IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 16, 2019, accepted August 30, 2019, date of publication September 6, 2019, date of current version September 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939812

No One Can Escape: A General Approach to Detect Tampered and Generated Image

KEJUN ZHANG[1,2], YU LIANG[1,2], JIANYI ZHANG[1], ZHIQIANG WANG[1,3], AND XINXIN LI[1,2]

[1]Beijing Electronic Science and Technology Institute, Beijing 10070, China
[2]Xidian University, Xi'an 710071, China
[3]The State Information Center, Beijing 100045, China

Corresponding authors: Jianyi Zhang (zjy@besti.edu.cn) and Zhiqiang Wang (wangzq@besti.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004100, in part by the Key Lab of Information Network Security, in part by the Ministry of Public Security under Grant C18612, and in part by the Fundamental Research Funds for the Central Universities under Grant 328201804 and Grant 328201908.

ABSTRACT Fake or tampered images pose a real problem in today's life. It is easy to unknowingly be drawn to an interesting image that is false. Recently, with the emergence of generative adversarial networks (GANs), it becomes much more easy to generate high-quality fake images in a very realistic way. However, the current digital image forensics algorithms mainly focus on the detection of traditional tampered images or need prior knowledge of the network structure of GANs. Hence, verifying the authenticity of an image is very challenging. In this paper, we propose a general method for simultaneously detecting tampered images, and GANs generated images. First, we use the Scharr operator to extract the edge information of the image. Then, we converted the edge image information matrix into the gray level co-occurrence matrix (GLCM) to scale the image without loss of image information. Finally, GLCM was fed into the deep neural network designed based on depthwise separable convolution for training. Compared with other methods, our model achieves a higher macro average of F1 score of 0.9865. Meanwhile, our method has better performance in detecting tampered images and has strong generalization ability for many GANs models.

INDEX TERMS Digital image forensics, generative adversarial networks, deep learning, convolutional neural networks.

I. INTRODUCTION

Over the past several years, digital image editing software has developed rapidly, such as photoshop. Convincing fake images can be created by an ordinary user with a little knowledge of how to use the image editing software. With image editing, people's lives become rich. However, malicious tampering will have a severe negative impact on individuals and even the government. At present, the digital image is facing a severe crisis of trust, which makes it very important to judge the authenticity or credibility of a digital image.

Generally, digital image forensics is divided into active forensics and passive forensics. In active forensics, special marks are needed, such as digital watermarks. And tampering can be detected by damaged image watermark. Active forensics requires image preprocessing before image release. Instead, passive forensics directly detects a digital image rely on some algorithms. In most passive forensics algorithms, image features are extracted after image preprocessing, and then SVM is used to classify [5]–[7], [27]. With the successful application of convolutional neural networks (CNN) in the field of image classification, CNN has been used to detect tampered images in some works [8], [15]. Among tampering techniques, splicing, removal, and copy-move are the most common editing operation. In this paper, we mainly discuss the detection of the above three tampering operations.

Moreover, with the development of deep learning, the field of image generation has achieved great success. In 2014, Goodfellow first introduced the generative adversarial networks (GANs) [1]. It consists of two parts, which are the generator and the discriminator. During the image generation process, the generator aims to produce fake images while the discriminator aims to distinguish between the real images and fake images. Via an adversarial process, the generator will be able to generate images that look like the real from scratch in the end [2]–[4], [16], [17], [26], [28]. Obviously, the fake

The associate editor coordinating the review of this manuscript and approving it for publication was Liehuang Zhu.

images created by this new technology has an enormous potential damage. Therefore, it is necessary to detect the images generated by the GANs.

In order to solve these problems, some detection algorithms have been proposed. However, these algorithms cannot detect tampered images and GANs generated images at the same time. Moreover, the detection algorithms for GANs generated images just have weak generalization ability. In this paper, we propose a method that can detect both tampered images and GANs generated images with a high macro average of F1 score of 0.9865. In addition, our model achieved good performance in just detecting tampered images and has strong generalization performance to a variety of GANs models in detection (the model is trained on the datasets generated by BigGANs and can detect the images generated by a variety of other GANs models). Our code is available at https://github.com/yuleung/image_forensics, and the dataset is also described in detail here.

The main contributions of this paper are summarized as follows:

- To our knowledge, this is the first time that a single model is used to detect tampered images simultaneously and GANs generated images.
- We find a general method which introduces a strong generalization ability into the detection of images generated by various GANs.
- We propose a structure with depthwise separable convolutions as the main component which with a smaller number of parameters compared with the one consists of traditional convolution.
- Compared with previous work, our method has good performance in detecting tampered images.

This paper is organized as follows. We discuss the related work of the detection of image tampering and GANs generated images in Section II. We explain our approaches in Section III, and describe our evaluation results in Section IV. Finally, Section V offers our conclusions.

