This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3301242

Pixel Bleach Network for Detecting Face Forgery under Compression

Congrui Li[#], Ziqiang Zheng[#], Yi Bin, Guoqing Wang, Yang Yang, Xuesheng Li, and Heng Tao Shen

***Abstract*—The existing face forgery algorithms have achieved remarkable progress in how to generate reasonable facial images and can even successfully deceive human beings. Considering public security, face forgery detection is of vital importance, making it essential to design face forgery detection algorithms to detect forgery images over the Internet. Despite the great success achieved by the existing Deepfake detection algorithms, they usually failed to achieve satisfactory Deepfake detection performance when deployed to handle the forgery videos in practice. One significant reason is compression. The videos over the Internet are inevitably compressed considering the transmission efficiency. The video compression results in significant Deepfake detection performance degradation for the existing Deepfake detection algorithms. To address this issue, in this paper, we propose a generic, simple yet effective "bleaching" pre-processing module based on the generative model and the high-level feature representations to produce a bleached image, which shares a similar appearance with the compressed images. The bleached images with recovered information can be identified accurately by the optimized Deepfake detection models without retraining. The proposed method has utilized a redesigned feature representation, which serves as a navigator to effectively and sufficiently alter the feature distribution in the high-dimensional space to remedy the difference between real facial images and forgery counterparts. Thus, the proposed method can successfully avoid misclassification. Comprehensive and extensive experiments are carried out on four low-quality Faceforensics++ datasets, demonstrating the effectiveness of our method in recovering the information loss caused by the compression artifacts across various backbones and compression.**

***Index Terms*—Deepfake detection, Robust Deepfake detection under compression, Adversarial learning**

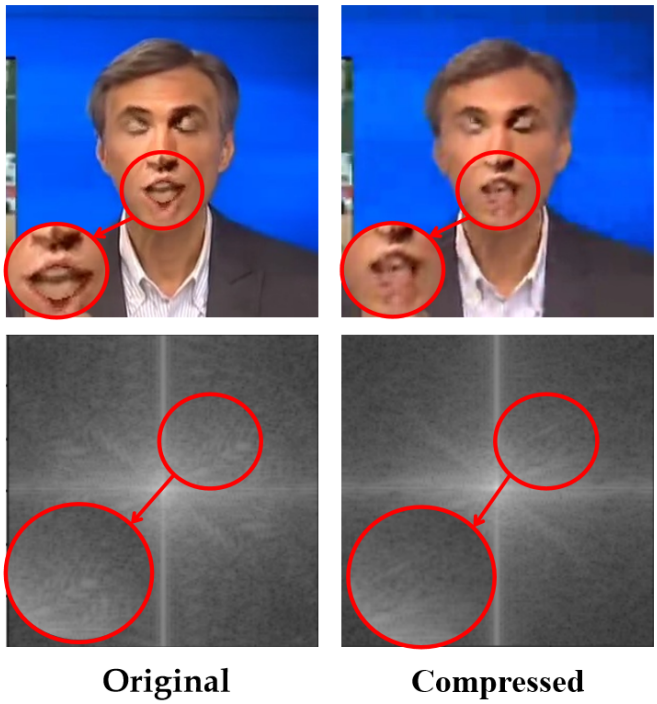


Fig. 1. The compression during the transmission procedures inevitably introduces serious perturbation to the face forgery images in both the spatial domain (top row) and frequency domain (bottom row).

I. Introduction

WITH the emergence of deep learning algorithms [1]–[6], lots of evolutionary progress has been achieved in computer vision fields. One remarkable visual application is the creation of forgery facial images (also referred to as Deepfakes), which targets to change or manipulate the facial expression of a person or even the identity information. Deepfakes can be created in numerous ways. Among all the Deepfakes algorithms, the notable one is to adopt the generative adversarial networks (GANs) [5], [7] for conducting the image or the video generation and alteration. The flourishing progress of the deep neural networks and the generative adversarial network [8] makes it increasingly difficult for human beings to distinguish a real facial image from the manipulated one. The synthesized forgery facial images or videos are so realistic that they cannot be detected by a non-suspecting person. When these facial image creation techniques are unrestrictedly adopted, it has arisen a boom of Deepfake APPs (*e.g.*, FaceApps, ZAO, *etc.*), making it available to the public to swap one's faces or manipulate the facial expressions following the Deepfake procedures.

The unauthorized/malicious usage of Deepfakes poses a serious threat to legal, political, and social systems as they can destroy the integrity of a person. The abuse of malicious attacks is inevitable and thus causes severe security and privacy issues. Therefore, it urges to promote the effectiveness of the Deepfake detection algorithms in various situations. To address these public issues caused by the Deepfake creation techniques, some Deepfake detection algorithms [9]–[12] have been proposed to alleviate the dilemma caused by the Deepfake images and videos. The existing Deepfake detection algorithms have indeed achieved satisfying Deepfake detection

C. Li, Z. Zheng, Yi Bin, Guoqing Wang, Yang Yang, and Heng Tao Shen are with the Center for Future Multimedia and School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Heng Tao Shen is also with the Peng Cheng Laboratory, China. Xuesheng Li is with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, China. (Email: dlyyang@gmail.com; Corresponding author: Yang Yang)

[#] Congrui Li and Ziqiang Zheng contribute to this work equally.

TABLE I
THE QUANTITATIVE COMPARISON OF DIFFERENT METHODS UNDER VARIOUS SETTINGS. THE TEXT IN RED, BLUE AND **BOLD** REPRESENTS THE TESTING ACCURACY ON HQ DATA, TESTING ACCURACY ON LQ DATA AND THE **ACCURACY DROP**, RESPECTIVELY.

Method	Faceforensics++ [20]	
	FaceSwap	Face2Face
MesoInception4 [14]	90.47 / 57.99 (**32.48**↓)	84.14 / 50.37 (**33.77**↓)
Xception [19]	98.41 / 70.95 (**27.46**↓)	97.94 / 50.82 (**47.12**↓)
F^3-Net [11]	98.20 / 71.35 (**26.85**↓)	97.04 / 51.62 (**45.32**↓)

performance under some specific settings.

To achieve accurate Deepfake detection, some specially designed convolutional neural network (CNN) backbones [13]–[15] have been proposed, which focus on the pixel-level modifications or the visual artifacts generated by the Deepfake algorithms. This line of work can obtain reasonable classification results on high-quality forensic data with visual artifacts. The detection performance will significantly degrade the forgery data essentially and deliberately designed to evade the tracing. Another line of the Deepfake detection algorithms is to shift the visual images to another domain (*e.g.*, frequency domain) [11], [16]–[18] to perform detection. After the transformations, the CNN models are optimized to detect some significant and prominent feature representations to achieve performance gain. However, the frequency domain based detection algorithms are fragile to compression. The high-frequency signals on the spectrum will be weakened if the origin Deepfake images are compressed as illustrated in Fig. 1. The compression results in catastrophic modifications to both the spatial and frequency representations of the Deepfake images, which will lead to the failure of Deepfake detection.

Considering that most of the existing Deepfake datasets are either drawn from the website (*e.g.*, Youtube) following the MPEG4.0 and H.264 formats or have been post-processed by compression for utility. The compression is general and universal among all the Deepfake datasets. The compression indeed lead to Deepfake detection performance degradation. The quantitative results are illustrated in Table I. When the Deepfake detection models trained on the high-resolution video data are tested on the low-resolution video data, current Deepfake detection algorithms (*e.g.*, Xception [19], MesoInception4 [14], and F^3-Net [11]) have an obvious performance drop under this setting. In this paper, we target to alleviate the influence of compression on Deepfake detection without retraining the whole Deepfake detection model.

We propose a generic plug-and-play pre-processing module to promote the Deepfake detection performance under the compression setting. In detail, the optimized Deepfake detection models based on high-quality forged facial images are given and frozen during the training procedure. When given the low-quality forged facial images (after various compressions) for evaluation, the trained neural networks would misclassify those compressed images. Our goal is to generate a compression remedy based on the feature representations of the compressed face forgery images in the frozen model. We have designed a bleach generator for generating bleach to correctly classify the wrongly classified samples. With the designed bleach generator module, we can achieve robust and accurate Deepfake detection even under compression. The entire framework of the proposed method is illustrated in Fig. 2. It is not trivial to achieve bleach generation because of the interplay between the compressed images and the frozen model. The generated bleach aims to correctly classify the failed examples while leaving the successfully classified examples undisturbed. The wrongly classified examples can be divided into two subclasses: false negatives (FN) and false positives (FP). The representations of these two categories will be distributed on either side of the origin position in the feature space during the optimization stage. It is a rigorous challenge to bleach both subclasses successfully. The proposed approach achieves a reasonable tradeoff between detecting both the low-quality compressed face forgery images and the original high-quality face forgery images.

