Constructing New Backbone Networks via Space-Frequency Interactive Convolution for Deepfake Detection

Zhiqing Guo, Zhenhong Jia, Liejun Wang, Dewang Wang, Gaobo Yang, and Nikola Kasabov, *Fellow, IEEE*

***Abstract*—The serious concerns over the negative impacts of Deepfakes have attracted wide attentions in the community of multimedia forensics. The existing detection works achieve deepfake detection by improving the traditional backbone networks to capture subtle manipulation traces. However, there is no attempt to construct new backbone networks with different structures for Deepfake detection by improving the internal feature representation of convolution. In this work, we propose a novel Space-Frequency Interactive Convolution (SFIConv) to efficiently model the manipulation clues left by Deepfake. To obtain high-frequency features from tampering traces, a Multichannel Constrained Separable Convolution (MCSConv) is designed as the component of the proposed SFIConv, which learns space-frequency features via three stages, namely generation, interaction and fusion. In addition, SFIConv can replace the vanilla convolution in any backbone networks without changing the network structure. Extensive experimental results show that seamlessly equipping SFIConv into the backbone network greatly improves the accuracy for Deepfake detection. In addition, the space-frequency interaction mechanism does benefit to capturing common artifact features, thus achieving better results in cross-dataset evaluation. Our code will be available at https://github.com/EricGzq/SFIConv.**

***Index Terms*—Deepfake detection, space-frequency interactive convolution, backbone network, manipulation traces.**

I. INTRODUCTION

IN recent years, with the continuous developments of artificial intelligence (AI), especially various generative models, the AI-powered Deepfake has made considerable progresses in manipulating face images and videos. While promoting some legitimate applications such as for entertainment and film production, Deepfake might also be used for malicious or illegal purposes [1], such as fabricating fake news to spread misinformation and mislead public opinions. In the community of image forensics, there is an urgent need to develop some Deepfake detection methods to expose Deepfake-enabled face forgeries.

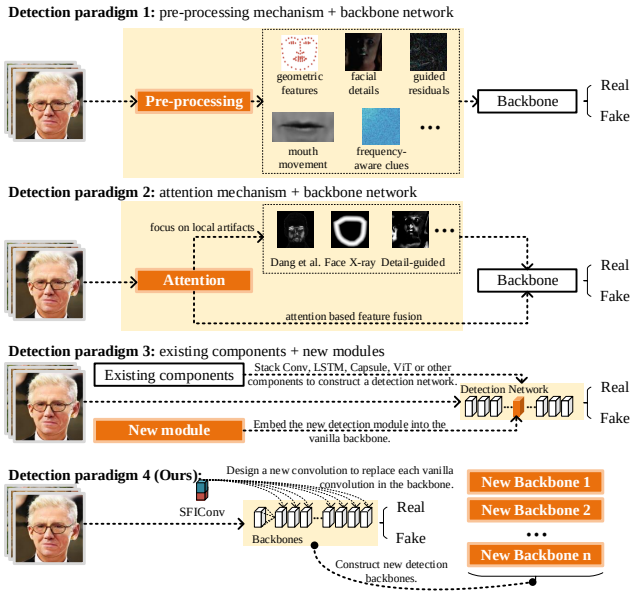

Fig. 1: Four deep learning based detection paradigms in Deepfake detection. To show the differences of the four paradigms more clearly, we use black and orange boxes to represent the existing and newly designed deep learning components respectively.

Deep learning has dominated various computer vision tasks such as image classification and semantic segmentation. Many backbone networks, which include AlexNet [1], VGGnet [2], ResNet [3], DenseNet [4] and EfficientNet [5], were designed to learn feature representations from image contents. However, Deepfake detection seriously depends on capturing subtle artifacts or texture changes, rather than salient image content features, such as identity, hair color and eyes. As we know, manipulation traces serve as the only clues to identify the authenticity of face images, whereas image content features have potential negative effects on feature learning for Deepfake detection.

Actually, existing deep learning based Deepfake detection works promote the traditional backbone networks to improve detection performance by suppressing content features and capturing subtle manipulation trace features. Specifically, existing works can be divided into three detection paradigms from the following points of view:

Z. Guo, Z. Jia and L. Wang are with the School of Computer Science and Technology, Xinjiang University, Urumqi, 830017, China. Email: {guozhiqing, jzhh, wljxju}@xju.edu.cn.

D. Wang and G. Yang are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China. Email: dewang_wang@126.com; yanggaobo@hnu.edu.cn.

N. Kasabov is with the School of Engineering, Computing and Mathematical Sciences, Auckland University of Technology, Auckland, 1010, New Zealand. Email: nkasabov@aut.ac.nz.

This work was supported in part by the Xinjiang Uygur Autonomous Region Tianshan Excellence Project under Grant 2022TSYCLJ0036, in part by the Scientific and Technological Innovation 2030 Major Project under Grant 2022ZD0115800, in part by the National Natural Science Foundation of China under Grant 62302427, 62261053, 61972143, 61972142, 62372164 and U1903213. (*Corresponding authors: Zhenhong Jia and Gaobo Yang*)

[1]https://www.ted.com/talks/danielle_citron_how_deepfakes_undermine_truth_and_threaten_democracy

- Using some pre-processing mechanisms to highlight manipulation clues from spatial-domain images, which are then fed into the backbone network for feature learning [6]–[9].
- Designing some attention mechanisms to force the backbone network to learn discriminant features directly from potential local artifact regions [8], [10], or using self-attention feature fusion to obtain better feature representations [11], [12].
- Constructing Deepfake detection networks by using some basic components such as convolution layer, Capsule, LSTM, etc. [13]–[16]. The alternative way is to design a new detection module, which can be embedded into the existing backbone network to promote feature learning [17]–[20].

In these three detection paradigms, the pre-processing mechanism can highlight potential manipulation clues, which will promote the feature learning of the backbone network, but it might also inevitably destroy some useful artifacts/traces due to the pre-processing operation. The attention mechanism forces the backbone network to learn feature representation from local artifact regions, but it is difficult to capture global information for image forensics. Furthermore, since the existing backbone networks for Deepfake detection are usually borrowed from common computer vision tasks, they do not address the inherent yet fundamental issue that their components are not specially designed for learning features from subtle manipulation traces.

In this work, we are motivated to address the above-mentioned drawbacks from a new perspective, namely constructing new backbone networks for Deepfake detection by designing space-frequency interactive convolution. Fig. 1 compares three existing detection paradigms with the proposed new detection paradigm in this work. As claimed above, the pre-processing mechanism is actually a double-edged sword, which usually converts spatial-domain images into frequency-domain, or only highlights partial key information for detection. However, the features learned from both spatial-domain and frequency-domain are complementary [12]. In this work, we exploit both the spatial-domain features with complete forgery clues and the high-frequency features with partial manipulation traces. These two types of features are embedded into the convolution layer via an interactive fusion for better manipulation trace extraction.

As we know, the manipulation traces are usually isolated dots or linear textures in fake face images, which exhibit in the form of drastic gray-scale changes. Thus, the manipulation traces can be regarded as high-frequency information. To capture the high-frequency manipulation trace features, a Multichannel Constrained Separable Convolution (MCSConv) is proposed, which can be embedded in the vanilla convolution layer to adaptively extract frequency domain information from any number of feature maps [2]. The proposed approach neither discards any spatial-domain features as the first category of works (preprocessing-based method), nor exploits only on local artifacts like the second category of works (attention-based method). Instead, the space-frequency features are interactively learned in the convolution layer to obtain better feature representations and expose potential artifacts in a global scope. This improves over the vanilla convolution for better Deepfake detection.

We distill the above insights and design a new convolution component, namely Space-Frequency Interactive Convolution (SFIConv). In addition, SFIConv is used to replace the vanilla convolution in the original backbone network to construct new backbone networks for Deepfake detection. Specifically, the novelties and contributions of this work are three-fold:

- A novel MCSConv module is designed to capture high-frequency tampering traces from Deepfake-powered fake face images. Similar to the pre-processing mechanism, MCSConv is embedded in the convolution layer to obtain cross-channel high-frequency manipulation features for different number of feature maps.
- A novel convolution component, namely SFIConv, is designed to improve the defects of vanilla convolution for Deepfake detection. Without adding extra parameters and reducing FLOPs, SFIConv can capture well manipulation traces by improving the internal feature representation of convolution. Moreover, the proposed SFIConv can be used to seamlessly replace the vanilla convolution in the backbone network designed for common visual tasks.
- With the help of MCSConv and SFIConv, we construct a variety of new backbone networks that are more suitable for Deepfake detection. Extensive experimental results show that the backbone network equipped with SFIConv significantly improves the accuracy for Deepfake detection tasks.

The rest of this paper is organized as follows. Section II briefly summaries the related works. Section III presents the proposed approach. Section IV reports the experimental results and provides some analysis. Conclusion is made in Section V.

