Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Full-defense framework: multi-level deepfake detection and source tracing*#

Hui SHI†1, Guibin WANG[1], Yanni LI†‡2, Rujia QI[1]

[1]School of Computer Science and Artificial Intelligence, Liaoning Normal University, Dalian 116021, China
[2]School of Management, Liaoning University of International Business and Economics, Dalian 116029, China
†E-mail: shihui_jiayou@lnnu.edu.cn; 841686948@qq.com

Received Nov. 17, 2024; Revision accepted Feb. 13, 2025; Crosschecked Aug. 29, 2025

Abstract: Deepfake poses significant threats to various fields, including politics, journalism, and entertainment. Although many defense methods against deepfake have been proposed based on either passive detection or proactive defense, few have achieved both passive detection and proactive defense. To address this issue, we propose a full-defense framework (FDF) based on cross-domain feature fusion and separable watermarks (SepMark) to achieve copyright protection and deepfake detection, combining the ideas of passive detection and proactive defense. The proactive defense module consists of one encoder and two separable decoders, where the encoder embeds one watermark into the protected face, and two decoders separately extract two watermarks with different robustness. The robust watermark can reliably trace the trusted marked face while the semi-robust watermark is sensitive to malicious distortions that make the watermark disappear after deepfake or watermark removal attack. The passive detection module fuses spatial- and frequency-domain features to further differentiate between deepfake content and watermark removal attacks in the absence of watermarks. The proposed cross-domain feature fusion involves substituting the "secondary" channels of spatial-domain features with the "primary" channels of frequency-domain features. Subsequently, the "primary" channels of spatial-domain features are used to replace the "secondary" channels of frequency-domain features. Extensive experiments demonstrate that our approach not only offers proactive defense mechanisms by using extracted watermarks, i.e., source tracing and copyright protection, but also achieves passive detection when there are no watermarks, to further differentiate between deepfake content and watermark removal attacks, thereby offering a full-defense approach.

Key words: Deepfake detection; Proactive defense; Source tracing; Cross-domain feature fusion; Watermark removal attack
https://doi.org/10.1631/FITEE.2401012 **CLC number:** TP309

1 Introduction

The rapid advancement of deepfake technology poses significant threats to individual privacy and societal trust. As synthetic media generation evolves, the risk of misinformation increases, raising concerns about digital content integrity and media credibility. This evolution threatens fields like journalism and privacy protection.

In response, research on deepfake defense has grown, focusing on passive detection and proactive defense techniques.

Passive detection relies mainly on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) (Li ZY et al., 2023). Liao et al. (2020) proposed a two-stream CNN to capture tampering artifacts and local noise residuals, effectively identifying operator chains and maintaining robustness under

‡ Corresponding author
* Project supported by the Liaoning Provincial Education Department Science Project (No. JYTMS20231039), the Liaoning Provincial Education Science Planning Project (No. JG22CB252), and the National Natural Science Foundation of China (Nos. 61976109 and 61601214)
# Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2401012) contains supplementary materials, which are available to authorized users
ORCID: Hui SHI, https://orcid.org/0000-0001-5029-7461; Yanni LI, https://orcid.org/0009-0002-5459-9744

JPEG compression. Wodajo and Atnafu (2021) introduced CviT, using CNNs and visual Transformers for facial classification. Chen H et al. (2023) proposed a Transformer-based self-supervised learning approach with data augmentation for improved detection. Shao et al. (2023) enhanced detection accuracy using a dual adaptation module with spatial attention in a pre-processed ViT network. Xu Y et al. (2023) used paired interaction learning with multi-channel Xception attention to improve tampering detection performance across color spaces. Zhang et al. (2024) introduced a spatial predictive module (SPM) and a temporal contrastive module (TCM) to enhance the natural consistency representation learning. Hu et al. (2024) proposed a novel deepfake detection model that can both recognize and localize unknown domain deepfake videos.

To address compression artifacts on social media, Liao et al. (2023) introduced the facial muscle motions (FAMM) framework, analyzing continuous facial landmarks to construct motion features and achieving high detection accuracy under various compression levels. Detection of post-processed image forgery remains challenging due to concealed artifacts. Chen JX et al. (2023) tackled this with the signal noise separation-based network (SNIS), isolating tampered regions and enhancing boundary information, demonstrating robustness against post-processing attacks. Guo et al. (2024) proposed space-frequency interactive convolution (SFIConv) to model manipulation cues from deepfake. However, passive detection methods often struggle with adversarial attacks and subtle manipulations.

Proactive defense includes active interference and active forensics. Active interference leverages adversarial perturbations to disrupt generative models like generative adversarial networks (GANs) (Ruiz et al., 2020). Dong et al. (2023) introduced TCA-GAN, generating transferable adversarial perturbations to undermine deepfake generation in black-box settings. Huang et al. (2022) proposed CMUA-Watermark to improve the universality of adversarial perturbations in white-box environments. However, traditional methods often fail against sophisticated attacks. Li YZ et al. (2024) presented LandmarkBreaker, disrupting facial landmark extraction to impede deepfake generation, followed by LandmarkBreaker++, which minimized perceptibility using gradient clipping and face masking. Balancing detection accuracy and computational efficiency remains a challenge in these integrated approaches.

Active forensics embeds traceable information to detect manipulated images. Sun et al. (2022) proposed FakeTracer, embedding traces in training data via autoencoders, which deepfake models inadvertently replicate, exposing manipulations. Zhao et al. (2023) introduced an identity watermarking framework with watermark injection and verification to protect facial identity features. However, proactive defenses lose effectiveness if removed and cannot directly verify the authenticity of suspicious images.

To achieve both passive detection and proactive defense, we propose a full-defense framework (FDF) based on cross-domain fusion and separable watermarks (SepMark). This framework ensures comprehensive protection before and after deepfake generation.

In summary, our contributions in this study are as follows:

1. We propose a novel FDF that integrates passive detection and proactive defense, providing defense before and after deepfake manipulation. The proposed FDF provides a comprehensive solution by using cross-domain fusion and deep SepMark.

2. The proposed FDF achieves source tracing and deepfake detection by using watermarks at different levels. The robust watermark can reliably trace the trusted marked face while the semi-robust watermark is sensitive to malicious distortions, making the watermark disappear after deepfake or watermark removal attacks. Even if the watermarks cannot be extracted, it can still detect whether the image is a forgery.

3. Cross-domain fusion technology improves accuracy. The proposed method can accurately identify subtle forgery features and inconsistencies.

4. Notably, it shows strong generalizability in adapting to diverse datasets and scenarios, underscoring its potential for real-world deepfake detection applications.

5. The proposed FDF provides strong robustness. The robust tracer and semi-robust detector exhibit a low bit error rate (BER) under 12 common attacks to trace image sources, while the semi-robust detector has a high BER under two deepfake attacks

and one watermark removal attack to distinguish the attack types.

2 FDF: passive detection and proactive defense model

2.1 Overview of FDF

We have integrated passive detection of deepfake and proactive defense for the first time, proposing a deepfake FDF based on cross-domain feature fusion and SepMark (Wu et al., 2023) (Fig. 1). The proactive defense module provides copyright authenticity and source tracing using two extracted watermarks. When the watermarks cannot be extracted, for example, when they have been maliciously removed by attackers or it is a forgery without watermarks, then the passive detection module will determine whether it is a forgery, thereby offering an FDF.