Besides, we have designed two additional loss functions to further align the feature spaces of the compressed face forgery images and the high-quality counterparts. The former compression remedy loss is responsible to formulate the constraint between the bleached feature representations and the target feature representations. The latter regularization loss aims to achieve a balance between recovering the information loss for the wrongly classified samples and guaranteeing true positive samples are not distracted. To demonstrate the effectiveness of our bleach generator, we conduct comprehensive experiments based on three widely used Deepfake detection backbones to verify the universality and effectiveness of the proposed approach. Extensive analysis and ablation studies are also conducted to demonstrate the stability of the proposed approach. The superior results of the proposed approach show that our method can succeed in bleaching wrongly classified images. To sum up, our contributions can be summarized as follows.

- We propose a universal image bleach generation module based on the thought of adversarial machining learning and combine a naive DCGAN [21] to bleach the compressed face forgery images. Two additional loss constraints are designed for better bleaching feature distribution to ensure the negative samples could be recovered correctly while keeping the labels of the positive samples unchanged.
- We regard the compression as a significant adversarial attack and we design a bleach generator to remedy the compression without retraining the Deepfake detection models. The proposed method is plug-and-play and generic, which could be extended to various Deepfake detection algorithms.
- The comprehensive and extensive experiments based on various detection backbones have been conducted on FaceForensics++ datasets. The experimental results have demonstrated the superiority of the proposed method.

The rest parts are organized as below. Section II briefly introduces the related work and Section III elaborates the proposed approach. Section IV presents the extensive experimental results on various datasets based on different backbones, followed by the discussions and conclusion in Section V.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3301242

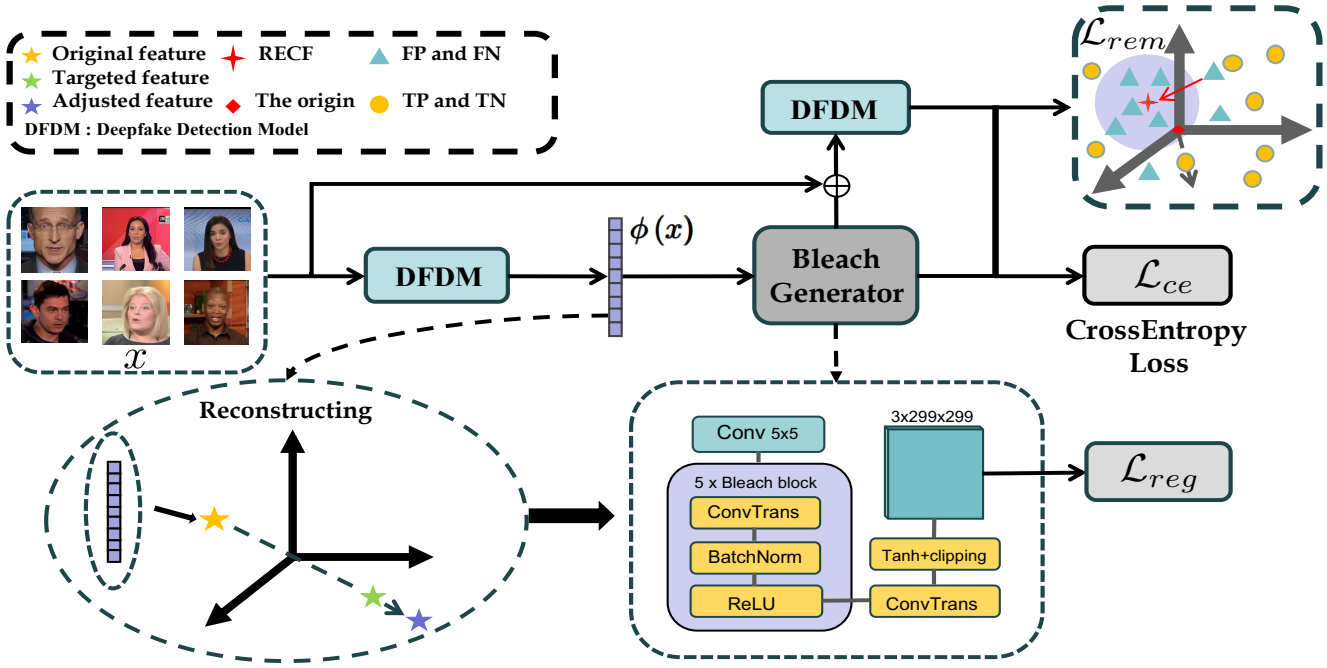


Fig. 2. Illustration of our method. We feed the low-quality (LQ) face forgery samples into the frozen Deepfake detection model (abbreviated as "DFDM"), *e.g.*, Xception network [19] trained on the high-quality (HQ) face forgery images. The feature representations of the last full-connected layer are the input of our "bleach generator" to synthesize a corresponding generated bleach. The generated bleach is then merged into the input forgery image to output the bleached image for recovering the information loss caused by the compression. The bleached images can be correctly identified by the frozen DFDM without retraining. The compression remedy loss $\mathcal{L}_{rem}$ is to align the distribution of LQ and HQ data while the regularization loss $\mathcal{L}_{reg}$ is designed for preventing the collapse of the bleached image. To achieve the trade-off between detecting the wrongly classified samples and guaranteeing true positive samples are not distracted, we construct a balanced offset training strategy to steadily optimize the whole framework.

II. Related Work

A. Deepfake Detection

There have been manipulated with four face forgery techniques: Face2Face, Deepfakes, FaceSwap, and Neural Textures. Some state-of-the-art generative models [22], [23] (*e.g.*, PGGAN [24], StarGAN [25] and StyleGAN [26] and *etc.*) can synthesize hyper-realistic fake face images with the resolution up to 1024×1024. These synthesized photo-realistic images are widely spread on the Internet, which brings great challenges to network security. Current Deepfake approaches especially refer to the manipulation of human faces and these forgery methods can be divided into two main categories: 1) one is to swap two faces from different people to change identity information; 2) the other is 3D face reconstruction and animation methods [27]–[29]. Face2Face [1] is to transfer specified action or expression to the targeted image and keep other attributes. Some algorithms [11], [17], [18] aimed to combine the statistics in the frequency domain as extra clues to boost the Deepfake detection accuracy. The representative F^3-Net [11] proposed a novel method to achieve Deepfake detection in the frequency domain. Besides, various methods based on different motivations have been proposed. For example, Wang *et al.* [30] introduced neuron monitoring to classify, which tracks the characteristics of fake images by capturing changes in activated neuron outputs through neuron coverage. Liu *et al.* [15] utilized an improved Gram block to record the rich texture information to successfully detect the fake images generated by StyleGAN [26], StarGAN [25], *etc* [24], [31]. Dang *et al.* [13] proposed a single attention network to guide the model to focus on the modified regions. Based on this, Zhao *et al.* [32] introduced multi-attention heads in the spatial domain to make the model focus on different local features and introduced a texture enhancement module to zoom in on the subtle artifacts. However, [32] suffers from the sensitivity to compression since the compression will inevitably lead to the changes of such subtle artifacts. The advanced architectures [33]–[37] (including the transformer architecture [34], [38], combining the additional text annotations [36] and designing the multiple instance embeddings [33]) were also introduced to the Deepfake detection field. Besides, to foster the improvement of Deepfake detection, some public datasets [20], [39] have been proposed to further promote the Deepfake detection performance. Different from these previous works, we aim to perform robust Deepfake detection under the compression setting.

B. Deepfake Detection under Compression

Some attempts of Deepfake detection under compression had been achieved in [40], [41]. The compression introduced redundant noise or perturbations to the face forgery images and made it more difficult to perform accurate Deepfake detection. [42] introduced the temporary constraint to promote the detection performance of the compressed face forgery LQ videos. The researchers [43] proposed to combine the paired LQ-HQ data for super-resolution and achieved improvement for the compressed face forgery images. However, this method heavily

TABLE II
LIST OF SYMBOLS AND ABBREVIATIONS MENTIONED IN OUR PAPER.