II. Related Works

A. Deepfake Forgery

The Deepfake-powered face forgeries can be divided into two categories, namely face identity forgery and face attribute forgery.

For face identity forgery, the most common way is to swap facial identity features. Zhu et al. [21] proposed the first mega pixel level face swapping method, which achieved a high-fidelity forgery effect. Zhang et al. [22] solved the problem that the existing methods can't keep the target face attributes well in face swapping, and realized efficient and realistic video face swapping. In addition, another way of face identity forgery is to directly generate face images with arbitrary identities by generative models [23]. For example, Karras et al. [24] designed the StyleGAN model to directly generate face images that do not exist in the world.

For face attribute forgery, manipulating expression attributes is usually harmful, simply because it will change the original expression semantics and convey false emotions such as happy and angry. Thies et al. [25] animated the facial expression

[2]Vanilla convolution usually represents the standard convolution used in backbone networks such as VGGNet and ResNet.

from the source video and re-rendered the facial expression attributes of the target video. Tripathy et al. [26] manipulated facial expression attributes by action unit representation, while keeping the original facial identity well. Moreover, another way of face attribute forgery is to change the style attributes in face images. For example, Choi et al. [27] proposed a unified model to manipulate face attributes such as hair color and gender in multiple domains. Wei et al. [28] proposed a hair editing interaction method to manipulate hair styles in face image.

Deepfake inevitably leaves some artifacts or subtle texture changes, which are the key clues to identify the authenticity of face images. However, vanilla convolution was originally designed for capturing image content features (such as cats, dogs, etc.), rather than subtle tampering traces. Thus, we are motivated to improve the original mode of vanilla convolution to better capture the subtle manipulation clues for Deepfake detection.

B. Deepfake Detection

The existing deep learning based Deepfake detection works can be divided into three basic detection paradigms. Although some detection methods may adopt the design ideas of multiple detection paradigms at the same time (e.g. [29] constructs a detection model by designing pre-processing mechanism and new modules, so it belongs to both detection paradigm 1 and 3), all methods are designed from the following three basic detection paradigms to capture manipulation clues.

Detection Paradigm 1: Pre-processing mechanism is a very common method, which provides potential forgery clues for backbone network to promote Deepfake detection. For example, Sun et al. [6] fed the precise geometric features of face images into the two-stream recurrent neural network to realize efficient and robust Deepfake detection. Zhu et al. [8] used the combination of direct light and identity texture disentangled from face images as facial details to detect subtle forgery patterns. Yang et al. [30] proposed a trace generator to promote the backbone network to track the potential manipulation traces. In our previous works, we also generated manipulation traces [31] and guided residuals [12] via pre-processing mechanism, so as to promote the backbone network to detect Deepfake forgery.

Detection Paradigm 2: Attention mechanism is also a common method, which promotes the backbone network to focus on local forgery regions or adaptive fusion features. For example, Li et al. [10] assumed that there was a blending step in face swapping, and designed an attention mechanism called "Face X-ray" to promote the network to pay attention to the blending boundary of faces. Luo et al. [32] also proposed a Residual-Guided Spatial Attention Module, which guided the low-level feature extractor to pay more attention to forgery traces from a new perspective. For attention based feature fusion, Chen et al. [11] proposed an RGB-Frequency Attention Module to obtain more comprehensive local feature representation. In our previous work, we also designed an Attention Fusion Mechanism to realize adaptive fusion of spatial and residual features [12].

Detection Paradigm 3: Stacking common basic components of deep learning or designing new modules to embed in the backbone network is also a popular design scheme of detection model. For example, Nguyen et al. [13] used capsule components to construct a Deepfake detection model. Ge et al. [14] cascaded a CNN-based encoder, a ConvGRU-based aggregator and a binary classifier to capture semantic changes in Deepfake videos. In addition, Zhao et al. [17] designed three key modules and inserted them into the backbone network to focus on different local parts, enhance subtle artifacts and aggregate different levels of features. Liu et al. [18] proposed a Gram Block that can be added to ResNet to extract global texture features. In the previous work, we also designed two modules, which can be embedded in the front end and each bottleneck layer of ResNet to help the backbone network capture manipulation clues from face images [20].

On the one hand, the previous detection paradigm studied how to help backbone network capture manipulation traces efficiently from different directions by designing pre-processing and attention mechanisms. On the other hand, Deepfake detection methods are constructed by using basic components of deep learning or new detection modules. Essentially, these methods are all using auxiliary mechanisms (such as pre-processing, attention, new modules) to improve the backbone network designed for common visual tasks, instead of directly constructing a new backbone network suitable for Deepfake detection. In this paper, we propose a new detection paradigm. That is, a SFIConv, which can focus on manipulation traces, is designed and used to construct backbone networks suitable for Deepfake detection. By changing the feature learning mode of vanilla convolution, SFIConv can be more suitable for capturing manipulation traces in Deepfake images.

III. Methodology

A. Overview

In the traditional backbone network, vanilla convolution has only one input and one output, which transforms an input tensor $T_{in} \in \mathbb{R}^{c_{in} \times h \times w}$ into an output tensor $T_{out} \in \mathbb{R}^{c_{out} \times h \times w}$, where c and $h \times w$ represent the channel number and spatial dimension of the feature tensor, respectively. In our SFIConv, we propose to seamlessly replace the vanilla convolution in the backbone network by extracting two types of features, namely spatial-domain features and high-frequency features. Thus, the input and output of SFIConv are designed into three different modes (see Fig. 2 for details) to meet the needs of different locations in the backbone network.

To explain how SFIConv replaces vanilla convolution, we divide the existing backbone network into three parts: "Head", "Body" and "Tail", as shown in Fig. 2. Firstly, the "Head" part refers to the first layer of the backbone network, which converts an input image (single channel or three channels) into multi-channel feature maps. Since SFIConv involves the division of the feature map into two parts in proportion, the number of input channels of SFIConv should be a divisible even value such as 32 or 64. In fact, the first convolution layer of existing backbone networks can usually convert the input image into feature maps with 32 or 64 channels. Thus,

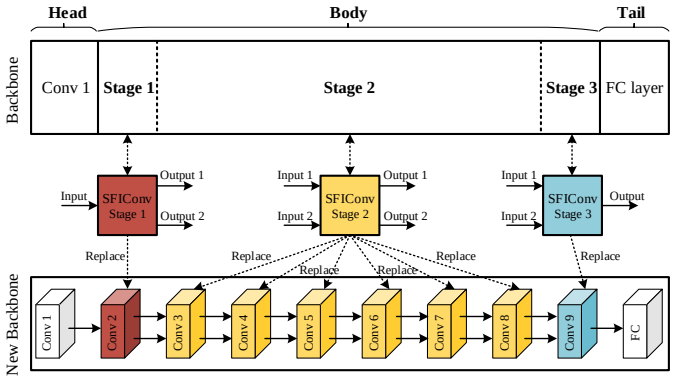

Fig. 2: Overview of building a new backbone network using SFIConv. Note that SFIConv only replaces the convolution layer in the existing backbone network, and does not change other operations such as pooling and activation functions.

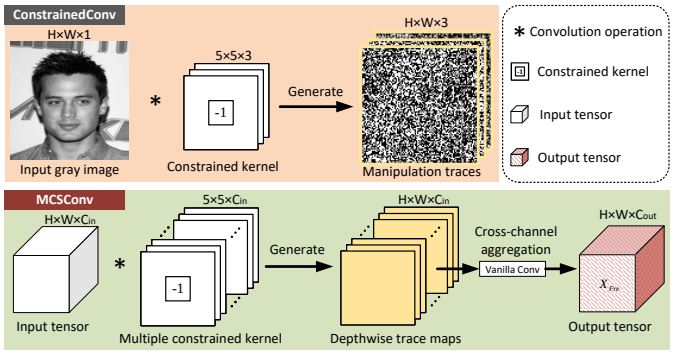

Fig. 3: Comparison between ConstrainedConv and MCSConv.

there is no need to change the first convolution layer when building a new backbone network. Secondly, the "Tail" part is a classification layer composed of fully connected (FC) layers, which does not need to be changed. That is, when SFIConv is used to replace vanilla convolution, the "Head" and "Tail" parts are unchanged. We only need to replace the vanilla convolution of the "Body" part with SFIConv, so as to construct a new backbone network suitable for Deepfake detection.

The "Body" part is also divided into three stages. As shown in Fig. 2, we name the SFIConvs with three input and output modes as "Stage 1", "Stage 2" and "Stage 3", respectively. The three stages are placed at different positions of the "Body" for the generation, interaction and fusion of space-frequency features, respectively. Each stage replaces a layer of vanilla convolution. Specifically, "Stage 1" divides a group of input feature maps into two groups of output feature maps to generate spatial-domain features and high-frequency features, respectively. "Stage 3" is to fuse two groups of features into one group of features, which is fed into the "Tail" part for classification. That is, only the first and last convolution layers of the "Body" part are replaced by "Stage 1" and "Stage 3", and their purposes are to separate and reintegrate features, respectively. All convolution layers in the middle of the "Body" will be replaced by "Stage 2". Designing three stages is to transmit two groups of different features in the backbone network without changing the original network topology and calculation cost. Next, we will present the details of SFIConv, including MCSConv and three "Stages".

