Zhao, F. Li, S. Liao, Y. Ding, and B. Zou, "Novel robust zero-watermarking scheme for digital rights management of 3D videos," *Signal Processing: Image Communication*, vol. 54, no. 1, pp. 140–151, 2017.

[31] T. Zhang, X. Wang, X. Xu, and C. L. P. Chen, "GCB-Net: graph convolutional broad network and its application in emotion recognition," *IEEE Transactions on Affective Computing*, vol. 8, pp. 1–12, 2019.

[32] S. Zhang, C. Gao, F. Chen, S. Luo, and N. Sang, "Group sparse-based mid-level representation for action recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 660–672, 2017.

[33] S. Cai, Y. Huang, B. Ye, and C. Xu, "Dynamic illumination optical flow computing for sensing multiple mobile robots

from a drone," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 8, pp. 1370–1382, 2018.

[34] R. Ke, Z. Li, J. Tang et al., "Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 54–64, 2018.

[35] H. Deng, U. Arif, K. Yang, Z. Xi, Q. Quan, and K.-Y. Cai, "Global optical flow-based estimation of velocity for multicopters using monocular vision in GPS-denied environments," *Optik*, vol. 219, no. 1, Article ID 164923, 2020.

[36] M. Sun, W. Sun, X. Zhang, Z. Zhu, and M. Li, "Moving vehicle detection based on optical flow method and shadow removal," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 322, pp. 447–453, 2020.

[37] C. Lv, Y. Wu, D. Fan, and X. Lu, "Fast registration of UAV aerial images based on improved optical-flow model combined with feature-point matching," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8875–8887, 2019.

[38] J. Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm*, Intel Corporation, Microprocessor Research Labs, Hillsboro, OR, USA, 2000.

[39] R. Chaudhry, A. Ravichandran, G. Hager et al., "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Miami, FL, USA, 2009.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6626974, 8 pages
https://doi.org/10.1155/2021/6626974

Research Article

Countering Spoof: Towards Detecting Deepfake with Multidimensional Biological Signals

Xinlei Jin, Dengpan Ye, and Chuanxi Chen

School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Dengpan Ye; yedp@whu.edu.cn

Received 27 December 2020; Revised 20 March 2021; Accepted 10 April 2021; Published 22 April 2021

Academic Editor: Beijing Chen

The deepfake technology is conveniently abused with the low technology threshold, which may bring the huge social security risks. As GAN-based synthesis technology is becoming stronger, various methods are difficult to classify the fake content effectively. However, although the fake content generated by GANs can deceive the human eyes, it ignores the biological signals hidden in the face video. In this paper, we proposed a novel video forensics method with multidimensional biological signals, which extracting the difference of the biological signal between real and fake videos from three dimensions. The experimental results show that our method achieves 98% accuracy on the main public dataset. Compared with other technologies, the proposed method only extracts fake video information and is not limited to a specific generation method, so it is not affected by synthetic methods and has good adaptability.

1. Introduction

With the rapid advancement of computer vision and digital content processing technology, face tampering is no longer limited to pictures, some deep learning technologies (e.g., deepfake) can be utilized to generate human faces in videos, which are very similar to natural face videos taken by using digital cameras, but it is difficult to distinguish them with the naked eyes. The recent research study by Korshunov [1] shows that fake videos can easily deceive the face recognition system, and some serious security risks, such as fake news, have been raised by them.

Deepfake technology is the result of scientific and technological progress and the rapid development of artificial intelligence technology, and it has broad application prospects. For example, deepfake technology is used in entertainment industries such as films, which can save time and labor costs. However, if this technology is abused by criminals, it will also cause a serious crisis, and it can even forge the speeches of world leaders, seriously endangering political security. Therefore, the forensics of deepfake videos is of great significance. At present, the forensics method of deepfake video is mainly based on intraframe or interframe information by analyzing the difference between real and fake videos.

In this paper, we propose a deepfake video forensics method based on multidimensional biological signals. Recent work shows that heart rate signals can be used to effectively distinguish between real and fake videos [2, 3]. Although GANs can generate fake content that deceive human eyes, it destroys the original biological signals of the real video, such as heart rate signals. Therefore, we can classify the real and fake videos by extracting and analyzing the biological signals in the videos. Our main contributions are as follows:

(1) We propose a synthetic video forensics method, which mainly analyses the different biological signals between real and fake videos to detect the spoofed content.

(2) We further explore the distinct information in the multidimensional scene to ensure the technological efficiency. That is, we utilize the RGB space to concentrate on the color variations, the YUV space to concentrate on brightness alteration, and the chrominance method to reduce noise effects.

(3) We analyzed the shortcomings of traditional photoplethysmography (PPG) and used a deep neural network to realize the classification of real and fake videos. The experimental results show that the deep models can reach high detection accuracy, which is about 98% on the main public dataset.

The rest of this paper is organized as follows. Section 2 introduces related work, including the development of PPG and deepfake video forensics. Section 3 describes the proposed method in detail. Section 4 shows the details and results of our experiment. In Section 5, we conclude and give the future work.

2. Related Work

2.1. Deepfake. Deepfakes are fake videos digitally manipulated to depict people saying and doing things that never actually happened. Deepfakes rely on neural networks that analyze large sets of data samples to learn to mimic a person's facial expressions and mannerisms. The process involves feeding footage of two people into a deep learning algorithm to train it to swap faces.

The overall pipeline of the basic deepfake is shown in Figure 1. The autoencoder is usually formed by two convolutional neural networks (the encoder and the decoder). The encoder converts the input target's face to a vector. There is only one single encoder regardless of the identities of the subjects to ensure the encoder captures identity-independent attributes such as facial expressions. On the other hand, each identity has a dedicated decoder, which generates a face of the corresponding subject from the vector. Specifically, an encoder-decoder pair is formed alternatively using encoder and decoder for input face of each subject, and their parameters are optimized to minimize the reconstruction errors. The parameter update is performed with the back-propagation until convergence. The training stage can be stated as

$$\begin{aligned} \min L_A &= \frac{1}{N}\sum_{i=1}\left\|F_i - D_A\left(E\left(F_i;\,\theta\right);\,\phi_A\right)\right\|^2, \\ \min L_B &= \frac{1}{N}\sum_{i=1}\left\|F_i - D_B\left(E\left(F_i;\theta\right);\phi_B\right)\right\|^2, \end{aligned} \tag{1}$$

where L denotes the loss value of the autoencoder; N is the number of input data of the network; F_i is the input face image; θ is the weight of encoder E; and Φ is the weights of decoder D.

In the converting stage, the trained decoder B is used to decode the latent vector of face A to obtain the face-swapping image of A. Similarly, we can use the trained decoder A to decode the latent vector of face B to obtain the face-swapping image of B. The converting stage can be stated as

$$\begin{aligned} F_A' &= D_B\left(E\left(F_A;\,\theta\right);\,\phi_B\right), \\ F_B' &= D_A\left(E\left(F_B;\theta\right);\phi_A\right), \end{aligned} \tag{2}$$

where F denotes the original face and F' denotes the fake face.

2.2. Biological Signals. Biological signal extraction was originally used in the medical field to detect whether the patient's heart rate (HR) or other signals are normal, so that the doctor can observe the abnormal biological signal of the patient in time. However, electrocardiogram (ECG) leads, pulse oximeters, and other detectors all require specific sensors to be connected to the human body. To avoid the use of intrusive sensors, computer vision researchers have proposed a method of noncontact remote HR measurements, based on observing subtle changes in color and motion in the RGB video, such as remote photoplethysmography (PPG) [4, 5].

Balakrishnan et al. [6] show that heart activity can cause head movements, which can be used to extract heart rate estimates from video streams. Tulyakov proposed a chrominance method, which can effectively improve the accuracy of heart rate estimation [5]. Niu proposed a remote heart rate estimation method based on deep learning and achieved good results [7].

2.3. Forgery Detection. To deal with the possible harm caused by deepfake videos, researchers are exploring effective methods to classify real and fake videos. Because deepfake is also a forgery of images, early detection methods can learn from the forgery detection method of images. Recently, a bunch of high-efficient detectors with the new algorithms have been proposed to improve the performance of tampering detection and localization [8, 9]. Also, in order to specifically detect deepfake forgery, researchers classify real and fake videos based on intraframe information, interframe information, or special artifact.

Nguyen et al. [10] proposed a capsule network that can detect various kinds of attacks, from presentation attacks using printed images and replayed videos to attacks using fake videos created using deep learning. It uses fewer parameters than traditional convolutional neural networks with similar performance. Do et al. [11] used a deep convolutional neural network (VGGFace) for detecting real/fake images from GANs. Afchar et al. [12] exploited features at a mesoscopic level, instead of purely microscopic and macroscopic features, and proposed mesonet and meso-4 net, which have a low number of parameters. Bonettini et al. [13] combined CNNs, attention layers, and siamese training and achieved good performance on DFDC. Li and Lyu [14] created negative data only using a simple image processing operation, rather than using deepfake to produce, and then used CNN models to classify the videos. Zhao [15] formulated deepfake detection as a fine-grained classification problem and proposed a new multiattentional deepfake detection network. Liu [16] proposed a novel Spatial-Phase Shallow Learning (SPSL) method, which combines the spatial image and phase spectrum to capture the upsampling artifacts of face forgery to improve the transferability.

Güera and Delp [17], based on temporal inconsistencies between frames, used CNN (frame feature extraction) and RNN (temporal sequence analysis) for real and fake video

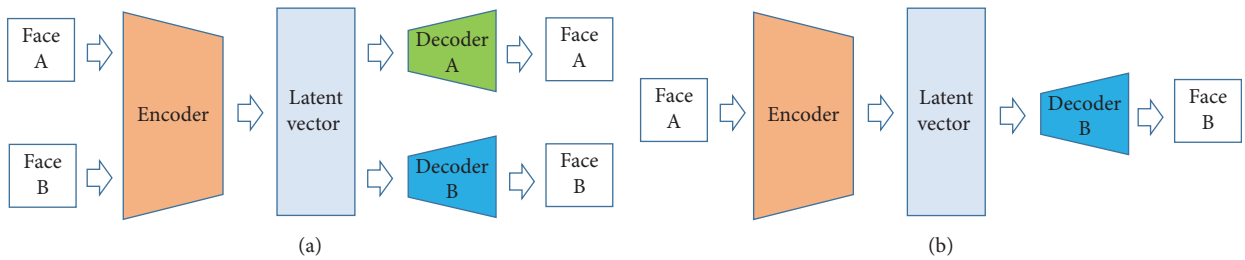


Figure 1: Overview of the deepfake procedure. (a) The training stage of deepfake. (b) The converting stage of deepfake.

classification. Sabir et al. [18] also proposed the CNN + RNN method, but they used face alignment and bidirectional recurrency.

Agarwal et al. [19] tracked facial and head movements and then extracted the presence and strength of specific action units and classified real and fake video by SVM. Li et al. [20] used CNN and RNN to detect abnormal blinking in fake videos. Yang et al. [21] classified real and fake videos by the inconsistency of 3D head poses. Li et al. [22] detected whether the input image can be decomposed into the blending of two images from different sources. Wang et al. based on monitoring neuron behaviors to spot AI-synthesized fake faces [23].

3. Method

In this section, we first analyze the discrepant biological signals between real and fake videos. Then, we point out the inefficiency of the traditional PPG method for detecting the deepfake video. Lastly, we propose a deepfake video forensics method based on the inconsistency of biological signals, and experimental results of the evaluation verify the effectiveness of our method.

3.1. Deepfake Detection with Biological Signals. Although PPG technology has been developed for a long time, it is not easy to extract heart rate signals in an unrestricted environment. We analyzed the method of manually extracting heart rate signals from the face video using computer vision; Figure 2 shows that these methods cannot distinguish fake videos from real videos. We selected a pair of real and fake videos from the DeepFakeDetection (DFD) dataset and used the Kalman filter [24] method to estimate heart rate signals from them. The result shows that the difference in heart rate signals between real and fake videos is not obvious.

Generally, to eliminate motion artifacts and noise caused by environmental changes and extract pure heart rate signals better, the videos are always processed by denoising and filtering. However, these technologies destroy the abnormal heart rate signals in the fake video, which cause the weak classification effect. Therefore, we map the video to ppg_map and classify it through the deep network to achieve the effect of deepfake video classification based on different heart rate extraction algorithms. In detail, given a video Vmmc5 (= {T1, T2, T3 ..... Tk}) including k frames, for each frame, we first extract the face and make face alignment. Then, the skin segmentation is performed to remove the influence of the background. Next, the face image is divided into n blocks (R1, R2, R3 ... Rn), which are independent on each other. Lastly, we calculate the signal value in each block from multidimension. The signal values of different blocks in the same frame are arranged in columns, and the signal values of the same block in different frames are arranged in rows to form our ppg_map. Then, these ppg_maps are used to train the CNN classification model, as shown in Figure 3.

In the process of generating ppg_map, it is necessary to avoid the adverse effects of the head movement and background of the characters. We will discuss this in detail in Section 3.2.

3.2. ppg_Map Generation. The beating of the human heart causes the periodic constriction of blood vessels, which affects the skin's reflection of light. This change is not easily detectable by the human eyes, but it can be detected and recorded by optical instruments. The facial area in the face video can well reflect the heart rate information of the human body. So, we located the facial area and extracted the biological signals.

3.2.1. Face Detection and Alignment. In order to make the detection faster and simpler, the Viola and Jones method [25] is utilized to detect human faces. However, because the faces in the video will not be fixed at a certain position and angle, we align the detected faces by rotating the face to keep both eyes at the same level. On the other hand, the face area detected by the Viola–Jones method is larger than the real face area and contains more background area; we further adjust the region of interest (ROI). In other words, we located 81 landmarks and used four points (1, 8, 15, and 71) as reference points to adjust the face region (Figure 4) to make the ROI include as many face regions as possible.

3.2.2. Skin Detection and Segmentation. The biological signals are extracted from the facial skin, so we reduce the negative influence of other nonskin areas, such as eyes, hair, and background areas. Meanwhile, this will also reduce the disturbance caused by eye blinking and lip motions. Consequently, in the video frame, we first adopt the skin detection algorithm to gain the main facial skin information. Then, as a mask, the skin area is used to extract the facial skin and remove the background and nonskin areas.

3.2.3. Blocks Division and Signal Extraction. Now, we have made the skin detection and segmentation to make

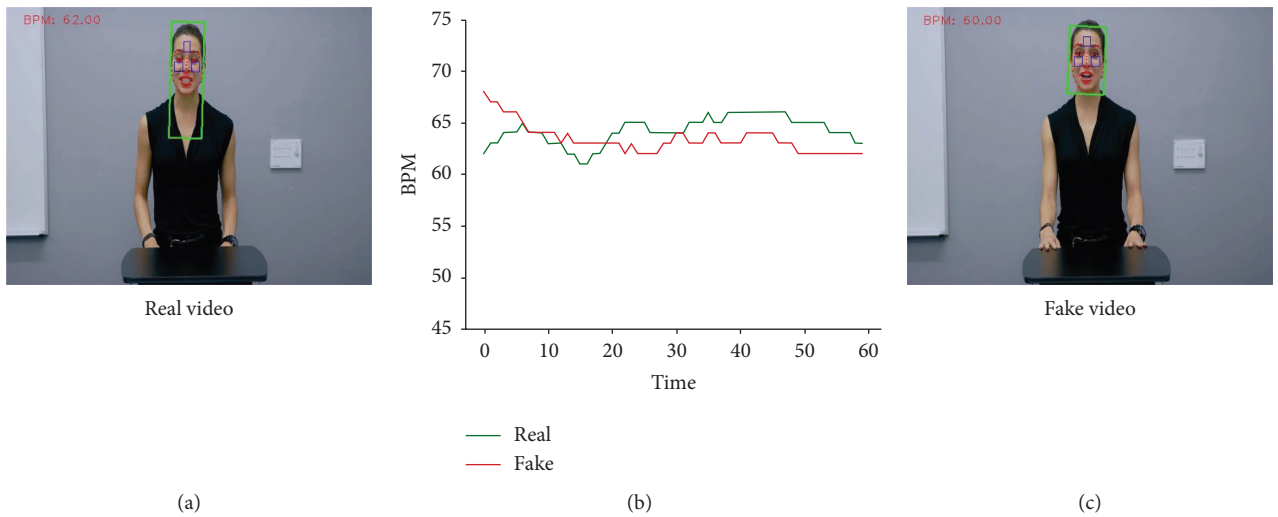


Figure 2: Comparison of real and fake video heart rate. The horizontal axis represents the number of frames of the video, and the vertical axis represents the detected heart rate.

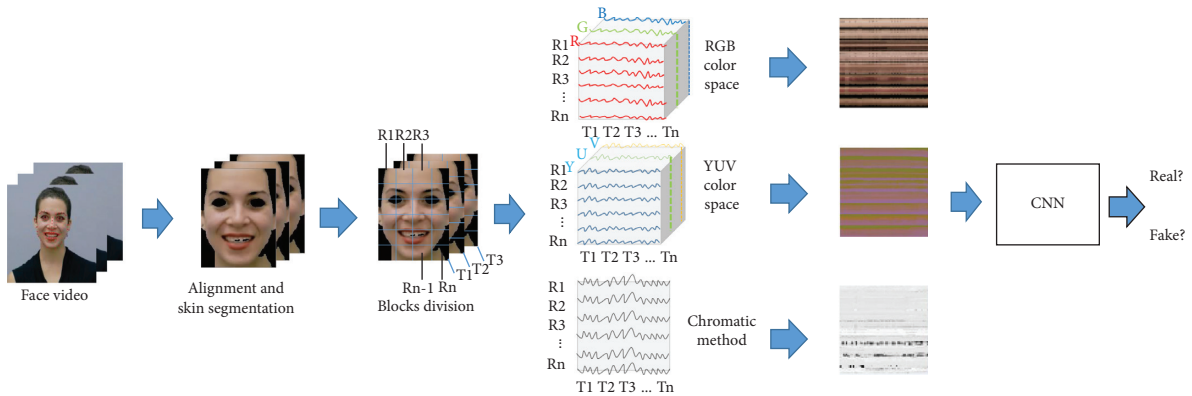


Figure 3: Overview of the proposed method.

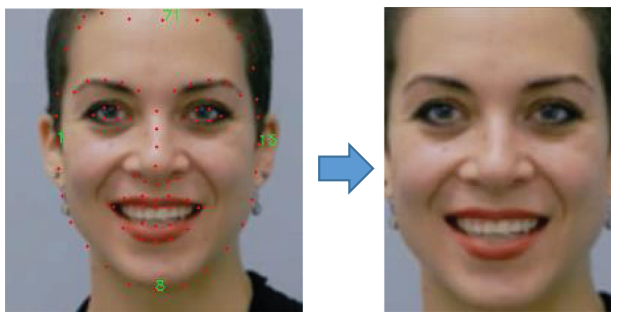


Figure 4: Refining the face ROI. We locate 81 landmarks on the face and use the four points (1, 8, 15, and 71) as benchmarks to further adjust the detected facial area to reduce the background area.

biological signals clearer. Then, the video frame is divided into m ∗ n blocks for extracting biological signals from every block. The PPG method mainly extracts heart rate signals from three dimensions [4–6]. That is, the RGB dimension intuitively reflects the changes in the color of the human face, the YUV dimension pays more attention to changes in brightness, and the chrominance dimension can effectively eliminate environmental artifacts and errors caused by head movement. So, we extract biological signals from RGB color space, YUV color space, and chrominance dimension.