Notation	Description
C23 (HQ) data	High-quality (quantization rate of 23) face forgery images
C40 (LQ) data	Low-quality (quantization rate of 40) face forgery images
TP/FP	True positive / False positive
TN/FN	True negative / False negative
RECF	Reconstructed and compensated feature
DFDM	Deepfake detection model
Generated bleach	Generated output from the feature representation extracted from DFDM
Bleached image	The combination of generated bleach and corresponding Deepfake image
J_ϕ	Jacobian of feature ϕ
$\mathbb{I}$	Prediction outputs calculated by $\frac{\Delta f}{\lvert\Delta\rvert}$
$\mathbf{I}_0$	Original image
$\mathbf{I}_{bl}$	Bleached image
$\mathcal{L}_{rem}$	Compression remedy loss
$\mathcal{L}_{reg}$	Regularization loss

suffers from the overfitting problem. The recent work [44] utilized optimal transport theory in knowledge distillation. To detect low-quality compressed Deepfake images effectively, [44] designed a teacher-student network, which has been trained from different quality images to guide the student network to learn discriminative features from the low-quality samples. Zhang *et al.* [45] proposed to adopt the self-supervision to achieve the decoupling for the Deepfake video detection. The spatial and temporal feature representations were combined [46] to promote the detection performance even under compression. Some specially designed networks [47] were designed to learn the robust feature representations under various settings. Huang *et al.* [47] observed the trace caused by imperfect upsampling algorithms within the GAN-synthesized process. Huang *et al.* [47] introduced the gray-scale fakeness prediction map to improve Deepfake detection accuracy. [48] designed a two-stream network to combine the paired LQ-HQ data for input and achieve robust Deepfake detection under compression. However, it requires paired data for training and evaluation. Haliassos *et al.* [49] proposed to combine the fine-grained lips presentations for better Deepfake detection. Even though some reasonable Deepfake detection results have been achieved in the existing methods [32], [38], they require to train the Deepfake detection models from scratch based on the LQ data or retraining the trained models optimized by the HQ data. Different from these methods, in this work, we regard the compression as one significant adversarial attack and we design a bleach generator to remedy the compression without retraining the Deepfake detection models. The proposed method is plug-and-play and generic, which could be extended to various Deepfake detection algorithms and achieve performance gain.

III. METHOD

A. Preliminary and Problem Formulation

We first illustrate the list of symbols and abbreviations mentioned in our paper in Table II to provide better readability. We go through some basic knowledge mentioned in our paper and the overview framework of the proposed method is shown in Fig. 2. With the frozen Deepfake detection model optimized on high-quality forgery images, the highly compressed Deepfake images initially pass through the detection. We obtain the high-level feature representations at the last fully connected layer as the input of the following bleach generator, generating the corresponding bleach image based on the feedback from the Deepfake detection model. We adopt a DCGAN-like architecture [21] to do up-sampling and non-linear transformation to get the same size output with the compressed Deepfake images. Finally, the generated bleach (also regarded as a perturbation) is merged into the compressed Deepfake images as the bleached images following the adversarial attack manner. The bleached images are then fed into the Deepfake detection model and can be identified accurately. In detail, the compression remedy loss $\mathcal{L}_{rem}$ is responsible to make the distribution of low-quality forgery images (C23 data) approach the distribution of high-quality forgery images (C40 data). Considering the imbalance between the FP and TN samples, we construct a balanced offset training strategy to alleviate the influence of imbalanced data distribution. To avoid potential misunderstandings, we provide detailed explanations about how the compression affects the Deepfake detection models and how to recover the information loss through the generated bleach.

Compression as an attack. We regard the compression as a critical adversarial attack to the forgery images and the Deepfake detection models. Given the pairs of $(x^*, y^*) \in \{\mathcal{X}^*, \mathcal{Y}\}$. The x^* is the original image input while y^* is the corresponding label. The adversarial attack [50] aims to generate an adversarial perturbation Δx and put the perturbation into x^* and generate the perturbed output $x = x^* + \Delta x$. The perturbed data could mislead the trained classifier F on $\{\mathcal{X}^*, \mathcal{Y}\}$ causing the misclassification: $F(x) \neq y^* = F(x^*)$. The compression leading to the modification of x^* is also a non-ignorable adversarial attack to the Deepfake detection algorithms.

Bleach to recover. An intuitive strategy to defeat the adversarial attack is to add a noise δ as a double attack to the perturbed output to remedy the misclassification. We design a *bleach generator* to recover those misclassified examples caused by compression and try to avoid a practically viable tradeoff between the probability of bleaching and overall accuracy loss. The **generated bleach** as the "compression remedy" could recover information loss caused by the compression and further promote the Deepfake detection performance.

How to bleach? We simply the Deepfake detection problem by focusing on the feature representation (from the last fully connected layer) of the probabilistic classifiers: f_y=$\langle w_y, \phi(x)\rangle$, where w_y represents the class-specific parameters belonging to class y and $\phi(x)$ represents the feature representations got from the frozen model. The perturbation or compression on the original images can lead to the shift of the feature representation from the original class to another class in the label space expressed as: $y \neq y^* = argmax_y \langle w_y, \phi(x) + \Delta\phi(x)\rangle$. Under this setting, we cannot obtain the mapping function from images to labels and we can only get the feature expression of the images. We design Δx, which can satisfy $\|\phi(x) + \Delta\phi - \phi(x + \Delta x)\|^2 \leq \mathcal{E}$. Thus, an intuitive strategy is to utilize Jacobian to linear ϕ at x:

$$\phi(x + \Delta x) = \phi(x) + J_\phi \Delta x + O\left(\|\Delta x\|^2\right), \quad (1)$$

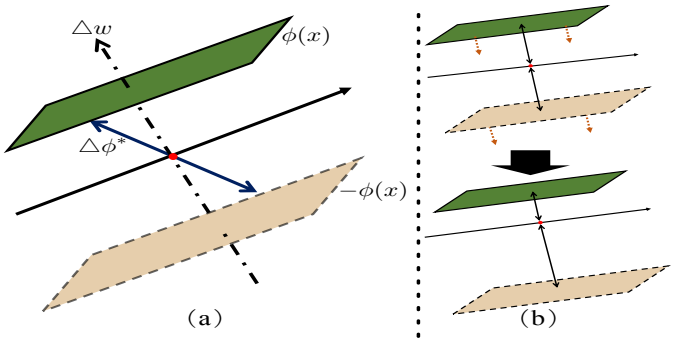


Fig. 3. The illustration of proposed reconstruction and compensation strategy. (a) illustrates the direct mapping of the feature to a symmetrical space (dashed box) centered on the origin to obtain $\Delta\phi^*$ in Equ 2. (b) shows the balanced offset training strategy to guarantee that the generated bleach can be effective and efficient leaned based on even the imbalanced distribution between FN and FP samples.

and we can approximately change Equ 1 to:

$$\Delta\phi^* = \phi(x + \Delta x) - \phi(x) = J_\phi \Delta x, \tag{2}$$

through this, we can turn this problem into a non-simultaneous linear problem. It will require lots of time and computational costs to directly solve this problem. We target to compute our generated bleach Δx to resort to the GAN [21] architecture, which uses random noise to generate an RGB image.

B. Bleach Generator

In this section, we will illustrate how the proposed bleach generator synthesizes the compression remedy. The LQ data are firstly fed into the classifier as shown in Fig. 2. We choose Xception [19] as the network backbone for illustration. Xception is the most universal classifier in the Deepfake detection task, which was first evaluated on the FaceForensics++ dataset [20] and now has become the standard Deepfake detection network backbone choice. We obtain the wrongly classified examples based on the frozen model and the corresponding feature representation $\phi(x)$ at the last fully connected layer. Then, $\phi(x)$ is fed into the bleach generator. The bleach generator consists of two parts: 1) feature reconstruction and compensation; 2) a generator to generate the same size bleach following the naive DCGAN manner [21].

Feature reconstruction and compensation. The network structure of feature reconstructing and compensation is shown in Fig. 3. We propose an effective method to map the original feature $\phi(x)$ to the symmetric point in the feature space. The procedure of from $\phi(x)$ to $\Delta\phi^*$ is described as:

$$\Delta\phi^* = -\phi(x) - \phi(x) = -2\phi(x), \tag{3}$$

the binary classification formulates a solution space spanned by $\Delta f = \langle \Delta w, \phi(x) \rangle$. For the two predicted labels, the Δf will be either positive or negative according to our observations. Since Δw has been fixed, the two solution spaces are only influenced by Δf, leading to two components distributed on both sides of the origin. It is difficult to optimizer both solution spaces because the two representations from the opposite side may conflict and disturb each other. In most cases, the true negative and false positive samples are highly imbalanced. The model will tend to focus on the dominant examples in this case. To mitigate this problem, we propose a balanced offset training strategy by controlling the distance of the feature from the origin. In detail, we aim to adjust the $\Delta\phi^*$ as follows:

$$\Delta\phi^* = -2\phi(x) + m \cdot \mathbb{I} \times \phi(x), \tag{4}$$

where $m = 0.7$ in our experiments. m is a constant value to measure the intensity of the offset and $\mathbb{I}$ controls which direction to mitigate the imbalance. $\mathbb{I}$ is calculated by the feature representation $\frac{\Delta f}{|\Delta|}$. Then we reshape the output $\Delta\phi^*$ to $8 \times 8 \times 32$. We adopt a naive DCGAN architecture to increase the dimension of this feature representation to obtain a $299 \times 299 \times 3$ generated bleach (the same size as the compressed images). Then the generated bleach is added to the compressed image and the output is fed into the Deepfake detection model to promote the Deepfake detection performance under the compression setting.