B. Multichannel Constrained Separable Convolution

Constrained convolution layer (ConstrainedConv) is a well-known tool, which is used to extract tampering traces in forged images by capturing the changes of local pixel relationships caused by image tampering operations [33]. In this work, we still use ConstrainedConv kernel to extract high-frequency information (i.e. manipulation traces) from any number of input feature maps. However, the original design of ConstrainedConv is to put it before the backbone network as a pre-processing layer, and it extracts manipulation traces related features from fixed single-channel gray image. Thus, the original ConstrainedConv is not applicable to SFIConv.

To embed the ConstrainedConv into SFIConv-Stage1, we have made some improvements and renamed it as Multichannel Constrained Separable Convolution (MCSConv), as shown in Fig. 3. In our MCSConv, each kernel W_c performs the following constraints of constrained convolution kernels:

$$\begin{cases} W_c(0,0) = -1, \\ \sum_{(x,y)\neq(0,0)} W_c(x,y) = 1, \end{cases} \tag{1}$$

where (x, y) represent the spatial index in W_c, and the central value is set to $(0, 0)$. We make the number of W_c be consistent with the channel number c_{in} of the input feature map. Then, c_{in} kernels in the MCSConv are convolved with the corresponding input feature maps, which can obtain c_{in} depthwise trace maps containing high-frequency information. However, these feature maps still do not effectively utilize the feature information of different channels at the same spatial position. Thus, we continue to use vanilla convolution $f(\cdot)$ to aggregate the high-frequency information across channels. This process is similar to the Depthwise Separable Convolution [34], but it is worth noting that MCSConv does not use the traditional 1×1 convolution when aggregating cross-channel information. Instead, the kernel size of $f(\cdot)$ in MCSConv is variable (e.g. 1×1, 3×3, 5×5, etc.). This is to keep it consistent with the original kernel size when replacing vanilla convolution in any backbone network, so as to avoid changing parameters and calculation cost of the original convolution. Compared with vanilla convolution, MCSConv only adds a few extra parameters and FLOPs when generating c_{in} depthwise trace maps [3], and other operations are consistent with vanilla convolution in terms of parameters and calculation cost.

C. Space-Frequency Interactive Convolution

To overcome the defects of vanilla convolution for Deepfake detection, we change the feature representation in convolution layer via space-frequency feature interaction. Vanilla convolution has only one input and one output. For convenience of description, the input and output of each vanilla convolution layer that needs to be replaced are denoted as $T_{in} \in \mathbb{R}^{c_{in}\times h\times w}$

[3]Parameters and FLOPs increase linearly with the number of W_c.

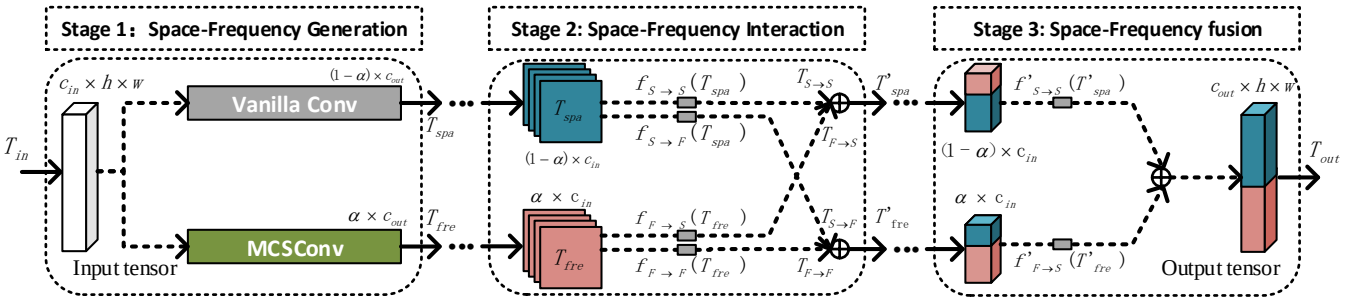

Fig. 4: Detailed structure of SFIConv. Since "Stage 1" and "Stage 3" are used for the generation and fusion of space-frequency features, they usually only need to be used once when building a new backbone network. The body part of the network can be connected by using multiple "Stage 2" with dual input and output. For the convenience of description, the input and output of each stage are represented by c_{in} and c_{out}. Note that we use gray rectangles to represent vanilla convolution operations.

and $T_{out} \in \mathbb{R}^{c_{out} \times h \times w}$, respectively. Fig. 4 shows the details of SFIConv. The three stages are detailed as follows.

SFIConv-Stage1: To achieve interaction between two types of features, we first need to use SFIConv-Stage1 to decompose the input feature tensor T_{in} into two groups of output feature tensors $\{T_{spa}, T_{fre}\}$ along the channel dimension.

In SFIConv-Stage1, we exploit vanilla convolution function $f(\cdot)$ and MCSConv to obtain $\{T_{spa}, T_{fre}\}$ from the input features T_{in}, respectively. To keep the calculation cost and parameters as consistent as possible, the convolution kernel size of $f(\cdot)$ and MCSConv in SFIConv-Stage1 is consistent with the original convolution to be replaced. In addition, we set the rational number $\alpha \in [0, 1]$ as a hyper-parameter to control the channels of $\{T_{spa}, T_{fre}\}$. Among them, $T_{spa} \in \mathbb{R}^{(1-\alpha)c_{out} \times h \times w}$ denotes the feature representation learned from the spatial-domain image with complete manipulation traces, and $T_{fre} \in \mathbb{R}^{\alpha c_{out} \times h \times w}$ represents the high-frequency features generated by MCSConv. As the outputs of SFIConv-Stage1, both T_{spa} and T_{fre} are fed into SFIConv-Stage2 for space-frequency feature interaction.

When constructing a new backbone network, we replace SFIConv-Stage1 with the second layer convolution in the backbone network, rather than the first layer. The reasons behind this include two points. On the one hand, the number of input channels of SFIConv-Stage1 is preferably even. We hope that the input features can be evenly decomposed (e.g. $\alpha = 0.5$) into two groups of features. The first layer convolution can usually transform the input image (single channel or three channels) into a set of feature maps with even channels, which will facilitate feature decomposition. On the other hand, extracting spatial-domain features and high-frequency features from more feature maps will be more conducive to feature learning and interaction. If only 3-channel features are used as input, we can only disassemble them into 1-channel and 2-channel feature maps. Obviously, the first layer convolution of the original backbone network also plays an important role: transforming the input image (single channel or three channels) into multi-channel (32, 64, etc.) feature maps. Thus, the first layer convolution does not need to be changed in our scheme.

SFIConv-Stage2: This stage is the core component of SFIConv for the interaction of space-frequency features. To compute the feature tensors $\{T_{spa}, T_{fre}\}$ from "Stage 1", SFIConv-Stage2 is designed as a convolution structure with dual-input and dual-output modes.

In SFIConv-Stage2, both T_{spa} and T_{fre} should conduct intra-feature updating and inter-feature interaction. There are four groups of convolution operations $\{f_{S\to S}, f_{S\to F}, f_{F\to F}, \{f_{F\to S}\}$ to generate self-updated features $\{T_{S\to S}, T_{F\to F}\}$ and interaction features $\{T_{S\to F}, T_{F\to S}\}$. Table I summarizes the relevant parameters and their definitions in this work. The specific operations are defined as follows

$$\begin{cases} updating: T_{S\to S} = f_{S\to S}(T_{spa}), \\ interaction: T_{S\to F} = f_{S\to F}(T_{spa}), \\ updating: T_{F\to F} = f_{F\to F}(T_{fre}), \\ interaction: T_{F\to S} = f_{F\to S}(T_{fre}), \end{cases} \tag{2}$$

where the input and output channels in each group of convolution operations are expressed in the form {input_channels, output_channels} as follows

$$\begin{cases} f_{S\to S} = \{c_{in} - (\alpha \cdot c_{in}), c_{out} - (\alpha \cdot c_{out})\}, \\ f_{S\to F} = \{c_{in} - (\alpha \cdot c_{in}), \alpha \cdot c_{out}\}, \\ f_{F\to F} = \{\alpha \cdot c_{in}, \alpha \cdot c_{out}\}, \\ f_{F\to S} = \{\alpha \cdot c_{in}, c_{out} - (\alpha \cdot c_{out})\}, \end{cases} \tag{3}$$

For the space-frequency feature interaction, we exploit the sum operation to fuse the self-updated features $\{T_{S\to S}, T_{F\to F}\}$ and the interaction features $\{T_{S\to F}, T_{F\to S}\}$, respectively. At "Stage 2", the output feature tensor $T'_{spa} \in \mathbb{R}^{(1-\alpha)c_{out} \times h \times w}$ and $T'_{fre} \in \mathbb{R}^{\alpha c_{out} \times h \times w}$ can be expressed as

$$T'_{spa} = \underbrace{\overbrace{T_{S\to S}}^{update} + T_{F\to S}}_{interaction} \tag{4}$$

$$T'_{fre} = \underbrace{\overbrace{T_{F\to F}}^{update} + T_{S\to F}}_{interaction}. \tag{5}$$

SFIConv-Stage3: After the interactive feature learning at "Stage 2", we should further fuse the separated features. The

TABLE I: Summary of Parameters.