(1) Color Space Dimension. We split the block into three channels of RGB (YUV) and calculate the pixel average of each channel for all blocks. Then, 3 sequences of length m ∗ n can be derived in a frame. Meanwhile, the same block in different frames is also changed with frames. When these procedures are employed in T frames, we can get a three-dimensional matrix with the shape $T * N * 3$, where T denotes the number of frames, N denotes the number of blocks, and 3 represents three channels (RGB or YUV). Each row of the matrix represents the change of the same block on different frames, and each column represents the changes of different blocks in the same frame.

(2) Chrominance Dimension. We calculate the average chrominance of each block [5]. For each pixel, the chrominance signal C is computed as the linear combination of two signals Xf and Yf:

$$C = \mathrm{Xf} - \alpha \mathrm{Yf},$$
$$\alpha = \frac{\sigma(\mathrm{Xf})}{\sigma(\mathrm{Yf})}, \tag{3}$$

where σ (Xf) and σ (Yf) denote the standard deviations of Xf and Yf. The signals Xf and Yf are band-passed filtered signals obtained, respectively, from the signals X and Y,

$$X = 3\mathrm{Rn} - 2\mathrm{Gn},$$
$$Y = 1.5\mathrm{Rn} + \mathrm{Gn} - 1.5\mathrm{Bn}, \tag{4}$$

where Rn, Gn, and Bn are the normalized values of the individual color channels. When we adopt the operations for all blocks and T frames, we can get a two-dimensional matrix with the shape $T * N$.

These matrices are stored as color maps (three-dimensional matrix) and grayscale maps (two-dimensional matrix) to form the corresponding ppg_map. Then, we move the sliding window to generate the next ppg_map the same way.

3.3. CNN-Based Classification. We use a CNN classifier to classify the generated ppg_map. The network consists of six convolutional layers, using the 'relu' activation function, followed by a flatten layer. There are two fully connected layers after convolutional layers. The last fully connected layer uses 'softmax' as the activation function and outputs the scores of the positive and negative classes. In order to avoid overfitting, we added a dropout layer, as shown in Figure 5.

For each dimension in Section 3.2.3, we trained the model and get the accuracy on the testing set. Furthermore, we combine the signals of three dimensions to make the final decision.

4. Results

In this section, we will introduce the details of our experiment. First, we describe the dataset we used. Then, we provide detailed experimental settings and the result of the experiment.

4.1. Dataset. We used three public datasets to train and test our method. For each dataset, we generated the ppg_maps and divided it into a training set, validation set, and testing set according to the ratio of 6 : 2:2. We optimize our model on the training set and validation set and then get the forensics accuracy on the testing set.

4.1.1. Face Forensics++. The FF++ dataset is proposed by Andreas [26], consisting of 1000 original video sequences that have been manipulated with four automated face manipulation methods: Deepfakes, Face2Face, FaceSwap, and NeuralTextures. 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. Owing to the Face2Face and NeuralTextures method in the FF++ dataset does not tamper the whole face (we obtain biological signals from the whole face, and when the tampering part is too small, it will reduce effectiveness of the method), we mainly verify our method on Deepfakes and FaceSwap datasets.

4.1.2. DeepFake Detection. The DFD dataset contains 363 original videos performed by actors and 3068 manipulated videos. These actors are required to perform different actions and then implement face-swap technology between different actors. To better extract the biological signals from the face, we chose a few specific actions, such as "podium speech happy" and "talking still." In these actions, the face is well facing the camera, and there are not too many interference factors. Therefore, we used 176 real videos and 754 fake videos from DFD. The biggest problem with the DFD dataset is the imbalance of positive and negative samples. So, we should expand the real video. The principle of expansion is any segment of the real video also is a real video. So we use the idea of sliding window to generate ppg_map. When processing real video, the stride of the sliding window is smaller than the length of sliding window, as shown in Figure 6. After expansion, we are equivalent to using 704 real videos and 754 fake videos.

4.1.3. UADFV. The UADFV dataset is proposed by Yang et al. [21], which contains 49 real videos and 49 fake videos. The average length of each video is about 11 seconds, and the resolution is 294×500 pixels.

4.2. Experiment Setting and Results. For generating ppg_map, we divide the face frame into $8 * 8$ blocks ($N = 64$) and used 64 frames ($T = 64$) to generate a ppg_map (which means the length of sliding window is 64), so the pixels of each ppg_maps are 64×64. Figure 7 shows a schematic of ppg_map.

We implemented this code on a workstation with four 2080Ti GPU cards. The model was trained using RMSprop for 160 epochs with a learning rate of 0.0004.

We used the Deepfakes dataset in FF++ (RGB dimension) to verify the effectiveness of the model. The accuracy and loss values of this model on the training set are shown in Figure 8. It can be seen from Figure 8 that as the epochs increase, the classification accuracy of the model is gradually increased, while the loss value is gradually decreased, and it stabilizes at 160 epochs, which illustrates the effectiveness of the model in this paper.

In order to prove the advantage of multidimensional signals, we analyzed the classification accuracy of single-dimensional signals and multidimensional signals, as shown in Table 1. The accuracy can be improved obviously when using multidimensional (M-D) signals.

4.3. Comparison. In order to verify the effectiveness of the method, a comparative experiment was carried out with the model mentioned in FaceForensics++, and the comparison results are shown in Table 2. The results show that our method has higher detection accuracy than other methods.

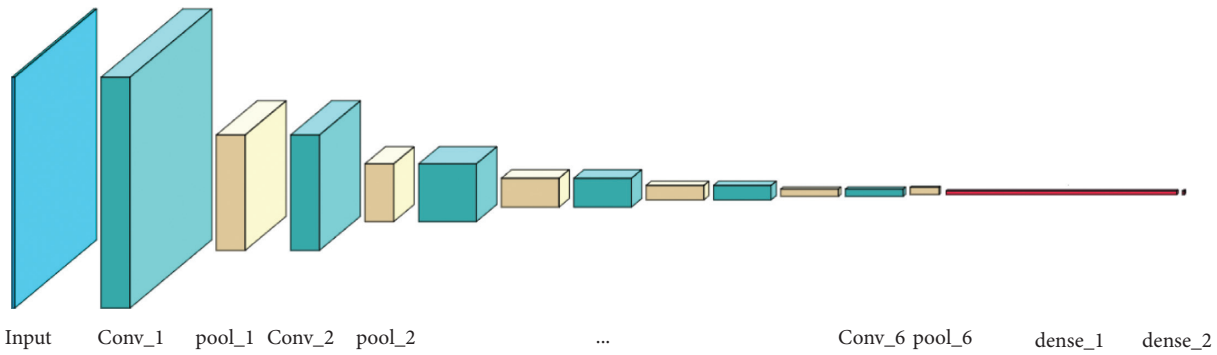


Figure 5: CNN architecture. We used six convolution layers with max pooling, followed by a flatten layer and dense layers.

Figure 6: Different ways to generate ppg_map for real and fake videos. (a) When dealing with fake video, the stride of the sliding window is equal to the length of the sliding window. (b) When dealing with a real video, the stride of the sliding window is smaller than the length of the sliding window.

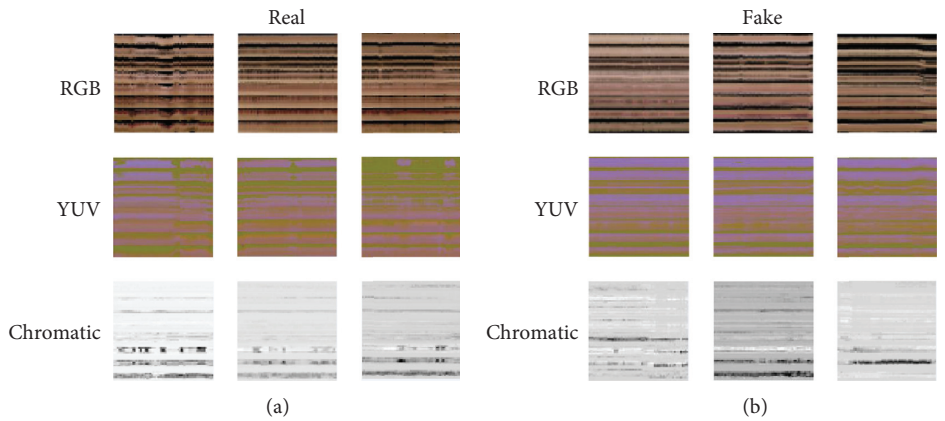


Figure 7: Schematic diagram of the ppg_map. (a) The ppg_maps generated by real videos. (b) The ppg_maps generated by fake videos.

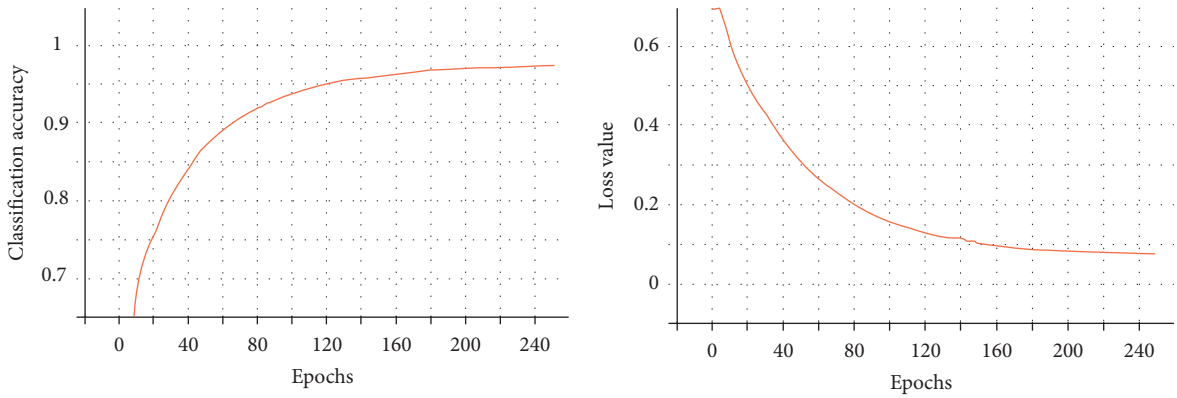


Figure 8: ((a), (b)) The changing curve of accuracy rates and loss value with training times, respectively.

Table 1: The accuracy on the testing set on different datasets.

Method	DFD (%)	UADFV (%)	Deepfakes (%)	FaceSwap (%)
Chrominance	86.74	87.23	81.23	79.25
YUV	90.21	89.36	94.23	88.26
BGR	94.26	88.30	96.20	90.84
M-D	**97.14**	**94.72**	**98.01**	**93.64**

Table 2: Comparison of experimental accuracy results by different models.

Method	Deepfakes (%)	Face2Face (%)
Afchar et al. [12]	87.3	56.2
Bayar and Stamm [27]	84.5	73.7
Rahmouni et al. [28]	85.5	64.2
Baek et al. [29]	71.8	68.6
Rossler et al. [30]	96.4	86.9
Dogonadze [31]	93.6	83.9
Ours	**98.01**	**93.94**

5. Conclusions

In this paper, we propose a forensics method based on biological signals, through a deep neural network to realize the classification of real and fake videos. The deepfake cannot effectively retain the biological signals in the face video. Consequently, we use multidimensional biological signals to analyze the differences between real and fake videos. However, some deepfake videos are hard to be exposed under the complicated conditions such as unstable character movements and complex scene switching. We hope that the deepfake detection in these scenarios could be solved effectively by using signal enhancement and denoising in the near future work.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China NSFC (grant numbers 62072343, U1736211), the National Key Research Development Program of China (grant numbers 2019QY(Y) 0206). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements.

References

[1] P. Korshunov and S. Marcel, "Deepfakes: a New Threat to Face recognition? assessment and detection," 2018, https://arxiv.org/abs/812.08685.

[2] U. A. Ciftci, I. Demir, and L. Yin, "Fakecatcher: detection of synthetic portrait videos using biological signals," 2020, http://arxiv.org/abs/1901.02212.

[3] V. Conotter, E. Bodnari, G. Boato, and H. Farid, "Physiologically-based detection of computer generated faces in video," in *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*, pp. 248–252, IEEE, Paris, France, October 2014.

[4] P. V. Rouast, M. T. P. Adam, R. Chiong, D. Cornforth, and E. Lux, "Remote heart rate measurement using low-cost RGB face video: a technical literature review," *Frontiers of Computer Science*, vol. 12, no. 5, pp. 858–872, 2018.

[5] S. Tulyakov, X. Alameda-Pineda, E. Ricci, L. Yin, J. F. Cohn, and N. Sebe, "Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2396–2404, Las Vegas, NV, USA, June 2016.

[6] G. Balakrishnan, F. Durand, and J. Guttag, "Detecting pulse from head motions in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3430–3437, Portland, OR, USA, June 2013.

[7] X. Niu, S. Shan, H. Han, and X. Chen, "Rhythmnet: end-to-end heart rate estimation from face via spatial-temporal

representation," *IEEE Transactions on Image Processing*, vol. 29, pp. 2409–2423, 2019.

[8] 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*, p. 1, 2020.

[9] B. Chen, X. Qi, Y. Zhou, G. Yang, Y. Zheng, and B. Xiao, "Image splicing localization using residual image and residual-based fully convolutional network," *Journal of Visual Communication and Image Representation*, vol. 73, Article ID 102967, 2020.

[10] 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, IEEE, May 2019.

[11] N. T. Do, I. S. Na, and S. H. Kim, "Forensics face detection from gans using convolutional neural network," 2018.

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

[13] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of CNNs," 2020, http://arxiv.org/abs/2004.07676.

[14] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," 2018, http://arxXiv.org/abs/:1811.00656.

[15] H. Zhao et al., "Multi-attentional deepfake detection," arXiv preprint arXiv:2103.02406 (2021).

[16] H. Liu, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain," 2021, http://arxXiv.org/abs/:2103.01856.

[17] D. Güera and E. J. 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, IEEE, Auckland, New Zealand, 2018, November.

[18] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *Interfaces (GUI)*, vol. 3, no. 1, 2019.

[19] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," in *Proceedingd of the CVPR Workshops*, pp. 38–45, Venice, Italy, 2019, June.

[20] Y. Li, M. C. Chang, and S. Lyu, "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, IEEE, Hong Kong, China, 2018, December.

[21] 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, IEEE, Brighton, UK, 2019, May.

[22] L. Li, J. Bao, T. Zhang et al., "Face x-ray for more general face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5001–5010, Seattle, WA, USA, August 2020.

[23] R. Wang, F. Juefei-Xu, L. Ma, and X. Xie, "FakeSpotter: a simple yet robust baseline for spotting AI-synthesized fake faces," 2019, http://arxXiv.org/abs/:1909.06122.

[24] S. K. A. Prakash and C. S. Tucker, "Bounded Kalman filter method for motion-robust, non-contact heart rate estimation," *Biomedical Optics Express*, vol. 9, no. 2, pp. 873–897, 2018.

[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR*, IEEE, Kauai, HA, USA, 2001, December.

[26] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: a large-scale video dataset for forgery detection in human faces," 2018, http://arxXiv.org/abs/:1803.09179.

[27] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pp. 5–10, Web Tokyo, Japan, 2016, June.

[28] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Proceedings of the IEEE Workshop on Information Forensics and Security (WIFS)*, IEEE, Rennes, France, January 2017.

[29] J.-Y. Baek, Y.-S. Yoo, and S.-H. Bae, "Generative adversarial ensemble learning for face forensics," *IEEE Access*, vol. 8, pp. 45421–45431, 2020.

[30] A. Rossler, D. Cozzollino, L. Verdoliva, C. Riess, J. Thies, and M. Niessnar, "Faceforensics++: learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea, February 2019.

[31] N. Dogonadze, O. Jana, and Ji Hou, "Deep face forgery detection," 2020, http://arxiv.org/abs/2004.11804.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5558873, 12 pages
https://doi.org/10.1155/2021/5558873

Research Article

Reversible Privacy Protection with the Capability of Antiforensics

Liyun Dou,[1] **Zichi Wang**,[1] **Zhenxing Qian**,[2] **and Guorui Feng**[1]

[1]*School of Communication and Information Engineering, Shanghai University, Shanghai, China*
[2]*Shanghai Institute of Intelligent Electronics and Systems, School of Computer Science, Fudan University, Shanghai, China*

Correspondence should be addressed to Zhenxing Qian; zxqian@fudan.edu.cn

Received 1 February 2021; Revised 8 March 2021; Accepted 18 March 2021; Published 12 April 2021

Academic Editor: Beijing Chen

In this paper, we propose a privacy protection scheme using image dual-inpainting and data hiding. In the proposed scheme, the privacy contents in the original image are concealed, which are reversible that the privacy content can be perfectly recovered. We use an interactive approach to select the areas to be protected, that is, the protection data. To address the disadvantage that single image inpainting is susceptible to forensic localization, we propose a dual-inpainting algorithm to implement the object removal task. The protection data is embedded into the image with object removed using a popular data hiding method. We further use the pattern noise forensic detection and the objective metrics to assess the proposed method. The results on different scenarios show that the proposed scheme can achieve better visual quality and antiforensic capability than the state-of-the-art works.

1. Introduction

Photo sharing has become a widespread user activity with the advent of intelligent mobile devices and online social networks (OSN). Image distributions cause privacy concerns and the requirement to modify permissions since the shared content contains sensitive data of users. By providing unique rights to selected communicating parties in OSN, users' security and privacy can be strengthened. A well-established form of privacy protection is to blur a part of an image, which can be achieved by various image processing techniques, for example, blurring, mosaic, masking, and object removal, as shown in Figure 1. In these methods, the first three must introduce a significant amount of distortion to hide the underlying content. Object removal provides more natural viewing conditions and is able to protect the content. This process is reversible such that the original data can be accessed with permissions [1].

After object removal in an image, the broken parts can be inpainted using the surrounding contents. Generally, image inpainting algorithms can be divided into three groups, including the statistical-based, the diffusion-based, the patch-based, and the deep generative models-based methods [2, 3]. Statistical methods use parametric models to describe textures but fail when additional intensity gradients are applied [4]. Diffusion-based methods propagate pixels from the known areas of the image [5–7] using smoothness priors; however, blurring occurs when large and high-frequency regions need to be inpainted. Patch-based and deep generative models are the most widely used, where the former fills the holes in the image using the patch from local or global search regions [8–12] and the latter exploits semantics learned from large-scale datasets [13–15]. None of the inpainting algorithms have considered the secrecy of the inpainted areas from the security perspective. The inpainted images are easy to be detected and located by forensic algorithms.

In this paper, we propose a new privacy protection scheme using image inpainting and data hiding, which realizes the antiforensics capability. When considering the undetectability of edge inpainting, we use the algorithm of the DFNet network [16]. The regions around the broken edge are inpainted twice, and the inpainting results are fused to achieve the capability of antiforensics. By combining image dual-inpainting and data hiding, a privacy protection scheme with antiforensics capability is realized. We combine local variation within and between channels and use the popular data hiding algorithm HILL [17] to embed the

Figure 1: Common privacy protection methods: (a) original image; (b) blurring; (c) mosaic; (d) masking; and (e) object removal.

protection data. The rest of this paper is organized as follows: we introduce the related works in Section 2. The proposed method is depicted in Section 3. Experimental results and analysis are provided in Section 4. Section 5 concludes the whole paper.

2. Related Works

In this section, we introduce the works that are related to the proposed method, including the image inpainting, the data hiding, and the image forensics.