C. Loss Functions

1) Compression Remedy Loss: Our target goal is to recover the information loss caused by the compression and promote the overall Deepfake detection accuracy. To recover the information loss, an intuitive way is to utilize the deep metric learning loss [51], [52] to provide some constraints to reduce the distance between bleached feature representation ϕ_{bl} and the target feature representation ϕ_{tar}. However, the feature distribution of the manipulated faces generated by different Deepfake algorithms changes from different algorithms. Even for the samples from the same distribution, the samples are also quite diverse. To build a universal bleach network that can be adopted on varied Deepfake data and diverse Deepfake detection models, we aim to minimize the distance between the bleached feature (denoted as ϕ_{bl}) and to obtain reconstructed and compensated feature representation.

Besides, we simultaneously push ϕ_{bl} away from the origin to force those true positive and false negative examples separate from false positive and false negative samples, in which we achieve the goal through a filtering mechanism as the following formula:

$$\mathcal{L}_{rem} = \frac{1}{\sqrt{B}} \sqrt{\sum_{1}^{N} \|\phi_{bl} - \phi_{tar}\|^2 * \mathbb{E}} - \frac{\tau}{\sqrt{B}} \sqrt{\sum_{1}^{B-N} \|\phi_{bl}\|^2 * (\mathbb{I} - \mathbb{E})}, \tag{5}$$

where N indicates the number of false positives and false negatives in a single batch and B is the batch size and $\mathbb{E}$ represents the distribution of false positives and false negatives to achieve the filtering. With this designed compression remedy loss, the model could learn to recover the information difference between the high-quality images and the low-quality images. The filtering mechanism can help obtain more effective samples for modeling these differences. In our experiments, we set $\tau = 0.2$ based on our observations to control the balance

between pushing and pulling in the compression remedy loss $\mathcal{L}_{rem}$.

2) Regularization Loss: Considering the generated image is added into every image, to achieve the trade-off between accurately identifying the false negative samples while not distracting the true positives, we design an effective way to normalize the generated bleach. We formulate this regularization loss as follows:

$$\mathcal{L}_{reg} = \sqrt{\frac{1}{M}\sum_{1}^{M} \|k \cdot \mathbf{I}_{bl} - \mathbf{I}_0\|^2}, \tag{6}$$

where M is the number of elements for the input image. $\mathbf{I}_{bl}$ is the bleached image and $\mathbf{I}_0$ represents the LQ face forgery image after compression. We introduce the hyper-parameter k to normalize/control the value of the generated bleach. A very large value of the generated bleach will lead to the corruption of the bleached image (the combination of the original LQ image and the synthesized bleach). The design of k could effectively alleviate the corruption of bleached images.

D. Final Objective Function

In this section, we discuss the final objective function adopted in our paper. Besides the proposed two above-mentioned loss functions, we also choose the cross-entropy loss $\mathcal{L}_{ce}$ to optimize the Deepfake detection model. The final loss function is described as:

$$\mathcal{L} = \mathcal{L}_{rem} + \alpha\mathcal{L}_{reg} + \gamma\mathcal{L}_{ce}, \tag{7}$$

where α and γ are hyper-parameters to balance the contribution of the proposed three components. Considering the imbalance between true negative and false positive samples, we also conduct a balanced offset training strategy to mitigate distribution differences between true negative and false positive samples in the wrongly classified examples. We set $\alpha = 1$ and $\gamma = 2$ in our all experiments. More ablation studies about the hyper-parameter selection will be included in Section IV-C.

IV. Experiments

A. Experimental Setting

1) Datasets: To validate the universal adaptability of our model, we conduct our experiments on the challenging FaceForensics++ [20] dataset. FaceForensics++ is the most widely applied dataset in most Deepfake detection algorithms, which contains 1,000 real videos collected from the Internet. Each real one is manipulated by 4 forgery approaches: DeepFakes, NeuralTexture [53], FaceSwap [54] and Face2Face [1], respectively. There are three subversions of the FaceForensics++ dataset according to three compression levels: 1) Raw (original quality), 2) HQ (quantization rate of 23), and 3) LQ (quantization rate of 40). We choose high-quality and low-quality versions to perform cross-compression Deepfake detection. For data split, we strictly follow the official train/val/test split (720/140/140) to perform a fair comparison with other methods.

2) Implement Details: For dataset pre-processing, we extract 300 random frames for each video for all training datasets. And we keep the same sampling for testing datasets to ensure a fair comparison. Following the experimental setting of [32], we use a state-of-art face extractor RetinaFace [55] on each extracted frame and save every aligned facial image as the size of 299×299 which is the same as the setting in FaceForensics++ [20]. To simulate the real situation, we choose the HQ (C23) version of the FaceForensics++ dataset to train the classifier and then test the trained classifier on the LQ (C40) version. We formulate all wrongly classified examples: FP (false positive) and FN (false negative) respectively based on the training set of the LQ data. After that, we randomly sample the correctly classified data to formulate a balanced offset training dataset for our bleach generator. We also combine these samples as the training datasets for our bleach generator and the parameters of the basic classifier are frozen during the whole training process. For our bleach generator, we optimize this module with Adam optimizer [56] with a learning rate of 0.0002 and the weight decay of $1e^{-8}$. We strictly follow the official experimental setting of the adopted Deepfake detection algorithms. All these experiments are mainly conducted on a machine with 4 Geforce RTX 2080Ti GPUs.

3) Backbones: In our paper, we choose three widely used Deepfake detection backbones (including Xception [19], MesoInception4 [14] and F^3-Net [11]) and two general image classification backbones (including ResNet-50 and EfficientNet-B4) to explore the effectiveness and generalization ability of our approach.

- **Xception** [19]. We first adopt a widely used Xception for the Deepfake detection, which is a DCNN architecture pre-trained on ImageNet dataset [57]. The separable convolutions and the residual connections provide a strong ability to achieve robust and accurate Deepfake detection.
- **MesoInception4** [14]. The MesoInception4 is also adopted in our experiments. Inspired by the inception neural network, this method adopts two inception modules [58] and two traditional convolution modules to boost the Deepfake detection accuracy.
- **F^3-Net** [11]. Finally, a recent representative F^3-Net utilized frequency domain information to mine subtle artifacts and compression error features. Two complementary modules: frequency-aware decomposition (FAD) and local frequency statistics (LFS) were introduced to decompose image components and local frequency domain statistical information. Finally, the dual-stream network features were fused by Mixblock. The FAD module integrated the components of several frequency bands to obtain a wider range of information. LFS adopted the local frequency statistics to calculate the average frequency response in patch regions to form a multi-channel characteristic diagram.
- **ResNet-50** [59] is a variant of the ResNet architecture, a widely used network backbone in image classification, object detection, and segmentation. The skip connections introduced by the ResNet architecture allow the model to bypass some of the layers in the network. This residual learning helps to alleviate the vanishing gradient problem.

TABLE III
EXPERIMENTAL RESULTS OF ACC. AND AUC UNDER VARIOUS FACE FORGERY SETTINGS: FACESWAP, FACE2FACE, DEEPFAKES AND NEURALTEXTURE. ALL THESE EXPERIMENTS ARE CONDUCTED BY USING 300 FRAMES FOR EACH VIDEO.

RECG	Backbone	FaceSwap		Face2Face		DeepFakes		NeuralTexture	
		Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC
×	Xception [19]	70.95	84.14	52.86	75.38	**86.96**	93.94	62.99	72.41
✓		**79.62**	**87.24**	**67.30**	**82.26**	86.87	**94.28**	**67.94**	**75.05**
×	MesoInception4 [14]	57.99	71.39	50.37	60.21	68.24	84.24	63.45	68.81
✓		**60.64**	**71.46**	**51.44**	**66.77**	**71.71**	**84.60**	**64.08**	**69.31**
×	F^3-Net [11]	71.35	84.93	51.62	61.41	86.76	**94.17**	65.01	73.52
✓		**77.55**	**86.27**	**59.82**	**65.96**	**87.63**	93.60	**71.89**	**78.25**
×	ResNet-50 [59]	73.15	86.07	56.60	78.09	87.54	95.26	64.09	74.70
✓		**80.47**	**88.49**	**66.65**	**79.98**	**88.31**	**95.29**	**69.69**	**76.49**
×	EfficientNet-B4 [60]	74.47	85.26	52.10	63.87	87.05	95.49	65.60	76.73
✓		**81.32**	**89.72**	**74.54**	**82.66**	**88.93**	**96.17**	**71.16**	**79.03**

ResNet-50 also includes batch normalization and ReLU activation functions, which help to improve the speed and accuracy of the model.

- **EfficientNet-B4** [60] is a widely-used neural network architecture designed to be both accurate and computationally efficient. It is a variant of the EfficientNet family of models that have achieved state-of-the-art performance on several computer vision tasks. EfficientNet-B4 optimized the network architecture for both depth, width, and resolution simultaneously and adopted a novel inverted bottleneck block to improve performance. These designs allow the network to achieve high accuracy while still being computationally efficient.