Parameters	Definition
$f_{S\to S}$	the convolution function of spatial-domain features, whose generates features are used to realize intra-feature update
$f_{S\to F}$	the convolution function of spatial-domain features, whose generates features are used to interact with updated high-frequency features
$f_{F\to F}$	the convolution function of high-frequency features, whose generates features are used to realize intra-feature update
$f_{F\to S}$	the convolution function of high-frequency features, whose generates features are used to interact with updated spatial-domain features
$T_{S\to S}$	updated spatial-domain features
$T_{S\to F}$	interaction features generated from the spatial-domain features, which are used to interact with the updated high-frequency features
$T_{F\to F}$	updated high-frequency features
$T_{F\to S}$	interaction features generated from the high-frequency features, which are used to interact with the updated spatial-domain features
$f'_{S\to S}$	convolution function of spatial-domain features in SFIConv-Stage3
$f'_{F\to S}$	convolution function of frequency-domain features in SFIConv-Stage3
$D_{S\to F}$	the spatial dimension of feature map $T_{S\to F}$
$D_{F\to S}$	the spatial dimension of feature map $T_{F\to S}$

fused features will be restored to a set of vanilla feature vectors, which can be fed into the FC layer for classification. SFIConv-Stage3 is embedded between the last SFIConv-Stage2 layer and the classification layer to re-aggregate the space-frequency features.

In SFIConv-Stage3, both the spatial-domain and frequency-domain features $\{T'_{spa}, T'_{fre}\}$ still perform convolution operations $\{f'_{S\to S}, f'_{F\to S}\}$ separately, which transform different input channels into the same output channel c_{out}. The input and output channels of convolution operations are defined as follows

$$\begin{cases} f'_{S\to S} = \{c_{in} - (\alpha \cdot c_{in}), c_{out}\}, \\ f'_{F\to S} = \{\alpha \cdot c_{in}, c_{out}\}. \end{cases} \tag{6}$$

Thus, the two types of features are fused as follows

$$T_{fusion} = f'_{S\to S}(T'_{spa}) + f'_{F\to S}(T'_{fre}) \tag{7}$$

where $T_{fusion} \in \mathbb{R}^{c_{out}\times h\times w}$ is the final output feature.

SFIConv changes only the feature representation inside the convolution layer, which captures well tampering traces. We also observe that the input and output of SFIConv are fully consistent with those of vanilla convolution. Thus, SFIConv can seamlessly replace vanilla convolution to construct new backbone networks for Deepfake detection.

Flexibility of SFIConv: The above is the standard form of SFIConv. It keeps almost the same parameters and calculation cost as vanilla convolution (the slightly difference is caused by MCSConv). The design of SFIConv is also flexible, which considers the tradeoff between computing performance and cost. Specifically, compressing the spatial dimension of feature map is an effective way to reduce the computational complexity in convolution layer. In general, the input and output dimensions of vanilla convolution layers are equal. For SFIConv, the input and output dimensions of each stage can either be kept consistent with the vanilla convolution (standard form) or different from the vanilla convolution (flexible form).

In the flexible form, low-dimensional feature maps can be used in the internal calculation of SFIConv, thus reducing the FLOPs of the whole backbone network. Specifically, the output dimension of "Stage 1", the input and output dimensions of "Stage 2", and the input dimension of "Stage 3" can all be changed. Only the initial input (input of "Stage 1") and the final output (output of "Stage 3") are consistent with the vanilla convolution that needs to be replaced.

There are two branches in SFIConv, namely, spatial-domain branch and frequency-domain branch. Here, we only reduce the feature dimension of frequency domain branch. Specifically, we use nearest neighbor interpolation to adjust the spatial dimension by introducing the rational number $\beta \in [0, 1]$ as the scaling factor. We first compress the spatial dimension of T_{fre} at "Stage 1". Note that features need to maintain the same feature dimension when interacting. Thus, when interacting with spatial-domain features, the dimension of high-frequency features needs to be enlarged, and when interacting with high-frequency features, the dimension of spatial-domain features needs to be reduced. Since the output dimension of "Stage 1" is changed, we should change the spatial dimensions $\{D_{S\to F}, D_{F\to S}\}$ of interaction feature $\{T_{S\to F}, T_{F\to S}\}$ in "Stage 2", which can be expressed as follows

$$\begin{cases} D_{S\to F} = I(T_{S\to F}, \beta), \\ D_{F\to S} = I(T_{F\to S}, \frac{1}{\beta}), \end{cases} \tag{8}$$

where $I(T, \beta)$ denotes the nearest neighbor interpolation with the parameter β [4] that is conducted on the feature map T. At "Stage 3", we still need to enlarge the dimension of high-frequency features to keep consistency with the dimension of spatial-domain features, so as to further integrate two types of features.

As a result, the computational cost of the backbone network is significantly reduced by compressing the feature dimension of the frequency-domain branch. Note that the feature dimension is only changed in the process of feature interaction, without changing the topological structure of the original backbone network.

IV. Experiments

A. Experimental Settings

1) Datasets: In this work, we selected four open yet challenging Deepfake face datasets, namely HFF [31], FF++ [35], DFDC [36] and CelebDF [37], for experimental evaluation. It includes one image dataset and three video datasets.

HFF [31]: It is an image dataset containing five types of fake face images, which are generated by four generative

[4] β is the scaling factor of nearest neighbor interpolation method, which is used to scale the spatial dimension of feature map. When $\beta \in [0, 1]$, the feature dimension is reduced. When $\beta > 1$, the feature dimension is enlarged.

models (including PGGAN [23], StyleGAN [24], Glow [38] and StarGAN [27]) and a computer graphics-based method (Face2Face [25]). The HFF dataset also contains three kinds of real face images with different resolutions. There are totally 155k face images that are divided into training set and testing set with the ratio of 4:1.

FF++ [35]: As the most popular face forensics dataset, it contains 1,363 real video sequences and 4,000 fake video sequences generated by four forgery methods (such as FaceSwap [39], DeepFake[5], Face2Face [25] and NeuralTextures [40]). The author uses H.264 compression to provide the datasets with two compression levels, namely, high quality (HQ) and low quality (LQ). To avoid the similarity between consecutive frames, we extract the same number of face frames from each video at equal intervals. For real video sequences, the face images are extracted from each video with interval $N_{iter} = 2$ and frame number $N_{frames} = 50$. For fake video sequences, the $N_{iter} = 2$ and $N_{frames} = 16$ are used to extract the fake face images. Thus, the number of both real and fake face images is more than 60k. We randomly select an integer of 120k face images (real: 60k, fake: 60k) from these images for experimental evaluation. The ratio of training set to testing set is 5:1.

DFDC [36]: It is a very large-scale face forensics dataset (more than 100k real and fake video sequences) generated by two unknown deepfake algorithms. In the experiment, we randomly select 2,891 real video sequences and 20,210 fake video sequences. For real video sequences, N_{iter} and N_{frames} are set to 2 and 35, respectively. For fake video sequences, $N_{iter} = 10$ and $N_{frames} = 5$. Thus, we also randomly select an integer of 120k face images (real: 60k, fake: 60k) from the extracted frames, which are divided into the training set and the testing set with the ratio of 5:1.

CelebDF [37]: It is a high-quality face forensics dataset, which is usually used for cross-dataset evaluation. There are 890 real and 5,639 fake video sequences. We use the parameters of $N_{iter} = 2$ and $N_{frames} = 100$ to extract face images from each real video sequence, and $N_{iter} = 2$ and $N_{frames} = 16$ to extract fake face images. There are totally 120k face images (real: 60k, fake: 60k) randomly selected from the extracted face frames for cross-dataset evaluation.

For the video datasets, we capture the face region from each video frame by using the Face Recognition Library[6], and the face frames without background information are saved for experiments. Note that all face images and frames are resized to 256×256 for training and testing.