One encoder and two decoders are included in the proactive defense module. In the encoder side, a batch of I_{co} and a batch of M_{en} are fed into a UNet-like structure to produce I_{en}. In the random forward noise pool (RFNP) end, various distortions are introduced to the noise pool for the combined training. In particular, both standard forward propagation and backward propagation are realized. In this way, I_{en} interacts with pseudo-noise which is sampled from different distortions. The decoder of SepMark consists of a robust decoder Tr and a semi-robust decoder De. The robust decoder Tr can extract the embedded watermark under any attack, while the semi-robust decoder De can extract the watermark under common distortion attacks but not under deepfake or watermark removal attacks.

When the watermarks cannot be extracted, the proposed passive detection module will further determine whether the image is a forgery.

Case 1: The robust decoder Tr receives the image noised by various distortions, and then extracts the robust watermark M_{tr} which is roughly the same as the original embedded watermark M_{en}. Therefore, the robust Tr can trace the source of the images for copyright protection regardless of attacks.

Case 2: The semi-robust decoder De receives the common and maliciously distorted images, and

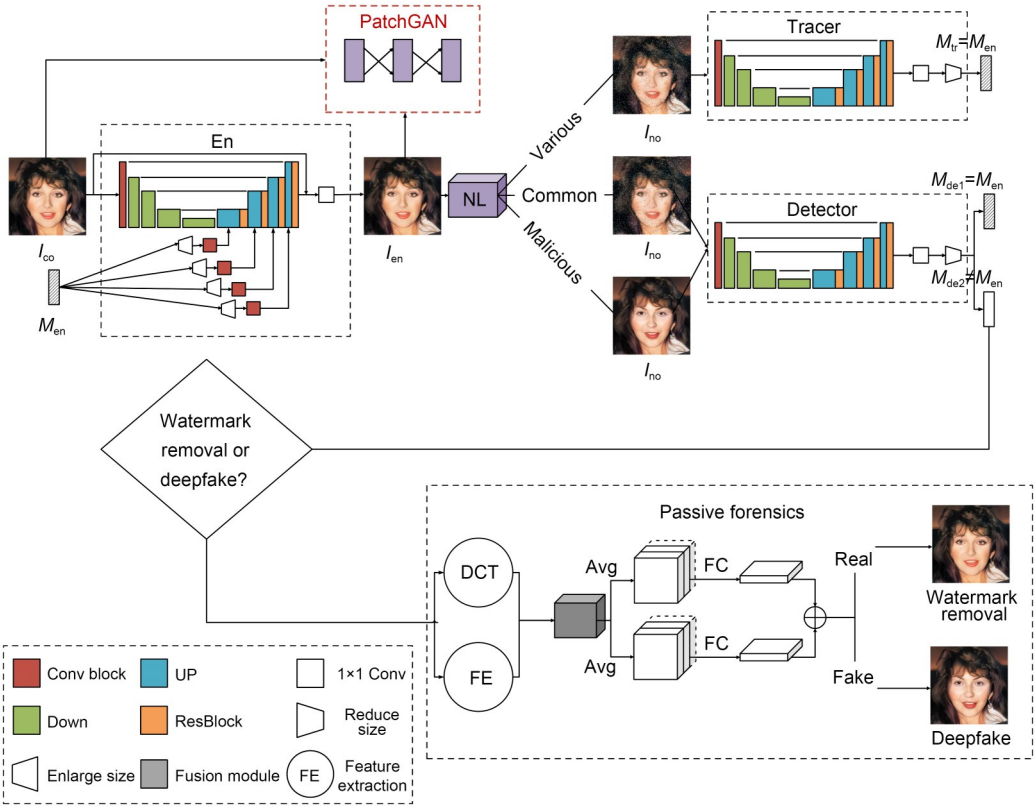

Fig. 1 Overall framework diagram (References to color refer to the online version of this figure)

then extracts semi-robust watermark M_{de}. If $M_{de}\approx M_{en}$ or $M_{de}\approx M_{tr}$, it indicates that the face image under common distortions is real and that the source is reliable.

Case 3: If $M_{de}\not\approx M_{en}$ or $M_{de}\not\approx M_{tr}$, then the face image under malicious distortions is unreliable. The passive detection module is used to further differentiate between deepfake content and watermark removal attacks.

2.2 Passive detection module based on cross-domain feature fusion

In social networks, compression is a common data processing method used to reduce file size, improve transmission efficiency, and save storage space. However, excessive compression can damage shallow texture features, disrupt pixel correlations, and render spatial information insufficient for effective feature extraction, making spatial–frequency domain fusion a key research area. Cross-domain feature fusion can take full advantage of spatial and frequency domains, better capture effective features, resist compression and other common attacks, provide more stable features, and improve the accuracy of deepfake detection.

The proposed passive detection module involves mainly data augmentation, dual-domain feature extraction, and cross-domain fusion. Data augmentation is used to enhance the diversity of training samples. Cross-domain feature extraction aims to capture both spatial- and frequency-domain features. For spatial features, a feature extraction network combined with a lightweight attention module is used to extract low-dimensional spatial characteristics. For frequency features, a separation extraction and fusion module is used, using discrete cosine transform and channel recombination to extract relevant components. To better use dual-domain features, we propose to exchange feature channels using cross-domain fusion. Through channel exchange and attention mechanisms, the network can focus more effectively on specific channels relevant to the task and adaptively adjust weights across different positions, thereby improving perception capabilities in various regions. Finally, the proposed module reaches the final deepfake detection decisions through average pooling layers, fully connected layers, and linear combination operations.

2.2.1 Random data augmentation strategy

The random data augmentation strategy plays a crucial role in deepfake detection. To enhance the model's generalization ability and robustness, we use random rotation, flipping, scaling, cropping, and noise operations to increase the diversity of training data and improve the model's performance under different conditions. Specifically, random rotation at various angles and horizontal/vertical flipping are applied to simulate different shooting angles and directions. Random scaling and cropping operations effectively simulate different shooting distances and perspective changes. Additionally, noise is introduced by adding random Gaussian or salt-and-pepper noise, simulating potential interference during image transmission and storage processes, thereby enhancing the model's robustness.

2.2.2 Spatial- and frequency-domain feature extraction

1. Spatial-domain feature extraction

We introduce CNNs to extract local low-level forgery features, including spatial locality and local textures. After data augmentation, the image $\boldsymbol{T}_i$ is fed into a 7×7 convolutional layer, followed by batch normalization and activation function processing, and finally a max-pooling operation to obtain 64-channel low-dimensional spatial-domain features $\boldsymbol{T}_s$. This process can be expressed as follows:

$$\boldsymbol{T}_s=\mathrm{MaxPool}\Big(\mathrm{Relu}\Big(\mathrm{BN}\Big(\mathrm{Conv}\big(\boldsymbol{T}_i\big)\Big)\Big)\Big), \quad (1)$$

where MaxPool denotes max-pooling, Relu represents the activation function, BN indicates batch normalization, and Conv denotes the convolution operation.