4) Evaluation Metrics: We report the classification accuracy (denoted as **Acc.**) and the area under the curve (**AUC**) on the low-quality version of the FaceForensics++ dataset. The classification/identification accuracy is the significant metric in most of the Deepfake detection tasks, directly measuring if the Deepfake detection model succeeds in accurately identifying the modified images. AUC is also proposed as the evaluation metric for different Deepfake detection experiments. We also report the number of false negative (FN) and false positive (FP) examples in our ablation studies to show the effectiveness of the proposed method.

B. Quantitative Comparison

1) FaceSwap and Face2Face: We conduct our experiments on two sub-forgery sets of the FaceForensics++ dataset. We first analyze the difference between 2 FaceForensics++ sub-dataset from the distribution of feature representation outputs. As shown in Fig. 4, in the vicinity of the origin (yellow rectangle area), the bias of distribution between failed and correct examples can be observed. The blue point cluster more densely, whereas the red point is sparse around the original point. Face2Face has a more complicated distribution of feature representations. Both positive and negative examples arrange densely around the original point.

Xception. XceptionNet [19] is a traditional convolutional neural network with separate convolutions and residual connections. Xception has shown its strong ability on performing robust and accurate face recognition and becomes a wildly used Deepfake detection algorithm. The batch size is set to 14 during the training procedure under this setting. The quantitative results are reported in Table III. As observed, the proposed method can heavily promote the Deepfake detection accuracy under compression without retraining the whole network. With the effective former bleaching module, the latter Deepfake detection model could discriminate the tiny differences between the real and forgery samples.

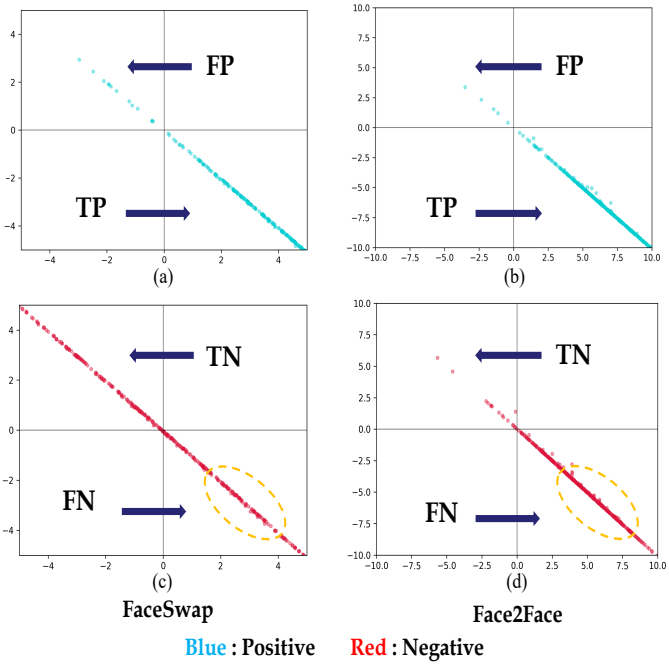


Fig. 4. The representation distribution of the images generated by FaceSwap and Face2Face based on the Deepfake detection backbone Xception. The red and blue points indicate the negatively predicted examples and positively predicted examples respectively. The points in the second and fourth quadrant shown in (a) and (b) represent FP and TP samples. The points in the second and fourth quadrant shown in (c) and (d) represent FN and TN samples. The region covered by the yellow dotted line describes the scenario in which a large number of TN samples are misclassified with high confidence. Almost all of the negative examples generated by Face2Face are misclassified with higher confidence, which demonstrates that the Xception classifier is extremely fragile when dealing with the highly compressed samples from the Face2Face dataset.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3301242

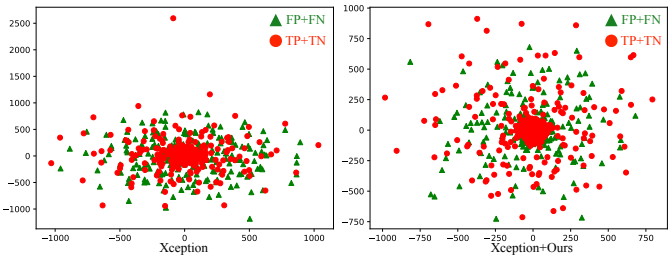


Fig. 5. The t-SNE visualization of feature representation ϕ at the last layer. The proposed method could centralize the FP and FN samples, which also supports that our method has a strong ability to bleach the wrongly classified samples.

We first analyze the robustness of the existing Deepfake detection algorithm Xception under compression. We report the representation distribution of the images generated by FaceSwap and Face2Face based on the Deepfake detection model Xception in Fig. 4, which represents a low robustness to the compression. For further explanation, we also provide the t-SNE visualization of the distribution of the representations from the last layer of Xception to show how our method contributes to improving the recognition performance. The distribution visualization is illustrated in Fig. 5. The proposed method could get a more reasonable distribution.

MesoInception4. We then perform experiments based on MesoInception4, which is another representative Deepfake detection network architecture. The dimension of the last layer of MesoInception4 is only 16. We introduced two convolutional modules to increase the dimension of the representations for stacking more information. The output after the two convolutional layers is expanded to 512 and then is reshaped to $4 \times 4 \times 32$, which is regarded as the input of DCGAN. The experimental results are also reported in Table III. The performance gain of the proposed method based on MesoInception4 is smaller than adopting the Xception as the backbone. We attribute this to the reason that the information contained in the 16-dimensional feature representation in MesoInception4 is limited. It is difficult to generate a $299 \times 299 \times 3$ bleach from the small size feature representation. In other words, it is also challenging and difficult to generate and model the information difference between the *original* and *compressed* Deepfake images from such abstract and limited feature representation.

F^3-Net. We combine the proposed method with F^3-Net and set the batch size to 16 on all datasets. Under this setting, F^3-Net did not normalize the output of the last layer. To avoid the collapse of the model, we empirically multiply the output with the additional hyper-parameter $s = 0.01$ to generate a more reasonable bleach. We will quantitatively and qualitatively discuss the influence of choosing different values of s in Section IV-C1. As reported in Table III, when $s = 0.01$, we can improve the detection accuracy from 51.62% to 59.82% on the challenging Face2Face dataset, and from 71.35% to 77.55% on FaceSwap images. Meanwhile, the AUC scores are also improved on both two forgery methods of compressed data.

ResNet-50 and EfficientNet-B4. Two widely used image classification backbones are also included. Following the same experimental setting, the quantitative experimental results are reported in Table III. The proposed method could achieve a very large performance improvement based on both ResNet-50 and EfficientNet-B4 backbones on the challenging Face2Face setting: from 56.60% to 66.65% accuracy improvement for ResNet-50 and from 52.10% to 74.54% accuracy improvement for EfficientNet-B4. The experimental results demonstrate that the proposed method is a general framework, which could lead to performance gains based on various network backbones. The proposed method works better performance gains when it is combined with compact network backbones (*e.g.*, Xception, ResNet-50 and EfficientNet-B4), since these designed network backbones could yield more compact feature representations that could store more information from the Deepfake images, leading to better performance.

2) DeepFakes and NeuralTexture: In this section, we provide more experimental results to demonstrate the effectiveness of the proposed method on the other two types of forgery approaches: *DeepFakes* and *NeuralTexture*.

DeepFakes. We first evaluate the performance of the proposed method on the images generated by the DeepFakes algorithms. Similarly, we choose the same five network backbones: MesoInception4 [14], Xception [19], F^3-Net [11], ResNet-50 [59] and EfficientNet-B4 [60] in our experiments. We report all the quantitative results in Table III. The proposed method is plug-and-play and could results in performance gains on both accuracy and AUC scores based on various network backbones.

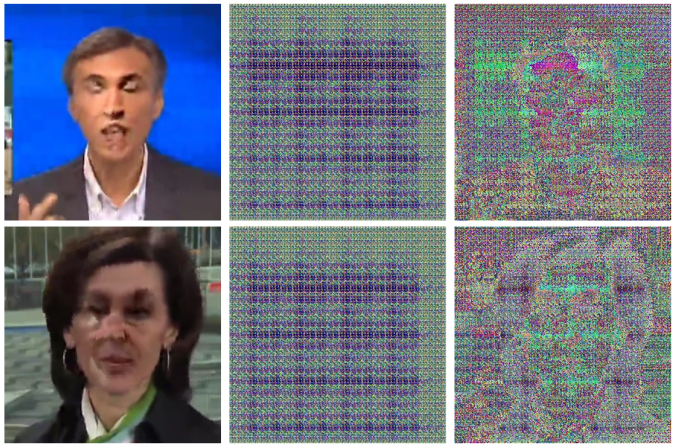
NeuralTexture. NeuralTexture [53] is an advanced face forgery approach compared with the other three methods. It is much more difficult to detect the face forgery videos generated by NeuralTexture since this method intentionally introduced the subtle small-scale artifacts, which are inconspicuous to both models and human beings. With the compression, these subtle visual artifacts are further weakened, making the face forgery detection problem extremely challenging. The proposed method can improve both Acc. and AUC scores based on all three backbones, which indicates that the proposed method is a general and effective module for various Deepfake detection algorithms.