2.1. Image Inpainting. Image inpainting is a method to fill the missing information in an image and is quite important in the field of image processing. Nowadays, the deep generative models-based methods are widely used in the field of image inpainting [14, 18–23]. Numerous methods can be divided into two categories [24]. One approach is to use an effective loss function or construct an attention model to fill in the missing regions to try to make the content more realistic. They use the content in the background to fill, and a better way is to fix the unknown region by partial convolution [18]. The other approach focuses on structural consistency. To ensure the continuity of the image structure, these approaches usually adopt edge-based contextual priors. For example, [19] designed an edge linking strategy that can well solve the image semantic structure inconsistency problem.

Regardless of the inpainting method, there is a discontinuous transition zone at the edge of the inpainting. This area will become a forensic object and thus easy to locate the inpainting area by someone who is interested, which is quite unsafe. In order to not only achieve a good visual effect but also secure safety, a smooth transition needs to be achieved in advance. An iterative method to optimize the pixel gradients in the edge transition regions is proposed in [25],. The quality of fusion depends on whether the incorporated content is consistent with the original content in terms of gradient changes. Thus, Hong et al. [16] design a learnable fusion block to implement pixel-level fusion in the transition region, which is named deep fusion network for image completion (DFNet). The results show that DFNet has superior performances, especially in the aspects of harmonious texture transition, texture detail, and semantic structural consistency.

2.2. Data Hiding. To further optimize the data embedding problem in information hiding, adaptive embedding algorithms are widely proposed. Among them, STC (Syndrome Trellis Coding) [26] based adaptive architectures are most preferred by researchers. This method uses a predefined distortion function to minimize the additive distortion between stego and cover. For the multiscale characteristics of the image space, the design of the distortion function has attracted more and more attention. For instance, Li et al. [17] proposed a new distortion function for image information hiding. The cost function is composed of a high-pass filter and two low-pass filters. The high-pass filter is used to locate the difficult-to-predict parts of an image and then employ the low-pass filters to make the low-cost values more clustered. Furthermore, the methods of MiPOD (Minimizing the Power of Optimal Detector) [27] and ASO (Adaptive Steganography by Oracle) [28] were proposed one after another. In addition, a number of distortion functions have been proposed for JPEG steganography as well, such as IUERD (Improved UERD) [29], UED (Uniform Embedding Distortion) [30], and RBV (Residual Blocks Value) [31].

In addition, some work uses machine learning algorithms to design steganalysis tools to detect steganography. Most of these approaches learn a general steganography model through a supervised strategy and then use it to distinguish suspicious images [32–35]. With the rapid development of deep learning, the performance of steganalysis has been greatly improved [36–38]. However, depth features still have limitations in steganalysis [39]. For example, the truncation and quantization operations in the feature extraction process are difficult to be learned by existing networks. Therefore, feature extraction is still a challenge in steganalysis, and many rich feature sets have been used for JPEGY steganalysis. The main available feature sets include JPEG rich-model [40], DCTR GFR (Gabor filter residuals) [41], and DCTR (Discrete Cosine Transform Residual) [42]. In the classification process, the ensemble classifier is considered to be effective in measuring the feature set [43, 44].

2.3. Image Forensics. Currently, there are two forensic methods of detecting image inpainting [45, 46]. In [45], the authors find that the Laplacian operations along the isophote direction in the inpainted regions are different from the other regions. Accordingly, the inpainted regions can be identified by exploring the changes of local variances between intra- and interchannels. In [46], noise pattern analysis is used to locate the inpainted regions. For the images captured by one camera, the noise patterns in each image are approximately the same and vice versa. Therefore, the noise pattern can be used as the fingerprint for a camera, which is widely adopted in image forensics.

The noise pattern analysis algorithm in [46] is popular. In this model, the pixel values can be constructed by ideal

pixel values, multiplicative noises, and various additive noises, which can be expressed by

$$I = f((I + K) \cdot O) + a, \tag{1}$$

where I and O are the actual pixel and ideal pixel value of the natural scene, a is the sum of various additive noises, $f(\bullet)$ is the camera processing like CFA interpolation, and K is the coefficient for noise pattern. In equation (1), the multiplicative noise $K{\cdot}O$ is the theoretical expression of the noise pattern, which is a multiplicative noise in the high frequencies related to the image contents. Generally, we can use a low-pass filter to remove the additive noises. The residual noise is then used to estimate the noise pattern [47], as shown in the following equation:

$$p = I - F(I), \tag{2}$$

where $F(\bullet)$ is the low-pass filter and p is the estimated noise pattern. The noise pattern can be used to distinguish the content from different images. Therefore, the inpainted region can be detected after extracting the noise pattern from each part of the image.

During inpainting, since there are limited pixels around the damaged regions, each diffusion is smoothed based on the surrounding pixels to accomplish the diffusion. Therefore, the pixels located in the inpainted region satisfy $I_t^n(i, j) = 0$, which means that the results of Laplacian operation on this position remain unchanged along the isophote direction after the diffusion-based inpainting. The Laplacian variation along the isophote direction can be calculated by

$$\delta_{\Delta I(i,j)} = \Delta I(i, j) - \Delta I(i_v, j_v), \quad \forall (i, j) \varepsilon I \tag{3}$$

where $\Delta I(i, j)$ is the (i, j)-th Laplacian value and $\Delta I(i_v, j_v)$ is the result of Laplacian operation on a virtual pixel on (i_v, j_v). The virtual pixel is located at the direction of $\nabla I^{\perp}(i, j)$, and its distance to the pixel $I(i, j)$ is identical to 1.

3. Proposed Method

In this section, we present an antiforensic framework to perform object removal in images using dual-inpainting and data hiding. As shown in Figure 2, the proposed framework contains four parts. We first select the protected area interactively and calculate the percentage of the area in the whole image. Then, the background with the missing protected area was inpainted. In order to achieve a satisfactory visual effect and be as forensic-free as possible, an image dual-inpainting algorithm is proposed, as shown in Figure 3 and described in Section 3.1–3.3. For the inpainted image, region segmentation is performed based on the changes of local variances between the intra- and interchannels. Meanwhile, the protected region is embedded into the background after converting it into a bitstream by combining the HILL embedding algorithm and considering the segmentation. On the recipient side, we can extract the embedded data, fuse it with the background image, and recover the original image.

3.1. Protection Region Selection. We interactively specify the area in an image to be protected, which also means that the hidden area is determined. After that, we calculate the number of the pixels to be hidden, including the values and coordinates of these RGB pixels. The pixels are converted into bit stream for embedding. We define the bits of each pixel as 5×9, in which "5" stands for pixel values in three channels, horizontal and vertical coordinate values, and "9" means that we convert each decimal to 9 bits. In a color image, information can be embedded in all three channels at each position. Thus, the maximum amount of embeddable information is three times the image size. The maximum embedding ratio T is calculated to be 6.66% per image. Let t be the proportion of the selected protection region. The proportion should be smaller than a predefined threshold T. An example of the interactive region selection is shown in Figure 4.

3.2. Background Processing. After specifying the protection area, we remove the contents in this area and inpaint the image. When inpainting large areas, it is often not possible to perfectly blend the inpainted area with the existing content, especially in the edge areas [16]. To fill this gap, the DFNet network [23] introduces a fusion block, which combines the structural and texture data and smoothly blends them during the inpainting process. As shown in Figure 5, I is the input image, F_k is the feature maps from k-th layer, and I_k is resize of I. The learnable function M is designed to extract the raw completion C_k from feature maps F_k, which is as follows:

$$C_k = M(F_k), \tag{4}$$

where M denotes the channel conversion operation, which converts n channel feature maps into 3-channel images under the condition of constant resolution.

In addition, another learning function A is used to generate the alpha composition map a_k:

$$a_k = A(F_k, I_k). \tag{5}$$

Map a_k usually is obtained by synthesis from a single channel or 3 channels for imagewise alpha composition. Previous experience has demonstrated that channelwise alpha composition performs better. A is a convolutional module which consists of 3 convolutional layers with kernel sizes of 1, 3, and 1, respectively. The final result I_k' is achieved by

$$I_k' = B(a_k, C_k, I_k) = a_k * C_k + (1 - a_k) * I_k. \tag{6}$$

The fusion block makes the image inpainted by the DFNet network almost visually free of edge discontinuity. Although the DFNet network achieves good visual results, it is not suitable for privacy protection since it can be easily localized for forensics. For example, pattern noise of the image detection reveals clear artifacts in the restoration edge area. To conceal these traces and achieve the privacy-preserving, further manipulation of the inpainting image is required.

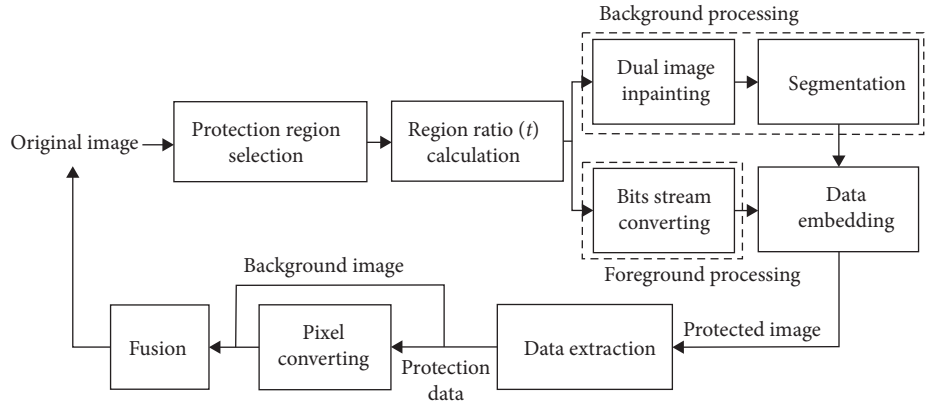


Figure 2: Architecture of the proposed scheme.

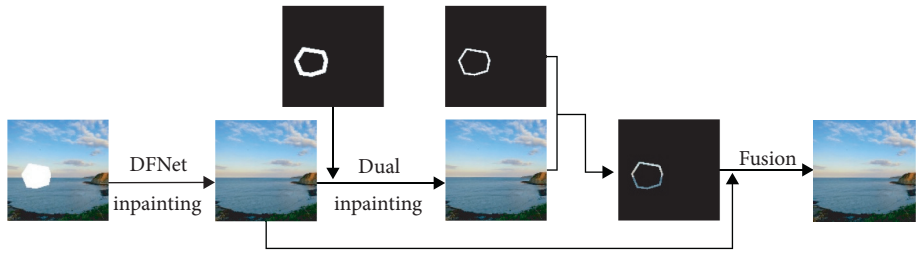


Figure 3: Dual-inpainting process architecture.

The detection area is mostly found in the edge area of the restoration, so we consider secondary processing of the edge area to eliminate the traces left during the restoration process. In this process, we used the mathematical morphology of the dilation operation and the erosion operation. In the dilation operation, the structural element B is used as an external window to increase the overall boundary of the target image. In the erosion operation, the structural elements serve as the internal windows to eliminate the boundary of the image. The dilation operation is expressed by equation (7) and erosion operation can be expressed by equation (8):

$$I \oplus B = \left\{(i, j) | B_{(i,j)} \cap I \neq \varnothing\right\}, \tag{7}$$

$$I \Theta B = \left\{(i, j) | B_{(i,j)} \subseteq I\right\}, \tag{8}$$

$$B_{(i,j)} = \{(x, y) | x = m + i, y = n + j, (m, n) \in B\}. \tag{9}$$

The specific dual-inpainting process is shown in Figure 3. Firstly, the background image should be inpainted using the DFNet network. Then, we apply a mathematical morphological dilation operation on the edges of the broken region mask map. Based on this mask map, secondary inpainting of the primary inpainted image is performed in the region. In addition, mathematical morphology erosion operation is then applied to the secondary inpainted region, leaving only a portion of the region close to the edge. Note that the dilation operation uses a larger size of structural elements than that of the erosion operation to ensure the results of the secondary inpainting of the lower edge are preserved. The results of the secondary inpainting of the edge region are fused with the primary repair map to obtain a graph of the experimental results of antiedging detection.

3.3. Area Segmentation and Data Hiding. To hide the secret data of the protection region, we employ the popular data hiding framework which can be achieved by STC [17]. We improve the popular cost function HILL for STC to fit the requirements in our method.

In the STC framework, the theoretical minimum steganography distortion D for the marked image with an embedding amount of γ (bits) can be defined as

$$D = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(p_{i,j}^{+} \rho_{i,j}^{+} + p_{i,j}^{-} \rho_{i,j}^{-}\right),$$

$$p_{i,j}^{+} = \frac{e^{-\lambda \rho_{i,j}^{+}}}{\left(1 + e^{-\lambda \rho_{i,j}^{+}} + e^{-\lambda \rho_{i,j}^{-}}\right)}, \tag{10}$$

$$p_{i,j}^{-} = \frac{e^{-\lambda \rho_{i,j}^{-}}}{\left(1 + e^{-\lambda \rho_{i,j}^{+}} + e^{-\lambda \rho_{i,j}^{-}}\right)},$$

where $p_{i,j}^{+}$ and $p_{i,j}^{-}$ are the probabilities of adding 1 or subtracting 1 on $c_{i,j}$, $0 < p_{i,j}^{+} + p_{i,j}^{-} < 1$, and $\rho_{i,j}$ stands for the distortion values used to measure the effects of modification. The parameter λ ($\lambda > 0$) is used to make the ternary data entropy of the modification probability identical to the capacity γ, as shown in the following equation:

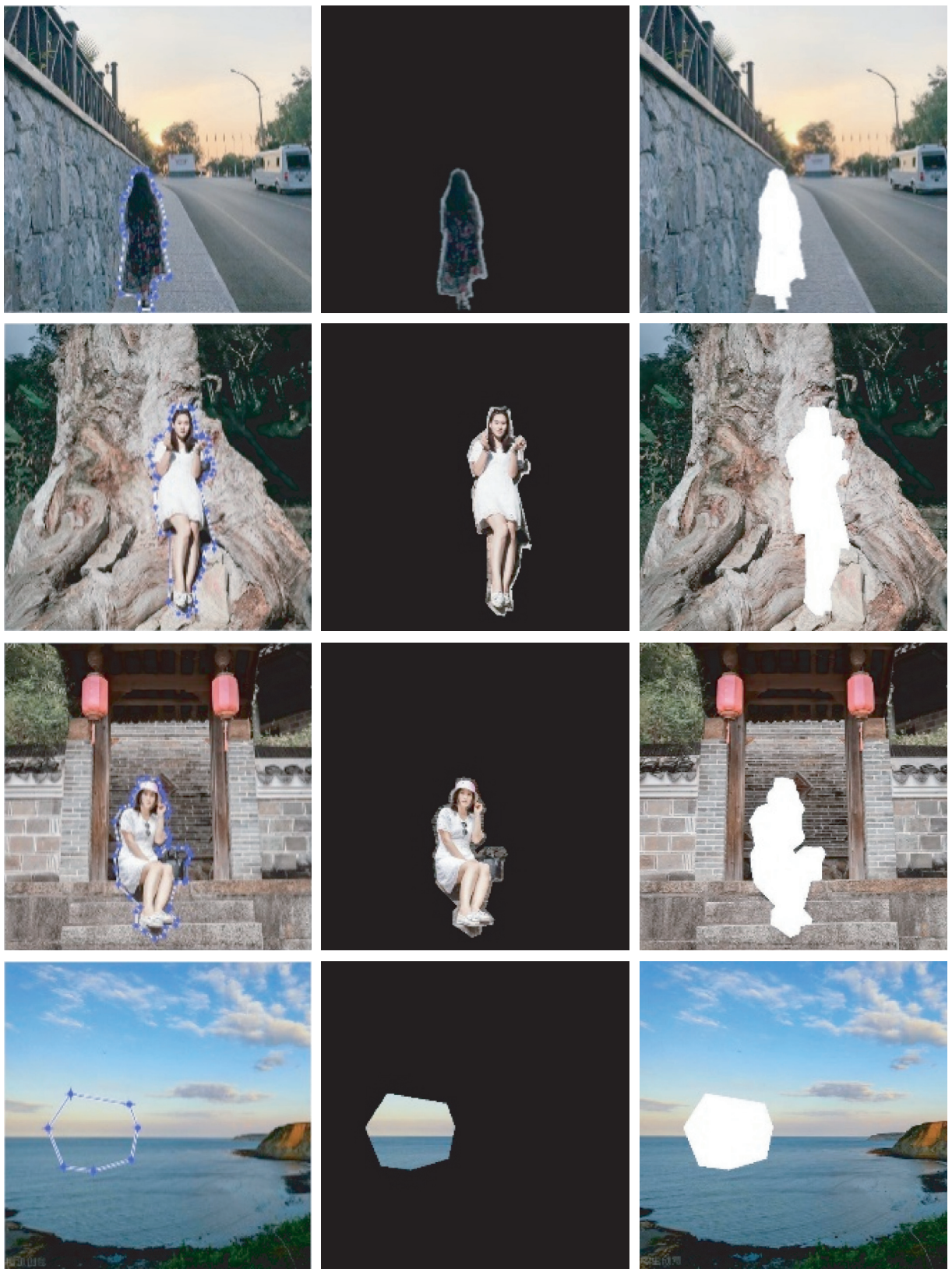


Figure 4: Protection region selection. (a) Interactive selection. (b) Protected areas. (c) Background image.

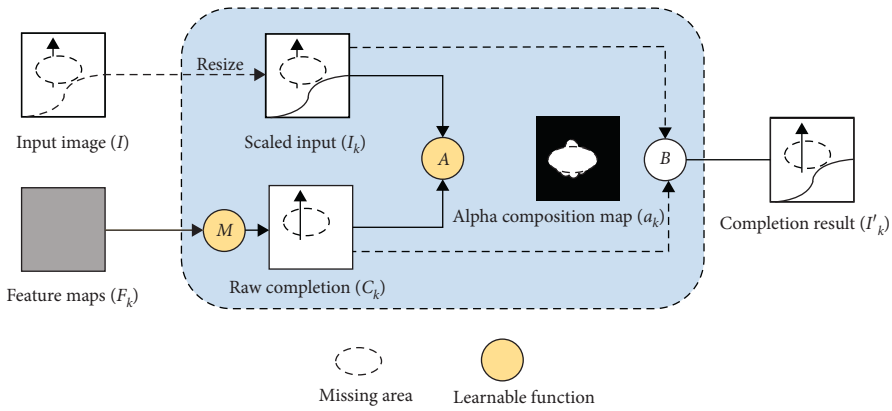


Figure 5: Illustration of fusion block.

$$-\sum_{i=1}^{M}\sum_{j=1}^{N}\left\{p_{i,j}^{+}\log_2 p_{i,j}^{+}+p_{i,j}^{-}\log_2 p_{i,j}^{-}+\left\{\left(1-p_{i,j}^{+}-p_{i,j}^{-}\right)\log_2\left(1-p_{i,j}^{+}-p_{i,j}^{-}\right)=\gamma. \right.\right. \tag{11}$$

To achieve the minimum distortion D, STC encoding is used. Let the secret bits $m = [m_1, m_2, \ldots, m_\gamma]^T \in \{0, 1\}^\gamma$, cover pixels $c = [c_1, c_2, \ldots, c_{MN}]^T$, and stego pixels $y = [y_1, y_2, \ldots, y_{\mathrm{MN}}]^{\mathrm{T}}$. Then, m can be embedded into c using

$$\begin{aligned}\mathrm{Emb}(c,m) &= \arg\min_{\mathbf{y}_l\in C(\mathbf{m})} D(c,y),\\ D(c,y) &= \sum_{c_i\neq y_i}\rho_i^{(y_i-c_i)},\end{aligned} \tag{12}$$

where $y_l \in \{0, 1\}^{MN}$ is the least significant bits of the stego image, $C(m) = \{z \in \{0, 1\}^{MN} | Hz = m\}$ is the companion set of m, and $H \in \{0, 1\}^{\gamma\times MN}$ is a predefined low-density parity test matrix related to embedding speed and embedding efficiency. The embedded bits m can be extracted simply by a matrix multiplication operation:

$$m = Hyl. \tag{13}$$

To fit the requirements in our method, we improve the popular cost function HILL for STC by combining variations within and between adjacent pixel channels. Specifically, we divide the cover image into four regions (marked with green, blue, black, and red in Figure 6) using the cost values of HILL and edge connectivity. The pixel complexity of the four regions decreases in the order of green, blue, black, and red. In other words, the green region has the most complex pixels and is the best embedding region for the whole image. Therefore, secret bits are embedded into the green region preferentially.