2. Frequency-domain feature extraction

Discrete cosine transform (DCT) is used in frequency-domain feature extraction. Specifically, the preprocessed image $\boldsymbol{T}_i$ is first converted from the RGB color space to the YCrCb color space. Subsequently, a convolution operation is applied, with the convolution layer's weights being initialized to discrete cosine coefficients to achieve DCT. This process can be represented as follows:

$$d=\mathrm{Conv}_{\mathrm{DCT}}\Big(\mathrm{RtoY}\big(\boldsymbol{T}_i\big)\Big), \quad (2)$$

where d denotes the DCT coefficient, $\mathrm{Conv}_{\mathrm{DCT}}$ represents the convolution operation on DCT coefficients, RtoY indicates the color space transformation operation, and $\boldsymbol{T}_i$ represents the preprocessed image.

For the 192-channel DCT coefficients, the first $3C$ channels are taken and divided into three groups, serving as the DCT coefficients for corresponding color channels Y, Cr, and Cb. Each group contains C channels, as follows:

$$\begin{cases} d_Y = d_Y[:, 0{:}C, :, :], \\ d_{\mathrm{Cr}} = d_{\mathrm{Cr}}[:, C{:}2C, :, :], \\ d_{\mathrm{Cb}} = d_{\mathrm{Cb}}[:, 2C{:}3C, :, :], \end{cases} \tag{3}$$

where d_Y, d_{Cr}, and d_{Cb} represent DCT coefficients for respective color channels Y, Cr, and Cb, d denotes the DCT coefficient, and C represents the number of channels per group. Next, specific frequency coefficients are selected. The first 22 channels are taken from the frequency coefficients of the Y channel, and the first 21 channels are taken from the frequency coefficients of the remaining color channels. The process is as follows:

$$\begin{cases} d_Y = d_Y[:, 0{:}22, :, :], \\ d_{\mathrm{Cr}} = d_{\mathrm{Cr}}[:, 0{:}21, :, :], \\ d_{\mathrm{Cb}} = d_{\mathrm{Cb}}[:, 0{:}21, :, :]. \end{cases} \tag{4}$$

Finally, the coefficients of the three color channels are merged to obtain 64-channel low-dimensional frequency-domain features $\boldsymbol{T}_{\mathrm{f}}$, as follows:

$$\boldsymbol{T}_{\mathrm{f}} = \mathrm{Concatenate}\left[d_Y, d_{\mathrm{Cr}}, d_{\mathrm{Cb}}\right], \tag{5}$$

where $\boldsymbol{T}_{\mathrm{f}}$ represents the 64-channel frequency-domain feature vector.

2.2.3 Cross-domain feature fusion

The cross-domain feature fusion module integrates spatial- and frequency-domain features (Fig. 2). Initially, spatial- and frequency-domain features undergo further feature extraction via 3×3 convolution layers and batch normalization layers. Subsequently, spatial and channel attention mechanisms are introduced to capture fine-grained and rich local features. The channel attention mechanism directs the network's focus towards specific channels that are crucial for the task, enhancing feature representation and minimizing redundant information. Conversely, the spatial attention mechanism helps the network better capture both local and global contextual information by adaptively adjusting weights at various positions, thereby improving the network's perception capabilities across different regions. The detailed process is formulated as follows:

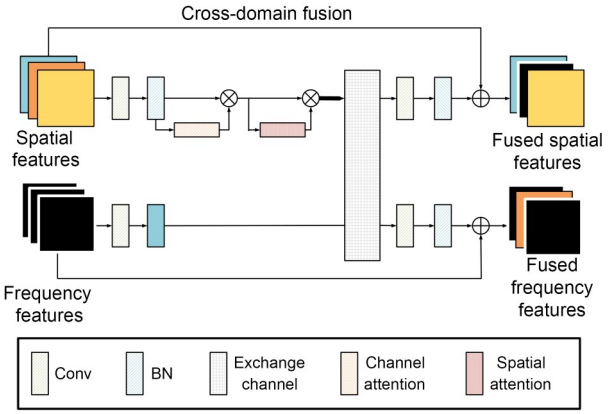

Fig. 2 Cross-domain feature fusion module

$$\boldsymbol{f}_{\mathrm{s}} = \mathrm{Att}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\boldsymbol{T}_{\mathrm{s}}\right)\right)\right), \tag{6}$$

$$\boldsymbol{f}_{\mathrm{f}} = \mathrm{BN}\left(\mathrm{Conv}\left(\boldsymbol{T}_{\mathrm{f}}\right)\right), \tag{7}$$

where $\boldsymbol{f}_{\mathrm{s}}$ and $\boldsymbol{f}_{\mathrm{f}}$ represent the further extracted spatial- and frequency-domain features respectively, and Att indicates the attention mechanism.

Cross-domain feature fusion is achieved by swapping the "important" channels between the spatial and frequency domains. Note that there is a direct correspondence between feature channels and their respective weights in the normalization layers. For instance, assuming that the shape of the feature vector is [1, N, 512, 512], the weights of the normalization layer are presented as a $1\times N$ one-dimensional matrix, with each feature channel corresponding to a specific weight value in the matrix. By analyzing the weights of the normalization layers and applying a predefined threshold, feature channels of spatial and frequency domains can be grouped according to their "importance."

Specifically, the absolute values of the normalization layer weights are first sorted in descending order, and the sorted indices are then obtained, as shown in Eqs. (8) and (9):

$$\boldsymbol{n}_{\mathrm{s}}=\mathrm{Sort}\left(\left|\boldsymbol{W}_{\mathrm{s}}\right|\right), \tag{8}$$

$$\boldsymbol{n}_{\mathrm{f}}=\mathrm{Sort}\left(\left|\boldsymbol{W}_{\mathrm{f}}\right|\right), \tag{9}$$

where $\boldsymbol{W}_{\mathrm{s}}$ and $\boldsymbol{W}_{\mathrm{f}}$ represent the normalized weight matrices of spatial- and frequency-domain features, respectively, $|\cdot|$ denotes the element-wise absolute value operation applied to the 64 individual scaling factors (α), one for each channel in the batch normalization layer, and $\boldsymbol{n}_{\mathrm{s}}$ and $\boldsymbol{n}_{\mathrm{f}}$ denote the indices of the sorted weight matrices.