In summary, even though the proposed method could improve the overall detection accuracy in most cases, we still face some challenges and there is a lot of room for us to improve the Deepfake detection accuracy. For instance, to improve the overall detection accuracy, the model tends to reduce the FP samples while failing to maintain the detection efficiency of false negative samples. We attribute this phenomenon to the reason that the Deepfake detection algorithms cannot distinguish the false positive and false negative samples in the feature space. When false positive and false negative samples are mixed very close, the model would inevitably misclassify the true negative into false negative while reducing the FP samples. This situation goes worse when the false positive and false negative samples are highly imbalanced.

Finally, to further compare the effectiveness of the proposed method with the existing state-of-the-art Deepfake detection algorithms, we follow the experimental setting of [44] and

TABLE IV
THE EXPERIMENTAL RESULT COMPARISON BETWEEN OUR METHOD AND PREVIOUS SOTA METHODS.

Dataset	Method	FaceForensics++ (C40) Acc.
FaceSwap	Rossler *et al.* [20]	88.09
	Dogonadze *et al.* [61]	**90.02**
	F^3-Net [11]	89.58
	Ours	89.56
Face2Face	Rossler *et al.* [20]	80.21
	Dogonadze *et al.* [61]	**83.44**
	F^3-Net [11]	81.48
	Ours	82.23
DeepFake	Rossler *et al.* [20]	92.43
	Dogonadze *et al.* [61]	**93.97**
	F^3-Net [11]	93.06
	Ours	92.38
NeuralTex.	Rossler *et al.* [20]	56.75
	Dogonadze *et al.* [61]	61.12
	F^3-Net [11]	61.95
	Ours	**75.96**



Original image Generated bleach Bleached image

Fig. 6. The original images, the synthesized bleach and the corresponding corrupted bleached images after adding the generated bleach into the original images.

report the experimental results in Table IV. We compare the proposed method with the recently advanced algorithms. As reported, the proposed method could achieve much better results than the existing algorithms on the most challenging NeuralTexture dataset. Under other settings, the proposed method could also achieve comparable performance with other algorithms.

C. Ablation Studies

1) Selection of s in F^3-Net: We discuss the influence of the scale of the generated bleach based on F^3-Net. The comparisons between the original images, the synthesized bleach and the bleached images are illustrated in Fig. 6. We observe that the image output will be corrupted if we directly add the bleach to the compressed image. Thus, to avoid this, we design a beach scale s to project the generated bleach into an appropriate scale. We conduct extensive experiments by using different values of s and report the quantitative results in Table V. When $s = 1.0$, there is a huge performance drop since the uncontrolled bleach contributes to the Deepfake detection model most, which results in the classifier cannot distinguish between positive and negative samples. Among all the settings, our method has achieved the best performance when $s = 0.01$.

TABLE V
THE EXPERIMENTAL RESULTS OF USING DIFFERENT VALUES OF SCALE s BASED ON F^3-NET.

RECG	Backbone	FaceSwap Acc.	FaceSwap AUC
$s = 1$	F^3-Net [11]	54.16	57.79
$s = 0.1$		71.83	80.46
$s = 0.001$		77.41	86.19
$s = 0.01$		**77.55**	**86.27**

TABLE VI
THE EXPERIMENTAL RESULTS OF USING DIFFERENT VALUES OF α AND λ BASED ON VARIOUS BACKBONES.

RECG	Backbone	FaceSwap ACC	FaceSwap AUC
$\alpha = 1, \lambda = 10$	Xception [19]	77.18	85.27
$\alpha = 1, \lambda = 5$		78.86	88.16
$\alpha = 1, \lambda = 2$		**79.62**	**87.24**
$\alpha = 1, \lambda = 10$	MesoInception4 [14]	57.62	70.19
$\alpha = 1, \lambda = 5$		59.27	70.01
$\alpha = 1, \lambda = 2$		**60.64**	**71.46**
$\alpha = 1, \lambda = 10$	F^3-Net [11]	76.73	85.89
$\alpha = 1, \lambda = 5$		77.14	86.32
$\alpha = 1, \lambda = 2$		**77.55**	**86.27**

2) Hyper-parameter Selection: We have conducted experiments of choosing different values of α and λ to obtain the optimal combinations of the hyper-parameters and we report the experimental results in Table VI. As illustrated in Table VI, we can achieve the best results when $\alpha = 1, \lambda = 2$.

3) Effectiveness of $\mathcal{L}_{rem}$: To validate the effectiveness of the proposed compression remedy loss, we design experiments to measure how $\mathcal{L}_{rem}$ affects the accuracy and the number of failed examples. We perform experiments on the FaceSwap dataset and keep other experimental settings the same to make a fair comparison. The results on the FaceSwap dataset are reported in Table VII. The results present the improvement of identification accuracy and AUC based on the backbone Xception on the FaceSwap dataset.

4) Effectiveness of Balanced Offset Training Strategy: As above-mentioned, the proposed balanced offset training strategy plays an important role to alleviate the imbalance between false positives and false negatives. To demonstrate this, we evaluate the proposed balanced offset training on the FaceSwap dataset. We compare the experimental results of using different values ($m = -0.7, 0, +0.7$ respectively) for our

TABLE VII
THE EFFECTIVENESS OF THE PROPOSED COMPRESSION REMEDY LOSS $\mathcal{L}_{rem}$.

$\mathcal{L}_{rem}$	Backbone	FaceSwap Acc.	FaceSwap AUC	FaceSwap FP	FaceSwap FN
×	Xception [19]	77.55	86.46	14,465	4,396
✓		**79.62**	**87.24**	10,247	6,873

TABLE VIII
THE EFFECTIVENESS OF THE PROPOSED BALANCED OFFSET TRAINING STRATEGY. WE CAN MAKE THE FN AND FP SAMPLES MORE BALANCED, WHICH CAN ALLEVIATE THE INFLUENCE OF THE DATA IMBALANCE.

RECG	Backbone	FaceSwap Acc.	FP	FN	FN/FP
$m = -0.7$	Xception [19]	78.52	13,305	4,735	0.356
$m = 0$		78.93	12,331	5,372	0.436
$m = 0.7$		**79.62**	10,247	6,873	0.671

TABLE IX
THE EFFECTIVENESS OF THE PROPOSED $\mathcal{L}_{reg}$ LOSS. B/I INDICATES THE PIXEL-LEVEL RATIO OF THE BLEACHED IMAGE OVER THE ORIGINAL IMAGE. IF WE SET NO CONSTRAINT FOR THE REGULARIZATION LOSS, THE MODEL WOULD COLLAPSE AND FAIL TO ACHIEVE MEANINGFUL BLEACHING.

$\mathcal{L}_{reg}$	Backbone	FaceSwap Acc.	FP	FN	B/I
×	Xception [19]	50.02	240	41,740	1.8211
✓		**79.62**	10,247	6,873	0.0147

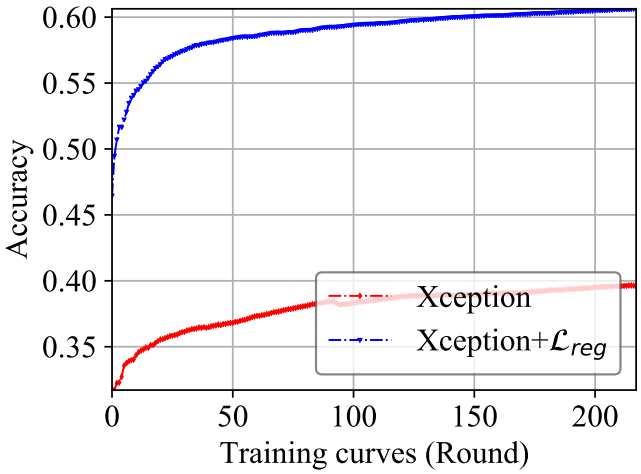


Fig. 7. The training curves of 1) using the proposed $\mathcal{L}_{reg}$ and 2) not using the $\mathcal{L}_{reg}$. Each round contains 20 iterations. As illustrated, the proposed $\mathcal{L}_{reg}$ can stabilize the training procedure and enhance the Deepfake detection performance.

balanced offset training strategy and the results are reported in Table VIII. The imbalance between false positives and false negatives is reduced by the proposed balanced offset training strategy. The proposed method provides a feasible solution for the extreme circumstances when the model cannot learn a bleached image due to a few false negative or false positive samples.