II. RELATED WORKS

In this section, we will briefly review the previous detection work on tampered images and the achievements of GANs in the field of image generation.

A. DIGITAL IMAGE TAMPERING DETECTION

Image tampering detection has always been a research hotspot, and researchers have done a lot of work. Wang *et al.* [5] proposed that splicing traces were easier to be detected in YCrCb color space than in RGB color space. Muhammad *et al.* [7] used a steerable pyramid transform (SPT) and local binary pattern (LBP) to extracted features. Agarwal and Chand [36] proposed a image tampering detection method in accordance with wavelet transform and texture descriptor. Wang *et al.* [6] proposed to model the edge image as a finite-state Markov chain and extract low dimensional feature vector from its stationary distribution for tampering detection. Bayar *et al.* [8] proposed a new type of CNN layer, called a constrained convolutional layer, which can suppress the content of a image and adaptively learn manipulation detection features. Rao *et al.* [15] initialized the first layer convolution kernel in their proposed CNN model with the basic high-pass filter set in a spatial rich model (SRM) [9]. Then they extracted image features using the pre-trained CNN model which trained in the training set. Zhou *et al.* [10] proposed a two-stream Faster R-CNN network for the detection task. Their model consists of two sub-networks, SRM noise stream sub-network and RGB stream sub-network, which were trained to extract different features respectively. Finally, the tampered area was detected by analyzing the two features. Moreover, Wang *et al.* [29] use a Dilated Residual Network variant(DRN-C-26) [30] to detect image warping applied to human faces.

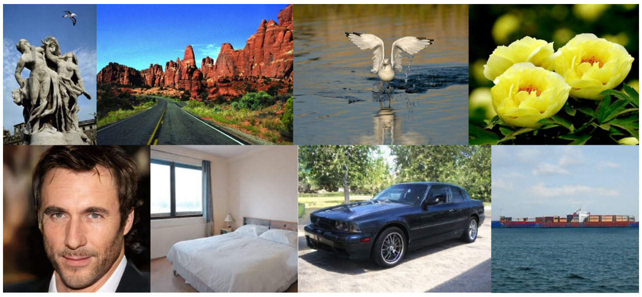


FIGURE 1. The first line is the tampered image, and the second line is the GANs generated image. For tampered images, the attacker will erase the tampering traces in the RGB image as much as possible for the attack.

B. GANs IN IMAGE GENERATION

Since GANs was first proposed in 2014, various variations of GANs were proposed. Unlike the original GANs which can only generate gray images of the handwritten digitals, the current GANs can generate multiple categories of realistic color images (as shown in the second line of Fig 1).

PGGAN [16] can generate 1024 × 1024 high-definition image. The key ideal of PGGANs is to grow both the generator and discriminator progressively: starting from a low resolution. SNGAN [17] used spectral normalization to stabilize the training process of the discriminator and got the generated images with better or equal quality. BigGANs [2] has been trained as the largest model of GANs and made the input of the generator amenable to a truncated distribution during training. The diversity and fidelity of the generated images have a lot of progress. StyleGAN [3] designed a new generator architecture used the ideal of style transformation, and it works well when the generated image contains only a single object. StackGAN [4] designed a two-stage generation process that generates corresponding images based on the description of the sentence. pix2pixhd [22] proposed a method for synthesizing high-resolution photo-realistic images from semantic label maps rely on the structure of multiple generators and multiple discriminators. CT-GAN [23] used GANs to add or remove evidence of

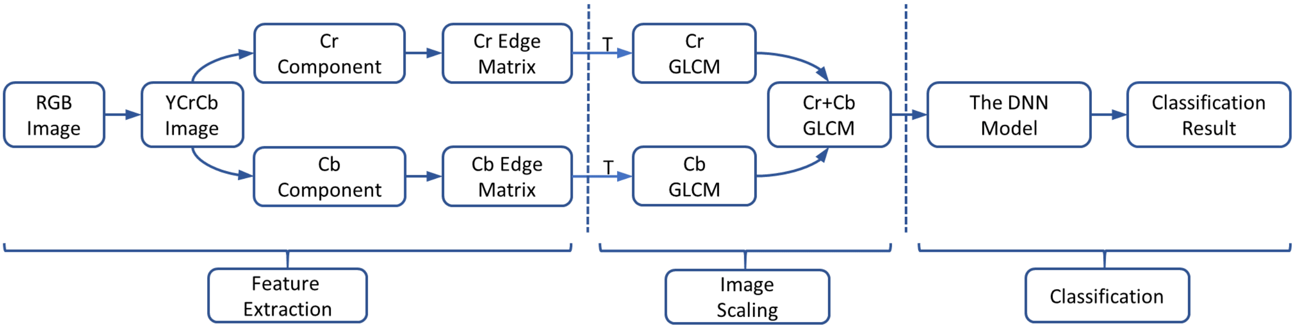


FIGURE 2. The overall framework of our proposed. In here, T is the truncation operation for edge information matrix.

medical conditions from volumetric (3D) medical scans. Moreover, researchers have also done some work on the forensics of GANs generated images [11]–[13], [31], [32], [35].