2) Evaluation Metrics: In this work, SFIConv is designed to improve the detection accuracy of backbone network. As we know, Deepfake detection is a binary classification task. We use the area under the receiver operating characteristic curve (AUC) and the Balanced Accuracy (BACC) [7] as two metrics for performance evaluation. In addition, SFIConv provides some flexibility, which can reduce the computational complexity by adjusting the hyper-parameter. Thus, the model parameters (Param.) and floating point operations (FLOPs) in different configurations are also provided.

[5]https://github.com/deepfakes/faceswap

[6]https://github.com/ageitgey/face_recognition

[7]The threshold of BACC is set to 0.5.

TABLE II: Quantitative Results of Ablation Studies on FF++ Datasets.

Networks	Hyper-parameters		Param. (M)	FLOPs (GMac)	FF++ HQ		FF++ LQ	
	α	β			AUC	BACC	AUC	BACC
Backbone	-	-	13.95	3.08	87.66	81.33	88.81	81.79
SFIC-ResNet26	0.25	0.25	13.95	2.47	82.48	74.82	83.84	76.27
		0.50	13.95	2.60	89.60	83.15	89.96	83.04
		0.75	13.95	2.80	90.88	84.14	90.92	84.43
		1.00	13.95	3.09	**96.38**	**89.57**	**96.47**	**89.96**
	0.50	0.25	13.95	1.86	77.61	70.42	78.09	70.59
		0.50	13.95	2.10	84.55	76.63	87.36	79.81
		0.75	13.95	2.51	87.57	80.33	88.11	80.53
		1.00	13.95	3.09	**95.73**	**88.41**	**96.18**	**89.45**
	0.75	0.25	13.95	1.24	76.81	69.27	77.45	70.32
		0.50	13.95	1.61	85.28	77.61	88.99	81.53
		0.75	13.95	2.23	88.23	80.68	89.69	82.55
		1.00	13.95	3.09	**94.69**	**89.48**	**95.37**	**90.53**

TABLE III: Replace Vanilla Convolution in Different Backbone Networks with SFIConv.

Networks	Hyper-parameters		Param. (M)	FLOPs (GMac)	FF++ HQ		FF++ LQ	
	α	β			AUC	BACC	AUC	BACC
MobileNet [34]	-	-	3.21	761M	71.80	65.24	72.14	65.96
SFIC-MobileNet	0.5	1.0	3.21	768M	**78.28**	**70.39**	**77.66**	**70.46**
ResNet26 [3]	-	-	13.95	3.08G	87.66	81.33	88.81	81.79
SFIC-ResNet26	0.5	1.0	13.95	3.09G	**95.73**	**88.41**	**96.18**	**89.45**
ResNet50 [3]	-	-	23.51	5.38G	84.05	76.67	85.13	78.13
SFIC-ResNet50	0.5	1.0	23.51	5.38G	**92.11**	**85.31**	**92.92**	**86.89**
ResNet101 [3]	-	-	42.51	10.25G	79.80	72.74	81.02	74.07
SFIC-ResNet101	0.5	1.0	42.51	10.25G	**85.33**	**77.82**	**89.13**	**82.12**
VGGNet13 [41]	-	-	9.41	14.68G	94.59	89.91	92.44	86.88
SFIC-VGGNet13	0.5	1.0	9.41	14.66G	**98.46**	**93.61**	**95.90**	**89.41**
VGGNet16 [41]	-	-	14.73	20.12G	94.08	88.81	92.80	87.19
SFIC-VGGNet16	0.5	1.0	14.73	20.11G	**97.36**	**93.02**	**94.46**	**88.67**
VGGNet19 [41]	-	-	20.04	25.57G	94.25	90.00	90.45	85.27
SFIC-VGGNet19	0.5	1.0	20.04	25.55G	**97.41**	**93.31**	**94.71**	**89.65**

3) Implementation Details: We use Adam optimizer with parameters ($\gamma_1 = 0.9$, $\gamma_2 = 0.999$) to train our model under the PyTorch framework. The initial learning rate L_r is set to 10^{-4}. After each training epoch, the L_r decays once according to $0.5 \times L_r$. All detection models are trained for 20 epoches, and the batch size is set to 64. To ensure that all detection models have the same weight initialization, we set the random seed to 7 to fairly compare the performance of detection models in different configurations. In the training, we also use data augmentation operations (such as horizontal flip, rotation and normalization) to increase the diversity of data.

B. Ablation Study

In SFIConv, we design two hyper-parameters α and β to control the channel ratio and the size of feature maps respectively, which are used to optimize feature representation and reduce FLOPs. In this subsection, a series of ablation experiments are conducted to show the influence of two hyper-parameters for SFIConv. In the experiments, the backbone network "ResNet26" and "FF++ dataset" are selected for experimental evaluation. We use SFIConv to replace the vanilla convolution in the backbone network to construct a new detection network, which is called "SFIC-ResNet26".

Table II reports the quantitative results of backbone network and SFIC-ResNet26 with different hyper-parameters. In all

TABLE IV: Generalization of Detection Methods to Unseen Face Manipulation Methods.

Methods	TEST-DF		TEST-FF		TEST-FS		TEST-NT	
	AUC	BACC	AUC	BACC	AUC	BACC	AUC	BACC
Meso-Incep [42]	53.94	47.14	58.83	53.21	47.99	49.36	62.37	**61.44**
Multi-task [43]	66.40	62.44	65.66	62.64	56.94	54.18	56.46	48.51
XceptionNet [35]	82.57	71.99	68.66	58.22	57.79	50.57	52.75	46.89
F^3-Net [44]	64.50	54.51	63.70	58.04	41.70	41.70	54.70	51.80
AMTENnet [31]	83.87	76.06	66.92	59.56	51.01	48.09	61.73	52.55
M2TR [45]	78.40	65.38	73.30	64.36	47.10	48.04	52.20	49.17
SFIC-ResNet26 (α=0.25,β=1.00)	75.03	67.22	61.80	54.99	52.33	49.63	47.40	44.86
SFIC-ResNet26 (α=0.50,β=1.00)	78.85	72.24	57.27	51.12	**60.49**	**56.19**	53.27	45.66
SFIC-VGGNet13 (α=0.25,β=0.75)	**86.97**	**77.09**	72.77	65.96	54.32	49.77	63.04	53.85
SFIC-VGGNet13 (α=0.50,β=1.00)	85.09	75.32	**78.30**	64.49	52.36	49.38	62.08	51.06
SFIC-VGGNet13 (α=0.75,β=0.75)	84.89	71.37	77.02	**66.54**	53.34	48.94	**68.93**	52.02

tables, the best results are marked in bold font. It can be observed that SFIConv can significantly improve the detection performance of the original backbone network without changing the model parameters and FLOPs. For hyper-parameter α, it will only change the number of channels with high-frequency features in SFIConv, without adding additional model parameters and FLOPs. And for the hyper-parameter β, it can adjust the size of the feature map in SFIConv to reduce FLOPs. As we know, some forgery clues will be lost in the process of compressing the feature map. Thus, the detection performance will be limited when β is not equal to 1. However, with the help of space-frequency feature interaction mechanism, SFIConv can still obtain better detection results than the vanilla convolution by only using less FLOPs. For example, when $\alpha = 0.25$ and $\beta = 0.75$, SFIC-ResNet26 improves AUC score by 3.22% and reduces FLOPs by 10% GMac compared with backbone network in HQ dataset. When $\alpha = 0.75$ and $\beta = 0.50$, similar AUC scores (88.81% vs 88.99%) are achieved on LQ dataset, while FLOPs decreased by 91.30% GMac.

To sum up, SFIConv can not only significantly improve the accuracy of backbone network in Deepfake detection without increasing the computational cost, but also have certain flexibility to realize the trade-off between detection performance and computational cost.

C. Different Backbone Networks with SFIConv

To further evaluate the effectiveness of SFIConv, we select some popular backbone networks as the baselines to check whether SFIConv can further boost different backbone networks in Deepfake detection.

Table III reports the results of seven baseline networks and corresponding SFIConv networks. From it, we can observe that when vanilla convolution is replaced by SFIConv, the accuracy of all baseline networks has been significantly improved. The above results confirm that SFIConv is a more suitable component than vanilla convolution in Deepfake detection. By decomposing the feature map in convolution layer and exploiting interactive fusion of space-frequency features, SFIConv overcomes the defects of vanilla convolution in modeling manipulation traces, so as to better capture manipulation traces in fake face images.

D. Generalization on Unseen Manipulations

Detecting fake faces generated by unseen manipulations is a challenging task. In this subsection, six representative works with source codes, which include Meso-Incep [42], Multi-task [43], XceptionNet [35], F^3-Net [44], AMTENnet [31] and M2TR [45], are selected as the baselines. Based on 100,000 face images in FF++ HQ training set, we divide the images generated by four face manipulation methods into source domain and target domain to perform the generalization experiment of unseen manipulations. Specifically, the target domain only contains one type of face forgery images for model testing, and the other three types of face forgery images are used as the source domain for model training.