5) Effectiveness of $\mathcal{L}_{reg}$: The regularization loss is a straightforward approach to provide the constraint for generating the pixel-level outputs to balance the ability of bleaching and promote the overall identification accuracy. A not-limited generated bleach will lead to the corruption of the bleached image. In detail, if false negative examples are dominant among all the training samples, the model will learn a large bleach for recovering the false negatives to true negatives with the assistance of classification loss. However, this phenomenon will bring about a rise of false positives since the model has resulted in the same influence to true positives. As discussed in previous work [62]–[65], even a small scale of noise can disrupt the classifier. In our experiments, we observe that the normalized bleach can effectively recover the information loss caused by compression. Hence, we set $k = 50$ in Equ 6 and achieve better performance as shown in Table IX. Besides, we also explored how the regularization loss affects the training process on the FaceSwap dataset. As shown in Fig. 7, the proposed $\mathcal{L}_{reg}$ could help preserve the stability of the training and promote identification accuracy.

6) Compression-agnostic: Furthermore, we also consider the utility of our method in the real-world setting. There are various formats of compressions during the transmission procedures. We have designed experiments on cross-compression tests. In detail, we adopt the trained bleach generator based on the data with a specific compression level and perform the Deepfake detection with another compression scale. Under this setting, we adopt low-quality data for training and high-quality data for testing. The quantitative results are reported in Table X. Please note, during the testing procedure, we do not retrain our bleach generator. We adopt various backbones to conduct the corresponding experiments while other experimental settings are the same. As reported, the proposed method has resistance to various compressions.

TABLE X
THE EXPERIMENTAL RESULTS OF OUR METHOD UNDER THE COMPRESSION-AGNOSTIC SETTING.

RECG	Training data	Backbone	C40 Acc.	C40 AUC
×	C23	Xception [19]	70.95	81.73
✓			**72.36**	**83.27**
×	C23	MesoInception4 [14]	57.99	68.83
✓			**59.05**	**72.56**
×	C23	F^3-Net [11]	71.35	85.61
✓			**71.84**	**86.35**

D. Discussions

1) Generalization Ability: We have also explored the generalization ability to the unseen compressed Deepfake images under the real-world setting. We first collect 360 Deepfake and real images from the Internet and these images are unseen for our trained Deepfake detection model. It is worth noting that the trained Deepfake detection model is **frozen** and without retraining during the inference stage. Furthermore, we simulate the real-world compression by uploading all the images to https://www.canva.cn/ for obtaining the compressed images. Then we download the compressed Deepfake images for testing. We provide the qualitative comparison between without and with compression in Fig. 8. As illustrated, the compression leads to the **patch-like** visual artifacts. Please zoom in for checking a more detailed comparison. The pre-trained Deepfake detection model only achieved **52.92**% accuracy, which indicates that the model fully failed to detect the Deepfake images under the compression setting. In contrast,

the proposed method could achieve **71.31**% accuracy even without retraining (71.31 vs. 52.92). The proposed method could achieve 18.4% accuracy improvement without retraining and could effectively detect the Deepfake images under the compression setting.

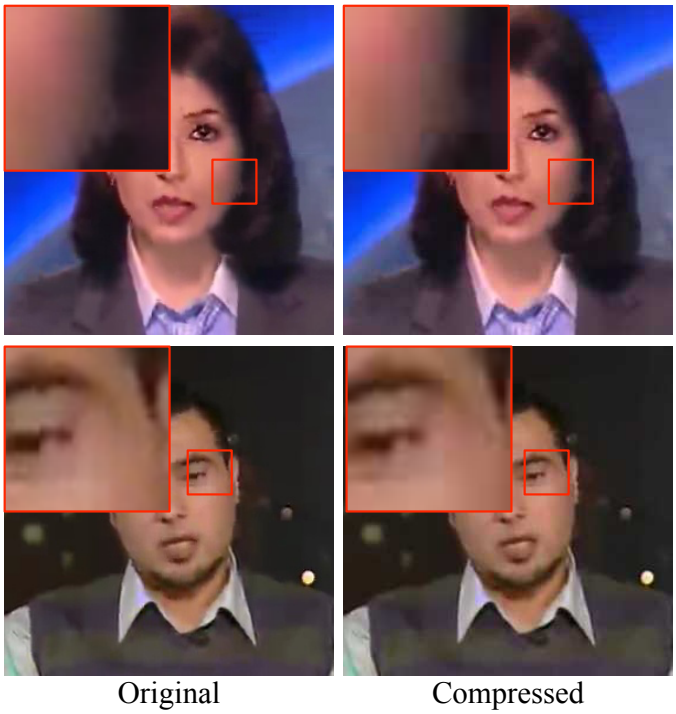


Fig. 8. The compression leads to patch-like visual artifacts and leads to the failure of the Deepfake detection. Best viewed in color.

2) Comparison with Previous Algorithms: We provide more detailed discussions about the comparison between the proposed method and existing Deepfake detection algorithms under the compression setting. A direct quantitative comparison with the existing algorithms [44], [48], [49], [66], [67] is illustrated in Table XI. The two stream algorithm [48] could achieve the best results among all the methods. It requires the paired HQ and LQ data for training and even for evaluation. However, it is not impossible to access such paired data for inference in real-world scenarios. Particularly, in most of the scenarios, we cannot obtain the original HQ data. Furthermore, during the inference stage, we are not given whether the input Deepfake images have been compressed. We conduct the experiments based on the same experimental setting and report the corresponding experimental results in Table XI. ADD [44] also requires the paired LQ-HQ data for training. At the inference procedure, the model should also know whether the Deepfake images are compressed and feed the input Deepfake images into the corresponding branch. As reported, the proposed method could achieve a competitive performance compared with the existing specially designed Deepfake detection algorithms under compression (especially, 75.96% accuracy under the challenging NeuralTexture setting). We argue that our method has two main advantages over the previous Deepfake detection algorithms under compression:

- The proposed method does **not** require paired LQ-HQ data for training. In the real-world setting, we usually do not have such paired data and only have the compressed LQ data. The original HQ data are not available for training.
- The proposed method is **compression-agnostic**. The previous algorithms should know whether the input Deepfake images are compressed in advance and then feed the input Deepfake images to the corresponding branch.

The proposed method could work effectively under a more practical and generalized setting. Besides, our method has also achieved various degrees of performance gains based on different network backbones as a plug-and-play module. To further demonstrate the effectiveness of the proposed method, we have also included three recent works (LipForensics [49], AMTEN [66] and FTDN [67]) for comparison, which are specially designed to perform Deepfake detection under the compression setting without using the paired LQ-HQ data for training. As reported, the proposed method could achieve the best Deepfake detection results among all these unpaired Deepfake detection algorithms.

[1]We follow the official construction of their provided codes https://github.com/ahaliassos/LipForensics to conduct a fair comparison and report the corresponding experimental results.

TABLE XI
ACCURACY COMPARISON WITH THE EXISTING DEEPFAKE DETECTION ALGORITHMS UNDER COMPRESSION. † INDICATES THAT IT REQUIRES THE PAIRED LQ-HQ DATA AS INPUT DURING BOTH THE TRAINING AND INFERENCE STAGES. THE BEST RESULTS UNDER THE **UNPAIRED** SETTING ARE IN BOLD. — INDICATES THAT THE RESULTS ARE NOT REPORTED.

Method	FaceSwap	Face2Face	DeepFakes	NeuralTexTure	Average
Two_stream† [48]	93.51	91.13	95.90	79.18	89.93
ADD† [44]	92.49	85.42	95.50	68.53	85.48
LipForensics[1] [49]	73.71	67.91	63.70	65.50	67.71
AMTEN [66]	-	-	-	-	84.16
FTDN [67]	-	**83.19**	83.55	-	-
Ours	**89.56**	82.23	**92.38**	**75.96**	**85.03**

V. CONCLUSION

In this study, we introduced a straightforward and efficient bleach module that can be seamlessly integrated with existing Deepfake detection algorithms to enhance the detection of face forgery under compression. Utilizing the feature representations generated by a pre-trained Deepfake detection model, our proposed bleach generator generates a bleach that can be added to the original Deepfake images, resulting in a bleached image that can be accurately identified by the Deepfake detection model. The core concept of our approach is to treat compression as a specific type of attack, and we employ a simple DCGAN architecture to generate the compression remedy. To address misclassifications caused by the compressed Deepfake images, we design two loss functions that modify the feature distribution in the feature space. We conduct extensive evaluations on four commonly used face forgery detection datasets, employing five different network backbones. The results demonstrate noticeable performance improvements compared to state-of-the-art detection models, validating the effectiveness of our proposed bleach design.

Additionally, we conduct comprehensive experiments to assess the effectiveness of each component of our proposed method.

VI. Acknowledge

This work was supported in part by the National Natural Science Foundation of China under grant U20B2063, 62220106008, and 62102070, and in part by the Sichuan Science and Technology Program under grant 2023NSFSC1392.