III. APPROACH

The model we to detect both tampered images and GAN generated images as Fig 2. It is mainly divided into three parts: feature extraction, image scaling, and classification. At first, we transform the image from RGB color space to the YCrCb color space and then use the Scharr operator to extract the image edge information of the Cr and Cb component. The gray level co-occurrence matrix (GLCM) [24] be followed, which was used to unify edge matrix of different sizes to the same size. Finally, the GLCM will be feed into a deep neural network based on depthwise separable convolution that we designed to get the classification results.

A. FEATURE EXTRACTION

YCrCb is a group in colors space just like RGB, where Y is the luminance component of the image, Cr and Cb are the chroma components. In general, the images that are stored on a hard drive and seen on the Internet are RGB images. For tampered images, the attacker will erase the tampering traces in the RGB image as much as possible for the attack (as shown in the first line of Fig 1). Therefore, it is challenging for people to discriminate whether an RGB image is a tampered image just using their eyes. However, from the first row of Fig 4, it can be seen that the tampering traces is more pronounced in the chromatic components (the spliced bird has a smoother edge than the other parts). Moreover, as shown in Fig 3, we found that there is also edge information between the foreground and background of the image generated by GANs or within the object. We convert the RGB image into the YCrCb color space and extract the Cr and Cb chrominance components.

And as shown in the Y component image in Fig 4, the Y component contains the main content details of the image, while the content details of the image will cover the edge information generated by tampering. The CrCb component contains less detailed information about the content of the image. This enables our model to focus more on the analysis of image edge information. To weaken the influence of the image content, and to extracted the key information of tampering. When the conversion is completed, we use the edge detection operator to obtain image edge information with the 3 × 3 kernel.

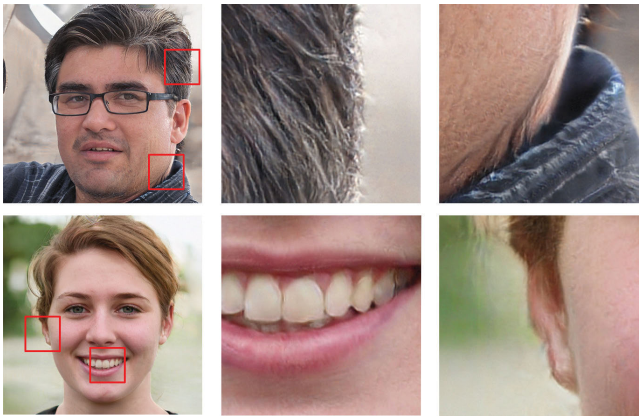


FIGURE 3. Fake face images generated by StyleGAN. The edges of hair, ears, collars, and teeth have distinct information.

The Sobel operator is one of the most important operators in image edge detection, and the Scharr operator is a variant of the Sobel operator, and it is more sensitive to edges than the Sobel operator. Scharr operator has two operators just like the Sobel operator, one for detecting vertical edges and another one for detecting horizontal edges. The specific form of its operator is as shown in Fig 5.

The process of extracting image edge information using the Scharr operator are as follows:

$$\begin{aligned}P_x(i,j) = |&v_{11}P(i+1,j+1) + v_{13}P(i+1,j-1)\\&+v_{21}P(i,j+1) + v_{23}P(i,j-1)\\&+v_{31}P(i-1,j+1) + v_{33}P(i-1,j-1)| \quad (1)\end{aligned}$$

$$\begin{aligned}P_y(i,j) = |&h_{11}P(i+1,j+1) + h_{12}P(i+1,j)\\&+h_{13}P(i+1,j-1) + h_{31}P(i-1,j+1)\\&+h_{32}P(i-1,j) + h_{33}P(i-1,j-1)| \quad (2)\end{aligned}$$

