A robust GAN-generated face detection method based on dual-color spaces and an improved Xception

Beijing Chen, Xin Liu, Yuhui Zheng, *Member, IEEE*, Guoying Zhao, *Senior Member, IEEE*, and Yun-Qing Shi, *Fellow, IEEE*

Abstract—**In recent years, generative adversarial networks (GANs) have been widely used to generate realistic fake face images, which can easily deceive human beings. To detect these images, some methods have been proposed. However, their detection performance will be degraded greatly when the testing samples are post-processed. In this paper, some experimental studies on detecting post-processed GAN-generated face images find that (a) both the luminance component and chrominance components play an important role, and (b) the RGB and YCbCr color spaces achieve better performance than the HSV and Lab color spaces. Therefore, to enhance the robustness, both the luminance component and chrominance components of dual-color spaces (RGB and YCbCr) are considered to utilize color information effectively. In addition, the convolutional block attention module and multilayer feature aggregation module are introduced into the Xception model to enhance its feature representation power and aggregate multilayer features, respectively. Finally, a robust dual-stream network is designed by integrating dual-color spaces RGB and YCbCr and using an improved Xception model. Experimental results demonstrate that our method outperforms some existing methods, especially in its robustness against different types of post-processing operations, such as JPEG compression, Gaussian blurring, gamma correction, and median filtering.**

Index Terms—**generated face, generative adversarial network, Xception, color space.**

I. Introduction

GENERATIVE adversarial networks (GANs) were first put forward by Goodfellow et al. in 2014 [1]. The basic idea is to train two adversarial networks: a generator, and a discriminator. They compete against each other to achieve the expected results. The generator generates images, while the discriminator determines whether the generated images are generated. Then, the purpose of generating high-quality images that the discriminator cannot recognize will be achieved after multiple iterations. Generating fake faces is one of the popular applications of GANs. Recently, with the rapid development of GANs, an increasing number of advanced GANs have been designed, such as DCGAN [2], WGAN [3], WGAN_GP [4], EBGAN [5], BEGAN [6], PGGAN [7], StyleGAN [8], DRGAN [9], CycleGAN [10], StarGAN [11], and Bridge-GAN [12]. Among them, the PGGAN and StyleGAN can generate high resolution (1024×1024) and high-quality face images that can even deceive human beings. The DRGAN is a conditional generative adversarial network that can automatically perform distortion rectification. The CycleGAN and StarGAN perform image-to-image translation and style alteration. The Bridge-GAN translates text descriptions to images with high content consistency. Regarding the local generation problem, Xu *et al*. [13] introduced edges into convolutional GAN-based inpainting and split the inpainting task into edge generation and edge-based image generation steps. If these generated faces were widely spread on the internet, they could cause security problems in politics, justice, criminal investigation, reputation protection, and other areas. Therefore, it is urgent to propose efficient methods for detecting these generated faces.

Recently, researchers have begun to explore the problem of generated face detection. GAN-generated face detection is a binary classification problem that determines whether a given face image is GAN-generated or not. The existing methods can be roughly divided into two categories: intrinsic feature-based and deep learning-based. *The former exploits the inconsistency of natural faces and GAN-generated faces in terms of facial attributes, texture information, color information, and other factors*. Yang *et al*. [14] extracted facial feature points and used support vector machine as the binary classifier. Matern *et al*. [15] distinguished natural images from generated images based on visual features, such as facial contours and eyes. Liao *et al.* [16] proposed a two-stream CNN network to detect both tampering artifact evidence and local noise residual evidence for image operator chain detection. McCloskey *et al*. [17] found that the color treatment of GANs was markedly different from that of real cameras, thus they utilized color cues to distinguish the synthesized images. Li *et al*. [18] found that chrominance components would expose more artifacts. Therefore, they exploited the chrominance components in the residual domain to detect generated images. In addition, Nataraj *et al*. [19] proposed using the images after color co-occurrence matrix pre-processing as input. However, all of these intrinsic attribute-based methods are based on hand-crafted features, which could limit their performance [20]. *Therefore, some deep learning-based methods have been introduced into the field of GAN-generated face detection*. Do *et al*. [21] employed VGGNet to distinguish between the generated faces and natural

This work was supported in part by the National Natural Science Foundation of China under Grants 62072251, U20B2065, and 61972206; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20211539; in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

Beijing Chen, Xin Liu, and Yuhui Zheng are with the Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China, also with School of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: nbutimage@126.com, 1273723636@qq.com, zheng_yuhui@nuist.edu.cn).

Guoying Zhao is with the Center for Machine Vision and Signal Analysis, University of Oulu, Oulu FI-90014, Finland (e-mail: guoying.zhao@oulu.fi).

Yun-Qing Shi is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark 07102, USA (e-mail: shi@njit.edu).

faces. Kong *et al*. [22] presented a new framework that attempts to reveal the real face hidden behind the fake face by using a convolutional neural network (CNN) to learn the joint information of the face and the audio. Chen *et al*. [23] exploited CNN to extract global and local features to detect GAN-generated faces. Mo *et al*. [24] found that the differences between the natural faces and the generated faces in the residual field were more obvious than those in the plain field. Therefore, they transformed the face images into the residual domain by a high-pass filter and then extracted features from the residual input by a CNN. Yang *et al*. [25] exploited a deep CNN to extract dynamic features of lips to distinguish fake faces. He *et al*. [26] exploited a shallow CNN to extract chrominance components in different color spaces to improve robustness. Chen *et al.* [27] proposed an improved Xception model for locally GAN-generated face detection. The Xception model was improved by introducing the feature pyramid network and Inception block with dilated convolution to obtain the multiscale and multilevel features for the small-generated face regions. Fu *et al*. [28] designed a dual-channel network to extract robust representations for detecting GAN-generated faces. Jia *et al* [29] exploited a dual-stream network to extract discriminative information from RGB and YCbCr spaces to detect 3D face spoofing. Liu *et al*. [30] used the Gram matrix and ResNet architecture [31] to extract global texture features to improve the generalization and robustness. Mi *et al*. [32] designed a self-attention-based algorithm to exploit the structural defects in GANs by taking advantage of the up-sampling process conducted by the transposed convolution operation. Based on the social cognitive process of the human brain, Fernando *et al*. [33] proposed hierarchical attention memory networks to detect fake faces. Hu *et al.* [34] applied a temporality-level stream to extract temporal correlation features and combined the frame-level stream to detect compressed Deepfake videos.

Most of the above-mentioned studies have achieved good performance in detecting the generated faces free of post-processing. However, in practical scenarios, the generated faces are often accompanied by some post-processing operations, such as JPEG compression and blurring. Unfortunately, most of the above-mentioned studies do not consider the robustness against the post-processing operations. Consequently, their performance will be degraded greatly when detecting post-processed generated faces. Data enhancement is a solution. However, in practice, the post-processing operation and its strengths are usually unknown and complex. Therefore, it is impossible to consider every post-processing step with arbitrary strengths by data enhancement. In addition, data enhancement will greatly increase the amount of data, which will affect the training speed and even lead to the overfitting or failure of the training. Therefore, it is very important to design a real robust model. To make the detection method as effective as possible in practical scenarios, a new robust detection method is proposed. Since Xception [35] has shown good performance in GAN-generated face detection [36, 37], it is used as the network benchmark. The main contributions are as follows.