4. Experimental Results

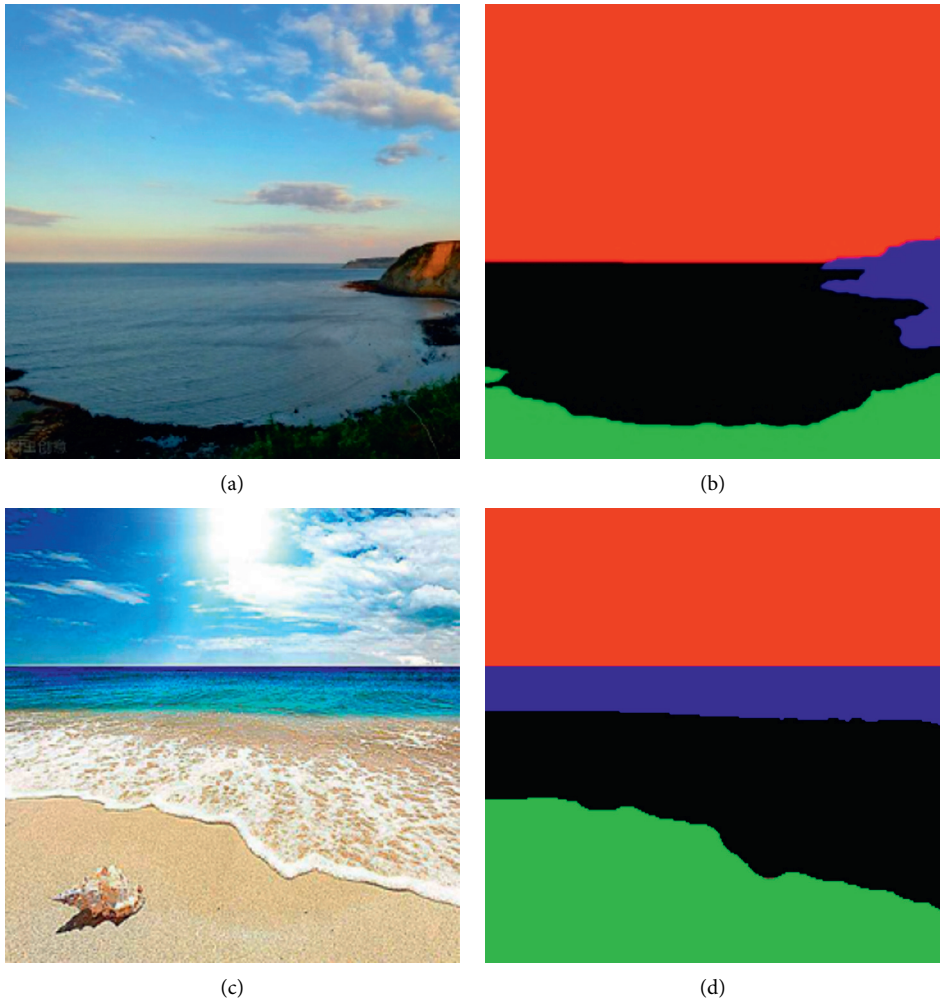
This section presents the experimental evaluation results. Firstly, we introduce the database employed and the corresponding parameters. Then, experiments for each part are presented in turn and their validity is demonstrated.

4.1. Performance for Antiforensics. To evaluate the performance of antiforensics, we randomly select images from the database for validation and interactively select the areas to be protected, as mentioned in Section 2.

In each image, the selection of the protected area is irregular shape generally. For later embedding of data, we strictly controlled the ratio of protected areas to the image to less than 6.66%. We use two separate forensic approaches for the forensic analysis of our results: one is pattern forensics by pattern noise, and the other one is based on changes between and within adjacent pixel channels.

Firstly, we select 50 landscape images sized 512×512 from Today's Headlines. As shown in Figure 7, we selected four of them, $I1$, $I2$, $I3$, and $I4$ in turn. Table 1 lists the space proportion t and the number of pixels to be embedded in the whole image of the corresponding protection area of the four images in Figure 7. Figure 7(c) shows the images after being inpainted based on DFNet, Figure 7(e) shows the images after being inpainted by our method, and Figures 7(d) and 7(f) show the pattern noise maps of Figures 7(c)~7(e), respectively. Comparing with the ground truth Figure 7(b), we find that Figure 7(d) has obvious traces at the repair edges, which makes the repair region easy to be forensically located. While our method overcomes this drawback well, it is difficult to forensically locate our tampered region from the pattern noise forensic aspect only. It shows that our aspect has a good antipattern noise forensic effect.

In Figure 8, we show the experimental results for five images ($M1$, $M1$, $M3$, $M4$, and $M5$) in the UCID database, sized 384×512. Table 2 lists the space proportion t and the number of pixels to be embedded in the whole image of the corresponding protection area of the four images in Figure 8. Two traditional methods and a deep learning method are used for comparison, where the traditional methods are edge-oriented and Delaunay-oriented provided by G'MIC [48], a full-featured open-source framework for image processing. The deep learning-based one is the DFNet method mentioned in [16].



(a) (b)

(c) (d)

Figure 6: Examples for area segmentation. (a) Original image. (b) The result of area segmentation.

Comparing from the subjective vision, both our experimental results and the deep learning method outperform the traditional method and achieve good visual connectivity at the edges. In particular, in row 7 of Figure 8, the effect at the red petal achieves a good visual effect after blending with the primary restored image by our secondary processing of the restored edges.

In addition, we localized the inpainted image for forensics by the forensic algorithm proposed in [46], as shown in the even rows of Figure 7. The traditional restoration-based algorithm is easy to be detected and located, and the DFNet-based restoration also achieved good antiforensic results. However, the images obtained by our method are more suitable to hide the area to be protected. In particular, the results are better when the area to be protected accounts for less than 4% of the whole image.

In Table 3, we show the F1 values of the five images in Figure 8, where a smaller F1 value indicates a worse ability to correctly locate the image and indicates that we have a better antiforensic effect. We can see from Table 3 that our method is superior in terms of objective indicators.

4.2. Experiment Setup. In our experiments, we use the free user-shared image dataset provided by Today's Headlines, which contains a large number of people landscapes, and various life images. We also use the UCID database. Based on the maximum amount of data that can be embedded in an image, it can be calculated that the size of the protected area must not exceed 6.66% of the whole image ($T = 6.66\%$) no matter how large the image size is. For the structural elements for the mathematical morphology of the background process, the circular structure is employed since it has a

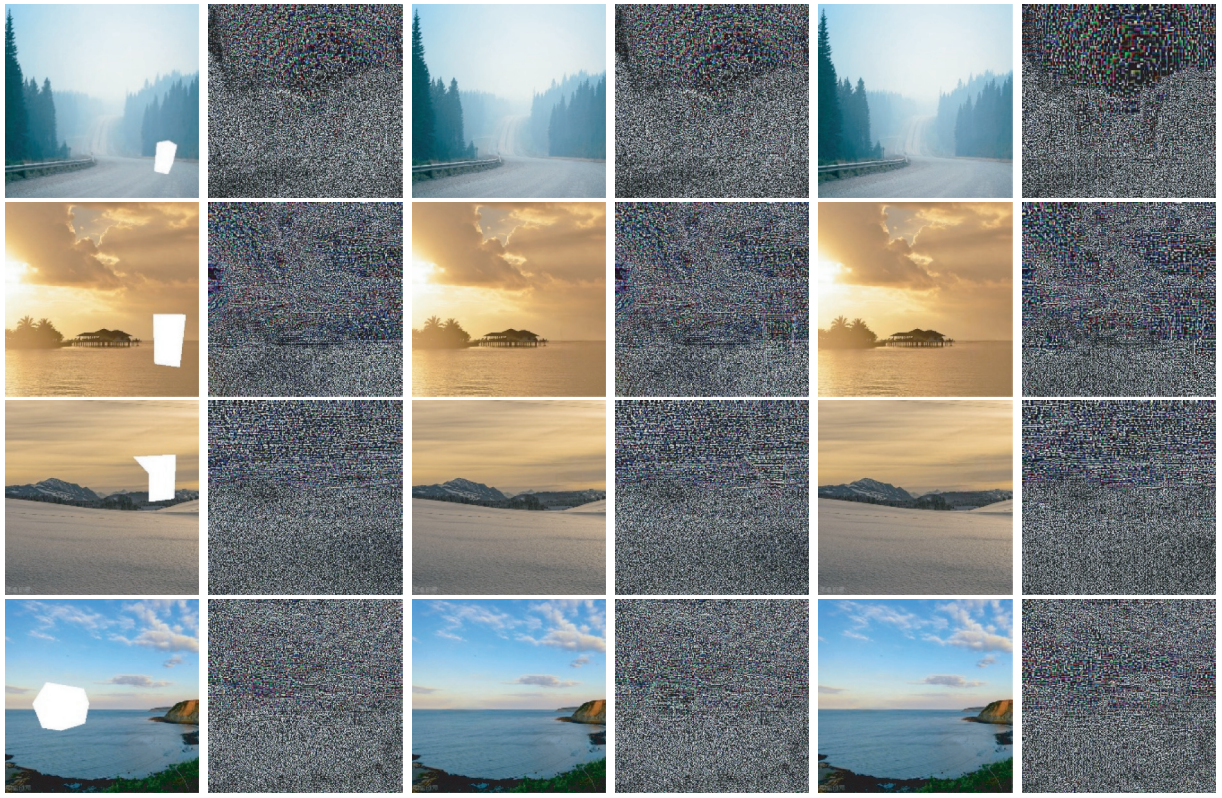


Figure 7: Examples from Today's Headlines. (a) Original image; (b) ground truth; (c) images inpainted via DFNet; (d) the pattern noise of (c); (e) images restored via our method; and (f) the pattern noise of (e).

Table 1: The percentage of protected areas in the whole image(t) and the total number of pixels in the protected area(p), $I1$, $I2$, $I3$, and $I4$ represent the four pictures in Figure 7, respectively.

Image	$I1$	$I2$	$I3$	$I4$
T	1.52%	3.99%	3.51%	5.53%
P	3985	10459	9201	14497

smoother edge where the structure size is 10 for the dilation operation and 5 for the erosion operation.

To evaluate the performance of image dual-inpainting against detection and localization, we adopt F1-score, peak signal-to-noise ratio (PSNR), and mean square error (MSE) objective indicators to evaluate the inpainting results:

$$\mathrm{F1} = \frac{2\mathrm{TP}}{(2\mathrm{TP} + \mathrm{FN} + \mathrm{FP})}, \tag{14}$$

where TP (true positive), FN (false negative), and FP (false positive) stand for the number of detected inpainted pixels, undetected inpainted pixels, and wrongly detected untouched pixels, respectively:

$$\begin{aligned} \mathrm{PSNR} &= 10 \times \log_{10} \frac{255^2 * \mathrm{MN}}{\sum_{i=1}^{M} \sum_{j=1}^{N} [B(i,j) - A(i,j)]^2}, \\ \mathrm{MSE} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [B(i,j) - A(i,j)]^2, \end{aligned} \tag{15}$$

where $A(i, j)$ and $B(i, j)$ are the original image and the inpainted image, respectively.

4.3. Reversibility Analysis. In this section, we show that our privacy protection method is effective during communication or sharing. Meanwhile, our method is fully reversible, which enables data to be extracted when it reaches the recipient side.

In Figure 9, we show five sets of comparisons between the recovered images and the original images. The first two of which are from the Today's Headlines database and the last three from the UCID database. In the prerecovery and embedding image operations, there is no damage or tampering to the regions other than the region to be protected. Therefore, under the condition of having the pixel values and coordinates of the region to be protected, the original images can be recovered.

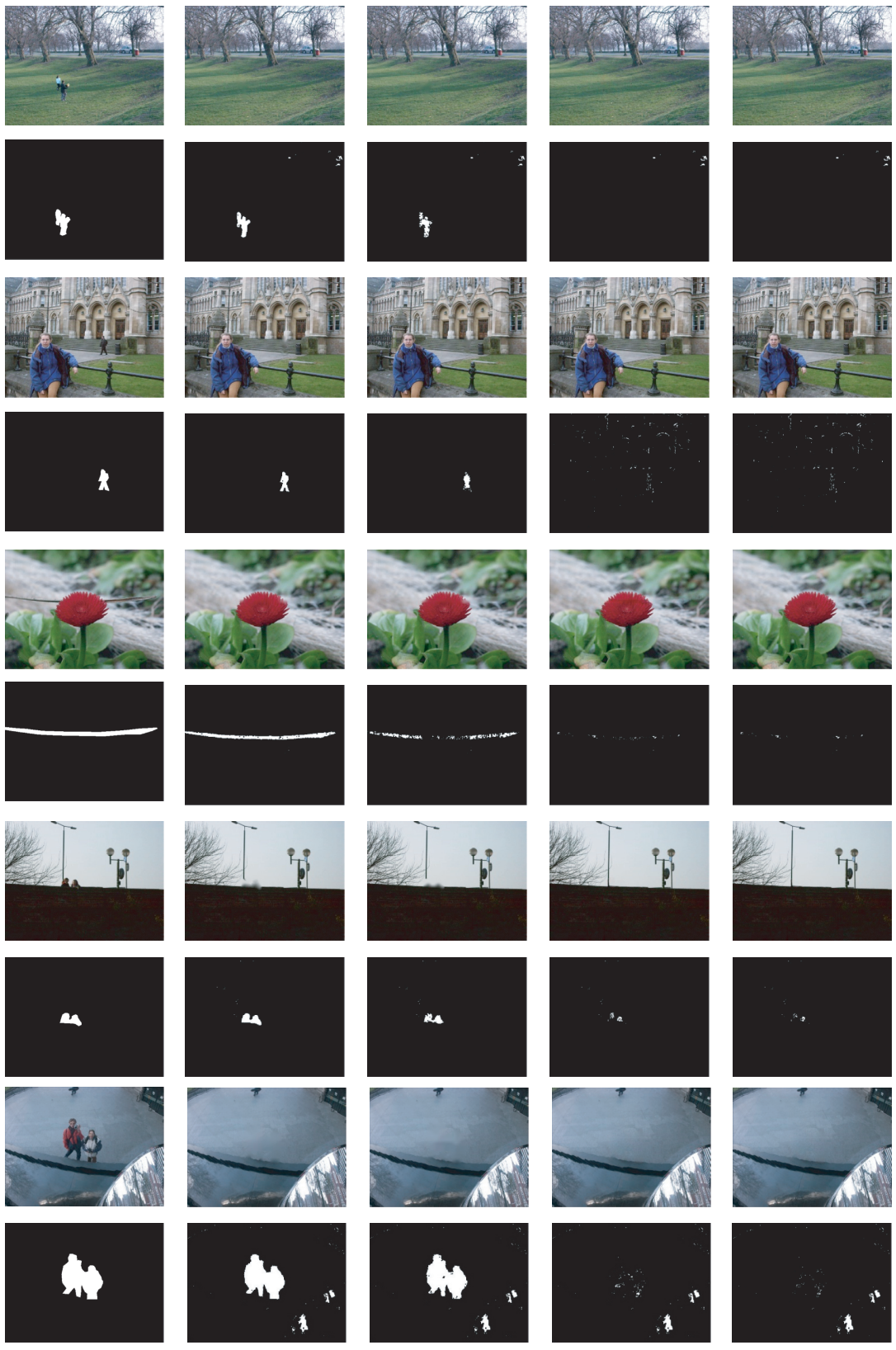


FIGURE 8: Examples from the UCID database. Rows 1, 3, 5, 7, and 9: from left to right, the first image is the original image, and the second to the fifth images represent the inpainted image by references [16, 48] and our method, respectively. Rows 2, 4, 6, 8, 10: from left to right, the first image is ground truth, and the second to the fifth images represent the localization result calculated by forensic algorithm 2.

Table 2: The percentage of protected areas in the whole image (t) and the total number of pixels in the protected area (p), $M1$, $M2$, $M3$, $M4$, and $M5$ represent the four pictures in Figure 8, respectively.

Image	$M1$	$M2$	$M3$	$M4$	$M5$
T	1.13%	0.77%	3.78%	0.87%	6.03%
P	2219	1513	7423	1707	11853

Table 3: F1-scores obtained on the UCID database for different inpainting algorithms and images.

Algorithm	Edge-oriented	Delaunay-oriented	DFNet	Dual-inpainting
$M1$	0.6948	0.8311	0.0045	0.0001
$M2$	0.6521	0.8512	0.0861	0.0040
$M3$	0.5242	0.8486	0.0981	0.0550
$M4$	0.7662	0.8992	0.2808	0.1762
$M5$	0.8945	0.9025	0.1297	0.0572

Figure 9: Examples for reversibility analysis. (a) Original images and (b) recovered images.

5. Conclusion

Currently, most of the privacy protection methods only focus on visual quality, while the real protection needs to be considered from the perspective of image security analysis. We propose a reversible privacy protection scheme using image dual-inpainting and data hiding, in which the original image can be perfectly recovered. Experimental results show that after the inpainting of the image with the removal of the area to be protected by the dual-inpainting algorithm, antiforensics for the two current methods for target removal forensics can be achieved. The later embedding and extraction of the protected region also achieve an effective combination of the two research directions of antiforensics and steganography. In addition, reversible privacy protection not only effectively stops snooping but also guarantees that the original image can be recovered when needed.

Data Availability

In our experiments, we use the free user-shared image dataset provided by Today's Headlines, which contains a large number of people landscapes and various life images. We also use the UCID database.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China (U20B2051).

References

[1] L. Yuan and T. Ebrahimi, "Image transmorphing with JPEG," in *Proceedings of the IEEE International Conference On Image Processing (ICIP)*, pp. 3956–3960, Quebec, Canada, September 2015.

[2] J. Yu, Z. Lin, J. Yang, and X. Shen, "Generative image inpainting with contextual attention," in *Proceedings of the CVF Conference on Computer Vision and Pattern Recognition CVPR*, Piscataway, NJ, USA, June 2018.

[3] P. Akyazi and P. Frossard, "Graph-based inpainting of disocclusion holes for zooming in 3d scenes," in *Proceedings of European Signal Processing Conference (EUSIPCO)*, Dublin, Ireland, September 2018.

[4] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," in *Proceedings of the 9th IEEE*

International Conference on Computer Vision (ICCV), pp. 305–312, Nice, France, October 2003.

[5] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 355–362, Kauai, HI, USA, December 2001.

[6] D. Tschumperle and R. Deriche, "Vector-valued image regularization with pdes: a common framework for different applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 506–517, 2005.

[7] M. Ghoniem, Y. Chahir, and A. Elmoataz, "Geometric and Texture Inpainting Based on Discrete Regularization on Graphs," in *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP)*, pp. 1349–1352, IEEE, Caen, France, 2009.

[8] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2002.

[9] S. Korman and S. Avidan, "Coherency sensitive hashing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1099–1112, 2016.