$$P'(i,j) = 0.5 \times P_x(i,j) + 0.5 \times P_y(i,j) \quad (3)$$

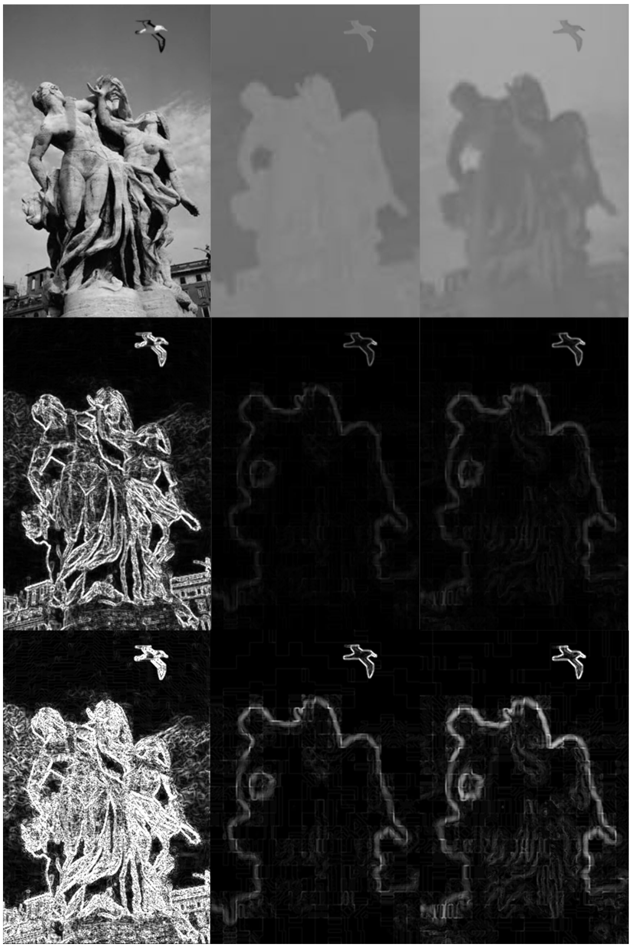


FIGURE 4. An example of splicing (the bird in the upper right corner of the image is cut from other photos). The first line from left to right is the Y, Cr and Cb component of the image in the YCrCb color space, and the second and third lines are the result of edge extraction using the Sobel operator and Scharr operator respectively corresponding to the first line.

$$\begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \qquad \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix}$$

FIGURE 5. The left is the operator in Scharr to do vertical edges detection, and the right is the operator in Scharr to do horizontal edges detection.

where $|\cdot|$ is the operation of taking absolute value. v_{ij} is the value of vertical edge detection operator located at (i, j). h_{ij} is the value of horizontal edge detection operator located at (i, j). $P(i, j)$ is the gray value of image located at (i, j). $P_x(i, j)$ and $P_y(i, j)$ are the edge information of the image filtered by Scharr operator in the vertical direction and horizontal direction respectively. $P'(i, j)$ combines horizontal edge information and vertical edge information.

The edge information results are shown in Fig 4. It can be seen that the tampering operation area has smoother and brighter edge information than the real area in the edge information matrix of the Cr component and the Cb component.

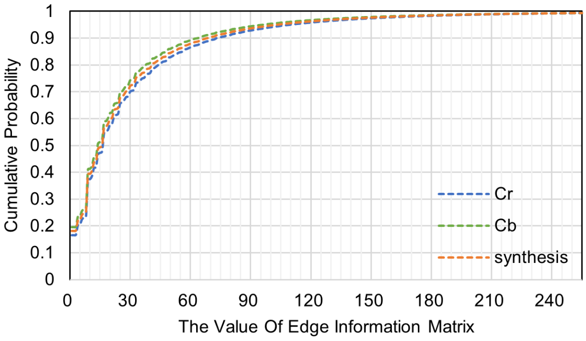


FIGURE 6. The cumulative probability of the values in the edge information matrix calculated from 12,369 images in the training set.

B. IMAGE SCALING

The texture of an image is formed by the repeated occurrence of the grayscale which distribution in a specific spatial location, and gray level co-occurrence matrix (GLCM) extracts the texture of the image by extracting characteristics of the gray spatial correlation. There are two reasons why we convert the image edge information matrix to the GLCM:

1) The pixels of an image are undetermined in the practical application of image forensics. Meanwhile, a CNN-based classifier usually requires the input data to have a specific size, and the detailed information of images is particularly important for detecting tampering operations. The size of the GLCM depends on the maximum gray value in the image. Therefore, GLCM can resize the edge information matrix to a uniform size without losing image details.
2) According to the characteristics of GLCM, the smooth edge corresponding to the tampered area and the rough edge corresponding to the untampered area in the edge image have different representations in GLCM.

The maximum value of edge information matrix obtained after Scharr operator filtering can up to 4080, but we find that the values of the edge information matrix of the image calculated by Scharr operator are mostly in a relatively small range, as shown in Fig 6. Therefore, we can choose a suitable threshold to truncate the large value of the edge information matrix with little impact on the performance of our model, which can reduce the size of the converted GLCM to minimize the complexity of our model. Truncation operation is according to the following rule:

$$P''(i,j) = \begin{cases} P'(i,j), & \text{if } P'(i,j) < T \\ T-1, & \text{if } P'(i,j) \geq T \end{cases} \tag{4}$$

where T is the threshold value, $P'(i, j)$ is the value of original image edge information matrix located at (i, j), and $P''(i, j)$ is the value of truncated image edge information matrix located at (i, j).