- The luminance and chrominance components are analyzed and compared to decide whether only two chrominance components used in [18, 26] or all three components need to be considered in the network input.
- Four color spaces are compared in terms of their robustness against post-processing to determine which color space is suitable for GAN-generated face detection.
- The Xception model is improved in robustness by introducing two technologies: convolutional block attention module (CBAM) [38, 39] and multilayer feature aggregation (MLFA) [40]. The CBAM can provide evidence for important information. The MLFA allows the capture of multilevel features to complement the missing details of deep representations.
- A robust dual-stream network model is designed by integrating the luminance component and chrominance components of dual-color spaces RGB and YCbCr as the inputs of the dual-stream network and using an improved Xception model. The proposed model performs better than some existing models, especially in robustness against different types of post-processing operations.

The remaining of this paper is organized as follows. Section II reviews the relevant works and technologies. Section III describes the structure of the proposed model in detail. The experimental results and analysis are presented in Section IV. Finally, Section V summarizes the paper.

II. Related works and related techniques

In this section, first, some related studies that use multiple color spaces for generated face detection and the network benchmark Xception [35] model are investigated; then, CBAM [38, 39] is introduced.

A. Detection methods using multiple color spaces

Other color spaces instead of RGB have been exploited to detect GAN-generated faces in recent years. Li *et al*. [18] pointed out that since GANs usually generate images in RGB space, they tend to follow the properties of natural images in RGB space and pay less attention to other color spaces. Therefore, Li *et al*. analyzed the generated images in the RGB, YCbCr, and HSV color spaces, and found that the chrominance components had a larger difference between natural and GAN-generated images than the luminance component. Therefore, they exploited the co-occurrence matrix to extract the features of chrominance components (H, S, Cb, and Cr) for classification. He *et al*. [26] pointed out that common post-processing operations would make the abnormal traces in the RGB space unreliable, while the statistical characteristics of chrominance information in other color spaces can be more distinguishable and robust. Therefore, they exploited a well-designed shallow CNN to extract the features of the chrominance components and then used the random forest classifier [41] for classification.

However, both of these studies [18, 26] use only chrominance components. Consequently, the luminance information is lost. However, the luminance information has better resistance to some post-processing operations than the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3116679, IEEE Transactions on Circuits and Systems for Video Technology

chrominance components, such as JPEG compression, and gamma correction. The detailed experimental analysis will be provided in Section III.

B. *Xception model*

The Xception is an improved version of Inception-v3 [42]. In [32], the authors believed that it was better to learn channel correlation and spatial correlation separately. Therefore, they used depthwise separable convolution to replace the ordinary convolution operation in Inception-v3. The depthwise separable convolution decomposes ordinary convolution into two processes: first, spatial convolution performs on each input channel independently, and then, pointwise convolution adopts a 1 × 1 kernel to convolve point by point. The architecture of the Xception model is provided in Fig. 1. As shown in Fig. 1, the Xception model consists of 14 blocks. The 14 blocks contain 3 common convolution layers and 33 depthwise separable convolution layers in total. All of the blocks expect the first and last blocks have linear residual connections around them.

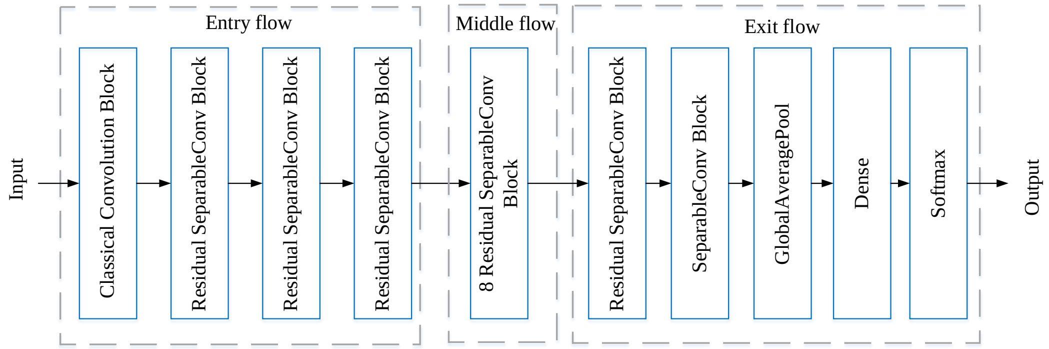

Entry flow
Middle flow
Exit flow
Input
Classical Convolution Block
Residual SeparableConv Block
Residual SeparableConv Block
Residual SeparableConv Block
8 Residual SeparableConv Block
Residual SeparableConv Block
SeparableConv Block
GlobalAveragePool
Dense
Softmax
Output

Fig. 1. Architecture of the Xception model.

C. *Convolutional block attention module*

In human vision, attention can filter out irrelevant information and enhance important information, leading people to focus on the local area of the whole scene [43]. The CBAM [38] is a lightweight module that sequentially infers the attention map along two independent dimensions (channel and space), and then multiplies the attention map with the input feature map. Fig. 2 is the overview of CBAM. CBAM can help the network to extract effective features by learning information to be emphasized or suppressed.

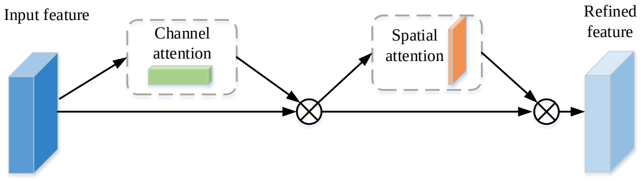


Fig. 2. Overview of the CBAM.

The channel attention is computed as,

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F))) + MLP(MaxPool(F)) \\ &= \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))), \end{aligned} \tag{1}$$

where $M_c \in R^{c\times 1\times 1}$ is the channel attention map of the input feature F, c represents the number of channels, σ denotes the sigmoid function, MLP is the multilayer perceptron, $AvgPool$ and $MaxPool$ represent the average pooling and maximum pooling, respectively, and $W_0 \in R^{c/r\times c}$ and $W_1 \in R^{c\times c/r}$ represent the MLP weights, where r is the reduction ratio.

The spatial attention is computed as,

$$\begin{aligned} M_s(F) &= \sigma(f^{7\times 7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7\times 7}(F^s_{avg}; F^s_{max})), \end{aligned} \tag{2}$$

where $M_s \in R^{H\times W}$ is the spatial attention map of the input feature F, $H\times W$ represents its size, and $f^{7\times 7}$ denotes a convolution operation with a filter size of 7 × 7.

III. Proposed Method

In this section, a robust dual-stream network model is proposed for detecting GAN-generated faces. Xception is used as a network benchmark since it has shown good performance in feature extraction of GAN-generated face detection [36, 37]. Here, the CBAM and MLFA modules are introduced into the Xception to improve its performance. In addition, the luminance component and chrominance components of dual-color spaces (RGB and YCbCr) are considered to be the network input after a detailed comparative analysis of the luminance component and chrominance components as well as different color spaces. Notice that the generated faces with different ages [44, 45] or expressions [46, 47] do not affect the GAN-generated face detection task because the detection performance mainly depends on the detection model itself and the training data provided. Therefore, we do not analyze the influence of age and expression.