Two zero vectors $\boldsymbol{Z}_{\mathrm{s}}$ and $\boldsymbol{Z}_{\mathrm{f}}$ with shapes matching those of the spatial- and frequency-domain features, respectively, are initialized. The detailed process is as follows:

First, we preserve the "important" feature channels in the spatial and frequency domains. In the weight matrices $\boldsymbol{W}_{\mathrm{s}}$ and $\boldsymbol{W}_{\mathrm{f}}$, elements that satisfy $\left|\boldsymbol{W}[i]\right|\geq\theta_w$ are referred to as "important weights," and their corresponding feature channels are considered "important" feature channels. These channels are then stored in $\boldsymbol{Z}_{\mathrm{s}}$ and $\boldsymbol{Z}_{\mathrm{f}}$, respectively. This process is described in Eqs. (10) and (11):

$$Z_{\mathrm{s}}[i]=\begin{cases}f_{\mathrm{s}}[i], & \text{if}\left|\boldsymbol{W}_{\mathrm{s}}[i]\right|\geq\theta_w,\\ 0, & \text{otherwise},\end{cases} \tag{10}$$

$$Z_{\mathrm{f}}[i]=\begin{cases}f_{\mathrm{f}}[i], & \text{if}\left|\boldsymbol{W}_{\mathrm{f}}[i]\right|\geq\theta_w,\\ 0, & \text{otherwise},\end{cases} \tag{11}$$

where $\boldsymbol{f}_{\mathrm{s}}$ and $\boldsymbol{f}_{\mathrm{f}}$ denote the spatial- and frequency-domain features to be fused respectively, θ_w represents the "importance" threshold, $\boldsymbol{W}_{\mathrm{s}}$ and $\boldsymbol{W}_{\mathrm{f}}$ are the batch normalized weight matrices of $\boldsymbol{f}_{\mathrm{s}}$ and $\boldsymbol{f}_{\mathrm{f}}$ respectively, and $\boldsymbol{Z}_{\mathrm{s}}$ and $\boldsymbol{Z}_{\mathrm{f}}$ are zero vectors.

Next, we fill the zero-value channels in $\boldsymbol{Z}_{\mathrm{s}}$ with the sorted feature channels from $\boldsymbol{f}_{\mathrm{f}}$ based on the sorted indices $\boldsymbol{n}_{\mathrm{f}}$. Similarly, the zero-value channels in $\boldsymbol{Z}_{\mathrm{f}}$ are filled with the sorted feature channels from $\boldsymbol{f}_{\mathrm{s}}$ based on the sorted indices $\boldsymbol{n}_{\mathrm{s}}$. The preliminarily fused spatial- and frequency-domain features are shown in Eqs. (12) and (13):

$$Z_{\mathrm{s}}[i]=\begin{cases}Z_{\mathrm{s}}[i], & \text{if}\,Z_{\mathrm{s}}[i]\neq 0,\\ f_{\mathrm{f}}\left[n_{\mathrm{f}}[i]\right], & \text{otherwise},\end{cases} \tag{12}$$

$$Z_{\mathrm{f}}[i]=\begin{cases}Z_{\mathrm{f}}[i], & \text{if}\,Z_{\mathrm{f}}[i]\neq 0,\\ f_{\mathrm{s}}\left[n_{\mathrm{s}}[i]\right], & \text{otherwise}.\end{cases} \tag{13}$$

Finally, $\boldsymbol{Z}_{\mathrm{s}}$ and $\boldsymbol{Z}_{\mathrm{f}}$ are treated as the new spatial- and frequency-domain features respectively. After activating features $\boldsymbol{Z}_{\mathrm{s}}$ and $\boldsymbol{Z}_{\mathrm{f}}$, additional convolution and batch normalization operations are applied. The resulting high-dimensional features are then added to the low-dimensional features $\boldsymbol{T}_{\mathrm{s}}$ and $\boldsymbol{T}_{\mathrm{f}}$ to form a residual connection, yielding the cross-fused spatial feature $\boldsymbol{F}_{\mathrm{s}}$ and frequency feature $\boldsymbol{F}_{\mathrm{f}}$:

$$\boldsymbol{F}_{\mathrm{s}}=\mathrm{Relu}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathrm{Relu}\left(\boldsymbol{Z}_{\mathrm{s}}\right)\right)\right)+\boldsymbol{T}_{\mathrm{s}}\right), \tag{14}$$

$$\boldsymbol{F}_{\mathrm{f}}=\mathrm{Relu}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathrm{Relu}\left(\boldsymbol{Z}_{\mathrm{f}}\right)\right)\right)+\boldsymbol{T}_{\mathrm{f}}\right). \tag{15}$$

We present our Channel_exchange code in Algorithm 1.

Algorithm 1 Channel_exchange

Input: features, bns, bn_threshold

1: Initialize feature1, feature2 to zeros

2: Compute absolute weights of BN layers:

bn1=abs(bns[0].weight)

bn2=abs(bns[1].weight)

3: Sort indices of bn1 and bn2 by descending order:

bn1_idx_big2small=sort_indices_desc(bn1)

bn2_idx_big2small=sort_indices_desc(bn2)

4: Preserve important features:

for each channel *i* in features[0].channels:

if bn1[*i*]>=bn_threshold then:

feature1[*i*]=features[0][*i*]

for each channel *j* in features[1].channels:

if bn2[*j*]>=bn_threshold then:

feature2[*j*]=features[1][*j*]

5: Exchange unimportant features:

exchange_list1={*i* | bn1[*i*]<bn_threshold}

exchange_list2={*j* | bn2[*j*]<bn_threshold}

for each *i* in exchange_list1:

feature1[*i*]=features[1][bn2_idx_big2small[*i*]]

for each *j* in exchange_list2:

feature2[*j*]=features[0][bn1_idx_big2small[*j*]]

6: Return feature1, feature2

Output: feature1, feature2

2.2.4 Final decision

$\boldsymbol{F}_{\mathrm{s}}$ and $\boldsymbol{F}_{\mathrm{f}}$ are fed to average pooling and fully connected layers, respectively, to derive two preliminary evidential results, d_1 and d_2. Finally, d_1 and d_2

are linearly combined to produce the final evidential result, as depicted in Eq. (16):

$$y=\alpha d_1+\beta d_2, \tag{16}$$

where α and β are parameters optimized during model training for the linear combination operation, and y represents the final result of the model.

2.3 Proactive defense module

The SepMark module (Wu et al., 2023) includes the encoder En, the noise layer NL, and the separable decoders Tr and De. The structure is shown in the supplementary materials (Fig. S1). An encoder En embeds a watermark into the image. The robust decoder Tr can resist various attacks, while the semi-robust decoder De is sensitive to malicious distortions and cannot extract a complete watermark. By comparing the extracted watermarks based on the robust decoder Tr and the semi-robust decoder De, the authenticity of the image can be determined (see the supplementary materials for details).

2.4 Loss function

The proactive defense module uses multiple loss functions to ensure the effectiveness of the encoder, decoder, and adversarial discriminator.

For the encoder En, the encoded image I_{en} should be visually almost identical to the original image I_{co}. An L_2 loss function is used as shown in Eq. (17), where ω represents the training parameters:

$$L_{En}=L_2\left(I_{co}, \mathrm{En}\left(\omega, I_{co}, M_{en}\right)\right)=L_2\left(I_{co}, I_{en}\right). \tag{17}$$

The robust decoder Tr extracts M_{tr}, which should be identical to M_{en} under any common attack. Here, ω_{co} represents common image attacks and sampling distortions, ω_{df} represents deepfake and watermark removal attacks, and NL denotes the noise layer.

$$L_{Tr}=L_2\left(M_{en}, \mathrm{Tr}\left(\omega, \mathrm{NL}\left(\omega_{co}+\omega_{df}, I_{en}\right)\right)\right)=L_2\left(M_{en}, M_{tr}\right), \tag{18}$$

where

$$I_{en}=\mathrm{En}\left(\omega, I_{co}, M_{en}\right). \tag{19}$$

Conversely, the semi-robust decoder De needs to be sensitive to the deepfake attack and watermark removal attack. The goal is that it should fail to extract the watermark under the deepfake attack and watermark removal attack but successfully extract the watermark M_{de} under common image attacks and sampling distortions, as shown below:

$$\begin{aligned} L_{De1}&=L_2\left(M_{en}, \mathrm{De}\left(\omega, \mathrm{Noise}\left(\omega_{co}, I_{en}\right)\right)\right)\\ &=L_2\left(M_{en}, M_{de}\right), \end{aligned} \tag{20}$$

$$L_{De2}=L_2\left(0, \mathrm{De}\left(\omega, \mathrm{Noise}\left(\omega_{df}, I_{en}\right)\right)\right)=L_2\left(0, M_{de}\right). \tag{21}$$

The adversarial discriminator's main role is to assess the visual quality of the encoded image I_{en} compared to that of the original image I_{co} and to guide the encoder in generating high-quality outputs.