Next, we calculate GLCM in four directions (0°, 45°, 90°, 135°) with an offset distance of 1 from the truncated edge information matrix of Cr component and Cb component.

Algorithm 1 Calculate GLCM in Four Directions (0°, 45°, 90°, 135°) With an Offset Distance of d From a Truncated Edge Information Matrix

```
Input: $Edge[M][N]$—The Cr component or Cb component edge information matrix truncated by T with size $M \times N$
Output: The GLCM in four directions (0°, 45°, 90°, 135°)
Initialize four matrices of size $T \times T$ to save $GLCM0^\circ$, $GLCM45^\circ$, $GLCM90^\circ$, $GLCM135^\circ$;
for $i = 0$ to $M - 1$ do
    for $j = 0$ to $N - 1$ do
        if $j < N - d$ then
            $GLCM0^\circ[Edge[i][j], Edge[i], [j + d]] \Leftarrow GLCM0^\circ[Edge[i][j], Edge[i], [j + d]] + 1$
        end if
        if $i < M - d$ and $j < N - d$ then
            $GLCM45^\circ[Edge[i][j], Edge[i + d], [j + d]] \Leftarrow GLCM45^\circ[Edge[i][j], Edge[i + d], [j + d]] + 1$
        end if
        if $i < M - d$ then
            $GLCM90^\circ[Edge[i][j], Edge[i + d], [j]] \Leftarrow GLCM90^\circ[Edge[i][j], Edge[i + d], [j]] + 1$
        end if
        if $i < M - d$ and $j > d - 1$ then
            $GLCM135^\circ[Edge[i][j], Edge[i + d], [j - d]] \Leftarrow GLCM135^\circ[Edge[i][j], Edge[i + d], [j - d]] + 1$
        end if
    end for
end for
Concatenate G0°, G45°, G90°, G135° together to get a matrix($G_Four$) of size $T \times T \times 4$
return $G_Four$
```

Finally, concatenate them together to get a matrix of size $T \times T \times 8$, which as the input of the deep neural network.

The specific algorithm of converting to GLCM is shown in Algorithm 1.

C. CLASSIFICATION BASED ON DEPTHWISE SEPARABLE CONVOLUTION

CNN has been widely used in computer vision (CV) tasks, and depthwise separable convolution has fewer parameters than traditional convolution. Moreover, some proposed networks based on depthwise separable convolution can also achieve excellent performance. Howard *et al.* [20] designed a lightweight network called MobileNet based on depthwise separable convolution. Chollet and FranÃ?gois [19] designed a network called Xception based on depthwise separable convolution, and excellent performance was achieved on several base datasets. In here, we designed a deep neural network based on depthwise separable convolution specifically for detecting tampered images and GANs generated images. Most convolution operations are depthwise separable convolution in our proposed network architecture. Our experiments show that compared with the network which all convolution operations use traditional convolution, and the architecture we designed to have slightly higher performance with fewer parameters. The specific network architecture is shown in Fig 7.