A. *Is the luminance component important?*

The most recent detection studies [14-34], except for two works [18] and [26], consider only the RGB color space. Regarding [18] and [26], although they consider other color spaces, they exploit only the chrominance components, discarding the luminance component. Is the luminance component beneficial to robustness against post-processing? This question will be answered in the following through feature similarity analysis and performance comparison. Notice that the YCbCr color space is considered. The reasons are as follows: (a) in [48], the authors pointed out that the YCbCr

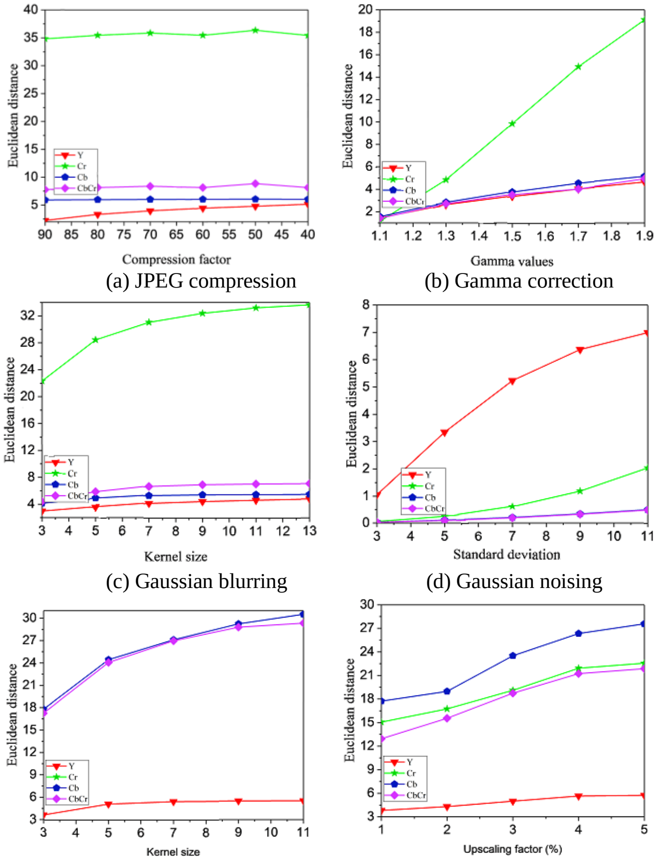
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3116679, IEEE Transactions on Circuits and Systems for Video Technology

space was perceptually uniform and had a good separation of luminance and chrominance. Therefore, they adopted the YCbCr space in color face detection; (b) the YCbCr space has been widely used in image and video compression standards, e.g., JPEG and MPEG. Therefore, the use of the YCbCr color space will enhance the robustness against JPEG compression.

Regarding the feature similarity analysis, a basic idea is that the stronger the similarity between the features extracted from the original image and those extracted from the corresponding post-processing image, the stronger the robustness. Specifically, the Xception model [35] is used to extract features in the YCbCr color space, and the Euclidean distance between the original image features and its corresponding post-processed image features are calculated. The Euclidean distance $d(X, Y)$ of two feature vectors $X = \{X_1, X_2, \ldots, X_n\}$ and $Y = \{Y_1, Y_2, \ldots, Y_n\}$ can be computed as follows,

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \quad (3)$$

The smaller the distance is, the more similar the original image features and the post-processed image features.



(a) JPEG compression (b) Gamma correction

(c) Gaussian blurring (d) Gaussian noising

(e) Median filtering (f) Resizing

Fig. 3. Average Euclidean distance values between the original image features and its corresponding post-processed image features under different types of post-processing operations with different strengths.

A total of 2,000 face images from the natural CelebA dataset [49] and their corresponding generated images by the PGGAN are considered for analysis. The generated face images are processed by some post-processing operations, such as JPEG compression, gamma correction, Gaussian blurring, Gaussian noising, median filtering and resizing. Then, the Euclidean distance is calculated between the post-processed generated image features and their corresponding original features. The average Euclidean distance values of 2,000 pairs of images are provided in Fig. 3. Fig. 3 shows that the luminance component is more robust to JPEG compression, median filtering, and resizing, while the chrominance components are more robust to Gaussian blurring and Gaussian noising. However, in most cases of actual scenarios, the post-processing operation is usually unknown. Therefore, it is a good choice to combine the chrominance and luminance components.

Furthermore, the performance of the chrominance component and luminance component are compared on the final GAN-generated face detection task. The YCbCr color space and the Xception model [35] are considered. The experimental datasets contain the natural CelebA dataset and its corresponding generated dataset by the PGGAN model. There are 202,599 natural images and 202,599 GAN-generated images. A total of 202,599 pairs were divided into training, validation, and testing sets at a ratio of 8:1:1. The evaluation metric *accuracy* is used to evaluate the performance of the chrominance component and luminance component. It can be computed by,

$$Accuracy = \frac{C_{number}}{T_{number}} \times 100\% \quad (4)$$

where C_{number} is the number of correctly classified face images, and T_{number} denotes the number of total face images.

The experimental results are shown in Table I. It can be seen from Table I that the chrominance components Cb and Cr have better robustness against Gaussian noising, while they only achieve an accuracy of 50% for JPEG compression. However, the luminance component Y is the opposite. Therefore, when the type of post-processing operation is unknown, considering both the chrominance component and luminance component is a good choice.

B. *Comparison of four color spaces*

The previous subsection shows that both the chrominance and luminance components are helpful for improving the performance in detecting post-processed GAN-generated faces. However, which color space is the optimal one? To address this question, in this subsection, four different color spaces (RGB, YCbCr, HSV, and Lab) are compared in terms of GAN-generated face detection. The experimental dataset is the same as in the previous subsection. The experimental results are shown in Table II. The results show that both the RGB and YCbCr spaces achieve better performance in terms of robustness than the other two color spaces, especially for JPEG compression. This finding further verifies the advantage of the YCbCr space in GAN-generated face detection. In addition, RGB and YCbCr have different effects on different post-processing operations. Therefore, dual-color spaces (RGB and YCbCr) are considered in the proposed method.

C. *Xception model combined with CBAM*

The CBAM module (Fig. 2) considers the importance of pixels not only in different channels but also in different positions of the same channel. In this paper, the CBAM module is applied to focus on the more important features in the feature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3116679, IEEE Transactions on Circuits and Systems for Video Technology

TABLE I
DETECTION ACCURACY (%) COMPARISON OF DIFFERENT COLOR COMPONENTS IN YCBCR COLOR SPACE

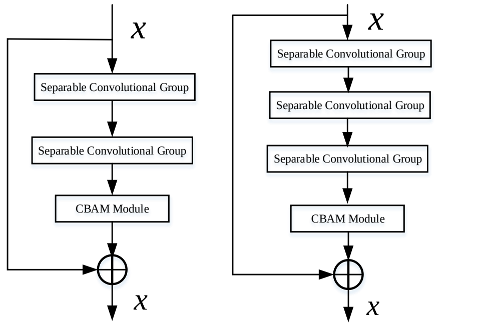
Color component	JPEG compression with different quality factors			Gamma correction with different gamma values			Median filtering with different kernel sizes		
	90	70	50	1.1	1.4	1.7	3×3	5×5	7×7
Y	99.1	89.9	75.5	99.7	99.5	98.8	92.3	70.8	64.7
Cb	50.0	50.0	50.0	99.6	99.4	91.3	83.7	50.0	50.0
Cr	50.0	50.0	50.0	99.7	99.5	92.4	60.2	50.0	50.0
CbCr	50.0	50.0	50.0	99.7	99.1	95.2	77.4	50.0	50.0
YCbCr	**97.1**	**91.4**	**83.8**	**99.6**	**98.4**	**97.1**	**94.7**	**75.4**	**66.6**