The adversarial discriminator's loss is

$$\begin{aligned} L_{Ad1}=&L_2\left(\mathrm{Ad}\left(\omega, I_{co}\right)-\overline{\mathrm{Ad}}\left(\omega, I_{en}\right), 1\right)\\ &+L_2\left(\mathrm{Ad}\left(\omega, I_{en}\right)-\overline{\mathrm{Ad}}\left(\omega, I_{co}\right), -1\right), \end{aligned} \tag{22}$$

where

$$\overline{\mathrm{Ad}}(I)=\frac{1}{B}\sum_{i=1}^{B}\mathrm{Ad}\left(I^{i\times 3\times H\times W}\right), \tag{23}$$

$$\mathrm{Ad}\left(\omega, I_{en}\right)=\mathrm{Ad}\left(\omega, \mathrm{En}\left(I_{co}, M_{en}\right)\right). \tag{24}$$

L_{Ad1} optimizes the discriminator using the relativistic average least squares GAN (RaLSGAN) loss to compare the relative authenticity between the encoded and original images.

Additionally, the discriminator is involved in updating the encoder:

$$\begin{aligned} L_{Ad2}=&L_2\left(\mathrm{Ad}\left(I_{co}\right)-\overline{\mathrm{Ad}}\left(I_{en}\right), -1\right)\\ &+L_2\left(\mathrm{Ad}\left(I_{en}\right)-\overline{\mathrm{Ad}}\left(I_{co}\right), 1\right), \end{aligned} \tag{25}$$

where

$$\mathrm{Ad}\left(I_{en}\right)=\mathrm{Ad}\left(\mathrm{En}\left(\omega, I_{co}, M_{en}\right)\right). \tag{26}$$

L_{Ad2} directly optimizes the encoder by minimizing the L_2 difference between the discriminator's output for I_{co} and I_{en}. This constraint forces the encoder to generate I_{en} with perceptual qualities indistinguishable from I_{co}, ensuring high visual quality.

The overall loss function of SepMark is expressed in Eq. (27), where λ_1–λ_5 are hyperparameters:

$$L_{\text{Total}}=\lambda_1 L_{\text{Ad2}}+\lambda_2 L_{\text{En}}+\lambda_3 L_{\text{Tr}}+\lambda_4 L_{\text{De1}}+\lambda_5 L_{\text{De2}}. \quad (27)$$

The passive detection module uses cross-entropy loss as the classification loss:

$$L_{\text{P}}=-\sum_i y_i \log\left(\hat{y}_i\right). \quad (28)$$

The overall loss function is

$$L_{\text{Total}}=\lambda_1 L_{\text{Ad2}}+\lambda_2 L_{\text{En}}+\lambda_3 L_{\text{Tr}}+\lambda_4 L_{\text{De1}}+\lambda_5 L_{\text{De2}}+\lambda_6 L_{\text{P}}. \quad (29)$$

The overall procedure of the method is given in the supplementary materials (Algorithm S1).

3 Experiments

3.1 Experimental setup

3.1.1 Training and test datasets

We selected the FaceForensics++ (FF++) dataset (Rössler et al., 2019) for training. The FF++ dataset includes 1000 real videos manipulated using four deepfake algorithms: Face2Face (F2F, Thies et al., 2019a), Faceswap (FS), DeepFakes (DF), and NeuralTextures (NT, Thies et al., 2019b). These videos and faces, sourced from 977 YouTube videos, are available at three quality levels: RAW, lightly compressed (C23), and heavily compressed (C40). For training, we used the RAW version of FF++ and divided it into training, validation, and test sets in an 8:1:1 ratio.

Additionally, we used the Celeb-DF (CDF) dataset (Li YZ et al., 2020) as a test set. It includes 590 real videos of individuals of diverse ages, races, and genders from YouTube, along with 5639 corresponding deepfake videos.

The training code is given in the supplementary materials (Algorithm S2).

3.1.2 Implementation details

All images were resized to 256×256 pixels, augmented, and normalized to $[0, 1]$ before being used. The network was trained using the Adam optimizer with an initial learning rate of 0.001 25, a batch size of 16, and for a total of 100 epochs. Hyperparameters λ_1, λ_2, λ_3, λ_4, λ_5, and λ_6 were set to 0.1, 1, 10, 10, 10, and 10, respectively. ResNet-18, initialized with pretrained weights from ImageNet, served as the backbone network for the passive detection module, while the proactive defense module was fine-tuned on the training dataset. Our method was implemented using the PyTorch framework and trained on an NVIDIA GeForce RTX 3090 GPU. Evaluation metrics including accuracy (ACC) and area under the ROC curve (AUC) were used for the passive detection module. For the proactive defense module, peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and learned perceptual image patch similarity (LPIPS) were used to assess the visual quality of the encoded images. Robustness was evaluated using the BER: the detector's BER under deepfake and watermark removal attacks should approach 50%, while the BER under common attacks for both the tracer and detector should approach 0%.

3.2 In-dataset testing

In-dataset tests were conducted on three different versions of FF++: RAW, C23, and C40. Table 1 shows the results of a comparison of ACC and AUC using different versions of FF++. Using the RAW dataset, our method showed the highest ACC of 98.69%, outperforming all the other methods. Our method

Table 1 Test results of various methods on the FF++ dataset

Method	ACC (%)			AUC (%)		
	RAW	C23	C40	RAW	C23	C40
DeepFake-Adapter (Shao et al., 2023)	–	**98.72**	**96.83**	–	–	–
Xception (Rössler et al., 2019)	98.26	95.73	81.73	–	–	–
MCX-API (Xu Y et al., 2023)	98.48	–	–	**99.68**	–	–
SPSL (Liu et al., 2021)	–	92.39	81.57	–	94.32	82.82
LiSiam (Wang J et al., 2022)	–	96.51	87.87	–	**99.13**	91.44
Ours (ResNet-18)	**98.69**	97.37	93.60	99.06	98.75	**96.55**

Bold values indicate the best results in each column. "–" indicates no corresponding experimental data

also showed high ACC up to 97.37% and 93.60% on the C23 and C40 datasets, respectively, surpassing the ACC of several existing methods, including Xception (Rössler et al., 2019), SPSL (Liu et al., 2021), and LiSiam (Wang J et al., 2022). Notably, even under heavy compression (C40), it achieved the highest AUC of 96.55%, surpassing all the other methods. Moreover, our method detected key tampering traces using frequency features, and even with the lightweight ResNet-18, it still performed well on the heavily compressed dataset. This high accuracy underscores the reliability of our detection method, which is essential for maintaining credibility and integrity in digital content. The results reflected the robustness of our approach across variable compression levels, demonstrating its suitability for real-world applications.