Table IV reports the detection results. From it, we can observe that the proposed approach generalizes well to the unseen manipulation method. Among the four kinds of manipulation methods, NT-generated face images usually have less visual artifacts, while the other three kinds of fake face images often have some obvious tampering traces. From Table IV, we can see that our method is better at capturing the common artifacts between different forgeries, thus achieving better generalization results on TEST-DF, TEST-FF and TEST-FS [8].

E. Intra-Dataset Evaluation

In this subsection, the proposed SFIConv is equipped to the existing backbone networks (such as ResNet26 [3] and VGGNet13 [41]) to construct new backbone networks (such as SFIC-ResNet26 and SFIC-VGGNet13) for comparison with the existing Deepfake detection works. All detection methods are trained from scratch on four datasets (including DFDC, HFF, FF++ HQ and FF++ LQ) to compare their performance fairly.

Table V reports the AUC score, BACC score, model parameters and FLOPs of the detection methods. From it, we

[8]TEST-XX indicates that the target domain of the generalization experiment is XX.

TABLE V: The Detection Results (%) of Forgery Detection Methods on Multiple Deepfake Face Datasets.

Methods	Year	Publication	Param.	FLOPs (Mac)	DFDC Dataset [36]		HFF Dataset [31]		FF++ Dataset [35] HQ		FF++ Dataset [35] LQ	
					AUC	BACC	AUC	BACC	AUC	BACC	AUC	BACC
Meso-Incep [42]	2018	WIFS	28.52K	60.18M	88.83	74.17	99.72	88.35	90.41	69.56	85.18	67.74
Multi-task [43]	2019	BTAS	305K	146M	80.05	80.05	95.29	95.29	81.44	81.43	77.50	77.50
XceptionNet [35]	2019	ICCV	20.81M	6.00G	96.28	89.83	99.91	98.53	94.50	86.72	92.02	83.53
F^3-Net [44]	2020	ECCV	21.17M	8.49G	87.50	79.86	95.70	89.65	88.00	80.02	83.00	75.21
AMTENnet [31]	2021	CVIU	9.88M	575M	92.04	83.96	99.94	98.43	90.88	81.63	87.25	78.40
M2TR [45]	2022	ICMR	>20.81M	>6.00G	97.20	90.27	97.00	91.14	94.50	86.88	92.10	82.31
SFIC-ResNet26 (α=0.25,β=1.00)	-	-	13.95M	3.09G	**97.44**	**91.24**	99.54	96.15	96.38	89.57	**96.47**	**89.96**
SFIC-ResNet26 (α=0.50,β=1.00)	-	-	13.95M	3.09G	97.21	90.98	99.02	94.32	95.73	88.41	96.18	89.45
SFIC-VGGNet13 (α=0.25,β=0.75)	-	-	9.41M	13.34G	96.39	89.62	99.99	99.29	96.88	90.07	93.36	85.18
SFIC-VGGNet13 (α=0.50,β=1.00)	-	-	9.41M	14.66G	96.65	90.15	**99.99**	**99.47**	**98.46**	**93.61**	95.90	89.41
SFIC-VGGNet13 (α=0.75,β=0.75)	-	-	9.41M	10.69G	97.11	90.94	99.99	99.25	97.47	91.38	93.57	85.91

can observe that the HFF dataset mainly contains high-quality face images generated by GANs, so all detection methods can achieve good detection accuracy. DFDC and FF++ are low-quality datasets composed of compressed video frames. The manipulation traces have been laundered by compression operation, which brings some difficulties to Deepfake detection.

For example, lightweight models (such as Meso-Incep and Multi-task) have limited ability to capture manipulation traces in low-quality data, and they only achieve low AUC and BACC scores. In addition, the pre-processing mechanism will further lead to the loss of partial forgery clues in low-quality data during the transformation from spatial domain to frequency domain. Thus, the pre-processing mechanism based detection methods (such as F^3-Net and AMTENnet) have not obtained competitive results on DFDC and FF++ datasets. In contrast, our method considers the complementarity of spatial-domain and frequency-domain features at the same time, thus avoiding the side effects caused by using only frequency-domain features. By replacing vanilla convolution with SFIConv, the backbone networks achieve better detection results on four datasets (marked in bold) than the baseline methods.

Among the baseline methods, XceptionNet and M2TR are large models with huge parameters and FLOPs. They effectively improve the detection results in low-quality datasets through complex model structure and forgery clue mining mechanism. In contrast, the backbone network equipped with SFIConv only uses smaller model parameters and FLOPs to achieve better detection results than the baseline methods. Especially, the model parameters and FLOPs of SFIC-ResNet26 are only about half that of M2TR[9], but it achieves better performance on four datasets. Since VGGNet13 itself has high FLOPs, SFIC-VGGNet13 also has high computational complexity. However, by adjusting the hyper-parameters, SFIC-VGGNet13 (α=0.75, β=0.75) significantly reduces the FLOPs by 37.14%, and achieves competitive detection results.

In the detection results of the FF++ dataset, we can also observe that five backbone networks equipped with SFIConv have achieved better detection results than the baseline methods. This further shows the superiority of SFIConv, which can effectively capture forgery clues even in low-quality datasets. Especially for SFIC-ResNet26, we notice that SFIC-ResNet26 achieves similar results on the FF++ HQ and FF++ LQ datasets, which are not affected by the compression operation. We speculate that the network structure of ResNet26 with shortcut connection can be combined with SFIConv more efficiently, so that SFIC-ResNet26 can directly capture visual artifacts. Visual artifacts are not easy to be changed by lossy compression operation, which makes SFIC-ResNet26 achieve better detection even on the FF++ LQ datasets.

[9]Please note that since the model parameters and FLOPs of M2TR method are difficult to calculate, we estimate the minimum value of M2TR according to the model structure (XceptionNet + Multiscale Transformer) for reference.

F. Cross-Dataset Evaluation

For Deepfake detection, cross-dataset evaluation is important for examining the generalization capability of different detection works towards different face forgeries. In this work, we adopt two cross-dataset evaluation protocols as follows:

1) Evaluation on DFDC and FF++: In Section IV-E, we have used six open source detection works for intra-dataset evaluation on four datasets such as DFDC, HFF, FF++ HQ and FF++ LQ. Since the HFF dataset is generated by GANs, it is essentially different from the DFDC and FF++ datasets that are mainly obtained by face swapping. Thus, only DFDC, FF++ HQ and FF++ LQ are used for cross-dataset evaluation. That is, the detection methods are trained on these three datasets and tested on the unseen dataset.

Table VI reports the cross-dataset evaluation results on three datasets. We can observe that the backbone network equipped with SFIConv generally achieves better generalization than the baselines. Especially for the FF++ dataset, when trained on one compression level and tested on the other compression level, the proposed method is significantly better than the baselines. That is, SFIConv is insensitive to the interference caused by compression in the same type of dataset. Especially for ResNet, SFIC-ResNet achieves more robust detection under compression interference. However, our method does not achieve the best results in some test scenarios (such as DFDC vs FF++ LQ). We speculate that there are few common features between DFDC and FF++ datasets, and the strong compression operation further increases the differences between the two datasets, thus weakening the generalization ability of SFIConv.

2) Evaluation on CelebDF: In this subsection, we introduce a widely-used benchmark, namely CelebDF, to examine

TABLE VI: Cross-Dataset Evaluation Results (%) on DFDC and FF++ Datasets.

Methods	Training on DFDC				Training on FF++ HQ				Training on FF++ LQ			
	FF++ HQ		FF++ LQ		DFDC		FF++ LQ		DFDC		FF++ HQ	
	AUC	BACC	AUC	BACC	AUC	BACC	AUC	BACC	AUC	BACC	AUC	BACC
Meso-Incep [42]	58.29	55.30	56.99	53.75	62.19	52.37	70.72	57.33	60.37	57.50	75.68	68.21
Multi-task [43]	55.53	55.53	55.69	**55.69**	50.74	50.74	64.41	64.41	49.79	49.79	56.06	56.06
XceptionNet [35]	55.24	53.22	55.24	53.08	62.38	58.10	79.15	70.29	60.54	56.56	85.02	74.58
F^3-Net [44]	56.30	51.87	56.80	54.57	62.20	55.43	66.80	61.57	61.40	**58.04**	72.70	66.03
AMTENnet [31]	57.59	55.51	56.89	55.19	64.76	59.32	78.03	70.07	63.27	55.19	82.19	68.75
M2TR [45]	55.70	53.91	54.70	53.49	63.30	56.17	73.70	67.22	57.50	55.42	83.10	74.76
SFIC-ResNet26 (α=0.25,β=1.00)	58.26	54.86	**58.68**	54.47	63.85	59.83	**87.74**	**75.89**	58.54	55.12	89.83	79.68
SFIC-ResNet26 (α=0.50,β=1.00)	57.85	54.56	57.58	53.60	60.62	57.03	87.45	75.43	58.16	55.54	**90.71**	**80.39**
SFIC-VGGNet13 (α=0.25,β=0.75)	58.53	53.53	56.34	54.12	**65.63**	58.13	82.40	74.17	63.54	57.32	89.09	76.48
SFIC-VGGNet13 (α=0.50,β=1.00)	58.58	55.66	57.01	54.52	63.15	57.70	76.99	70.16	62.11	51.72	83.49	63.78
SFIC-VGGNet13 (α=0.75,β=0.75)	**62.13**	**58.39**	57.96	55.37	65.20	**59.98**	80.79	72.97	**64.12**	54.76	88.16	73.03

TABLE VII: Effectiveness of SFIConv in Cross-Dataset Evaluation for ResNets.