Color component	Gaussian blurring with different kernel sizes			Gaussian noising with different std values			Resizing with different upscaling factors		
	3×3	5×5	7×7	3	5	7	1%	3%	5%
Y	95.8	87.7	65.3	98.6	90.1	98.6	90.2	78.2	69.8
Cb	92.2	65.7	56.3	99.4	99.9	99.4	90.5	60.2	50.0
Cr	77.1	66.3	57.2	99.9	99.4	99.9	79.3	52.9	50.0
CbCr	86.5	71.2	65.1	99.9	99.5	99.9	91.2	70.5	60.4
YCbCr	**96.8**	**91.2**	**81.1**	**98.3**	**97.4**	**98.3**	**93.7**	**82.3**	**71.5**

TABLE II
DETECTION ACCURACY (%) COMPARISON OF DIFFERENT COLOR SPACES

Color spaces	JPEG compression with different quality factors			Gamma correction with different gamma values			Median filtering with different kernel sizes		
	90	70	50	1.1	1.4	1.7	3×3	5×5	7×7
RGB	96.9	90.1	81.5	98.9	98.2	95.4	94.8	76.6	65.6
YCbCr	97.1	91.4	83.8	99.6	98.4	97.1	94.7	75.4	66.6
HSV	61.4	57.5	50.0	99.1	98.5	97.8	85.3	52.7	50.0
Lab	52.7	50.0	50.0	99.3	99.1	98.2	93.2	61.3	52.2

Color spaces	Gaussian blurring with different kernel sizes			Gaussian noising with different std values			Resizing with different upscaling factors		
	3×3	5×5	7×7	3	5	7	1%	3%	5%
RGB	96.4	90.8	80.7	98.2	97.1	92.5	93.5	81.7	70.1
YCbCr	96.8	91.2	81.1	98.3	97.4	91.5	93.7	82.3	71.5
HSV	94.2	58.3	52.1	99.6	99.1	98.2	91.2	69.4	56.8
Lab	96.5	84.2	75.3	99.7	99.5	98.9	72.5	59.7	55.9


x
Separable Convolutional Group
Separable Convolutional Group
CBAM Module
x
x
Separable Convolutional Group
Separable Convolutional Group
Separable Convolutional Group
CBAM Module
x

(a). Different dimensions (b). Same dimension

Fig. 4. Architecture of the CBAM-residual block for the input and output with same/different dimensions.

map. It is inserted into each residual block of the Xception model as the CBAM-residual block shown in Fig. 4. In each CBAM-residual block, a separable convolutional group contains a separable convolutional layer, a batch normalization (BN) layer [50], and a ReLU activation layer. The 1×1convolution is used when the dimensions of the input feature map and the output feature map are not the same, as shown in Fig. 4(a). The structure of Fig. 4(b) is suitable for the case where the input feature map and the output feature map have the same dimensions. To gain a deeper understanding of the improvement of the CBAM, Fig. 5 presents the classification activation maps [51] before and after the CBAM. As shown in Fig.5, the features after the CBAM module are more obvious and more reasonable.

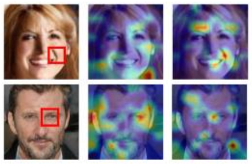


Fig. 5. Classification activation maps before and after the CBAM. The second column is before the CBAM, while the third column is after the CBAM.

D. Main framework of the proposed method

Based on the above discussion and analysis, a robust dual-stream network model is proposed for GAN-generated face detection by considering the luminance component and chrominance components of dual-color spaces (RGB and YCbCr) and using Xception, CBAM, and MLFA. The MLFA module aggregates multilayer features from different convolutional layers (blocks). It can make the corresponding model capable of finding inter-modal correlations not only at a fine level but also at a coarse level. With the deepening of the network, some shallow features, which have an important role in GAN-generated face detection, will be lost. As shown in Fig. 6, there is a large difference between the shallow and deep features. The deep features are more abstract, while the shallow features are more direct, focusing on some unnatural areas directly, especially for the post-processed images. Therefore, the MLFA module is introduced to save the features extracted from the shallow layers, and then fuse them with the deep features to enrich the features and improve the performance. In fact, similar ideas are also used in the U-Net model [52], which has been extensively used in the field of image segmentation.

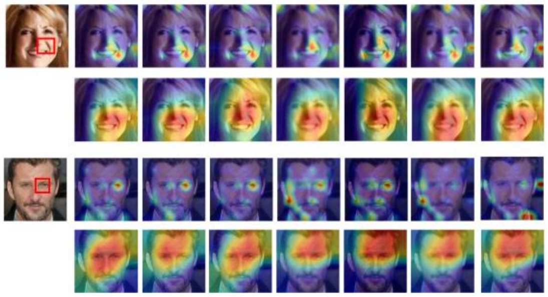


Fig. 6. Classification activation maps of the shallow layer and deep layer under different post-processing operations. For each image, the first row is for the shallow layer, while the second row is for the deep layer. For each row, from left to right, images correspond to the original image and its maps, the maps of JPEG compression, Gaussian noising, Gaussian blurring, gamma correction, median filtering and resizing.

An overview of the main framework is presented in Fig. 7. As shown in Fig. 7, the proposed detection method has two streams (RGB stream and YCbCr stream). In the proposed model, the inputs represented by two different color spaces are thrown into the detection module to obtain the classification scores; then, the scores are fused by the average method. The details of the detection module are given in Fig. 8. As shown in Fig. 8, the detection module is based on an improved Xception that combines the CBAM and MLFA. It contains five submodules with 16 different blocks, B_1-B_{16}. The pre-processing submodule (B_1-B_2) uses an inception block where the convolution kernel size and activation function are 3×3 and ReLU, respectively. The information content (B_3-B_5) submodule aims to learn shallow information. It is composed of three CBAM-residual blocks. Since the input features and output features of the CBAM-residual block have different dimensions, a 1×1 convolution is considered (Fig. 4(a)). The deep semantic submodule is used to extract high-level semantic information. It has ten CBAM-residual blocks (B_6-B_{16}). The network parameters of each block are shown in Table III. Notice that the meaning of the number below each block is as follows: (a) m×1 represents a vector, where m represents the length of the vector; (b) n represents the number of channels in the feature map. The MLFA submodule focuses on extracting multi-level features. It fuses the features extracted by the pre-processing submodule and the information content submodule as the shallow features for the subsequent decision submodule. The decision submodule first merges the deep semantic features and shallow features and then applies two FC layers to reduce the feature dimensionality. Finally, the widely used softmax loss is considered for classification,

$$L_{soft\,max} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^{T}x_i+b_{y_i}}}{\sum_{j=1}^{n}e^{W_j^{T}x_i+b_j}} \tag{5}$$

where N and n represent the batch size and the number of categories, respectively, x_i and y_i denote the deep feature and the label of i-th sample, W and b are the weight vector and the bias term, respectively. The goal is to minimize the loss $L_{softmax}$.

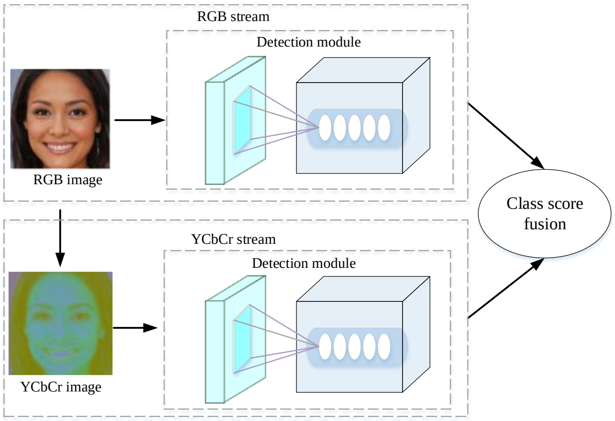


Fig. 7. Overview of the proposed detection method.