Table 2 shows that our method achieved perfect accuracy (100%) in detection using the FS dataset, surpassing all the other algorithms, demonstrating the effectiveness of the proposed deepfake detector. It also performed competitively in detection using the NT dataset, with an accuracy of 99.90%, also exceeding the performance of other algorithms. Although our detection accuracy in the F2F dataset was 99.15%, which was slightly low, our method still outperformed MesoNet (Afchar et al., 2018), and it achieved a comparable result, with a detection ACC of 98.07%, on the DF dataset.

Table 2 Accuracy of different methods applied to various forgery types based on the FF++ dataset (RAW)

Method	ACC (%)			
	DF	FS	F2F	NT
DeepFake-Adapter (Shao et al., 2023)	**99.84**	99.76	99.33	95.97
Xception (Rössler et al., 2019)	99.59	99.14	**99.61**	99.36
MesoNet (Afchar et al., 2018)	99.24	98.15	98.35	97.96
Ours (ResNet-18)	98.07	**100**	99.15	**99.90**

The best results are in bold

Table 3 shows that our method achieved superior performance on the C23 compressed dataset, with the highest accuracy of 98.75% for F2F and 95.39% for NT, outperforming other algorithms. It also excelled in detection in the FS dataset, achieving an accuracy of 99.17%, surpassing Xception's 98.36% (Rössler et al., 2019) and F3Net's 98.51% (Qian et al., 2020). Although there was a slight reduction in ACC for the DF dataset at 97.31%, compared to other methods, our approach still maintained high effectiveness.

Table 3 Accuracy of different methods applied to various forgery types based on the FF++ dataset (C23)

Method	ACC (%)			
	DF	FS	F2F	NT
Xception (Rössler et al., 2019)	**98.85**	98.36	98.23	94.50
F3Net (Qian et al., 2020)	98.43	98.51	98.34	93.22
MTD-Net (Yang et al., 2021)	98.64	**99.76**	98.42	94.60
Ours (ResNet-18)	97.31	99.17	**98.75**	**95.39**

The best results are in bold

Table 4 presents the performance on the heavy compression C40 dataset. Our method continued to demonstrate robust detection capabilities, achieving a high detection ACC of 95.58% in the DF dataset, 97.71% in FS, 93.46% in F2F, and 88.33% in NT. Although it performed slightly more poorly than DeepFake-Adapter (Shao et al., 2023), it significantly surpassed SPSL (Liu et al., 2021) and GRCC (Liang et al., 2022). Notably, our method maintained balanced performance across various compression levels, underscoring its robustness.

Table 4 Accuracy of different methods applied to various forgery types based on the FF++ dataset (C40)

Method	ACC (%)			
	DF	FS	F2F	NT
DeepFake-Adapter (Shao et al., 2023)	**99.84**	**99.76**	**99.33**	**95.97**
SPSL (Liu et al., 2021)	93.48	92.26	86.02	76.78
GRCC (Liang et al., 2022)	95.43	93.99	88.94	77.19
Ours (ResNet-18)	95.58	97.71	93.46	88.33

The best results are in bold

3.3 Cross-dataset testing

We evaluated the generalization performance of our proposed passive detection module through cross-dataset testing. The model was trained on the FF++ dataset and tested on the CDF and DFDC datasets. Furthermore, we extended cross-dataset evaluation to a CNN-generated image dataset (Wang SY et al., 2020) and an AI-generated image dataset (Lu et al., 2023). The results are summarized in Table 5. Our method achieved the highest AUC (77.51%) on the CDF dataset, demonstrating the best generalization performance. On the DFDC dataset, our AUC reached 70.58%, which was slightly lower than that achieved by Shao et al. (2023) but higher than those of all other

Table 5 Results from cross-dataset comparisons on various datasets

Method	AUC (%)			
	CDF	DFDC	CNN-generated image dataset	AI-generated image dataset
DeepFake-Adapter (Shao et al., 2023)	71.74	**72.66**	–	–
Xception (Rössler et al., 2019)	65.50	58.81	–	–
SPSL (Liu et al., 2021)	76.88	66.16	–	–
MTD-Net (Yang et al., 2021)	70.12	–	–	–
FreNet (Tan et al., 2024)	77.46	–	–	–
NoiseDF (Wang TY and Chow, 2023)	75.89	63.89	–	–
PRLE+EfficientNet (Cheng et al., 2023)	70.67	–	–	–
Ours (ResNet-18)	**77.51**	70.58	**75.24**	**77.18**

The best results are in bold

methods. Although our AUC was 2.08 percentage points (PPs) lower than that of Shao et al. (2023) on the DFDC dataset, it was 5.77 PPs higher on the CDF dataset. To further validate the generalization ability, we conducted tests on the latest AIGC datasets, which have not been tested in any other studies. Our AUC values reached 75.24% on the CNN-generated image dataset and 77.18% on the AI-generated image dataset. These results highlighted the robustness and generalizability of our approach compared to existing methods. This superior performance shows the effectiveness of our passive detection module in adapting to diverse datasets and scenarios, underscoring its potential for real-world deepfake detection applications.

3.4 Ablation studies

We conducted ablation experiments on the cross-domain feature fusion module and attention module of the passive detection model using the FF++ dataset (C23, C40). We evaluated the effectiveness based on the ACC under different parameter settings.

3.4.1 Lack of cross-domain feature fusion

We first removed the spatial- or frequency-domain fusion module and retrained the network for testing. Removing either the spatial domain or the frequency domain greatly degraded performance (Table 6). Using only the spatial-domain or only the frequency-domain features greatly reduced ACC compared to using cross-domain fusion. Specifically, cross-domain feature fusion improved accuracy by 16.12 PPs compared to the spatial domain only and 9.62 PPs compared to the frequency domain only, based on the C23 dataset. Similarly, on the C40 dataset, it improved ACC by 13.11 PPs compared to the spatial domain only and 8.40 PPs compared to the frequency domain only. The average improvements reached 14.62 and 9.02 PPs, respectively. This demonstrated that cross-domain fusion enhances the accuracy and generalizability of deepfake detection.

Table 6 Impact of cross-domain fusion on network performance

Domain	ACC (%)		
	C23	C40	Average
Spatial domain only	81.25	80.49	80.87
Frequency domain only	87.75	85.20	86.47
Cross-domain fusion	**97.37**	**93.60**	**95.49**

The best results are in bold

3.4.2 Lack of attention mechanism

To demonstrate the necessity and effectiveness of incorporating the attention mechanism, we conducted performance tests and comparisons on networks with or without the attention mechanism. Networks with the attention mechanism achieved ACC improvements of 3.81 PPs on the C23 dataset and 3.14 PPs on the C40 dataset, resulting in an average enhancement of 3.48 PPs (Table 7). This proved that attention mechanism plays an important role in improving detection accuracy.