[10] O. Meur, J. Gautier, and C. Guillemot, "Examplar-based inpainting based on local geometry," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP)*, pp. 3401–3404, IEEE, Brussels, Belgium, October 2011.

[11] K. He and J. Sun, "Statistics of patch offsets for image completion," in *Computer Vision–ECCV*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Springer, Berlin, Heidelberg, pp. 16–29, 2012.

[12] P. Buyssens, M. Daisy, D. Tschumperlé, and O Lezoray, "Exemplar-based inpainting: technical review and new heuristics for better geometric reconstructions," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 24, no. 6, pp. 1809–1824, 2015.

[13] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," *Advances in Neural Data Processing Systems*, pp. 341–349, 2012.

[14] S. Iizuka, E. Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 107, 2017.

[15] R. Yeh, C. Chen, T. Lim, and A. G. Schwing, M. Hasegawa-Johnson, M. N. Do, Semantic image inpainting with deep generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 6882–6890, Honolulu, HI, USA, 2017.

[16] X. Hong, P. Xiong, and R. Ji, "Deep fusion network for image completion," 2019, http://arxiv.org/abs/1904.08060.

[17] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210, Paris, France, October 2014.

[18] G. Liu, F. Reda, K. Shih et al., "Image inpainting for irregular holes using partial convolutions," 2018, http://arxiv.org/abs/1804.07723.

[19] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edge connect: generative image inpainting with adversarial edge learning," 2019, http://arxiv.org/abs/1901.00212.

[20] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," *Computer Vision and Pattern Recognition (CVPR)*, http://arxiv.org/abs/1604.07379, 2016.

[21] Z. Lin, X. Liu, Q. Huang et al., "Contextual-based image inpainting: Infer, match, and translate," 2018, http://arxiv.org/abs/1711.08590.

[22] C. Luo, W. Zuo, M. Wang, Z. Hu, and H. Zhang, "Semantic image inpainting with progressive generative networks," *ACM Multimedia*, pp. 1937–1947, 2018.

[23] T. Yian, L. Alexander, G. Schwing et al., "Semantic image inpainting with deep generative models," 2018, http://arxiv.org/abs/1607.07539.

[24] M. Bertalmio, G. Sapiro, V. Caselles et al., "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 417–424, ACM Press/AddisonWesley Publishing Co, Minneapolis, MN, USA, July 2000.

[25] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, 2003.

[26] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[27] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.

[28] S. Kouider, M. Chaumont, and W. Puech, "Adaptive steganography by oracle (ASO)," in *Proceedings of the IEEE International Conference On Multimedia and Expo*, pp. 1–6, San Jose, CA, USA, July 2013.

[29] Y. Pan, J. Ni, and W. Su, "Improved uniform embedding for efficient JPEG steganography," in *Proceedings of the 2016 International Conference on Cloud Computing and Security*, pp. 125–133, Nanjing, China, June, July 2016.

[30] L. J. Guo, J. Q. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Trans. Data Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.

[31] Q. Wei, Z. Yin, Z. Wang et al., "Distortion function based on residual blocks for JPEG steganography," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 17875–17888, 2018.

[32] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[33] V. Holub and J. Fridrich, "Random projections of residuals for digital image steganalysis," *IEEE Transactions on Data Forensics and Security*, vol. 8, no. 12, pp. 1996–2006, 2013.

[34] T. Denemark, V. Sedighi, V. Holub et al., "Selection-channel-aware rich Mmodel for steganalysis of digital images," in *Proceedings of the IEEE International Workshop on data Forensics and Security*, pp. 48–53, Atlanta, GA, USA, December 2014.

[35] V. Holub and J. Fridrich, "Phase-aware projection model for steganalysis of JPEG images," *SPIE, media watermarking, security, and Forensics*, vol. 9409, pp. 94090T–940911, 2015.

[36] M. Chen, V. Sedighi, M. Boroumand et al., "JPEG-phase-aware convolutional neural network for steganalysis of JPEG images," in *in Proceedings of the 5th ACM Workshop On Data Hiding and Multimedia Security*, pp. 75–84, Philadelphia, PA, USA, June 2017.

[37] G. Xu, "Deep convolutional neural network to detect J-UNIWARD," in *Proceedings of the 5th ACM Workshop On Data Hiding and Multimedia Security*, pp. 67–73, Philadelphia, PA, USA, June 2017.

[38] J. Zeng, S. Tan, B. Li et al., "Large-scale JPEG Image steganalysis using hybrid deep-learning framework," *IEEE Trans. data Forensics and Security*, vol. 13, no. 5, pp. 1200–1214, 2018.

[39] B. Li, Z. Li, S. Zhou et al., "New steganalytic features for spatial image steganography based on derivative filters and threshold lbp operator," *IEEE Trans. data Forensics and Security*, vol. 13, no. 5, pp. 1242–1257, 2018.

[40] J. Kodovsky and J. Fridrich, "Steganalysis of JPEG images using rich models," *International Society for Optics and Photonics*, vol. 8303, 2012.

[41] X. F. Song, F. L. Liu, C. F. Yang et al., "Steganalysis of adaptive JPEG steganography using 2D gabor filters," in *Proceedings of the 3rd ACM Workshop on Data Hiding and Multimedia Security*, pp. 15–23, New York, NY, USA, June 2015.

[42] C. Xia, Q. Guan, X. Zhao et al., "Improving GFR steganalysis features by using gabor symmetry and weighted histograms," in *Proceedings of the 5th ACM Workshop on Data Hiding and Multimedia Security*, pp. 55–66, Philadelphia, PA, USA, June 2017.

[43] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Data Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2014.

[44] F. Li, X. Zhang, B. Chen, and G. Feng, "JPEG steganalysis with high-dimensional features and bayesian ensemble classifier," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 233–236, 2013.

[45] J. Lukas, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," *Proceedings of SPIE Electronic Imaging*, vol. 6072, pp. 362–372, 2006.

[46] H. Li, W. Luo, and J. Huang, "Localization of diffusion-based inpainting in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3050–3064, 2017.

[47] M. Chen, J. Fridrich, and M. Goljan, "Digital imaging sensor identification (further study)," *Proceedings of SPIE Electronic Imaging*, vol. 6505, Article ID 65050P, 2007.

[48] G'MIC, "GREYC's magic for image computing," http://gmic.eu.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6652727, 16 pages
https://doi.org/10.1155/2021/6652727

Research Article

Towards Face Presentation Attack Detection Based on Residual Color Texture Representation

Yuting Du,[1] **Tong Qiao**,[1,2] **Ming Xu**,[1] **and Ning Zheng**[1]

[1]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China*
[2]*Zhengzhou Science and Technology Institute, Zhengzhou 450001, China*

Correspondence should be addressed to Ming Xu; mxu@hdu.edu.cn

Received 23 December 2020; Revised 7 February 2021; Accepted 27 February 2021; Published 16 March 2021

Academic Editor: Beijing Chen

Most existing face authentication systems have limitations when facing the challenge raised by presentation attacks, which probably leads to some dangerous activities when using facial unlocking for smart device, facial access to control system, and face scan payment. Accordingly, as a security guarantee to prevent the face authentication from being attacked, the study of face presentation attack detection is developed in this community. In this work, a face presentation attack detector is designed based on residual color texture representation (RCTR). Existing methods lack of effective data preprocessing, and we propose to adopt DW-filter for obtaining residual image, which can effectively improve the detection efficiency. Subsequently, powerful CM texture descriptor is introduced, which performs better than widely used descriptors such as LBP or LPQ. Additionally, representative texture features are extracted from not only RGB space but also more discriminative color spaces such as HSV, YCbCr, and CIE 1976 L∗a∗b (LAB). Meanwhile, the RCTR is fed into the well-designed classifier. Specifically, we compare and analyze the performance of advanced classifiers, among which an ensemble classifier based on a probabilistic voting decision is our optimal choice. Extensive experimental results empirically verify the proposed face presentation attack detector's superior performance both in the cases of intradataset and interdataset (mismatched training-testing samples) evaluation.

1. Introduction

Face authentication technology is widely deployed in real life. However, most existing face authentication systems are vulnerable to presentation attacks (PAs). For clarity, the bona fide and the PA samples are illustrated in Figure 1. Generally speaking, compared with the bona fide faces, the PA samples are generated by presenting spoofing artifacts toward face authentication system.

Since deep learning (DL) shows its outstanding potential in resolving image classification tasks, numerous DL-based methods are proposed by utilizing deep networks to extract deep features from images such as [1–6]. It is known that DL-based methods can achieve excellent performance when obtaining enough training data, but in face presentation attack detection task, the diversity and amount of training data is often not satisfied, and overfitting is also a vexing problem. To enable a presentation attack detection system be applicable to various environment, domain adaptation [7] manner is explored to resolve the overfitting. Moreover, similar to the two-stream strategy utilized in copy-move forgery [8], there is also two-stream-based method for learning fusion features to resolve PA detection problem [9].

Compared with DL-based methods, hand-crafted feature-based methods pay more attention to extract predefined specific patterns, which are more explainable. We can mainly divide these techniques into three categories: motion-related cue [10–13], image quality [14–16], and texture-based analysis [17–24]. Motion-related cue-based methods are highly robust in some specific cases, but the generalization ability is not satisfactory. Image quality artifact-based methods are not robust enough and computationally complex. By contrary, the performances of texture-based analysis methods are more preferable.

It is known that, in image forensics field, effective data preprocessing can obviously improve the algorithm's performance. For example, in [25], a Laplacian filter is used for input enhancement. And, in [26], the Schmid filter is used to

Figure 1: Cropped example face images extracted from the FASD. From the left to the right: genuine face, print attack, and replay attack, respectively.

enhance texture information. However, to the best of our best knowledge, in face antispoofing field, there is still a lack of effective measure of preprocessing. In this work, a novel perspective is introduced that nuisance noise can interference extracting representative features from face images, and we introduce a wavelet-based filter to preprocess the original image, which can successfully make the model perform better. The assumption is inspired by that in the process of using image sensors such as CCD and CMOS to capture images; due to the influence of the sensor material properties, electronic components, and circuit structure, various noises will be introduced, such as Gaussian noise, salt and pepper noise, speckle noise, shot noise, and white noise. However, such noise does not seem to be helpful for face PA detection. Therefore, analytical experiments are conducted to investigate how the difference changed between the bona fide and the PA faces by using residual (noise-free) images instead of original images (see Tables 1 and 2). For more intuitive, discrete wavelet transform is applied to conduct a similarity-based analysis, which is specifically described in Figure 2. By applying a discrete wavelet filtering (DW-filtering), compared with the original image, the similarity between the bona fide face and the PA from residual image is the lowest, meaning that the features extracted from both bona fide face and PA from residual image can be more discriminative than the others. Besides, since the effectiveness of texture analysis in color spaces is verified in [21], which utilizes two local texture descriptors (CoALBP and LPQ) and one classifier such as SVM, an assumption can be further drawn that if a high efficient classifier such as ensemble one, together with more discriminative descriptors for color residual texture representation is adopted, the performance of the detector can be further improved. The contributions of this paper can be summarized as follows:

In RGB space, luminance and chrominance information cannot be effectively characterized. However, the concerning color information stored in different channels is of importance for generating more discriminative color features. Therefore, many works consider extracting features by using HSV, YCbCr space, or fusion of them. Nevertheless, for the differentiability of various color channels and the best combination of them, there is still a lack of deep exploration. In the following sections, we have conducted extensive analytical experiments and in-depth discussions on this issue. A total of four color spaces are taken into account, namely, RGB, HSV, YCbCr, and LAB.

Existing methods lack of effective data preprocessing. In fact, an effective preprocessing operation can significantly improve the performance of the detector. In the preprocessing stage of this work, we propose to adopt DW-filter for obtaining residual image, which effectively

Table 1: The Chi-square distances (i.e., d_{χ^2}) for different color channels in original images. Larger d_{χ^2} value is indicated in bold compared to Table 2.

Color channel	FASD	RAD	MSU
RGB-R	154.0	115.1	94.5
RGB-G	278.3	120.7	103.6
RGB-B	323.3	130.3	**114.1**
HSV-H	1062.7	766.0	717.0
HSV-S	404.4	**242.4**	304.7
HSV-V	188.1	115.4	**100.8**
YCbCr-Y	**253.6**	198.2	**103.8**
YCbCr-Cb	191.6	311.4	141.3
YCbCr-Cr	147.5	206.3	127.5
LAB-L	235.6	**120.1**	102.3
LAB-A	151.2	177.7	146.8
LAB-B	182.7	212.3	151.8

Table 2: The Chi-square distances (i.e., d_{χ^2}) for different color channels in residual images. Larger d_{χ^2} value is indicated in bold compared to Table 1.

Color channel	FASD	RAD	MSU
RGB-R	**165.0**	**118.6**	**99.3**
RGB-G	**289.4**	**122.9**	106.6
RGB-B	**328.9**	**130.6**	**114.4**
HSV-H	**1123.4**	**942.6**	**818.7**
HSV-S	**482.2**	239.7	**307.2**
HSV-V	**203.7**	**126.7**	99.4
YCbCr-Y	253.1	**199.3**	103.6
YCbCr-Cb	**200.7**	**313.0**	**253.4**
YCbCr-Cr	**246.9**	**212.8**	**250.1**
LAB-L	**239.6**	120.0	**107.4**
LAB-A	**418.7**	**325.3**	**287.5**
LAB-B	**198.7**	**213.8**	**262.6**

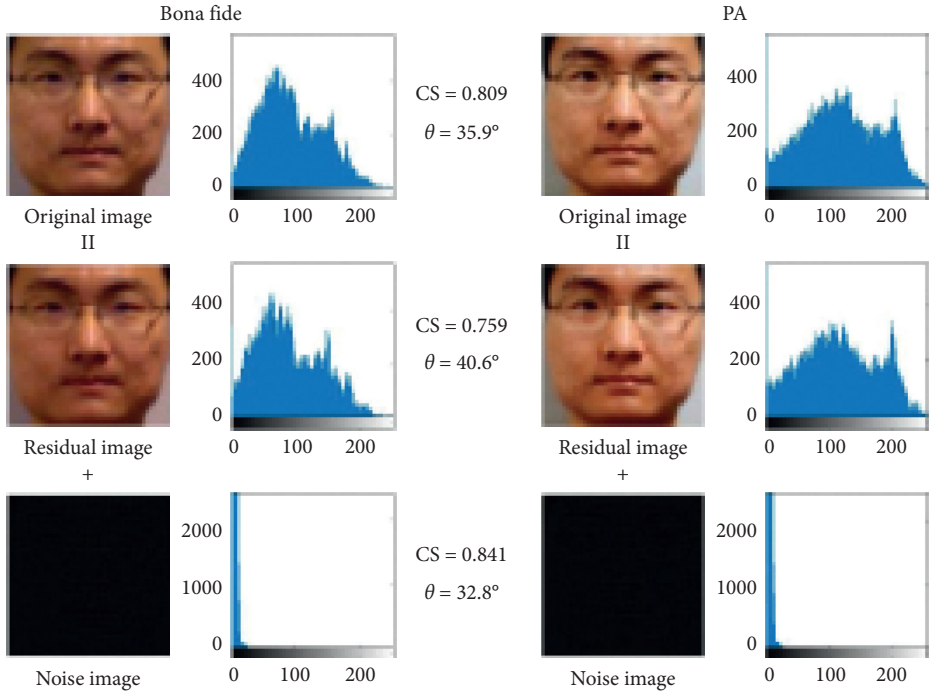


Figure 2: The CS of the bona fide and the PA samples. Residual image is obtained by DW-filter, where "Residual Image = Original Image – Noise Image." θ is the angle corresponding to the CS, which is inversely proportional to the image similarity.

alleviates the interference caused by nuisance noise while retaining valuable information for presentation attack detection. Meanwhile, extensive analytical experiments are conducted to further verify the effectiveness of the utilization of the residual image.

Among texture-based arts, the optimal choice of the descriptor is not well investigated. Thus, we mainly describe and analyze five widely used texture descriptors, namely, the CM, LBP, CoALBP, LPQ, and BSIF. According to the experimental results, the CM feature outperforms others in color spaces. Accordingly, our proposed RCTR is constructed relying on the powerful CM feature extracted in color channels of the residual image.

Most existing hand-crafted feature-based methods use single classifier such as SVM, which cannot always perform well. In this work, the performance of three widely adopted classifiers is well investigated, including LDA, SVM, and XGBoost. And, an ensemble classifier based on the probabilistic voting decision is designed. In the case of inter- or intradataset testing, our RCTR-based detector that employs the ensemble classifier shows satisfactory performance.

The remainder of this paper is organized as follows. In Section 2, the related works are presented. In Section 3, our proposed approach is described in detail. Three benchmark face presentation attack datasets are introduced in Section 4. In Section 5, we provide comprehensive experimental results and analysis. Last but not least, concluding remarks are drawn in Section 6.

2. Related Works

To address the challenge introduced by face presentation attacks, many presentation attack detection techniques have been proposed, which can be arbitrarily formulated into two categories: deep learning-based methods and hand-crafted feature-based methods. The specific overview is extended as follows.

2.1. Deep Learning-Based Methods. Deep learning can achieve promising results in the field of computer vision, which is also very effective when tackling face presentation attack detection task. In [2], CNN is utilized to extract deep features, and SVM is employed instead of fully connected layers for classification. Atoum et al. [27] present a two-stream network architecture to learn patch-based and depth-based features, and the classification result is determined by the fusion scores of both two streams. Rather than merely extracting spatial feature, a 3D-CNN structure is proposed in [6] to exploit the spatial-temporal features, which can capture more visual cues that are indeed useful for face presentation attack detection task. Meanwhile, a domain generalization regularization approach is incorporated for further enhancing the model generalization ability. Previous deep learning-based face presentation attack detection approaches formulate the task as a binary classification problem. Liu et al. [28] emphasize the importance of auxiliary supervision. Specifically, a CNN-RNN architecture is proposed to utilize depth map information and rPPG

(remote Photoplethysmography) signs, which can both exploit spoof patterns across spatial and temporal domains. In [29], an augmented dataset is collected in a specific image synthesis way, which can further improve the robustness of the model.

DL-based methods usually have superior classification accuracy when training and testing samples belong to similar scenes. However, due to heavily relying on a large-scale well-designed dataset, the performance of many DL-based methods will sharply decrease when dealing with mismatched training and testing samples. Poor generalizability is more serious in earlier DL-based methods [3]. And, in recent works [30–32], such defect is significantly improved.

2.2. Hand-Crafted Feature-Based Methods. The methodologies in this category mainly rely on defining specific patterns in advance for extracting discriminative features. Given that face presentation attack samples tend to be static, motion analysis-based schemes are developed, such as eye blinking [10], mouth movement [11], and just holistic face region movement analysis [13]. In general, the biometric information can be successfully obtained by analyzing the optical flow in specific areas of the image. Although the motion-related cue-based methods perform well when dealing with print attack, they may fail to complete the task of replay attack detection, where the motion-related cue for presentation attack detection can be easily inferred. Besides, image quality also can be a vital measurement toward face presentation attack detection. Galbally et al. [15] propose to resolve presentation attacks by calculating prominent factors among 25 image quality metrics. Di et al. [16] introduce an image distortion analysis countermeasure by evaluating four presentation attack patterns: specular reflection caused by display device, image blurriness, chromatic distribution variation, and poor color diversity. However, due to heavy computation, these methods are not efficient enough. It is worth mentioning that although various hand-crafted feature-based methods are proposed, there is still a lack of effective preprocessing to further improve the performance of the detector.