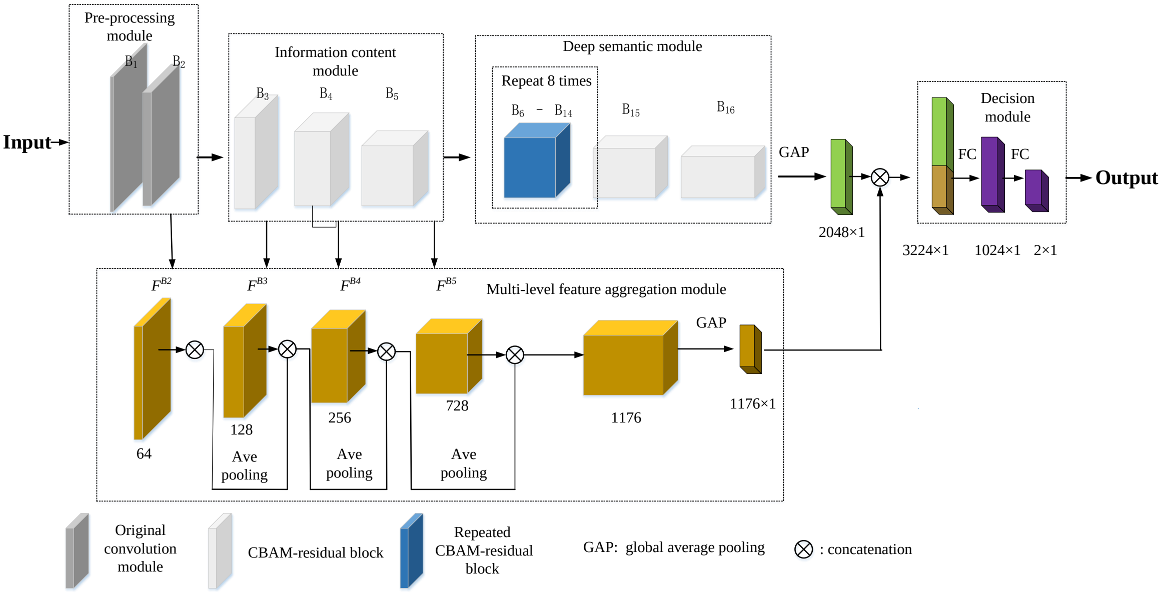


Fig. 8. Details of the proposed detection module.

TABLE III
NETWORK PARAMETERS OF 16 BLOCKS B_1-B_{16} IN THE FIG. 8

Layers	Name	Parameters
B_1	Conv2D_1	Filters = 32, Kernel_size=3, Strides = 2, Padding "Valid"
B_2	Conv2D_2	Filters = 64, Kernel_size=3, Strides = 2, Padding "Valid"
	Residual_Conv2d_1	Filters = 128, Kernel_size=1, Strides = 2, Padding "Same"
B_3	SeparableConv2d_1	Filters = 128, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_2	Filters = 128, Kernel_size=3, Strides = 1, Padding "Same"
	MaxPooling2D_1	Kernel_size=3, Strides = 2, Padding "Same"
	Add_1	(MaxPooling2D_1, Residual_Conv2d_1)
	Residual_Conv2d_2	Filters = 256, Kernel_size=1, Strides = 2, Padding "Same"
B_4	SeparableConv2d_3	Filters = 256, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_4	Filters = 256, Kernel_size=3, Strides = 1, Padding "Same"
	MaxPooling2D_2	Kernel_size=3, Strides = 2, Padding "Same"
	Add_2	(MaxPooling2D_2, Residual_Conv2d_2)
	Residual_Conv2d_3	Filters = 728, Kernel_size=1, Strides = 2, Padding "Same"
B_5	SeparableConv2d_5	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_6	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	MaxPooling2D_3	Kernel_size=3, Strides = 2, Padding "Same"
	Add_3	(MaxPooling2D_3, Residual_Conv2d_3)
B_6-B_{14}	Residual	Add_4
	SeparableConv2d_7	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_8	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_9	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	Add_4	(SeparableConv2d_9, Residual)
B_{15}	Residual_Conv2d_4	Filters = 1024, Kernel_size=1, Strides = 2, Padding "Same"
	SeparableConv2d_10	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_11	Filters = 728, Kernel_size=3, Strides = 1, Padding "Same"
	MaxPooling2D_4	Kernel_size=3, Strides = 2, Padding "Same"
	Add_5	(MaxPooling2D_4, Residual_Conv2d_4)
B_{16}	SeparableConv2d_12	Filters = 1536, Kernel_size=3, Strides = 1, Padding "Same"
	SeparableConv2d_13	Filters = 2046, Kernel_size=3, Strides = 1, Padding "Same"

The pseudo-code of the proposed method can be summarized in Algorithm 1.

Algorithm 1: Pseudo-code of the proposed method

```
Input: Image $\mathbf{I_{RGB}}$
1: if step == Train do
    2: Convert RGB image $\mathbf{I_{RGB}}$ into YCbCr color space as $\mathbf{I_{YcbCr}}$;
    3: Extract shallow features in the pre-processing submodule and information content submodule of four different blocks $B_2$, $B_3$, $B_4$, $B_5$ as $F_S^{B2}$, $F_S^{B3}$, $F_S^{B4}$, $F_S^{B5}$;
    4: Fusion the shallow features in the MLFA submodule as $F_{MS} = Concatenate(F_S^{B2}, F_S^{B3}, F_S^{B4}, F_S^{B5})$;
    5: Extract the deep semantic features from $F_S^{B4}$ in the deep semantic submodule as $F_D$;
    6: Reduce the dimensions of the shallow features and deep semantic features as $F_{GMS} = GAP(F_{MS})$, $F_{GD} = GAP(F_D)$;
    7: Merge the shallow features $F_{GMS}$ and deep semantic features $F_D$ as $F_C = Concatenate(F_{GMS}, F_{GD})$;
    8: Obtain the probability value of the YCbCr stream as $P_{YCbCr}$ = fully connected $FC(F_C)$;
    9: Repeat steps 3-8 for the RGB color space, obtaining the probability value of the RGB stream $P_{RGB}$;
    10: Predicate results as $P = Argmax(Average(P_{RGB}, P_{YCbCr}))$;
    11: Calculate softmax as $L_{softmax} = Softmax\_Loss(P, Label)$;
    12: Minimize $L_{softmax}$, and update the parameters of both the RGB stream and YCbCr stream by back propagation;
13: end if
14: if step == Test do
    15: Calculate the probability values of two streams as $P_{RGB} = RGB\_stream(\mathbf{I_{RGB}})$, $P_{YCbCr} = YCbCr\_stream(\mathbf{I_{YCbCr}})$;
    16: Predicate result $P = Argmax(Average(P_{RGB}, P_{YCbCr}))$;
17: end if
Output: Predicted result (natural or GAN-generated)
```

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, several experiments are conducted to demonstrate the effectiveness of the proposed method. Before the experiments, experimental datasets and implementation details are described. Then, ablation experiments are conducted to show the improvements on the Xception and dual-color spaces. Finally, comparative experiments with state-of-the-art work are performed to test the robustness against different types of post-processing operations.