References

[1] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2387–2395.

[2] R. Natsume, T. Yatagawa, and S. Morishima, "Rsgan: face swapping and editing using face and hair representation in latent spaces," *arXiv preprint arXiv:1804.03447*, 2018.

[3] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 818–833.

[4] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy, "Reenactgan: Learning to reenact faces via boundary transfer," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[5] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7184–7193.

[6] Z. Zheng, Y. Hu, Y. Bin, X. Xu, Y. Yang, and H. T. Shen, "Composition-aware image steganography through adversarial self-generated supervision," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[7] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *Advances in Neural Information Processing Systems (Neurips)*, vol. 29, pp. 613–621, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems (Neurips)*, vol. 27, 2014.

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

[10] 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 (TPAMI)*, 2020.

[11] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 86–103.

[12] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face x-ray for more general face forgery detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5001–5010.

[13] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5781–5790.

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

[15] Z. Liu, X. Qi, and P. H. Torr, "Global texture enhancement for fake face detection in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8060–8069.

[16] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, "Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6458–6467.

[17] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 772–781.

[18] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 3247–3258.

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258.

[20] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1–11.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[22] J. Song, J. Zhang, L. Gao, Z. Zhao, and H. T. Shen, "Agegan++: Face aging and rejuvenation with dual conditional gans," *IEEE Transactions on Multimedia (TMM)*, vol. 24, pp. 791–804, 2021.

[23] Z. Zheng, Z. Yu, H. Zheng, Y. Yang, and H. T. Shen, "One-shot image-to-image translation via part-global learning with a multi-adversarial framework," *IEEE Transactions on Multimedia (TMM)*, vol. 24, pp. 480–491, 2021.

[24] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[25] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8789–8797.

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

[27] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," in *SIGGRAPH Asia*, 2011, pp. 1–10.

[28] F. Liu, R. Zhu, D. Zeng, Q. Zhao, and X. Liu, "Disentangling features in 3d face shapes for joint face reconstruction and recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5216–5225.

[29] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment." *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 183–1, 2015.

[30] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang, and Y. Liu, "Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces," *arXiv preprint arXiv:1909.06122*, 2019.

[31] Z. Zheng, Y. Bin, X. Lu, Y. Wu, Y. Yang, and H. T. Shen, "Asynchronous generative adversarial network for asymmetric unpaired image-to-image translation," *IEEE Transactions on Multimedia*, 2022.

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

[33] X. Li, Y. Lang, Y. Chen, X. Mao, Y. He, S. Wang, H. Xue, and Q. Lu, "Sharp multiple instance learning for deepfake video detection," in *ACM International Conference on Multimedia (ACM MM)*, 2020, pp. 1864–1872.

[34] S. A. Khan and H. Dai, "Video transformer for deepfake detection with incremental learning," in *ACM International Conference on Multimedia (ACM MM)*, 2021, pp. 1821–1828.

[35] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-forensics for face swapping videos via adversarial training," *IEEE Transactions on Multimedia (TMM)*, 2021.

[36] 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 (TPAMI)*, 2021.

[37] L. Zhang, T. Qiao, M. Xu, N. Zheng, and S. Xie, "Unsupervised learning-based framework for deepfake video detection," *IEEE Transactions on Multimedia (TMM)*, 2022.

[38] J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, Y.-G. Jiang, and S.-N. Li, "M2tr: Multi-modal multi-scale transformers for deepfake detection," in *Proceedings of the 2022 International Conference on Multimedia Retrieval*, 2022, pp. 615–623.

[39] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "Wilddeepfake: A challenging real-world dataset for deepfake detection," in *ACM International Conference on Multimedia (ACM MM)*, 2020, pp. 2382–2390.

[40] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3207–3216.

[41] P. Yu, J. Fei, Z. Xia, Z. Zhou, and J. Weng, "Improving generalization by commonality learning in face forgery detection," *IEEE Transactions on Information Forensics and Security (TIFS)*, 2022.

[42] J. Hu, X. Liao, W. Wang, and Z. Qin, "Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. PP, no. 99, pp. 1–1, 2021.

[43] L. Sun, H. Zhang, X. Mao, S. Guo, and Y. Hu, "Super-resolution reconstruction detection method for deepfake hard compressed videos," *Journal of Electronics and Information Technology*, vol. 43, no. 200531, p. 2967, 2021.

[44] S. Woo *et al.*, "Add: Frequency attention and multi-view based knowledge distillation to detect low-quality compressed deepfake images," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 122–130.

[45] J. Zhang, J. Ni, and H. Xie, "Deepfake videos detection using self-supervised decoupling network," in *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[46] M. Li, B. Liu, Y. Hu, L. Zhang, and S. Wang, "Deepfake detection using robust spatial and temporal features from facial landmarks," in *IEEE International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2021, pp. 1–6.

[47] Y. Huang, F. Juefei-Xu, Q. Guo, Y. Liu, and G. Pu, "Fakelocator: Robust localization of gan-based face manipulations," *IEEE Transactions on Information Forensics and Security (TIFS)*, 2022.

[48] S. Cao, Q. Zou, X. Mao, D. Ye, and Z. Wang, "Metric learning for anti-compression facial forgery detection," in *ACM International Conference on Multimedia (ACM MM)*, 2021, pp. 1929–1937.

[49] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5039–5049.

[50] K. Roth, Y. Kilcher, and T. Hofmann, "The odds are odd: A statistical test for detecting adversarial examples," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 5498–5507.

[51] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[52] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 499–515.

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

[54] J. Zhang, X. Zeng, Y. Pan, Y. Liu, Y. Ding, and C. Fan, "Faceswapnet: Landmark guided many-to-many face reenactment," *arXiv preprint arXiv:1905.11805*, vol. 2, 2019.

[55] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[60] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 6105–6114.

[61] N. Dogonadze, J. Obernosterer, and J. Hou, "Deep face forgery detection," *arXiv preprint arXiv:2004.11804*, 2020.

[62] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 387–402.

[63] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[64] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[65] P. Tabacof, J. Tavares, and E. Valle, "Adversarial images for variational autoencoders," *arXiv preprint arXiv:1612.00155*, 2016.

[66] 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, p. 103170, 2021.

[67] Y. Sun, Z. Zhang, I. Echizen, H. H. Nguyen, C. Qiu, and L. Sun, "Face forgery detection based on facial region displacement trajectory series," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 633–642.

Congrui Li received his B.S. degree from Huazhong University of Science and Technology. He is currently a student at the Center for Future Media, University of Electronic Science and Technology of China, Chengdu, China. His research interests include Deepfake detection and computer vision.

Ziqiang Zheng received his B.Eng. degree in communication engineering from the Ocean University of China in 2019. He is currently a research assistant with the Center for Future Media, the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include multimedia content analysis and computer vision.

Yi Bin is currently with the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He received his Ph.D. degree from UESTC in 2020. His research interests include multimedia analysis, vision understanding, and deep learning.

Guoqing Wang (Member, IEEE) received a Ph.D. degree from The University of New South Wales, Australia, in 2021. He is currently with the School of Computer Science and Engineering, University of Electronic of Science and Technology of China. He has authored and co-authored more than 40 scientific articles at top venues, including IJCV, IEEE TIP, IEEE TIFS, ICCV, ACM MM, etc. His research work at UNSW has been recognized as the Australian Dean's Award for Outstanding Ph.D. Theses. His research interests include machine learning and unmanned system, with special emphasis on cognition and embodied agents.

Yang Yang (M'16) received a bachelor's degree from Jilin University, Changchun, China, in 2006, a master's degree from Peking University, Beijing, China, in 2009, and a Ph.D. degree from The University of Queensland, Brisbane, Australia, in 2012, all in computer science. He is currently with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include multimedia content analysis, computer vision, and social media analytics.

Xuesheng Li received his B.E. degree in electrical engineering and automation in 2001 from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, and his M.E. and a Ph.D. degree in mechatronics in 2004 and 2012 from UESTC. Since 2004, Dr.Li has been an assistant and associate professor at the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China. His present research interests include nonlinear acoustics, artificial intelligence, the Internet of Things, ultrasonic actuators, piezoelectric transducers, and other electronic mechanical devices.

Heng Tao Shen is the Dean of the School of Computer Science and Engineering, and the Executive Dean of AI Research Institute at the University of Electronic Science and Technology of China (UESTC). He obtained his BSc with 1st class Honours and Ph.D. from the Department of Computer Science, the National University of Singapore in 2000 and 2004 respectively. His research interests mainly include Multimedia Search, Computer Vision, Artificial Intelligence, and Big Data Management. He is/was an Associate Editor of ACM Transactions of Data Science, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, IEEE Transactions on Knowledge and Data Engineering, and Pattern Recognition. He is a Fellow of ACM, IEEE, and OSA.