Table 7 Impact of attention mechanism on network performance

Attention	ACC (%)		
	C23	C40	Average
Without	93.56	90.46	92.01
With	**97.37**	**93.60**	**95.49**

The better results are in bold

3.5 Visual quality test

We used the average PSNR, SSIM, and LPIPS of encoded images as metrics for visual quality assessment. The proactive defense module showed superior performance in terms of PSNR and SSIM, which were as high as 44.29 dB and 0.9672 (Table 8). Although in terms of LPIPS, CIN (Ma et al., 2022) gave impressive objective visual quality with a value of 0.0006, its invertible neural network was not sufficiently flexible to be compatible with our architecture of deep separable watermarking, and it had non-ideal robustness when encountering quantization noise. This further proved that the proactive defense module has good visual effects and comprehensive performance.

3.6 Robustness test

Robust tracers should have a low BER under common attacks to trace image sources, while semi-robust detectors should maintain a low BER under common attacks but a high BER under the deepfake attack and watermark removal attack, to distinguish the attack types. Table 9 presents 15 attacks, including 12 common, 2 deepfake, and 1 watermark removal attack (Tian et al., 2024). The common attacks included Identity (no modification), JPEG compression (quality factor set to 75), Resize (downscaled to 50% of the original size), Gaussian Blur (5×5 kernel), Median Blur (3×3 kernel), Brightness (increased by 30%), Contrast (enhanced by a factor of 1.5), Saturation (boosted by 25%), Hue (adjusted by 20

Table 8 Visual quality of the encoded image I_{en}

Model	Image size	Message length	PSNR↑ (dB)	SSIM↑	LPIPS↓
MBRS (Jia et al., 2021)	128×128	30	33.05	0.8106	0.0141
CIN (Ma et al., 2022)	128×128	30	42.41	0.9628	**0.0006**
PIMoG (Fang et al., 2022)	128×128	30	37.73	0.9407	0.0086
Ours	128×128	30	38.51	0.9588	0.0028
FaceSigns (Neekhara et al., 2024)	256×256	128	32.33	0.9211	0.0260
Ours	256×256	128	**44.29**	**0.9672**	0.0079

The best results are in bold

Table 9 Robustness test for distorted image I_{no}

Attack type	BER (%)					
	MBRS (Jia et al., 2021)	CIN (Ma et al., 2022)	PIMoG (Fang et al., 2022)	FaceSigns (Neekhara et al., 2024)	Our proactive defense module	
					Tracer	Detector
Identity	0.0	0.0	0.0366	0.0136	0.0	0.0
JPEG (75)	0.2597	2.7514	19.5562	0.8258	0.2136	0.2172
Resize (50%)	0.0	0.0	0.0767	1.0726	0.0744	0.0
Gaussian Blur (5×5)	0.0	22.7786	0.1169	0.1671	0.0372	0.0
Median Blur (3×3)	0.0	0.0307	0.0992	0.0977	0.0372	0.0
Brightness (30%)	0.0	0.0	1.3443	10.8196	0.0	0.0
Contrast (1.5)	0.0	0.0	0.8121	0.0334	0.0	0.0
Saturation (25%)	0.0	0.0	0.0803	0.7113	0.0	0.0
Hue (20)	0.0	0.0	0.1523	8.3780	0.0744	0.0
Dropout (10%)	0.0	0.0	0.4828	17.5615	0.1860	0.0
Salt Pepper (0.02)	0.0	0.0378	2.3667	12.3238	0.1860	0.0
Gaussian Noise (mean=0, deviation=0.01)	0.0	0.0	12.7396	7.0697	1.078	0.1413
Deepfake	–	–	–	–	6.250	44.37
FaceSwap	19.3744	48.5068	8.6745	49.9463	8.593	49.06
Watermark removal	–	–	–	–	9.100	48.55

"Tracer" denotes the BER between M_{tr} and M_{en}, while "Detector" denotes the BER between M_{de} and M_{en}

degrees), Dropout (10% of pixels set to zero), Salt & Pepper Noise (at a rate of 0.02), and Gaussian Noise (mean=0, deviation=0.01).

Deepfake attacks were generated using FaceSwap and deepfake from the FF++ dataset. For the watermark removal attack, an advanced CNN network specifically trained for watermark removal was used (Tian et al., 2024), iteratively minimizing residual signals to mimic real-world watermark tampering scenarios. The results showed that our tracer and detector had BERs close to 0%, reaching nearly optimal robustness against common attacks. However, our detector's BER approached 50%, close to random guesses, under the deepfake attack and watermark removal attack, while the tracer's BER remained near 0%. Unlike the other methods, our method could successfully distinguish deepfake attacks from watermark removal attacks.

4 Conclusions

In this paper, we propose a novel FDF, integrating passive detection and proactive defense mechanisms. In terms of proactive defense, the robust decoder can extract the watermark under various common attacks for copyright verification and source tracing, while the semi-robust decoder is sensitive to malicious attacks, rendering watermark extraction impossible under deepfake and watermark removal attacks.

When a watermark cannot be detected, it indicates that the image has been maliciously attacked—either the watermark has been removed or it is a deepfake. To distinguish between these two possibilities, we propose a passive detection module based on cross-domain feature fusion. This enhancement allows for the differentiation between watermark removal and deepfake that traditional methods might fail to achieve, ensuring reliable source tracing and tamper detection. Our experimental results show high visual quality and accuracy in detecting various types of forgeries and watermark removal across different compression levels, demonstrating the effectiveness and generalizability of our approach.

Future work will involve more advanced adversarial training, transfer learning, and localization capabilities to improve the model's performance.

Contributors

Hui SHI designed the research. Hui SHI and Guibin WANG processed the data. Hui SHI and Yanni LI drafted the paper. Rujia QI helped organize the paper. Hui SHI and Yanni LI revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

All the data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Afchar D, Nozick V, Yamagishi J, et al., 2018. MesoNet: a compact facial video forgery detection network. Proc IEEE Int Workshop on Information Forensics and Security, p.1-7. https://doi.org/10.1109/WIFS.2018.8630761

Chen H, Lin YZ, Li B, et al., 2023. Learning features of intra-consistency and inter-diversity: keys toward generalizable deepfake detection. *IEEE Trans Circ Syst Video Technol*, 33(3):1468-1480. https://doi.org/10.1109/TCSVT.2022.3209336

Chen JX, Liao X, Wang W, et al., 2023. SNIS: a signal noise separation-based network for post-processed image forgery detection. *IEEE Trans Circ Syst Video Technol*, 33(2): 935-951. https://doi.org/10.1109/TCSVT.2022.3204753

Cheng H, Guo YY, Wang TY, et al., 2023. Towards generalizable deepfake detection by primary region regularization. https://arxiv.org/abs/2307.12534

Dong JH, Wang Y, Lai JH, et al., 2023. Restricted black-box adversarial attack against deepfake face swapping. *IEEE Trans Inform Forens Secur*, 18:2596-2608.