A. *Experimental datasets*

In the experiment, the CelebA dataset [49] including 10,177 identities and 202,599 aligned face images is used as the natural face dataset. We use the function get_frontal_face_detector in the open source Dlib library [53] to find faces in a face image, crop the facial regions by removing the background, and then resize the cropped regions to a resolution of 128×128. These processed natural face images are used to train PGGAN [7], generating 202,599 face images with a size of 128×128. Here, the PGGAN is considered because it can generate high-quality face images and has been used in many existing studies [14, 18, 26, 28, 30]. Finally, 202,599 pairs of natural and generated images are obtained. In the experiments, 202,599 pairs were

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3116679, IEEE Transactions on Circuits and Systems for Video Technology

divided into training, validation, and testing sets at a ratio of 8:1:1. In addition, to test the robustness, several widely used post-processing operations are performed on the testing set. They are JPEG compression with different compression quality factors 90, 70 and 50, gamma correction with different gamma values 1.1, 1.4 and 1.7, Gaussian blurring with different kernel sizes 3×3, 5×5 and 7×7, median blurring with different kernel sizes 3×3, 5×5 and 7×7, Gaussian noising having different standard deviation values 3, 5 and 7, as well as resizing with the upscaling factor as 1%, 3%, 5% and then cropping the 128×128 central region. Some samples in the experimental datasets are provided in Fig. 9.

Fig. 9. Some samples in the experimental datasets. From left to right, the columns are the natural images in CelebA, the fake images generated by the PGGAN model, and the post-processed natural/fake images by JPEG compression, Gaussian noising, Gaussian blurring, gamma correction, median filtering and resizing.

B. Implementation details

All of the methods are implemented with Keras and a single 11GB GeForce GTX 1080 Ti, i7-6900K CPU, 64GB RAM. Parameters in convolution kernels are initialized by using a truncated normal distribution with zero mean and $\sigma = 0.01$. The size of the minibatch is set to 10 and the optimizer is the Adam method [54]. The initial learning rate is set to 1.0e^{-5} and the learning rate decay is 1.0e^{-6}. The source code is available at *https://github.com/imagecbj/A-robust-GAN-generated-face-detection-method-based-on-dual-color-spaces-and-an-improved-Xception*.

C. Ablation experiments

First, the effects of the MLFA and CBAM on the Xception are evaluated. Four methods, i.e., Xception, Xception+CBAM, Xception+MLFA, and the proposed method (Xception + CBAM + MLFA), are compared. Notice that here only a single RGB color space is considered. The experimental results are shown in Table IV. The results show that both the MLFA module and CBAM are helpful to improve the robustness against the post-processing especially when using the CBAM module and MLFA module together.

Then, the performance of dual-color spaces is tested. Therefore, the performance of the single RGB space, single

TABLE IV
ABLATION EXPERIMENTAL RESULTS FOR THE MLFA AND CBAM

Methods	JPEG compression with different quality factors			Gamma correction with different gamma values			Median filtering with different kernel sizes		
	90	70	50	1.1	1.4	1.7	3×3	5×5	7×7
Xception	96.9	90.1	81.5	98.9	98.2	95.4	94.8	76.6	65.6
Xception+CBAM	97.1	89.9	80.0	99.1	98.8	96.1	95.6	81.9	72.5
Xception+MLFA	97.5	91.2	85.3	98.9	98.4	95.7	93.2	82.4	73.9
Xception+CBAM+MLFA	**97.7**	**92.5**	**86.4**	**99.4**	**99.2**	**96.7**	**95.8**	**83.4**	**74.1**

Methods	Gaussian blurring with different kernel sizes			Gaussian noising with different std values			Resizing with different upscaling factors		
	3×3	5×5	7×7	3	5	7	1%	3%	5%
Xception	96.4	90.8	80.7	98.2	97.1	92.5	93.5	81.7	70.1
Xception+CBAM	95.2	91.3	81.4	98.9	97.8	92.5	93.9	80.5	68.7
Xception+MLFA	95.4	91.2	80.9	98.5	97.7	91.8	91.9	82.5	72.7
Xception+CBAM+MLFA	**95.6**	**91.8**	**82.3**	**99.2**	**98.0**	**93.3**	**94.2**	**82.8**	**69.4**

TABLE V
ABLATION EXPERIMENTAL RESULTS FOR DUAL-COLOR SPACES

Color spaces	JPEG compression with different quality factors			Gamma correction with different gamma values			Median filtering with different kernel sizes		
	90	70	50	1.1	1.4	1.7	3×3	5×5	7×7
RGB	97.7	92.5	86.4	98.7	97.7	96.7	95.8	83.4	74.1
YCbCr	98.5	93.2	85.5	99.3	99.1	97.1	97.2	84.5	73.1
RGB+CbCr	68.2	57.7	53.7	99.5	99.3	98.9	95.8	58.6	51.8
RGB+ YCbCr	**99.5**	**96.6**	**89.4**	**99.6**	**99.6**	**98.5**	**98.4**	**87.5**	**76.8**

Color spaces	Gaussian blurring with different kernel sizes			Gaussian noising with different std values			Resizing with different upscaling factors		
	3×3	5×5	7×7	3	5	7	1%	3%	5%
RGB	95.6	91.8	82.3	99.2	98.0	91.3	94.2	82.8	69.4
YCbCr	97.6	94.9	83.5	99.3	98.6	91.1	95.4	86.4	69.3
RGB+CbCr	97.9	77.4	56.1	99.9	99.7	98.6	96.8	85.3	61.6
RGB+ YCbCr	**98.3**	**96.3**	**85.2**	**99.7**	**99.5**	**94.3**	**97.2**	**88.5**	**71.3**

YCbCr space, and RGB + YCbCr dual spaces are compared in Table V. Notice that here RGB + YCbCr is also considered to evaluate the influence of the dropping of the luminance component. It can be observed from Table V that: (a) after the fusion of the two spaces, the results for various post-processing operations have been significantly improved; (b) the results further verify that the luminance component greatly improves the detection performance.

D. Comparative experiments with state-of-the-art work

To evaluate the effectiveness of the proposed method, the proposed method is compared with other methods, including Xception [35], Li's method [18], He's method [26], Liu's method [30], Fu's method [28] and Mi's method [32]. Among these compared methods, three methods (He's, Fu's, and Mi's) are also designed for robust detection. First, these methods are compared in the case of the original face images without post-processing. The detection accuracies of the different methods are shown in Table VI. It can be observed from the Table VI that all methods perform well with accuracies higher than 99% in free of post-processing. Then, seven methods are compared for their robustness against different post-processing operations. Table VII presents the comparative results. It can be seen from this table that: (a) the proposed method outperforms other methods overall due to the consideration of dual-color spaces and the use of MLFA, which considers both the deep semantic features and shallow features; (b) the Li's method and He's method perform well in detecting images post-processed by Gaussian noising. This finding is consistent with the conclusion from Fig. 3 and Table I that the chrominance component has better resistance to Gaussian noising than the luminance component. The Li's method [18] and He's method [25] exploit only the chrominance information.