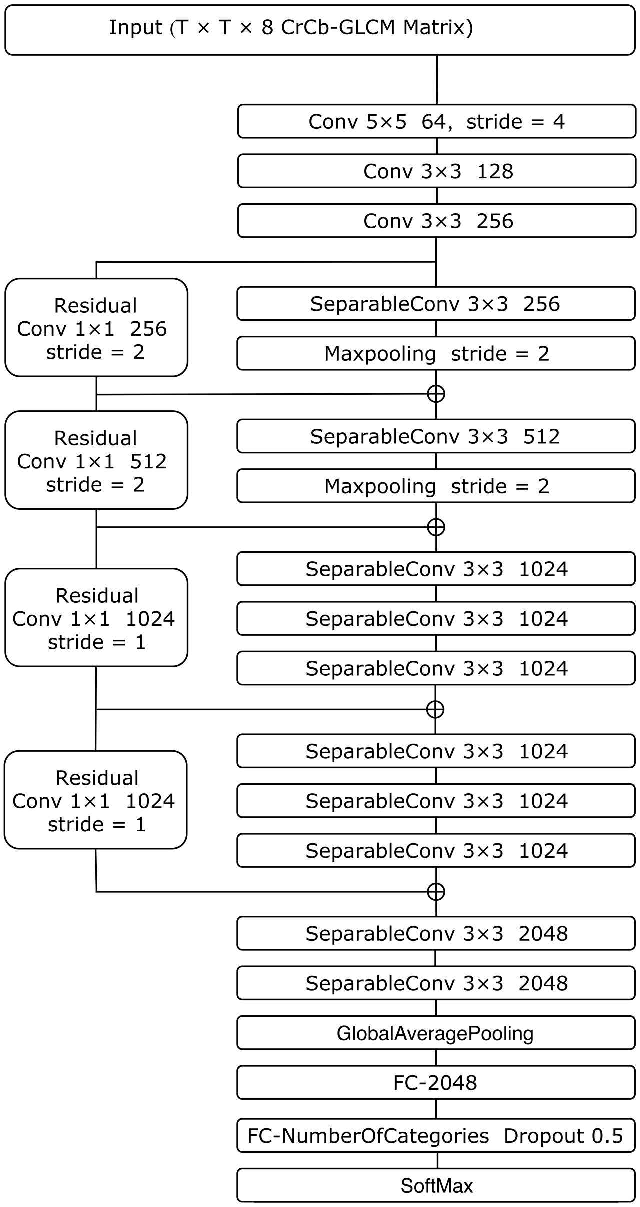


FIGURE 7. The network architecture we proposed. Default setting stride is equal to 1 and padding is the SAME padding in convolution operation, and Relu activate function be used. Conv is traditional convolution, SeparableConv is depthwise separable convolution, and Residual is residual block. Note that all Convolution and depthwise separable convolution layers are followed by batch normalization [18] (not included in the diagram).

Moreover, we made the following considerations to design the network architecture:

1) Since GLCM extracted from the edge information matrix, this will cause that the most of elements in GLCM to be zero (In the training set, we calculated that approximately 83.76% of the items are zero). To reduce the complexity of the neural network, we design the first layer of the network as a 5×5 convolution with a stride of 4.

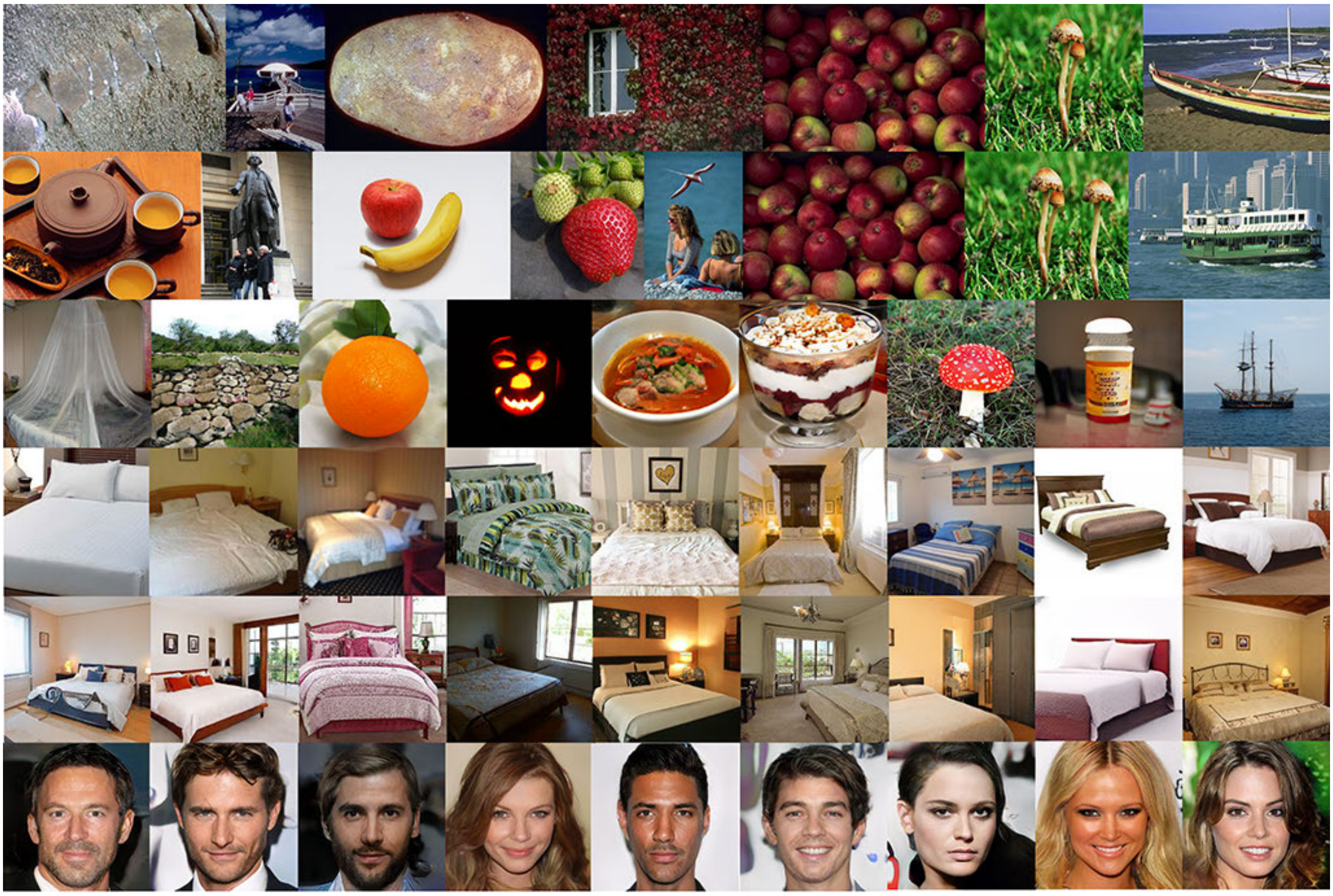


FIGURE 8. Some images in the datasets. From top to bottom, the line is real images in CASIA 2.0, tampered images in CASIA 2.0, images generated by BigGANs, images in LSUN (bedroom), images generated by StyleGAN, images generated by PGGAN.