Networks	Hyper-parameters		Training Dataset	CelebDF Dataset	
	α	β		AUC	BACC
ResNet26	-	-	FF++ dataset	67.29	61.18
w/ SFIConv	0.25	0.50		68.30	61.38
w/ SFIConv	0.25	0.75		70.27	61.84
w/ SFIConv	0.25	1.00		**70.68**	**63.01**
w/ SFIConv	0.50	1.00		67.42	60.86
w/ SFIConv	0.75	1.00		67.73	61.58
ResNet50	-	-	FF++ dataset	61.75	57.61
w/ SFIConv	0.25	0.50		64.83	59.91
w/ SFIConv	0.25	0.75		66.17	60.84
w/ SFIConv	0.25	1.00		**68.90**	**61.98**
w/ SFIConv	0.50	1.00		63.14	58.40
w/ SFIConv	0.75	1.00		64.59	59.86
ResNet101	-	-	FF++ dataset	64.66	57.11
w/ SFIConv	0.25	0.50		66.83	60.37
w/ SFIConv	0.25	0.75		68.96	**60.78**
w/ SFIConv	0.25	1.00		**69.03**	59.93
w/ SFIConv	0.50	1.00		64.96	58.72
w/ SFIConv	0.75	1.00		67.22	60.54

TABLE VIII: Effectiveness of SFIConv in Cross-Dataset Evaluation for VGGNets.

Networks	Hyper-parameters		Training Dataset	CelebDF Dataset	
	α	β		AUC	BACC
VGGNet13	-	-	FF++ dataset	61.59	55.30
w/ SFIConv	0.25	0.50		65.30	57.95
w/ SFIConv	0.25	0.75		66.06	61.31
w/ SFIConv	0.25	1.00		64.32	60.65
w/ SFIConv	0.50	1.00		60.29	56.04
w/ SFIConv	0.75	1.00		**70.85**	**64.74**
VGGNet16	-	-	FF++ dataset	63.59	59.60
w/ SFIConv	0.25	0.50		62.55	57.28
w/ SFIConv	0.25	0.75		67.74	61.33
w/ SFIConv	0.25	1.00		**70.95**	**64.47**
w/ SFIConv	0.50	1.00		64.63	58.88
w/ SFIConv	0.75	1.00		68.71	62.55
VGGNet19	-	-	FF++ dataset	60.75	56.16
w/ SFIConv	0.25	0.50		66.65	58.53
w/ SFIConv	0.25	0.75		**67.15**	**60.54**
w/ SFIConv	0.25	1.00		64.70	60.08
w/ SFIConv	0.50	1.00		61.67	55.16
w/ SFIConv	0.75	1.00		62.08	58.02

whether the proposed SFIConv improves the generalization of backbone networks. To reduce the evaluation error as much as possible, we randomly select 120k face images from all video sequences in CelebDF for experimental evaluation. Consistent with previous works, the proposed detection method is also trained on the FF++ HQ dataset and tested on the CelebDF dataset.

We select ResNet and VGGNet as the baselines to evaluate the effectiveness of SFIConv in backbone networks with different depths and structures. Table VII and VIII report the results of cross-dataset evaluation on different backbone networks, where the best results are marked by bold fonts. The proposed SFIConv effectively improves the generalization performance on six backbone networks with different depths and structures. Actually, SFIConv benefits from its space-frequency interaction mechanism, which captures well the common artifacts in various face forgeries.

We also select 13 detection methods as the baselines for comparisons. Note that the AUC scores of the baselines are directly obtained from the results reported in their references. Table IX reports the detection results in cross-dataset evaluation. Only replacing the vanilla convolution in ResNet26 and VGGNet16, namely SFIC-ResNet26 and SFIC-VGGNet16, achieves the AUC scores of 70.7% and 71.0%, which are close to the results of most SOTA methods. This also shows that the backbone network constructed by SFIConv can achieve good generalization performance. Although there is still a gap with the best cross-domain detection method, considering that we have not used additional mechanisms to enhance the cross-domain detection ability, SFIConv still has great potential to improve the generalization capability.

G. Preliminary Investigations on Attention Region

To reveal the difference between vanilla convolution and SFIConv in mining forgery clues, we generate some Average Forgery Attention Maps (AFAMs) following the visualization method [52] to observe the differences among various detection networks. Usually, there are some deviations in the visualization range on a single image. Thus, only visualizing a single image does not accurately reflect the difference of

TABLE IX: AUC Score (%) of Cross-Dataset Evaluation Results on CelebDF Dataset.

Methods	Year	Publication	Training Dataset	CelebDF Dataset [37]
SMIL [46]	2020	MM	FF++ dataset	56.3
F^3-Net [44]	2020	ECCV	FF++ dataset	65.2
Two-branch [47]	2020	ECCV	FF++ dataset	73.4
Face X-ray [10]	2020	CVPR	FF++ dataset	74.2
Luo et al. [32]	2021	CVPR	FF++ dataset	79.4
MADD [17]	2021	CVPR	FF++ dataset	67.4
LTW [48]	2021	AAAI	FF++ dataset	64.1
SPSL [49]	2021	CVPR	FF++ dataset	76.9
M2TR [45]	2022	ICMR	FF++ dataset	65.7
DCL [50]	2022	AAAI	FF++ dataset	**81.0**
MC-LCR [51]	2022	KBS	FF++ dataset	71.6
GocNet [20]	2023	ESWA	FF++ dataset	67.4
LDFnet [16]	2023	TCSVT	FF++ dataset	65.7
SFIC-ResNet26 (α=0.25, β=1.00)	-	-	FF++ dataset	70.7
SFIC-VGGNet16 (α=0.25, β=1.00)	-	-	FF++ dataset	71.0

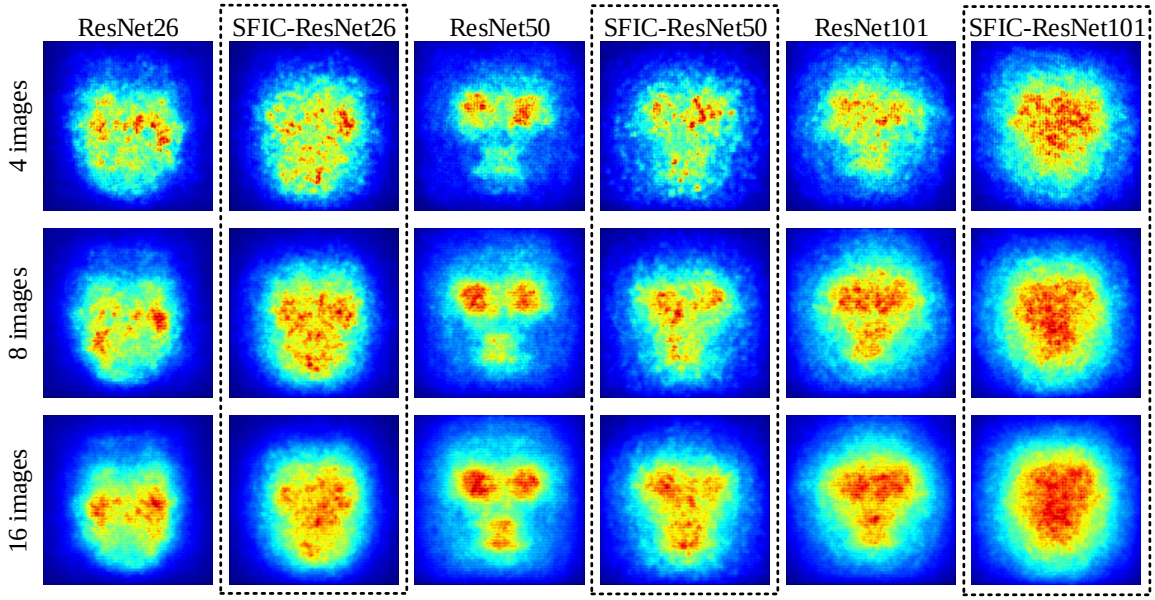

Fig. 5: Visualization of the attention regions. The visualization results of backbone network equipped with SFIConv are marked with dashed boxes.

the attention range. For AFAMs, they can reflect the average attention distribution of multiple fake face images, which avoids the deviation of single image. In our experiment, we randomly selected 4, 8 and 16 fake face images from different video sequences of the FF++ dataset. AFAMs are calculated on three groups of fake face images with different numbers to measure the scope of mining forgery clues.