TABLE VI
DETECTION ACCURACIES (%) OF DIFFERENT METHODS FOR ORIGINAL IMAGES

Methods	Accuracy
Li's [18]	99.7
He's [26]	99.8
Xception [35]	99.8
Liu's [30]	99.2
Fu's [28]	99.5
Mi's [32]	99.4
Proposed	**99.9**

TABLE VII
DETECTION ACCURACIES (%) OF DIFFERENT METHODS AGAINST DIFFERENT POST-PROCESSING OPERATIONS

Methods	JPEG compression with different quality factors			Gamma correction with different gamma values			Median filtering with different kernel sizes		
	90	70	50	1.1	1.4	1.7	3×3	5×5	7×7
Li's [18]	88.5	54.3	51.2	99.7	99.3	97.2	88.2	68.5	60.8
He's [26]	89.6	57.2	52.1	99.8	98.7	96.1	89.3	70.4	60.1
Xception [35]	96.9	90.1	81.5	98.9	98.2	95.4	94.8	76.6	65.6
Liu's [30]	89.8	70.3	65.8	95.5	93.4	92.2	78.5	68.3	64.2
Fu's [28]	89.2	68.3	61.8	99.5	99.3	98.2	71.2	62.1	57.6
Mi's [32]	97.8	90.2	83.2	99.9	99.5	98.3	80.1	73.2	63.2
Proposed	**99.5**	**96.6**	**89.4**	**99.6**	**99.6**	**98.5**	**98.4**	**87.5**	**76.8**

Methods	Gaussian blurring with different kernel sizes			Gaussian noising with different std values			Resizing with different upscaling factors		
	3×3	5×5	7×7	3	5	7	1%	3%	5%
Li's [18]	94.1	84.2	76.3	99.5	98.1	94.3	90.2	78.2	69.8
He's [26]	94.3	89.5	78.1	99.9	99.8	96.7	90.5	60.2	50
Xception [35]	96.4	90.8	80.7	98.2	97.1	92.5	79.3	52.9	50
Liu's [30]	94.5	89.8	79.5	97.2	94.6	88.7	91.2	70.5	60.4
Fu's [28]	80.5	76.1	69.4	97.1	85.3	62.8	89.7	88.5	87.3
Mi's [32]	95.2	90.1	80.9	99.1	80.1	68.7	89.3	65.2	56.7
Proposed	**98.3**	**96.3**	**85.2**	**99.7**	**99.5**	**94.3**	**97.2**	**88.5**	71.3

E. Visualization and analysis of results

To gain a deeper understanding of why the network is effective, we further exploited classification activation maps [46] to reveal the areas that are used as evidence for generated face detection by the proposed method. The classification activation maps are shown in Fig. 10. It can be easily seen that: (a) the main areas recognized by the proposed method are the skin and hair areas; (b) the proposed method can still pay attention to these main areas though the testing images are post-processed. This result shows that the proposed method has strong robustness.

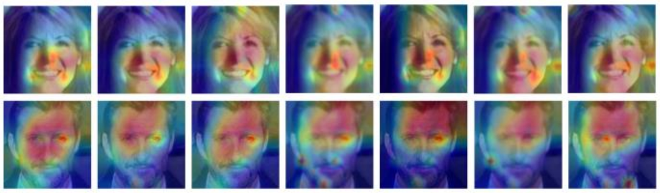


Fig. 10. Class activation maps from the trained proposed model in the RGB color space. From left to right, the column shows the GAN-generated images, and the post-processed images by JPEG compression, Gaussian noising, Gaussian blurring, gamma correction, median filtering and resizing.

V. CONCLUSION

In this paper, an improved GAN-generated face detection method is proposed. It considers both the luminance component and chrominance components of dual-color spaces (RGB and YCbCr). In addition, it is based on an improved Xception model with the CBAM and MLFA. Experimental results show that our method achieves better performance than some existing methods in robustness against different types of post-processing operations. Certainly, for GAN-generated face detection, the generalization ability under a cross-dataset test is also very important. Therefore, a robust method with strong generalization ability is a goal for our future work.

REFERENCES

[1] Goodfellow I, Pouget-Abadie J, and Mirza M, et al. “Generative adversarial nets,” In: Proceedings of the 28th Conference on Neural Information Processing Systems (NeurIPS 2014), pp. 2672-2680, 2014.

[2] Radford A, Metz L, and Chintala S, et al. “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.

[3] Arjovsky M, Chintala S, and Bottou L, “Wasserstein generative adversarial networks,” In: Proceedings of the 34th International Conference on machine learning. PMLR, pp. 214-223 2017.

[4] Gulrajani I, Ahmed F, and Arjovsky M, et al. “Improved training of wasserstein GANs,” Advances in Neural Information Processing Systems (NIPS2017), pp. 5767-5777, 2017.

[5] Junbo Z, Michael M, and LeCun Y, “Energy-based generative adversarial networks,” In: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), 2017.

[6] Berthelot D, Schumm T, and Metz L, “Began: Boundary equilibrium generative adversarial networks,” arXiv preprint arXiv:1703.10717, 2017.

[7] Karras T, Aila T, and Laine S, et al. “Progressive growing of GANs for improved quality, stability, and variation,” In: Proceedings of the 2018 International Conference on Learning Representations (ICLR2018), 2018.

[8] Karras T, Laine S, and Aila T, “A style-based generator architecture for generative adversarial networks,” In: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2019), pp. 4401-4410, 2019.

[9] Liao K, Lin C, and Zhao Y, et al, “DR-GAN: Automatic radial distortion rectification using conditional GAN in real-time,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 3, pp. 725-733, 2019.

[10] Zhu J Y, Park T, and Isola P, et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks,” In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV2017), pp. 2223-2232, 2017.

[11] Choi Y, Choi M, and Kim M, et al. “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” In: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2018), pp. 8789-8797, 2018.

[12] Yuan M, and Peng Y, “Bridge-GAN: interpretable representation learning for text-to-image synthesis”. IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 11, pp. 4258-4268, 2019.

[13] Xu S, Liu D, and Xiong Z, “E2I: generative inpainting from edge to image”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 4, pp. 1308-1322, 2020.

[14] Yang X, Li Y, and Qi H, et al. “Exposing GAN-synthesized faces using landmark locations,” In: Proceedings of the 2019 ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec2019), pp. 113-118, 2019.

[15] Matern F, Riess C, and Stamminger M, “Exploiting visual artifacts to expose deepfakes and face manipulations,” In: Proceedings of the 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW2019), pp. 83-92, 2019.

[16] Liao X, Li K, Zhu X, et al. “Robust detection of image operator chain with two-stream convolutional neural network,” IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 5, pp. 955-968, 2020.

[17] McCloskey S, and Albright M, “Detecting GAN-generated imagery using color cues,” arXiv preprint arXiv:1812.08247, 2018

[18] Li H, Li B, and Tan S, et al. “Identification of deep network generated images using disparities in color components,” Signal Processing, vol. 174, pp. 107616-1-12, 2020.

[19] Nataraj L, Mohammed T M, and Manjunath B S, et al. “Detecting GAN generated fake images using co-occurrence matrices,” Electroc Imaging, vol. 2019, no. 5, pp. 532-1-532-7, 2019.

[20] Liu B, and Pun C M. “Locating splicing forgery by fully convolutional netsworks and conditional random field,” Signal Processing: Image Communication, vol. 66, pp. 103-112, 2018.

[21] Do N T, Na I S, and Kim S H, “Forensics face detection from gans using convolutional neural network,” In: Proceeding of 2018 International Symposium on Information Technology Convergence (ISITC2018), 2018.

[22] Kong C Q, Chen B L, and Yang W H, et al. “Appearance matters, so does audio: revealing the hidden face via cross-modality transfer”, IEEE Transactions on Circuits and Systems for Video Technology, 2021. DOI: 10.1109/TCSVT.2021.3057457.