2) After the edge information matrix is converted to GLCM, the features of edges will be scattered throughout the GLCM. Therefore, In the deeper part of the network architecture, after obtaining a small feature map, we repeated convolution operations several times to fully extract image features.

IV. EVALUATION

We test the performance of detecting tampered images and GANs generated images at the same time in various situations, and we calculated the accuracy of our model in detecting tampered images and compared it to other models designed specifically for tampered images. Moreover, we test the generalization ability of our model in detecting tampered images and GANs generated images separately.

A. DATASET

All of the datasets we used in our experiments are listed below, and part of these images are shown in Fig 8.

CACIA 2.0: CACIA 2.0 dataset [21] contains 7491 authentic and 5123 tampered color images. The images in this database are with different size, various from 240 × 160 to 900 × 600 pixels, and the images have different formats: BMP, TIFF and JPEG images which with different Q factors. In this database, images have different scenes, and the tampered area has different post-processing. Specifically, some operations include resizing, deforming, and rotating taken to the splicing region before pasting to a final generation. Moreover, blurring operation are done along the boundary area of the tampered region or other than the boundary area of the tampered region. This database can represent most of the splicing tampering operations in real life.

GPIR dataset: GPIR dataset [33] contains 80 images with realistic copy-move forgeries. All these images have size 768 × 1024 pixels, while the forgeries have arbitrary shapes aimed at obtaining visually satisfactory results.

COVERAGE dataset: COVERAGE [34] contains 100 original-forged image pairs where each original contains multiple similar-but-genuine objects. It makes the discrimination of forged from genuine objects highly challenging. And six types of tampering were employed for forged image generation.

BigGANs dataset: BigGANs [2] was proposed in 2018. It is currently the best model in terms of integrated image diversity and fidelity. We used the BigGANs generator pretrained on the ImageNet Dataset with truncation threshold 0.4 to generated 1000 categories of images (16 images for each category, including 8 images with the resolution of

256 × 256 and 8 images with the resolution of 512 × 521, for a total of 16,000 images). The pre-trained model downloaded from TensorFlow-Hub.

LSUN Bedroom dataset (256 × 256)**:** LSUN dataset[14] contains around one million labeled images for each of 10 scene categories and 20 object categories. In our experiment, we selected images with a resolution of 256 × 256 in the scene categories of the bedroom from this database.

PGGAN dataset: We downloaded 10,000 images of the bedroom generated by PGGANs [16] which trained on LSUN dataset, and the resolution of the images is 256 × 256.

SNGAN dataset: SNGAN [17] was proposed in 2018. We downloaded 7150 images of the dog and cat which generated by SNGANs, and images with the resolution of 128×128 and images with the resolution of 256 × 256 are half each.

StyleGAN dataset: StyleGAN [3] was proposed in 2019. It works well when generating a single object. We downloaded 10,000 images of the bedroom generated by StyleGANs which trained on LSUN dataset, and the resolution of the images is 256 × 256.

B. EXPERIMENTAL DETAILS

If no special instructions are given, all the experiments were implemented using TensorFlow framework and trained on a single NVIDIA GTX2080TI GPU. ADAM optimizer is used to minimize the cross entropy loss with an initial learning rate of 0.0005, and decay of learning rate 0.85 every 600 steps, a minibatch size of 56, a batch normalization decay parameter of 0.95, and a weight decay(L2 regularization) parameter of 0.0001.

Here, we have a few more details to explain.

We use the following rules to convert the RGB color space to the YCrCb color space:

$$\begin{aligned} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ Cr &= (R - Y) \times 0.713 + 128 \\ Cb &= (B - Y) \times 0.564 + 128 \end{aligned} \tag{5}$$

In our experiments, detecting tampered images and GANs generated images at the same time is a multi-classification problem, we used the Macro-F1 score to evaluate our model.

$$Macro = P = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{6}$$

$$Macro - R = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{7}$$

$$Macro - F1 = \frac{2 \times Macro - P \times Macro - R}{Macro - P + Macro - R} \tag{8}$$

where P_i is the precision of class i, R_i is the recall of class i, n is the number of categories.

When we evaluate our model with accuracy, we use the following rule:

$$Accuracy = \frac{CNumber}{TNumber} \tag{9}$$

where *CNumber* is the number of correctly classified images, *TNumber* is the total number of images.