In addition, the effectiveness of texture descriptors in resolving face presentation attack problems has been verified by some works. For instance, multiscale local binary pattern (MSLBP) descriptor is designed for face presentation attack detection in [17], and a novel facial texture representation is introduced by using the spatial and temporal extensions of the local binary pattern (LBP-TOP) [33]. Besides, it is worth noting that Boulkenafet et al. [21] present a novel and appealing face presentation attack countermeasure by using color texture features, based on the assumption that gray-scale images are often used to display illuminance information, while more helpful color information are discarded. In fact, the RGB image cannot completely separate the luminance and chrominance signals while color texture features can be well extracted from HSV and YCbCr spaces. It is well-known that print attacks utilize photos of legitimate users to fool the face recognition system, while replay attacks often utilize electronic device such as mobile or tablet. Due to the restriction of the limited color gamut, the fake faces presented on the display device often show color degradation.

The effectiveness of texture descriptors and color space features in resolving face presentation attack detection task are verified. However, the discriminative features are generally extracted from original pixels in spatial domain, which are more or less impacted by nuisance noise introduced during image capturing. Besides, the study of combining various texture features within different color spaces to achieve the optimal color texture features still remains open in this community. Additionally, to the best of our knowledge, one single classifier cannot always bring optimal prediction results, compared with the powerful ensemble classifier. In virtue of our theoretical and empirical analysis in this paper, those negative factors can lead to bad detection results when training samples are mismatched with testing samples. To address those challenges, dependent of residual image via DW-filtering, it is proposed to design a high efficient ensemble face presentation attack detector based on RCTR.

3. Proposed Method

In this section, we specifically present the RCTR-based face presentation attack detection method. For clarity, let us first illustrate the overall framework in Figure 3. First of all, face alignment is applied to calibrate the face region from full frame. Next, a DW-filter is utilized to process the high-frequency coefficients in order to obtain more discriminative residual image. Then, the residual image is transformed from RGB into another color space (e.g., YCbCr). Subsequently, texture descriptor is applied to extract rich texture information, in which a comprehensive representation is constructed by combining optimal descriptor feature vectors, namely, RCTR. Finally, we design an ensemble classifier with the effective strategy of probabilistic voting decision, which can successfully complete the task of face presentation attack detection.

3.1. Analysis of Color Space. The samples of PA face are passed through different cameras or printing mediums (such as photos, mobiles, and tablets), so they can actually be called a kind of recaptured image. Therefore, we can assume that when generating PA samples, inherent differences in color channel between the bona fide and PA images are introduced during the recapturing process. This is due to the color gamut caused by the display medium and other defects in color reproduction, such as display imperfection, or noise signals. Compared to bona fide face samples, the camera used to capture the target face photos also brings about imperfect color reproduction. Thus, it is reasonable to use color images instead of gray-scale images for face presentation attack analysis. RGB is widely used, but other color spaces are also worthy of attention. Because color component and luminance component cannot be perfectly characterized in RGB space, it can be better discriminated in other space such as HSV. There are various color spaces

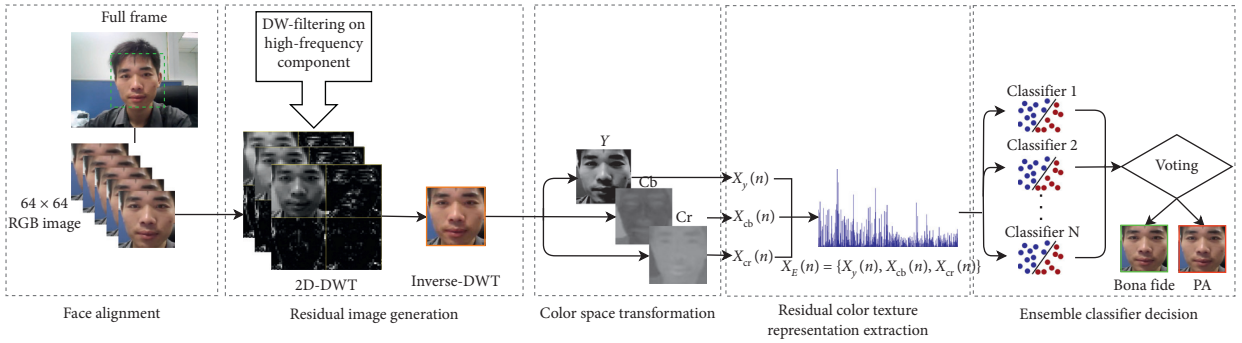


Figure 3: A pipeline of our proposed face presentation attack detection method, and YCbCr space is used here as an instance.

which have been proposed, and we consider analyzing bona fide face and PA images in four different color spaces: RGB, HSV, YCbCr, and LAB. Therefore, a metric is designed to examine which color space or channel is more distinguishable and details of the metric are as follows.

Firstly, for given image I with the size $l \times l$, the correlation coefficient between the adjacent pixels in each color component I^c ($c \in \{R, G, B_1, H, S, V, Y, \mathrm{Cb}, \mathrm{Cr}, L, A, B_2\}$) are calculated, which can be formulated as

$$m_i^c = \frac{\sum_{j=0}^{l-1}\sum_{k=0}^{l-1}\left(I_{j,k}^c - \overline{I}^c\right)\left(I_{j+1,k+1}^c - \overline{I}^c\right)}{\sqrt{\sum_{j=0}^{l-1}\sum_{k=0}^{l-1}\left(I_{j,k}^c - \overline{I}^c\right)^2 \sum_{j=0}^{l-1}\sum_{k=0}^{l-1}\left(I_{j+1,k+1}^c - \overline{I}^c\right)^2}}, \tag{1}$$

where $\overline{I}^c$ represents the mean pixel value of I^c. For simplicity, we only consider the diagonally adjacent pixels. It can be drawn that the larger the m_i^c, the higher the relevance between the adjacent pixel values of given $\overline{I}^c$.

Subsequently, for given bona fide face image set, m_i^c of each image is calculated and the corresponding histogram H_{bf}^c can be constructed. And, for the given PA image set, the histogram H_{pa}^c can be obtained in the same way. Then, Chi-square distance is used to measure the similarity between the two histograms, which can be formulated as

$$d_{\chi^2}\left(H_{bf}^c, H_{pa}^c\right) = \sum_b \frac{\left(H_{bf}^c(b) - H_{pa}^c(b)\right)^2}{H_{bf}^c(b) + H_{pa}^c(b)}, \tag{2}$$

where b is the bin index of the histogram. Similarly, the larger the d_{χ^2}, more significant the difference between the bona fide images and the PA images.

To evaluate the disparities between the bona fide face and the PA face in each color component, 10000 bona fide face images and 10000 PA face images are extracted from FASD, RAD, and MSU dataset, respectively, to perform analytical experiments. As introduced above, $m_i^c s$ of all images is calculated, the corresponding histograms H_{bf}^c and H_{pa}^c are obtained, and their $d_{\chi^2} s$ are also calculated, which can be seen in Table 1. Throughout the results of the three datasets, the d_{χ^2} values in RGB space are relatively stable (the maximum is 323.3, and the minimum is 94.5); this is because color components and luminance components are not well separated. As for the results on FASD, it can be observed that when using H channel, the d_{χ^2} value is 1062.7, which is significantly larger than any other channel. And, the result of the S channel is 404.4, which is the second largest. As for the V channel, the d_{χ^2} value is relatively small. This is meaning that the bona fide faces and the PA images are more distinguishable in color components (i.e., H and S channel) than in luminance component (i.e., V channel). As for YCbCr and LAB spaces, the differences between color component and luminance component are not as obvious as in HSV space. Similar conclusions can also be drawn from the results of RAD and MSU dataset.

Besides, only conducting analytical experiments are not enough to predict the actual situation; thus, extensive experiments are conducted to further investigate the benefit of color spaces transforming for face presentation attack detection (see Figure 4, for details).

3.2. Generation of the Residual Image. Face presentation attacks are implemented by printing human faces on various display media, such as A4 paper, mobile, and tablet screen. Though bona fide or PA samples are presented toward face authentication system, the nuisance noise is unavoidably introduced during image capturing process. A reasonable assumption can be made that nuisance noise existing in the face image, including bona fide and PA samples, might more or less impact the effectiveness of presentation attack detection, while the features extracted from the residual face image are more discriminative than that of original face image. Therefore, we propose to apply DW-filter for residual image extraction. It is important to study whether applying DW-filtering preprocessing operation in our scheme is effective to suppress nuisance noise from face image and meanwhile helpful to learn color texture features for presentation attack detection. To visually verify our hypothesis, we conduct the face image similarity-based analysis (see Figure 2 for illustration). By applying DW-filter, we segment the original face image to residual and noise one. Meanwhile, the statistical histogram of the pixels of each image is used to evaluate the similarity between two classes of face images, which is measured by the CS (cosine similarity):

$$\mathrm{CS}(X, Y) = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=0}^{n=255} x_i \times y_i}{\sqrt{\sum_{i=0}^{n=255}(x_i)^2} \times \sqrt{\sum_{i=0}^{n=255}(y_i)^2}}, \tag{3}$$

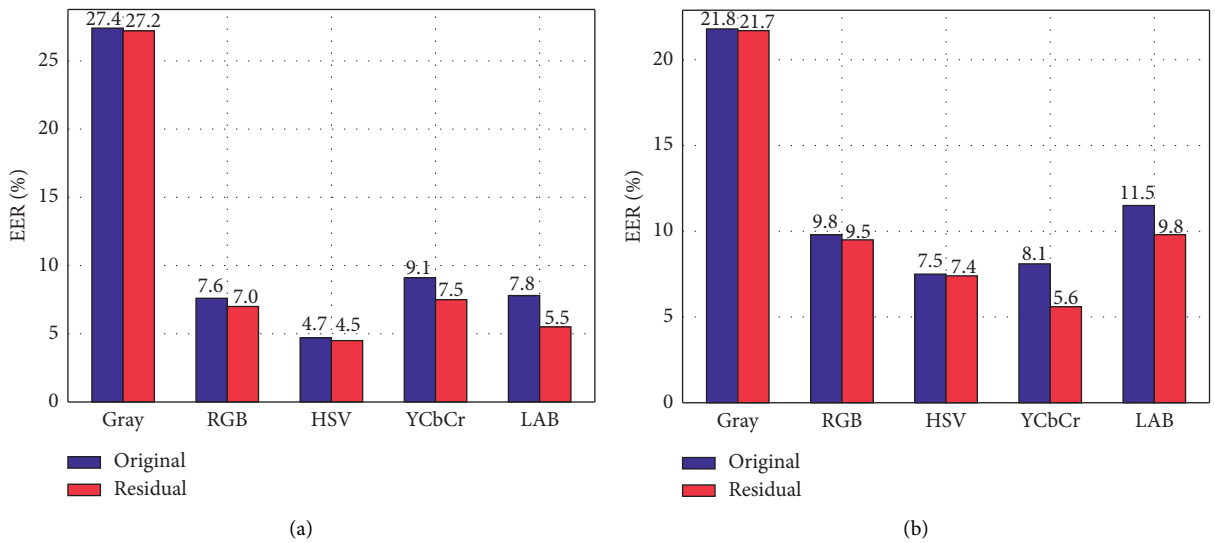


FIGURE 4: The EER results of the CM feature extracted in various color spaces from both original images and residual images. (a) FASD. (b) MSU.

where $\|\cdot\|$ denotes the 2-norm and x_i or y_i represents the frequency in ith gray level of histogram from compared images. In Figure 2, we can observe that the CS between the noise images of bona fide and PA faces is 0.841. Meanwhile, we also observe that the CS between original images is 0.809, larger than 0.759 from residual images. That is because the noise components in face images are filtered out, which makes the inherent defects introduced by presentation attack operation to be more discriminative. In addition, we can also notice that the CS of noise image is higher than that of the original images, which further proves the interference effect of nuisance noise.

To further verify the effectiveness of the use of residual image, similar analytic experiments following the settings in Section 3.1 are conducted; the only difference is that the residual image is used instead of the original image (see Table 2, for illustration). It can be observed that compared to Table 1, most d_{χ^2} values for residual images are generally larger than that for original images; only a few color channels show a slight decrease (all larger d_{χ^2} values are indicated in bold in the table). Specifically, when using original images on RAD, the d_{χ^2} value of H channel is 766.0, and this value is increased to 942.6 when using residual images. Furthermore, for residual images, color components become more distinguishable in YCbCr and LAB spaces. Specifically, the d_{χ^2} value of Cb channel for residual images is 246.9, while the counterpart for original images is 147.5. And, the d_{χ^2} value of A channel for residual images is 418.7, while the counterpart for original images is just 151.2.

Based on the above analysis, we can draw that the discrimination between the bona fide and the PA faces can be further enhanced by adopting residual image instead of the original one. That is undoubtedly beneficial for presentation attack detection. Thus, prior to feature extraction such as residual color texture representation in this paper, it can hold true that we first proceed the preprocessing by using an effective filter.

The proposed algorithm needs to preprocess an inquiry face image by filtering. DW-filter serves as a useful tool to preliminary acquire the residual image (see Figure 2 for instance). DW-filter has performed its powerful advantage at decomposing high and low frequencies [34]. The application of 2D-DWT in image processing is mainly to decompose the inquiry image through multiscale decomposition. A 2D-DWT process over an original image I with the size $l \times l$ can be formulated by

$$f_{2D-\mathrm{DWT}}(I) = \begin{bmatrix} I_{\mathrm{LL}} & I_{\mathrm{HL}} \\ I_{\mathrm{LH}} & I_{\mathrm{HH}} \end{bmatrix}, \tag{4}$$

where the original image I is decomposed into four sub-images: I_{LL}, I_{HL}, I_{LH}, and I_{HH} with the size $l/2 \times l/2$. I_{LL} corresponds to the approximation component (low frequency) of the image, while the remaining three I_{HL}, I_{LH}, and I_{HH} correspond to the horizontal detail component, vertical detail component, and diagonal detail component, respectively. As shown in Figure 2, when performing DWT filtering, the similarity between genuine face and fake face is reduced. In this case, the noise component is weakened after filtering, while the valuable information for presentation attack detection is preserved.

In particular, let us conduct DW-filtering proposed in [35], which can be formulated by

$$W_\lambda = \begin{cases} \mathrm{sgn}(|w| - \lambda), & |w| \geq \lambda, \\ 0, & |w| < \lambda, \end{cases} \tag{5}$$

where W represents the wavelet coefficients to be filtered, $\mathrm{sgn}(\cdot)$ is the sign function, and λ is the given threshold. In this work, we take the thresholding as a filter to preprocess face images. For instance, the sqtwolog threshold can be calculated by

$$\lambda = 2\sqrt{2\log(l)}. \tag{6}$$

Specifically, let us introduce the process of the DW-filtering based on 2 layer decomposition in three steps:

The widely adopted haar wavelet base is selected, and the given original face image is decomposed by applying MALLAT decomposition algorithm [34]. Accordingly, the wavelet coefficients of each layer are successfully obtained.

Based on the given threshold λ, the high-frequency components obtained by decomposing each layer are quantized, while the low-frequency component remains unchanged.

By means of MALLAT reconstruction algorithm, the low-frequency component of the 2nd layer after decomposition and the high-frequency components of each layer are reconstructed by inverse DWT, and finally, the residual face image by wavelet thresholding is generated.

3.3. Feature Extraction by Texture Descriptor. Based on the previous analysis, we decide to extract texture features from multiple color channels in residual images. It should be noted that color texture features are obtained by applying descriptors not only in gray-scale image but also in color channels. That is because the color image can provide more valuable information for presentation attack detection, which is beneficial to improve detector's robustness and accuracy. In this work, the co-occurrence matrix [36] is employed, which is widely used in image texture analysis. Moreover, widely adopted descriptors such as the LBP [37], LPQ [38], CoALBP [39], and BSIF [40] are also introduced. In this section, we mainly overview these descriptors.

3.3.1. CM. The co-occurrence matrix (CM) describes the distribution of intensity and information about the relative position of adjacent pixels in the image, which can measure the correlation among adjacent pixels and hence gather valuable information from recurrent micropatterns. Before calculating CM, for given image I, first-order differential operator is applied to suppress the image content, namely,

$$\widehat{I}(x, y) = I(x, y) - I(x, y + 1), \tag{7}$$

where (x, y) denotes the pixel coordinate and $\widehat{I}$ is the resulting image. It should be noted that only horizontal difference is considered here. As a result, the dynamic range of the image content is much narrower so that more reliable statistical description can be carried out. Subsequently, a truncating operation is conducted because there are too many distinct element values in the original image, which could result in huge dimension of the CM feature vector. The truncated image is calculated as follows:

$$T(x, y) = \begin{cases} \gamma, & \widehat{I}(x, y) \geq \gamma, \\ \widehat{I}(x, y), & -\gamma < \widehat{I}(x, y) < \gamma, \\ -\gamma, & \widehat{I}(x, y) \leq -\gamma, \end{cases} \tag{8}$$

where $\gamma > 0$ is the truncation threshold, and the result T is then used to compute the CM. Typically, a d order CM of the 2D array T can be obtained by

$$\begin{aligned} \mathrm{CM}(\theta_1, \theta_2, \ldots, \theta_d) = \frac{1}{N} \sum 1[&T(x, y) = \theta_1, T(x + \Delta x, y + \Delta y) = \theta_2, \ldots, \\ &T(x + (d-1)\Delta x, y + (d-1)\Delta y) = \theta_d], \end{aligned} \tag{9}$$

where $\theta_1, \theta_2, \ldots, \theta_d$ are the index, $1(\cdot)$ is the indicator function, N is the normalization factor, and Δx and Δy are the offsets. The effectiveness of the CM is validated in steganography detection [36] and face recognition [41]. However, in face presentation attack detection field, the use of the CM is not well explored.

3.3.2. LBP. The Local Binary Patterns (LBP) perform very well when depicting image structure information such as edges. The LBP is obtained via comparing each central pixel to its neighborhood one in the block, where the LBP features are described as a binary sequence, which can be formulated by

$$\mathrm{LBP}_{p,r}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, s(k) = \begin{cases} 1, & \text{if } k \geq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where g_c denotes the value at the central pixel coordinate (x_c, y_c), while g_p, $p \in \{0, 1, 2, \ldots, P-1\}$, represents the value of the neighboring pixel in the block, and r denotes the radius. For instance, when $r = 1$, P equals to 8. Then, the binary patterns are collected by statistical histograms to represent the image texture information. In general, high robustness toward luminance variation, rotation invariance, and low-computational complexity are the advantages of LBP descriptor. When a face image is tested, we cannot guarantee that it is correctly presented in front of a digital camera of presentation attack detector. Thus, the robustness of resisting rotation attack is crucial. However, the LBP feature contains only intensity relationships between adjacent pixels and lack of spatial relationship information, which raises the performance limitation.