[23] Chen B, Tan W, and Wang Y, et al. “Distinguishing Between Natural and GAN-Generated Face Images by Combining Global and Local Features”, Chinese Journal of Electronics, 2021. DOI: 10.1049/cje.2020.00.372.

[24] Mo H, Chen B, and Luo W, “Fake faces identification via convolutional neural network,” In: Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec2018), pp. 43-47, 2018.

[25] Yang C Z, Ma J, and Wang S L, et al. “Preventing deepFake attacks on speaker authentication by dynamic lip movement analysis”, IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1841-1854, 2021.

[26] He P, Li H, and Wang H, “Detection of fake images via the ensemble of deep representations from multi color spaces,” In: Proceedings of the 2019 IEEE International Conference on Image Processing, pp. 2299-2303, 2019.

[27] Chen B, Ju X, Xiao B, et al. “Locally GAN-generated face detection based on an improved Xception,” Information Sciences, vol. 572, pp. 16-28, 2021.

[28] Fu Y, Sun T, and Jiang X, et al. “Robust GAN-face detection based on dual-channel CNN network,” In: Proceedings of the 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). pp. 1-5, 2019.

[29] Jia S, Li X, and Hu C, et al. “3D face anti-spoofing with factorized bilinear coding,” IEEE Transactions on Circuits and Systems for Video Technology, 2020.

[30] Liu Z, Qi X, and Torr P H S, “Global texture enhancement for fake face detection in the wild,” In: Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2020), pp. 8060-8069, 2020.

[31] He K, Zhang X, and Ren S, et al., “Deep residual learning for image recognition,” In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), pp. 770-778.

[32] Mi Z, Jiang X, and T, et al. “GAN-generated image detection with self-attention mechanism against GAN generator defect”, IEEE Journal of Selected Topics in Signal Processing, vol. 14, no.5, pp. 969-981, 2020.

[33] Fernando T, Fookes C, and Denman S, et al. “Detection of fake and fraudulent faces via neural memory networks”, IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1973-1988, 2021.

[34] Hu J, Liao Xin, Wang W, et al. “Detecting compressed Deepfake videos in social networks using frame-temporality two-stream convolutional network”, IEEE Transactions on Circuits and Systems for Video Technology, 2021. DOI:10.1109/TCSVT.2021.3074259.

[35] Chollet F, “Xception: Deep learning with depthwise separable convolutions,” In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017), pp. 1251-1258, 2017.

[36] Rössler A, Cozzolino D, and Verdoliva L, et al. “Faceforensics++: Learning to detect manipulated facial images,” arXiv preprint arXiv:1901.08971, 2019.

[37] Marra F, Gragnaniello D, and Cozzolino D, et al. “Detection of gan-generated fake images over social networks,” In: Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR2018), pp. 384-389, 2018.

[38] Woo S, Park J, and Lee J Y, et al, “Cbam: Convolutional block attention module,” In: Proceedings of the 2018 European Conference on Computer Vision (ECCV2018), pp. 3-19, 2018.

[39] Chen B, Tan W, and Coatrieux G, et al. “A serial image copy-move forgery localization scheme with source/target distinguishment”, IEEE Transactions on Multimedia, 2020. DOI: 10.1109/TMM.2020.3026868.

[40] Parkin A, and Grinchuk O, “Recognizing multi-modal face spoofing

with face recognition networks," In: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW2019), pp. 1617-1623, 2019.

[41] Breiman L, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[42] Szegedy C, Vanhoucke V, and Ioffe S, et al. "Rethinking the inception architecture for computer vision," In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), pp. 2818-2826, 2016.

[43] Lin M, Chen Q, and Yan S, "Network in network," arXiv preprint arXiv:1312.4400, 2013.

[44] Tang J, Li Z, and Lai H, et al. "Personalized age progression with bi-level aging dictionary learning." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 905-917, 2017.

[45] Shu X, Tang J, and Lai H, et al. "Personalized age progression with aging dictionary," In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 3970-3978, 2015.

[46] Tian Y L, Takeo K, and Jeffrey F C. "Facial expression analysis," In: Handbook of face recognition, Springer, New York, pp. 247-275, 2005.

[47] Bartlett M S, Littlewort G, and Fasel I, et al. "Real time face detection and facial expression recognition: development and applications to human computer interaction," In: Proceedings of the 2003 Conference on Computer Vision and Pattern Recognition Workshop. IEEE, (CVPRW) vol. 5, pp. 53-53, 2003.

[48] Hsu R L, Abdel-Mottaleb M, and Jain A K. "Face detection in color images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, 2002.

[49] Liu Z, Luo P, and Wang X, "Deep learning face attributes in the wild," In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV2015), pp. 3730-3738, 2015.

[50] Ioffe S, and Szegedy C, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[51] Zhou B, Khosla A, and Lapedriza A, et al. "Learning deep features for discriminative localization," In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), pp. 2921-2929, 2016.

[52] Ronneberger O, Fischer P, and Brox T, "U-Net: convolutional networks for biomedical image segmentation," In: Proceedings of the 2015 International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI), pp. 234-241, 2015.

[53] D. King, "Dlib c++ library", Access on: http://dlib.net/dlib/image_processing/frontal_face_detector_abstract.h.html#get_frontal_face_detector, 2018.

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

Beijing Chen received the Ph.D. degree in Computer Science in 2011 from Southeast University, Nanjing, China. Now he is a Professor in the School of Computer, Nanjing University of Information Science & Technology, China. His research interests include color image processing, image forensics, image watermarking, and pattern recognition. He serves as an Editorial Board Member of the Journal of Mathematical Imaging and Vision.

Xin Liu received the B.S. degree in Applied Chemistry in 2017 from Nanjing University of Information Science & Technology, Nanjing, China. He is currently pursuing the M.S. degree in Computer Science from Nanjing University of Information Science & Technology, Nanjing, China. His research interests include image processing, and image forensics.

Yuhui Zheng received the Ph.D. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2009. From 2014 to 2015, he was a Visiting Scholar at the Digital Media Laboratory, School of Electronic and Electrical Engineering, Sungkyunkwan University, South Korea. He is currently a Professor at the School of Computer, Nanjing University of Information Science & Technology. His research interests cover image processing, pattern recognition, and remote sensing information system.

Guoying Zhao received the Ph.D. degree in Computer Science from the Chinese Academy of Sciences, Beijing, China, in 2005. She is currently a full Professor with Center for Machine Vision and Signal Analysis, University of Oulu, Finland. She has authored or coauthored more than 240 articles in journals and conferences. Her articles have currently over 15,600 citations in Google scholar (H-index 57). She is a Fellow of the IAPR. She has served as the area chairs for several conferences. She is an Associate Editor of Pattern Recognition, IEEE Transactions on Circuits and Systems for Video Technology, and Image and Vision Computing journal.

Yun-Qing Shi (M'88–SM'92–F'05) the Ph.D. degree from the University of Pittsburgh, USA. He has been with the New Jersey Institute of Technology, USA, since 1987. He has authored/co-authored more than 300 papers, one book, five book chapters, and an Editor of ten books, three special issues, and 13 proceedings, and holds 30 U.S. patents. His research interests include data hiding, forensics and information assurance, visual signal processing, and communications. He has served as an Associate Editor of the IEEE Transactions on Signal Processing and the IEEE Transactions on Circuits and Systems (II). He serves as an Associate Editor of the IEEE Transactions on Information Forensics and Security, and an Editorial Board Member of a few journals. He is a member of a few IEEE technical committees.