Fang H, Jia ZY, Ma ZH, et al., 2022. PIMoG: an effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. Proc 30th ACM Int Conf on Multimedia, p.2267-2275. https://doi.org/10.1145/3503161.3548049

Guo ZQ, Jia ZH, Wang LJ, et al., 2024. Constructing new backbone networks via space-frequency interactive convolution for deepfake detection. *IEEE Trans Inform Forens Secur*, 19:401-413. https://doi.org/10.1109/TIFS.2023.3324739

Hu J, Liao X, Gao DF, et al., 2024. Delocate: detection and localization for deepfake videos with randomly-located tampered traces. https://arxiv.org/abs/2401.13516

Huang H, Wang YT, Chen ZY, et al., 2022. CMUA-Watermark: a cross-model universal adversarial watermark for combating deepfakes. Proc AAAI Conf on Artificial Intelligence, p.989-997. https://doi.org/10.1609/AAAI.V36I1.19982

Jia ZY, Fan H, Zhang WM, 2021. MBRS: enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression. Proc 29th ACM Int Conf on Multimedia, p.41-49. https://doi.org/10.1145/3474085.3475324

Li YZ, Yang X, Sun P, et al., 2020. Celeb-DF: a large-scale challenging dataset for deepfake forensics. IEEE/CVF

Conf on Computer Vision and Pattern Recognition, p.3204-3213. https://doi.org/10.1109/CVPR42600.2020.00327

Li YZ, Sun P, Qi HG, et al., 2024. LandmarkBreaker: a proactive method to obstruct deepfakes via disrupting facial landmark extraction. *Comput Vis Image Underst*, 240: 103935. https://doi.org/10.1016/j.cviu.2024.103935

Li ZY, Zhang XH, Pu YW, et al., 2023. A survey on multimodal deepfake and detection techniques. *J Comput Res Dev*, 60(6):1396-1416 (in Chinese). https://doi.org/10.7544/issn1000-1239.202111119

Liang JH, Shi HF, Deng WH, 2022. Exploring disentangled content information for face forgery detection. Proc 17th European Conf on Computer Vision, p.128-145. https://doi.org/10.1007/978-3-031-19781-9_8

Liao X, Li KD, Zhu XS, et al., 2020. Robust detection of image operator chain with two-stream convolutional neural network. *IEEE J Sel Top Signal Process*, 14(5): 955-968. https://doi.org/10.1109/JSTSP.2020.3002391

Liao X, Wang YM, Wang TY, et al., 2023. FAMM: facial muscle motions for detecting compressed deepfake videos over social networks. *IEEE Trans Circ Syst Video Technol*, 33(12):7236-7251. https://doi.org/10.1109/TCSVT.2023.3278310

Liu HG, Li XD, Zhou WB, et al., 2021. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.772-781. https://doi.org/10.1109/CVPR46437.2021.00083

Lu ZY, Huang D, Bai L, et al., 2023. Seeing is not always believing: benchmarking human and model perception of AI-generated images. Proc 37th Conf on Neural Information Processing Systems.

Ma R, Guo MX, Hou Y, et al., 2022. Towards blind watermarking: combining invertible and non-invertible mechanisms. Proc 30th ACM Int Conf on Multimedia, p.1532-1542. https://doi.org/10.1145/3503161.3547950

Neekhara P, Hussain S, Zhang XQ, et al., 2024. FaceSigns: semi-fragile watermarks for media authentication. *ACM Trans Multim Comput Commun Appl*, 20(11):337. https://doi.org/10.1145/3640466

Qian YY, Yin GJ, Sheng L, et al., 2020. Thinking in frequency: face forgery detection by mining frequency-aware clues. https://arxiv.org/abs/2007.09355

Rössler A, Cozzolino D, Verdoliva L, et al., 2019. FaceForensics++: learning to detect manipulated facial images. IEEE/CVF Int Conf on Computer Vision, p.1-11. https://doi.org/10.1109/ICCV.2019.00009

Ruiz N, Bargal SA, Sclaroff S, 2020. Disrupting deepfakes: adversarial attacks against conditional image translation networks and facial manipulation systems. Proc 16th Europe Conf on Computer Vision, p.236-251. https://doi.org/10.1007/978-3-030-66823-5_14

Shao R, Wu TX, Nie LQ, et al., 2023. DeepFake-Adapter: dual-level adapter for deepfake detection. https://arxiv.org/abs/2306.00863

Sun P, Li YZ, Qi HG, et al., 2022. FakeTracer: exposing DeepFakes with training data contamination. Proc IEEE Int Conf on Image Processing, p.1161-1165. https://doi.org/10.1109/ICIP46576.2022.9897756

Tan CC, Zhao Y, Wei SK, et al., 2024. Frequency-aware deepfake detection: improving generalizability through frequency space learning. https://arxiv.org/abs/2403.07240

Thies J, Zollhöfer M, Nießner M, 2019a. Deferred neural rendering: image synthesis using neural textures. *ACM Trans Graph*, 38(4):66. https://doi.org/10.1145/3306346.3323035

Thies J, Zollhöfer M, Stamminger M, et al., 2019b. Face2Face: real-time face capture and reenactment of RGB videos. *Commun ACM*, 62(1):96-104. https://doi.org/10.1145/3292039

Tian CW, Zheng MH, Jiao TC, et al., 2024. A self-supervised CNN for image watermark removal. *IEEE Trans Circ Syst Video Technol*, 34(8):7566-7576. https://doi.org/10.1109/TCSVT.2024.3375831

Wang J, Sun YL, Tian JH, 2022. LiSiam: localization invariance Siamese network for deepfake detection. *IEEE Trans Inform Forens Secur*, 17:387-398. https://doi.org/10.1109/TIFS.2022.3186803

Wang SY, Wang O, Zhang R, et al., 2020. CNN-generated images are surprisingly easy to spot for now. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8692-8701. https://doi.org/10.1109/CVPR42600.2020.00872

Wang TY, Chow KP, 2023. Noise based deepfake detection via multi-head relative-interaction. Proc AAAI Conf on Artificial Intelligence, 37(12):14548-14556. https://doi.org/10.1609/aaai.v37i12.26701

Wodajo D, Atnafu S, 2021. Deepfake video detection using convolutional vision Transformer. https://arxiv.org/abs/2102.11126

Wu XS, Liao X, Ou B, 2023. Separable watermark module for dual-domain fusion in deepfake detection. Proc Int Conf on Cyber Security and Privacy, p.234-245.

Xu Y, Raja K, Verdoliva L, et al., 2023. Learning pairwise interaction for generalizable deepfake detection. Proc IEEE/CVF Winter Conf on Applications of Computer Vision Workshops, p.1-11. https://doi.org/10.1109/WACVW58289.2023.00074

Yang JC, Li AY, Xiao S, et al., 2021. MTD-Net: learning to detect deepfakes images by multi-scale texture difference. *IEEE Trans Inform Forens Secur*, 16:4234-4245.

Zhang DC, Xiao ZH, Li SK, et al., 2024. Learning natural consistency representation for face forgery video detection. https://arxiv.org/abs/2407.10550

Zhao Y, Liu B, Ding M, et al., 2023. Proactive deepfake defence via identity watermarking. Proc IEEE/CVF Winter Conf on Applications of Computer Vision, p.4591-4600. https://doi.org/10.1109/WACV56688.2023.00458

List of supplementary materials