3.3.3. CoALBP. For the sake of compensating the missing spatial relationship information in the LBP features, the co-occurrence of adjacent local binary patterns (CoALBP) is proposed in [39]. In this method, two simplified LBP configurations, denoted as LBP (+) and LBP (×), are

introduced. LBP (+) considers two horizontal and two vertical pixels, while LBP(×) considers four diagonal pixels. Before calculating the co-occurrence information of LBPs, each LBP is transformed to its vector form by using Kronecker delta:

$$V_i(B) = \delta_{i,l(\mathrm{lbp}(B))},$$
$$\delta_{a,b} = \begin{cases} 1, & \text{if } a \neq b, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $i \in \{0, 1, 2, \ldots, n-1\}$, n is the number of neighbor pixels, B is the position vector in an image intensity I, and $l(\mathrm{lbp}(\cdot))$ denotes a decimal number label of $\mathrm{lbp}(\cdot)$. For example, if the given binary sequence is 0010, the corresponding label is 2. If all possible LBP label values are in the range $[0, N]$ $(N = 2^n)$, an $N \times N$ autocorrelation feature matrix H can be calculated by

$$H(D) = \sum_{B \in I} V(B) V(B + D)^T, \tag{12}$$

where D is the displacement vector between two LBPs. Four displacement vector are set as follows: $D_1 = (\Delta B, 0)^T$, $D_2 = (\Delta B, \Delta B)^T$, $D_3 = (0, \Delta B)^T$, and $D_4 = (-\Delta B, \Delta B)^T$, which correspond to the direction of 0°, 45°, 90°, and 135°. At last, the four resulting matrices are concatenated to form the final CoALBP feature. It should be noted that although the CoALBP descriptor preserves more spatial information than LBP, the high dimension of CoALBP feature increases the computation cost of training a classifier.

3.3.4. LPQ. The local phase quantization (LPQ) is originally proposed by [38] to solve the problem of inaccurate classification caused by image blurring. The LPQ descriptor uses local phase information, which is extracted through the short time Fourier transform (STFT) based on the square region. The resulting STFT within the region of $g \times g$ surrounding the central pixel position m from the given image is defined by

$$F_{\mathbf{u}}(m) = w_u^T \mathbf{x}, \tag{13}$$

where w_u represents the basis vector of the 2D discrete Fourier transform at the frequency u and $\mathbf{x}$ denotes the vector containing all pixels in the region of $l \times l$. Specifically, the Fourier complex coefficients are calculated at four 2D frequencies: $u_0 = (s, 0)^T$, $u_1 = (s, s)^T$, $u_2 = (0, s)^T$, and $u_3 = (s, s)^T$, where s is a small scalar and $s \ll 1$. Then, the basic LPQ feature can be formulated by

$$Q(m) = [\mathrm{RC}\{Q^c(m)\}, \mathrm{IC}\{Q^c(m)\}],$$
$$Q^c(m) = \left\{F_{u_0}(m), F_{u_1}(m), F_{u_2}(m), F_{u_3}(m)\right\}, \tag{14}$$

where RC{·} and IC{·} mean to return the real component and imaginary component of a complex number, respectively. In addition, each element of $Q(m)$ is quantized as a binary sequence by a preliminary defined function. At last, the resulting binary sequence is represented as decimal integer values in the range [0, 255] and collected into feature histogram, which is similar to LBP. While LPQ is known to possess invariance to blurring effects, as discussed in [16], it is possible that image blurring is relevant to face presentation attack.

3.3.5. BSIF. Without loss of generality, the optimal selection of local features can effectively capture the relevant structure characteristics of the image. Alternatively, the binarized statistical image features (BSIF) [40] are adopted in a manner, in which an inquiry image is convolved with a linear filter, and then, the binary code of the filter response is obtained. By means of independent component analysis (ICA), the weight values of the filters are learned from a set of natural image patches by maximizing the statistical independence of the filter responses. Given an image block C and a bank of linear filters with the same size, the convolutional response r_i is computed by

$$r_i = C * W_i, \tag{15}$$

where W_i denotes the filter, $i \in \{1, \ldots, n\}$. Specifically, in this work, 8 filters are used (i.e., $n = 8$). And, then, the binarized feature is obtained:

$$b_i = \begin{cases} 1, & \text{if } r_i \geq 0, \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

It should be noted that the filter W_i has been well-trained by learning a set of heterogenous natural images which is different from the face images. Therefore, the BSIF features can avoid tedious filter design and parameter tuning. Moreover, the BSIF descriptor is capable of serving as a general descriptor to deal with various presentation attack scenarios in the practical detection.

3.4. Design of the Classifier. After extracting valid features, an efficient and accurate classifier is supposed to design. Various classifiers are adopted in face presentation attack detection (see [12, 42–44], for instance). In general, the monotone classifier structure equipped with fixed parameters possibly leads to the deviation of classification results. In order to achieve high level detection accuracy and generalization ability, we intend to investigate the following classifiers and select the optimal scheme of designing a classifier based on the proposed color residual texture representation.

3.4.1. LDA. Linear discriminant analysis (LDA) is a supervised approach that is widely adopted in the field of face recognition [45] and face presentation attack detection [12], which can be used for both dimensionality reduction and classification. The objective of LDA is to find a proper projection that maximizes the between-class scatter matrix and minimizes the within-class scatter matrix in the projective feature space. In the past, the image data was directly used as input, but when dealing with the high-dimensional face data, LDA often suffers from the small sample size problem. In this work, we extract texture descriptors with strong expressiveness from face images and relatively low

dimension features are extracted. Then, LDA can also be used as a classifier to be considered.

3.4.2. SVM. Support vector machine (SVM) is a kind of classifier of generalized model for binary classification tasks based on supervised learning. By utilizing the kernel method, nonlinear classification tasks can also be accomplished. Due to the outstanding property of sparsity and robustness, SVM is often used when resolving face recognition missions [46]. The decision boundary of SVM is the maximum margin hyperplane for the solution of learning samples. Furthermore, SVM uses hinge loss functions to calculate empirical risks and adds regularization terms to the solution system to optimize structural risks. Face presentation attack detection can be considered as a binary classification task, and support vector machines are classifiers with the potential to cope with such task. More importantly, the feature size obtained by our hand-crafted feature-based method is relatively large, and SVM performs well when learning high-dimensional feature vectors.

3.4.3. XGBoost. By optimizing the boosting algorithm on the basis of gradient boosting decision tree (GBDT), extreme gradient boosting (XGBoost) has been employed to resolve the classification and regression problems in many fields [47]. In fact, XGBoost is still based on the tree model. Hundreds of tree models with low classification accuracy are combined to iterate continuously, and each iteration generates a new tree. XGBoost adds a regular term to the cost function to control complexity. From the perspective of bias-variance trade off, the regular term reduces the variation of the model, makes the learned model simpler, and prevents overfitting. When conducting face presentation attack detection, a detector based on XGBoost classification possibly produces superior generalization ability dealing with heterogenous data.

3.4.4. Ensemble Classifier. As [48] states, to make an ensemble decision, constituent classifiers should be heterogenous, and meanwhile, their classification performances should be comparable. Accordingly, three base classifiers (LDA, SVM, and XGBoost) are selected in our well-designed ensemble classifier. Actually, we have also tried other kinds of classifiers, such as Naive Bayesian and Decision Tree. However, these two classifiers are not adopted in our design due to unsatisfying performance. The scheme of voting decision can be referred to as a soft voting, which is not a simple majority rule. Specifically, the average of the probability that all model prediction samples are in a certain class is taken as the threshold, and the corresponding class with the highest probability will bring the final prediction result. As Figure 5 illustrates, Classifier 1 and Classifier 2 both predict the test sample "Bona Fide," and only Classifier 3 outputs "PA," while after the soft voting decision, the final result is still "PA." The experimental results in Section 5.4 also can verify that our carefully designed voting scheme produces better performance than using single classifier.

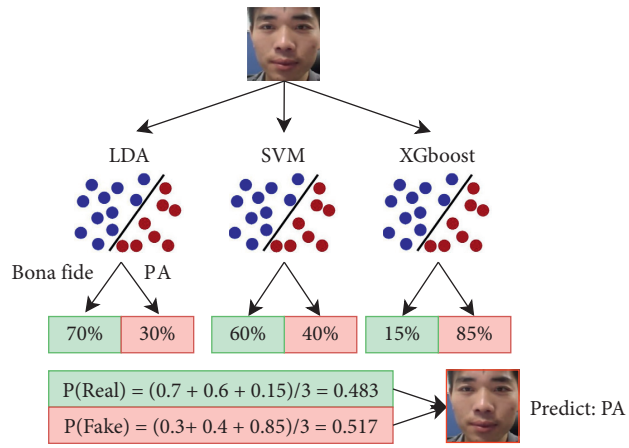


Figure 5: A toy example of the voting decision. For clarity, three base classifiers are used here.

4. Description of the Benchmark Datasets

In this work, four challenging benchmark datasets are used to evaluate our proposed detector: CASIA Face Antispoofing Dataset (FASD), Replay-Attack Dataset (RAD), MSU Mobile Face Spoof Dataset (MSU), and ROSE-YOUTU Face Liveness Detection Dataset (ROSE). For clarity, a summary of the four datasets is illustrated in Table 3. Detailed descriptions of the four datasets are given as follows.

4.1. CASIA Face Antispoofing Dataset. The CASIA Face Antispoofing Dataset [49], released in 2012, consists of 600 video clips from 50 different clients. There are three attack types involved. (1) *Warped Photo Attack*. The photograph of the legitimate client is presented to the camera, and the movement of the face is simulated by bending the photo. (2) *Cut Photo Attack*. The eye area in the face photo is cut out and a person blinks behind the paper hole. (3) *Replay Video Attack*. High-resolution video of face is displayed on a tablet. There are three imaging quality level used to record the whole real accesses and spoofing attacks. (1) *Low-quality*, with 640×480 resolution, captured by a cheap USB-camera. (2) *Normal-quality*, with 480×640 resolution, captured by another USB-camera better than the former. (3) *High-quality*, with 1280×720 resolution, captured by a Sony NEX-5 camera. The recordings of the total 50 clients are established, in which 20 clients are split into training set and remaining 30 clients into testing set.

4.2. Replay-Attack Dataset. The Idiap Replay-Attack Dataset [19], released in 2012, includes 1200 video recordings of both real accesses and spoofing attacks from 50 subjects. The video recordings are collected at two different stationary conditions. (1) *Controlled*. Uniform background scenes and lighting equipment are applied. (2) *Adverse*. Background is not uniform and only natural day-light illuminates. Under the same environments, each client is taken two high-resolution photos with Canon PowerShot SX150 IS and iPhone 3GS, respectively. These recordings are utilized to fabricate the spoofing attack samples. In total, there are three attack

Table 3: A summary of the four publicly available face spoofing datasets FASD, RAD, MSU, and ROSE.

Dataset	Release time	Subjects	Video number	Acquisition camera	Attack scenarios
FASD	2012	50	600 (150 genuine, 450 fake)	Low-quality camera (640 × 480) Normal-quality camera (480 × 640) Sony NEX-5 camera (1280 × 720)	(1) Warped photo (2) Cut photo (3) Replay video
RAD	2012	50	1200 (200 genuine, 1000 fake)	MacBook 13″ camera (320 × 240)	(1) Print (2) Mobile (3) High-def
MSU	2014	55 (35 available)	280 (110 genuine, 330 fake)	MacBook air 13″ camera (640 × 480) Google nexus 5 camera (720 × 480)	(1) Printed photo (2) Replayed video
ROSE	2018	20	3350 (500 genuine, 2850 fake)	Hasee phone (640 × 480) Huawei phone (640 × 480) iPad 4 (640 × 480) iPhone 5s (1280 × 720) ZTE phone (1280 × 720)	(1) Printed paper (2) Video replay (3) Masking

scenarios. (1) *Print*. High-resolution face photos are printed on A4 papers and displayed in front of the camera. (2) *Mobile*. High-resolution pictures and videos are displayed on an iPhone screen. (3) *High-def*. The photographs and videos are shown on an iPad screen with 1024 × 768 resolution. All recordings of 50 clients are partitioned into three disjoint subsets: (1) *Train*, (2) *Development*, and (3) *Test*, with 15, 15, and 20 clients, respectively.

4.3. MSU Mobile Face Spoof Dataset. The MSU Mobile Face Spoof Dataset [16], released in 2014, consists of 440 video clips of genuine and fake faces taken from 55 clients in total, while 280 recordings corresponding to 35 clients' subset are available. Two types of cameras are used to collect the data: a built-in camera of Macbook Air 13," referred to as laptop camera, with 640 × 480 resolution and a front-facing camera of Google Nexus 5, referred to as Android camera, with a resolution of 720 × 480. There are two spoofing attack types included. (1) *Printed Photo*. To generate the printed attack samples; a HD photograph of the client's face is captured by the Canon 550D camera, with 5184 × 3456 resolution. Then, the photo is printed on an A3 paper using a HP color printer. (2) *Video Replay*. The video of the client's face is first recorded using a Canon 550D camera and an iPhone 5S back-facing camera. The Canon camera is used to capture a HD video with 1920 × 1088 resolution, which is replayed on an iPad Air screen. And, the iPhone 5S is used to capture another HD video with 1920 × 1080 resolution, which is replayed on the iPhone 5S screen.

4.4. ROSE-YOUTU Face Liveness Detection Dataset. The ROSE-YOUTU Face Liveness Detection Dataset [7], released in 2018. ROSE dataset consists 3350 videos from 20 clients. For each client, there are 150–200 video clips with the average duration about 10 seconds. Five types of mobiles are used to collect the dataset: a Hasee smart-phone with the resolution of 640 × 480, a Huawei smart-phone with a resolution of 640 × 480, an iPad 4 with the resolution of 640 × 480, an iPhone 5s with resolution of 1280 × 720, and a ZTE smart-phone with resolution of 1280 × 7200. Three spoofing attack types are considered: (1) printed paper attack: to generate fake samples; still printed paper and quivering printed paper (A4 size) are used, (2) video replay attack: face videos are displayed on Lenovo LCD screen and Mac screen, and (3) masking attack: masks with and without cropping are presented.

5. Experimental Results and Analysis

5.1. Experimental Setup. As prior works [19, 21, 50], the face video recordings in FASD, RAD MSU, and ROSE datasets are split into single-face region frame, and frame-based experiments are conducted. All face images are normalized into 64 × 64 size after face alignment; the facial landmarks are localized by using Dlib 19.14.0 [51]. The parameter settings of the descriptors are shown as follows: when extracting the CM feature, two first-order differential operators are applied (in horizontal direction and vertical direction), the truncation threshold $\gamma = 2$, and the order is set as $d = 3$. And, the offsets are chosen as $(\Delta x, \Delta y) \in \{(0, 1), (1, 0)\}$. As for LBP feature, the parameters $P = 8$ and $R = 1$. As for CoALBP feature, LBP (+) is used with radius $R = 1$ and the corresponding $\Delta B = 2$. The parameters for the LPQ descriptor are $g = 7$ and $s = 1/7$. At last, the filter size of BSIF features is set as 7 × 7. The dimension of the texture feature extracted by using the CM, LBP, CoALBP, LPQ, and BSIF on single channel is 75, 59, 1024, 256, and 256, respectively. Additionally, scikit-learn toolkit [52] is used for model training and parameter fine-tuning.

In the following experiments, equal error rate (EER) is used as a metric. In general, a threshold is adopted to calculate the false reject rate (FRR) and the false accept rate (FAR). When these two rates are equal by adjusting the threshold, the common value is referred to as EER. Besides, HTER also serves as another metric for evaluation (advised on RAD), which can be formulated by

$$\mathrm{HTER} = \frac{\mathrm{FAR}(\tau, D) + \mathrm{FRR}(\tau, D)}{2}, \tag{17}$$

where τ is the value of the EER estimated on the dataset D. It should be noted that the smaller EER or HTER represents the better detection result.

5.2. Validation of the Residual Color Texture Representation. In this section, the CM descriptor is used as an instance to verify the effectiveness of employing RCTR. Both

benchmark FASD and MSU are used for testing. In Figure 4, the EER of the CM features extracted from gray-scale image, RGB, HSV, YCbCr, and LAB spaces are presented, where the SVM classifier is used. As can be clearly observed, the results obtained by using residual images are generally better than that of using original images both on the two datasets. Thus, it can hold true that, by using the residual image instead of the original image, the interference of nuisance noise can be effectively reduced, while more discriminative features for presentation attack detection can be extracted. More importantly, the effectiveness of color space transforming can also be verified in Figure 4. When considering the EER of the CM features extracted from residual images, the worst result is shown in the case of gray scale both on FASD and MSU. Besides, the lowest EER on FASD is 4.5% when using HSV space, and the best performance on MSU is 5.6% in the case of YCbCr.

5.3. Performance Comparison of Different Texture Descriptors. In this part, the performance of the LBP, CoALBP, LPQ, BSIF, and CM descriptors are evaluated on FASD, where SVM classifier is employed, as shown in Figure 6. It can be observed that the EERs of the CM descriptor (brown column) is obviously lower than that of the other four types of descriptors in the cases of RGB, HSV YCbCr, and LAB, and the CoALBP descriptor (red column) performs best in the case of gray scale. Since the performance of all descriptors is relatively poor in gray-scale space, we only consider using RGB, HSV, YCbCr, and LAB spaces. Thus, the CM descriptor is selected to construct the final RCTR.

5.4. Evaluation of Different Classifiers. Subsequently, the EER results of the CM features on benchmark FASD by employing different classifiers are presented, as shown in Table 4. And, for fair comparison, the average EERs of each classifier is also presented. It can be observed that, basically, our proposed ensemble classifier maintains the lowest EER in most cases except in gray scale. Moreover, the average EER of ensemble classifier is 10.7%, which is still the lowest among four powerful classifiers. Obviously, our proposed probabilistic decision-based ensemble classifier can perform better than using single classifier such as LDA, SVM, or XGBoost.

5.5. Fusion of the Residual Color Texture Representation. In this section, the fusion performance of color spaces for RCTR is well-explored. A total of four color spaces are considered, namely, RGB, HSV, YCbCr, and LAB. As discussed above, the CM descriptor is selected to extract texture features from residual images to construct the RCTR, and the ensemble classifier is employed. Extensive experiments based on different color space fusions are conducted, in which the benchmark FASD and MSU are used for evaluation, as can be seen in Table 5. Furthermore, the performance of the combination of only color components is also explored. Specifically, {H,S,Cb,Cr} means the RCTR extracted from H, S, Cb, and Cr channels.

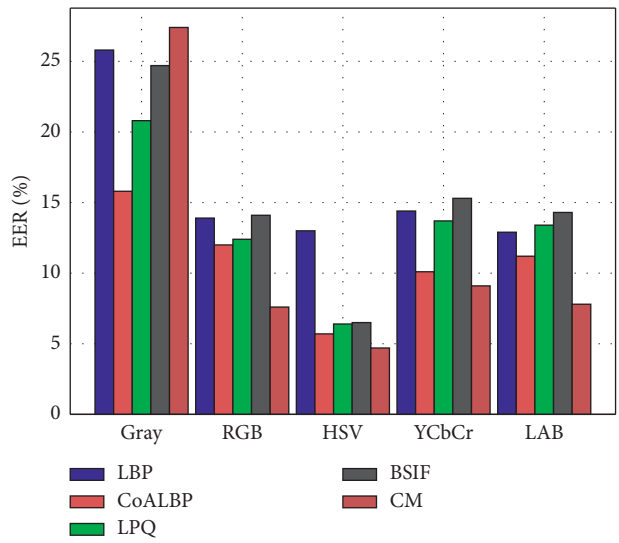


Figure 6: The EER results of the LBP, CoALBP, LPQ, BSIF, and CM features extracted from various color spaces.

Table 4: The EER results of the CM features extracted from various color spaces when using LDA, SVM XGBoost, and Ensemble Classifier.

Color space	LDA	SVM	XGBoost	Ensemble
Gray	30.6	27.4	**26.1**	27.2
RGB	8.3	7.6	6.9	**6.3**
HSV	5.5	4.7	5.6	**4.4**
YCbCr	9.7	9.1	9.0	**8.1**
LAB	8.5	7.8	9.7	**7.5**
Average	12.5	11.3	11.5	**10.7**

Table 5: The performance of various color space combinations of RCTR when employing ensemble classifier.