Fig. 5 shows the visualization results, in which the first to third rows represent AFAMs calculated on different numbers of images. From this, it can be observed that the original backbone network tends to focus on the local region in the face image. However, after being equipped with SFIConv, the new backbone networks further expand the attention range and capture the global forgery clues in the face image. These results reveal the intrinsic reason why SFIConv is more suitable for capturing manipulation traces than vanilla convolution from the visual perspective.

V. Conclusion

In this work, we propose a novel SFIConv to address the inherent issue that vanilla convolution can not effectively capture the subtle manipulation traces for Deepfake detection. SFIConv fuses spatial-domain features and high-frequency information generated by MCSConv in an interactive manner to construct stronger feature representation containing manipulation traces. With flexible design, SFIConv not only reduces the FLOPs, but also greatly improves the accuracy of backbone network for Deepfake detection. The extensive experiments show that SFIConv can serve as an efficient component to seamlessly replace the vanilla convolution in existing backbone networks, which significantly promotes Deepfake detection without changing the model structure.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations,*, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[5] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[6] Z. Sun, Y. Han, Z. Hua, N. Ruan, and W. Jia, "Improving the efficiency and robustness of deepfakes detection through precise geometric features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3609–3618, 2021.

[7] Z. Guo, L. Hu, M. Xia, and G. Yang, "Blind detection of glow-based facial forgery," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7687–7710, 2021.

[8] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Z. Li, "Face forgery detection by 3d decomposition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2929–2939, 2021.

[9] Z. Guo, G. Yang, D. Wang, and D. Zhang, "A data augmentation framework by mining structured features for fake face image detection," *Computer Vision and Image Understanding*, vol. 226, p. 103587, 2023.

[10] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5001–5010, 2020.

[11] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji, "Local relation learning for face forgery detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1081–1088, 2021.

[12] Z. Guo, G. Yang, J. Chen, and X. Sun, "Exposing deepfake face forgeries with guided residuals," *IEEE Transactions on Multimedia*, 2023.

[13] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," *arXiv preprint arXiv:1910.12467*, 2019.

[14] S. Ge, F. Lin, C. Li, D. Zhang, J. Tan, W. Wang, and D. Zeng, "Latent pattern sensing: Deepfake video detection via predictive representation learning," in *ACM Multimedia Asia*, pp. 1–7, 2021.

[15] C. Miao, Z. Tan, Q. Chu, N. Yu, and G. Guo, "Hierarchical frequency-assisted interactive networks for face manipulation detection," *IEEE Transactions on Information Forensics and Security*, 2022.

[16] Z. Guo, L. Wang, W. Yang, G. Yang, and K. Li, "Ldfnet: Lightweight dynamic fusion network for face forgery detection by integrating local artifacts and global texture information," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

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

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

[19] C. Kong, B. Chen, H. Li, S. Wang, A. Rocha, and S. Kwong, "Detect and locate: Exposing face manipulation by semantic-and noise-level telltales," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1741–1756, 2022.

[20] Z. Guo, G. Yang, D. Zhang, and M. Xia, "Rethinking gradient operator for exposing ai-enabled face forgeries," *Expert Systems with Applications*, vol. 215, p. 119361, 2023.

[21] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4834–4844, June 2021.

[22] L. Zhang, H. Yang, T. Qiu, and L. Li, "Ap-gan: Improving attribute preservation in video face swapping," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2226–2237, 2022.

[23] 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.

[24] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[25] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2387–2395, 2016.

[26] S. Tripathy, J. Kannala, and E. Rahtu, "Facegan: Facial attribute controllable reenactment gan," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1329–1338, 2021.

[27] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797, 2018.

[28] T. Wei, D. Chen, W. Zhou, J. Liao, Z. Tan, L. Yuan, W. Zhang, and N. Yu, "Hairclip: Design your hair by text and reference image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18072–18081, 2022.

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

[30] J. Yang, S. Xiao, A. Li, W. Lu, X. Gao, and Y. Li, "Msta-net: Forgery detection by generating manipulation trace based on multi-scale self-texture attention," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4854–4866, 2022.

[31] 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.

[32] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16317–16326, 2021.

[33] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, 2018.

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[35] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *International Conference on Computer Vision (ICCV)*, 2019.

[36] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," *arXiv preprint arXiv:1910.08854*, 2019.

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

[38] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in neural information processing systems*, pp. 10215–10224, 2018.

[39] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3677–3685, 2017.

[40] 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.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

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

[44] 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*, pp. 86–103, Springer, 2020.

[45] 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*, pp. 615–623, 2022.

[46] 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 *Proceedings of the 28th ACM international conference on multimedia*, pp. 1864–1872, 2020.
[47] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed, "Two-branch recurrent network for isolating deepfakes in videos," in *European Conference on Computer Vision*, pp. 667–684, Springer, 2020.
[48] K. Sun, H. Liu, Q. Ye, J. Liu, Y. Gao, L. Shao, and R. Ji, "Domain general face forgery detection by learning to weight," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 2638–2646, 2021.
[49] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 772–781, 2021.
[50] K. Sun, T. Yao, S. Chen, S. Ding, J. Li, and R. Ji, "Dual contrastive learning for general face forgery detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 2316–2324, 2022.
[51] G. Wang, Q. Jiang, X. Jin, W. Li, and X. Cui, "Mc-lcr: Multimodal contrastive classification by locally correlated representations for effective face forgery detection," *Knowledge-Based Systems*, p. 109114, 2022.
[52] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14923–14932, 2021.

Zhiqing Guo received the Ph.D. degree in Computer Science and Technology from Hunan University in 2023. At present, he has been introduced to the Xinjiang University as an A-level Young Talent to engage in teaching and scientific research. He is the PI of several projects such as Natural Science Foundation of China (NSFC). His research interests include digital media forensics, computer vision and deep learning. He has published several research papers in international journals such as IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, ACM Transactions on Multimedia Computing Communications and Applications, Expert Systems with Applications, Computer Vision and Image Understanding, etc.

Zhenhong Jia received the B.S. degree from Beijing Normal University, Beijing, China,in 1987, and the M.S. and the Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China,in 1995. He is currently a full professor of information and communication engineering with Xinjiang University, and the director in the Autonomous Region Key Laboratory for Signal and Information Processing. His research interests include digital image processing, and photoelectric information detection and sensors. He has published research papers in international journals such as IEEE Transactions on Fuzzy Systems, IEEE Transactions on Circuits and Systems for Video Technology, Information Sciences, etc.

Liejun Wang received his Ph.D. degree in the School of Information and Communication Engineering from Xi'an Jiaotong University in 2012. He is currently a full professor with the School of Computer Science and Technology, Xinjiang University. His research interests include wireless sensor networks, computer vision, and natural language processing. He has published research papers in international journals such as IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Geoscience and Remote Sensing, IEEE Transactions on Cybernetics, IEEE Transactions on Vehicular Technology, etc.

Dewang Wang received his M.S. degree from College of Computer Science and Information Technology of Guangxi Normal University, Guilin, China, in 2019. Currently, he has been pursuing the Ph.D. degree with the Hunan University, Changsha, China. His research interests include data hiding, steganography, and deep learning. He has published research papers in international journals such as ACM Transactions on Multimedia Computing Communications and Applications, etc.

Gaobo Yang received the Ph.D. degree in Communication and Information System from Shanghai University in 2004. He is a professor in Hunan University, China. He made an academic visit to University of Surrey, UK, from August 2010 to August 2011. He is the PI of several projects such as Natural Science Foundation of China (NSFC), Special Pro-phase Project on National Basic Research Program of China (973) and program for New Century Excellent Talents (NCET) in university. His research interests include image and video signal processing, digital media forensics. He has published many research papers in international journals such as IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Broadcasting, ACM Transactions on Multimedia Computing, Communications, and Applications, etc.

Nikola Kasabov (Fellow, IEEE) received MSc degree in computing and electrical engineering and a Ph.D. degree in mathematical sciences from the Technical University of Sofia, Bulgaria, in 1971 and 1975, respectively. He is the Founding Director of KEDRI and Professor of knowledge engineering with the School of Computing and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand. He also holds a George Moore Chair Professor of data analytics at the University of Ulster UK and Honorary Professorships at the Teesside University UK and the University of Auckland NZ. His major research interests include information science, computational intelligence, neural networks, bioinformatics, and neuroinformatic. He has published research papers in international journals such as IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Cognitive and Developmental Systems, etc.