Color space fusion	FASD EER	MSU EER	Average EER
RGB + HSV	2.4	5.3	3.95
RGB + YCbCr	3.5	3.2	3.35
HSV + YCbCr	2.1	2.3	2.20
RGB + HSV + YCbCr	**1.6**	2.5	2.05
HSV + YCbCr + LAB	2.0	2.3	2.15
All spaces	1.8	**2.0**	**1.90**
{H,S,Cb,Cr}	4.9	5.5	5.20

As shown in Table 5, when combining the features of all four color spaces, the optimal performance of RCTR can be achieved on MSU (with the EER of 2.0%). As for FASD, when combining RGB, HSV, and YCbCr spaces, the lowest EER (1.6%) is obtained. Meanwhile, the EER of the RCTR extracted from {H,S,Cb,Cr} is 4.9% and 6.5%, respectively, which is not as good as combining all color spaces.

When considering the average value, the EER when combining all four spaces is the lowest (1.9%). And, it can be clearly observed that when combining three color or four spaces, the EERs of the detector are generally lower than those only combining two spaces. Then, we can draw that, in most cases, by combining the RCTR features of more color spaces, the performance of our face PA detector can be further improved.

5.6. Intradataset Performance in Comparison with the State of the Art. In this section, we evaluate the performance for identifying the bona fide and the PA images in the case that the training and testing data are matched. Tables 6–8 present the experimental results of our proposed method and the state-of-the-art techniques ([12, 16, 21, 33, 53, 54, 55, 56, 57] for hand-crafted feature-based methods and [2, 3, 9, 27, 58, 59] for DL-based methods). The results of the RCTR combining four color spaces are used for comparison. It should be noted that, as reported in [16], because only a portion samples (high-quality) of FASD were used for evaluation, for fair comparison, the result is not listed in Table 6. And, since the EER is not adopted in [16, 55], we only cite HTER results on RAD.

From Table 6, we can observe that DRL-FAS [59] outperforms other methods both on FASD and RAD. Our method outperforms most methods except [59] on FASD and shows competitive performance on RAD. Motion Mag algorithm [12] also achieves the best HTER on RAD, but suffers significant degradation when testing on FASD. Meanwhile, it can be seen that the performance of most hand-crafted feature-based approaches and DL-based methods are satisfactory on RAD. The reason lies on that when collecting the data of RAD; the photo capture condition is relatively simple, i.e., only one kind of camera is adopted.

As can be clearly observed in Table 7, our RCTR-based detector achieves the lowest EER on MSU (2.0%). And, from Table 8, it is observed that DRL-FAS [59] achieves best performance on ROSE, with an EER of 1.8%. Our method gets the second place, with an EER of 10.7%, which is the best performance among hand-crafted feature-based methods [21, 57] and better than DL-based method [60]. In conclusion, these results indicate that the bona fide and the PA images can be accurately identified by employing our proposed RCTR-based ensemble classifier.

5.7. Interdataset Performance Comparison with the State of the Art. To evaluate the performance of the detector when training and testing samples are mismatched, cross testing among all three datasets is conducted. The HTER results of our RCTR-based detector when combining all four spaces and only using H, S, Cb, and Cr channels are presented in Table 9. It is observed that SSR-FCN [61] performs best when training on FASD and testing on RAD (with an HTER of 19.9%), but in another case, the performance of SSR-FCN is relatively poor (41.9%). When training on RAD and testing on FASD, auxiliary [28] outperforms other methods (with the EER of 28.4%). As for our proposed method (RCTR-{H, S, Cb, Cr}), the HTER is 31.8% and 39.6%, respectively, which significantly outperforms the methods proposed in [3, 12, 33, 55] while comparable with outstanding arts in [9, 28, 29, 59, 60]. It is worth noting that the HTER of RCTR-all spaces is higher than RCTR-{H,S,Cb,Cr}. This phenomenon can be explained as follows: when capturing the face records, the scene's brightness condition of different datasets is not consistent, so the RCTR feature extracted in complete color spaces containing the luminance

Table 6: Performance comparison with the state-of-the-art methods on FASD and RAD. "-" represents that the results are not available.

Method	FASD	RAD	
	EER	EER	HTER
LBP-TOP [33]	10.0	7.9	7.6
LDP-TOP [53]	8.9	2.5	1.8
Motion Mag [12]	14.4	0.2	**0.0**
IDA [16]	—	7.4	—
Dynamic [54]	21.8	5.3	3.8
Spectral cubes [55]	14.0	—	2.8
CVLBC [56]	6.5	1.7	0.8
Color LBP [57]	7.1	0.9	4.9
Color [21]	2.1	0.4	2.8
Deep CNN [3]	7.4	6.1	2.1
Partial CNN [2]	4.5	2.9	4.3
LBP-Net [58]	2.5	0.6	1.3
Fusion CNN [27]	2.7	0.8	0.7
MobileNet + attention [9]	4.2	0.1	0.3
ResNet + attention [9]	3.1	0.2	0.4
DRL-FAS [59]	**0.2**	**0.0**	**0.0**
RCTR-all spaces (ours)	1.8	0.7	2.1

Table 7: Performance comparison with the state-of-the-art methods on MSU.

Method	EER
LBP + SVM baseline	14.7
DoG-LBP + SVM baseline	23.1
IDA [16]	8.5
LDP-TOP [53]	6.5
Color LBP [57]	10.6
Color [21]	4.9
RCTR-all spaces (ours)	**2.0**

Table 8: Performance comparison with the state-of-the-art methods on ROSE.

Method	EER
LBP + SVM baseline	34.1
Color LBP [57]	27.6
Color [21]	13.9
De-spoofing [60]	12.3
DRL-FAS [59]	**1.8**
RCTR-all spaces (ours)	10.7

Table 9: Interdataset testing comparison on the FASD dataset versus the RAD in terms of HTER.

Method	Train FASD	Test RAD	Train RAD	Test FASD	Average
LBP-TOP [33]	49.7		60.6		55.2
Motion Mag [12]	50.1		49.7		49.9
Spectral cubes [55]	34.4		50.0		42.2
Deep CNN [3]	48.5		45.5		47.0
Auxiliary [28]	27.6		**28.4**		28.0
De-spoofing [60]	28.5		41.1		34.8
STASN [29]	31.5		30.9		31.2
MobileNet + attention [9]	30.0		33.4		31.7
ResNet + attention [9]	36.2		34.7		35.5
DRL-FAS [59]	28.4		33.2		30.8
SSR-FCN [61]	**19.9**		41.9		**27.0**
RCTR-all spaces (ours)	37.1		42.0		39.6
RCTR-{H,S,Cb,Cr} (ours)	31.8		39.5		35.7

Table 10: Interdataset testing comparison with color texture-based methods on FASD, RAD, and MSU datasets in terms of HTER.

Method	Training / Testing	FASD		RAD		MSU		Average
		RAD	MSU	FASD	MSU	FASD	RAD	
Color LBP [57]		47.0	36.6	39.6	35.2	49.6	42.0	41.7
Color [21]		**30.3**	20.4	**37.7**	34.1	46.0	**33.9**	33.7
RCTR-{H,S,Cb,Cr} (ours)		31.8	**19.1**	39.5	**29.0**	**41.3**	34.4	**32.5**

information is not as good as in the color component, i.e., H, S, Cb, and Cr.

Besides, it can be observed that when training on FASD and testing on RAD, the result is better than training on RAD and testing on FASD. The reason lies in FASD which has more types of cameras and more attack scenarios; thus, the detector is more robust. However, the manner of collecting the recordings of RAD dataset is relatively simple, and the lack of diversity in training data leads to poor performance of the detector when testing on new dataset.

In Table 10, more comprehensive experiments are conducted to compare our method with other color texture-based methods [21, 57]. It can be seen that when training on FASD and testing on MSU, the HTER of our proposed detector is lowest. Similarly, our proposed method performs best in half of the cases. In addition, the average HTER of our proposed detector is 32.5%, which is also the lowest. The well performance of our proposed algorithm using color residual texture representation when testing on mismatched samples can be attributed to the generalization ability of the CM feature and the highly robust ensemble classifier.

5.8. Performance versus Training Set Scale. In this part, we investigate on how the scale of training data impacts the performance of the proposed method. Specifically, the training set scale is increased from 10% to 90%, with a step of 10%, and the remaining data are used for validation. 10-folds' validation experiments are conducted, and each experiment randomly selects face images to form the training set; the average of the results are taken as the final result. Prediction accuracy (ACC for short) is used as metric, that is, the ratio of correct predictions to the total testing samples. As illustrated in Figure 7, as the scale of training data increases, the ACC of our proposed presentation attack detector is gradually improved. And, when using only 10% training data, the ACC of our RCTR-based detector on all three datasets is higher than 95.5%. The empirical study indicates that our proposed method can achieve excellent prediction accuracy with a small-scale training data. In addition, since DL-based methods are data-driven, so the performance of them is likely to be unsatisfactory when there is insufficient training data.

5.9. Time Complexity Analysis. We conduct time consumption statistical experiments to analyze the processing time. All methods considered are implemented by using Matlab2017a and Python 3.6 on an Intel Core i7 2.8 GHz CPU and 16 GB RAM PC. A total of 500 videos are used, and the number of frames of each video is between 300 and 400.

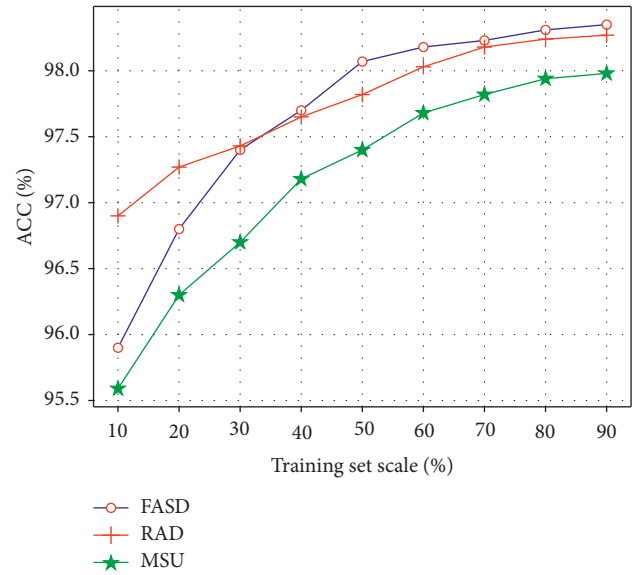


Figure 7: Performance of the proposed method versus the training set scale.

Table 11: Processing time (per video) of our method and some baseline methods.

Method	Time (second)
LBP + SVM baseline	10.3
Color LBP [57]	12.2
Color [21]	21.9
RCTR-all spaces (ours)	15.6

The average processing time of each video is recorded, which is shown in Table 11. It can be observed that our method can achieve a competitive time consumption compared with other methods (with an average processing time of 15.6 second), which indicates the good real-time detection ability of the proposed method. Furthermore, our method has better detection accuracy compared with other methods.

6. Conclusion

In this paper, we propose a RCTR-based detector to address the challenge raised by face PA. First, by considering the nuisance noise existing in face image, a DW-filter is applied to eliminate such interference, after which more discriminative residual images are obtained. Next, the RGB image should be transformed to more representative spaces such as HSV, YCbCr, and LAB. Dependent on the powerful texture descriptor CM, the RCTR feature is extracted from multiple color channels. Besides, an ensemble classifier is carefully designed based on a probabilistic voting rule to make the prediction. Extensive analytical experiments are conducted

to verify the effectiveness of transforming color space and employing residual image. Four challenging benchmark datasets FASD, RAD, MSU, and ROSE are used to evaluate our proposed method, and our proposed RCTR-based detector shows preferable performance in the cases of both intradataset and interdataset testing.

Data Availability

All data, models, or code generated or used during the study are available from the corresponding author upon request.

Disclosure

Yuting Du and Tong Qiao are co-first authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yuting Du and Tong Qiao contributed to the work equally.

Acknowledgments

This work was funded by the Cyberspace Security Major Program in National Key Research and Development Plan of China under grant no. 2016YFB0800201, the Natural Science Foundation of China under grant no. 61702150, the Public Research Project of Zhejiang under grant no. LGG19F020015, and the Key Research and Development Plan Project of Zhejiang Province under grant no. 2017C01065.

References

[1] L. Feng, L.-M. Po, Y. Li et al., "Integration of image quality and motion cues for face anti-spoofing: a neural network approach," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 451–460, 2016.

[2] L. Lei, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *Proceedings of the International Conference on Image Processing Theory Tools & Applications*, pp. 1–6, Montreal, Canada, December 2016.

[3] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *Computer Science*, vol. 9218, pp. 373–384, 2014.

[4] N. N. Lakshminarayana, N. Narayan, N. Napp, S. Setlur, and V. Govindaraju, "A discriminative spatio-temporal mapping of face for liveness detection," in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, Delhi, India, February 2017.

[5] Z. Xu, L. Shan, and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in *Proceedings of the Asian Conference on Pattern Recognition (ACPR)*, Beijing, China, May 2015.

[6] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2639–2652, 2018.

[7] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018.

[8] 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. 25, 2015.

[9] H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson, and S. Z. Li, "Attention-based two-stream convolutional networks for face spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 578–593, 2019.

[10] P. Gang, S. Lin, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, November 2017.

[11] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, "Real-time face detection and motion analysis with application in "liveness" assessment," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 548–558, 2007.

[12] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition Workshops*, Columbia, SC, USA, June 2013.

[13] B. Wei, L. Hong, L. Nan, and J. Wei, "A liveness detection method for face recognition based on optical flow field," in *Proceedings of the International Conference on Image Analysis and Signal Processing*, Marrakech, Morocco, July 2009.

[14] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: application to iris, fingerprint, and face recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 710–724, 2014.

[15] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *Proceedings of the International Conference on Pattern Recognition IEEE Computer Society*, Stockholm, Sweden, August 2014.

[16] W. Di, H. Hu, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics & Security*, vol. 10, no. 4, pp. 746–761, 2015.

[17] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proceedings of the International Joint Conference on Biometrics (IJCB)*, pp. 1–7, Washington, DC, USA, October 2011.

[18] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "LBP-top based countermeasure against face spoofing attacks," in *Proceedigs of the Computer Vision—ACCV 2012 Workshops*, Daejeon, Korea, November 2012.

[19] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proceedings of the IEEE International Conference of the Biometrics Special Interest Group (BIOSIG)*, Darmstadt, Germany, September 2012.

[20] J. Komulainen, A. Hadid, and M. Pietikainen, "Context based face anti-spoofing," in *Proceedings of the IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, Redondo Beach, CA, USA, October 2014.

[21] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.

[22] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "An investigation of local descriptors for biometric spoofing

detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 849–863, 2015.

[23] S. R. Arashloo, J. Kittler, and W. Christmas, "Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2396–2407, 2015.

[24] J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection with component dependent descriptor," in *Proceedings of the International Conference on Biometrics (ICB)*, pp. 1–6, Halmstad, Sweden, June 2016.

[25] B. Chen, X. Qi, Y. Zhou, G. Yang, Y. Zheng, and B. Xiao, "Image splicing localization using residual image and residual-based fully convolutional network," *Journal of Visual Communication and Image Representation*, vol. 73, Article ID 102967, 2020.

[26] P. He, X. Jiang, T. Sun, and H. Li, "Computer graphics identification combining convolutional and recurrent neural networks," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1369–1373, 2018.

[27] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *Proceedings of the IEEE International Joint Conference on Biometrics*, Denver, CO, USA., April 2017.

[28] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: binary or auxiliary supervision," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CL, USA, September 2019.

[29] X. Yang, W. Luo, L. Bao et al., "Face anti-spoofing: model matters, so does data," in *Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CL, USA, September 2019.

[30] K. Ali, "Unknown presentation attack detection against rational attackers," 2010, https://arxiv.org/abs/2010.01592.

[31] H. Feng, Z. Hong, H. Yue et al., "Learning generalized spoof cues for face anti-spoofing," 2005, https://arxiv.org/abs/2005.03922.

[32] Y. Jia, J. Zhang, S. Shan, and X. Chen, "Single-side domain generalization for face anti-spoofing," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.

[33] T. Freitas Pereira, J. Komulainen, A. Anjos et al., "Face liveness detection using dynamic texture," *Eurasip Journal on Image & Video Processing*, vol. 1, no. 2, 2014.

[34] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[35] S. G. Chang, B. Bin Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

[36] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[37] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[38] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *Image and Signal Processing*, pp. 236–243, Springer, Berlin, Germany, 2008.

[39] R. Nosaka, Y. Ohkawa, and K. Fukui, "Feature extraction based on co-occurrence of adjacent local binary patterns," in *Advances in Image and Video Technology*Springer, Berlin, Germany, 2011.

[40] J. Kannala and E. Rahtu, "Bsif: binarized statistical image features," in *Proceedings of the Pattern Recognition (ICPR), 21st International Conference on Pattern Recognition*, Tsukuba, Japan, November 2012.

[41] A. Eleyan and H. Demirel, "Co-occurrence matrix and its statistical features as a new approach for face recognition," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 19, no. 1, pp. 97–107, 2011.

[42] X. Song, X. Zhao, L. Fang, and T. Lin, "Discriminative representation combinations for accurate face spoofing detection," *Pattern Recognition*, vol. 85, pp. 220–231, 2018.

[43] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: spoof detection on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2268–2283, 2016.

[44] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing using speeded-up robust features and Fisher vector encoding," *IEEE Signal Processing Letters*, vol. 24, pp. 141–145, 2017.

[45] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data—with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[46] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June 2015.

[47] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CL, USA, August 2016.

[48] R. P. W. Duin, "The combining classifier: to train or not to train?" in *Proceedings of the Object Recognition Supported by User Interaction for Service Robots*, Quebec City, Canada, August 2002.

[49] Z. Zhang, J. Yan, S. Liu, L. Zhen, and S. Z. Li, "A face antispoofing database with diverse attacks," in *Proceedings of the International Conference on Biometrics (ICB)*, Seoul, Korea, August 2012.

[50] X. Tan, L. Yi, J. Liu, and J. Lin, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Proceedings of the Computer Vision—ECCV*, Heraklion, Crete, August 2010.

[51] D. E. King, "Dlib-ml: a machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. 3, pp. 1755–1758, 2009.

[52] A. Swami and R. Jain, "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 10, pp. 2825–2830, 2012.

[53] Q. T. Phan, D. T. Dang-Nguyen, G. Boato et al., "FACE spoofing detection using LDP-TOP," in *proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Phoenix, AZ, USA, 2016.

[54] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, 2015.

[55] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4726–4740, 2015.

[56] X. Zhao, Y. Lin, J. Heikkila et al., "Dynamic texture recognition using volume local binary count patterns with an application to 2D face spoofing detection," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 552–566, 2017.

[57] J. K. Z. Boulkenafet and A. Hadid, "Face anti-spoofing based on color texture analysis," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 2636–2640, Quebec City, Canada, September 2015.

[58] L. Li, X. Feng, Z. Xia, X. Jiang, and A. Hadid, "Face spoofing detection with local binary pattern network," *Journal of Visual Communication and Image Representation*, vol. 54, pp. 182–192, 2018.

[59] R. Cai, H. Li, S. Wang, C. Chen, and A. C. Kot, "DRL-fas: a novel framework based on deep reinforcement learning for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 937–951, 2020.

[60] A. Jourabloo, Y. Liu, and X. Liu, "Face de-spoofing: anti-spoofing via noise modeling," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 290–306, Glasgow, Scotland, March 2018.

[61] D. Deb and A. K. Jain, "Look locally infer globally: a generalizable face anti-spoofing approach," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1143–1157, 2020.