C. DETECTION PERFORMANCE

We experimented on the CACIA 2.0 and BigGANs datasets to test the performance of detecting tampered images and GANs generated images at the same time in various situations. Specifically, we tested the performance difference of different threshold values used in edge information matrix, of different color space, of different deep neural networks for classification and the performance difference between Sobel operator and Scharr operator. In addition, We test the performance of two methods proposed in [11], [31] and the XceptionNet which has the best performance in detecting GANs generated images among the multiple methods tested in [32].

We randomly selected 5123 images from 7491 real images and 5123 tampered images from CASIA 2.0, and 5123 images were randomly selected from the 16,000 images of BigGANs dataset. Then, we randomly selected 4123 images from each class, and total 12,369 images as the training set, and the remaining 1000 images from each class and total of 3000 images as the test set. The experimental results are shown in Table 1.

We found that the best results were achieved with the truncation value of 192, that the truncation value below 192 resulted in a performance reduction, and that the truncated value above 192 did not bring much higher performance. In addition, both for tampered images and GANs generated images, we found that the Scharr operator used in our method has better performance than the Sobel operator in dealing with the edge detection tasks for fake images detection. Compared with traditional convolutional network model and other classic networks model, the deep neural network model based on depthwise separable convolution has better classification performance. Compared with the component in other color space, Cr and Cb components in YCrCb color space can perform feature extraction better in fake images detection tasks. And compared with the method proposed in [11], [31] and XceptionNet, our method achieves better performance.

Meanwhile, it can be noticed that, under some particular conditions, our model reached very high precision and recall(it even to be 100%). And with different parameters, our model all works well in detecting images generated by BigGANs. Hence, with these results, we can conclude with confidence that our model can extract the characteristic of the images generated by GANs very well.

Our general model also has good performance in detecting tampered images compared to other methods which specially designed to detect tampering. The experimental results are shown in Table 2.

In the best combination in the Table 1, if calculate the accuracy according to the rule (9), only real images and tampered images are considered, and GANs generated images are ignored, the detection accuracy of our model trained on

TABLE 1. Results of various experimental conditions: edge extraction operator, the threshold used in edge information matrix, Color space and its corresponding components used to extract edge extraction, which deep networks model are used for classification (SepConv$_{wp}$ represents that most convolution operations are depthwise separable convolution in our proposed network architecture, and Conv2$_{wp}$ represents that all convolution operations use traditional convolution in our proposed network architecture. Resnet18 is the network architecture proposed in [25]). Tp represents the tampered image, GANs represents the GANs generated image, Au represents the real image, and their subscript R represents recall, and subscript P represents precision.

Threshold; Color Space; Model; Operator		Tp_R	$GANs_R$	Au_R	$Macro_R$	Tp_P	$GANs_P$	Au_P	$Macro_P$	$Macro - F1$
SepConv$_{wp}$; Scharr; CrCb	T = 160	98.80%	**100%**	96.70%	98.50%	96.86%	99.80%	98.80%	98.51%	0.9851
	T = 192	**99.40%**	**100%**	96.50%	98.63%	96.69%	**99.90%**	**99.38%**	**98.66%**	**0.9865**
	T = 288	99.10%	**100%**	96.80%	98.63%	96.97%	**99.90%**	99.08%	98.65%	0.9864
T =192; SepConv$_{wp}$; Scharr	AB(LAB)	97.60%	**100%**	95.20%	97.60%	95.78%	99.11%	97.94%	97.61%	0.9761
	UV(LUV)	97.80%	99.80%	**98.50%**	**98.70%**	95.98%	99.50%	97.91%	97.80%	0.9825
	RGB	92.50%	99.60%	86.70%	92.93%	87.43%	98.61%	93.03%	93.02%	0.9298
T =192; CrCb; Scharr	Conv2$_{wp}$	99.00%	**100%**	96.30%	98.43%	96.49%	99.80%	99.07%	98.46%	0.9844
	Resnet18	85.89%	**100%**	97.50%	94.43%	**98.73%**	96.90%	88.72%	94.78%	0.9461
T =192; CrCb; SepConv$_{wp}$; Sobel		96.00%	99.90%	94.30%	96.73%	94.67%	99.40%	96.13%	96.73%	0.9673
XceptionNet [19]		95.30%	99.30%	74.90%	89.83%	82.94%	92.89%	95.78%	90.54%	0.9018
Mo et al. [11]		96.40%	98.30%	84.80%	93.17%	85.92%	98.01%	96.91%	93.61%	0.9339
Nataraj et al. [31]		85.20%	98.60%	58.90%	80.90%	67.19%	99.19%	79.81%	82.07%	0.8148

TABLE 2. Performance comparison between our method and other methods in CASIA2.0 dataset.

Method	Training dataset	Accuracy
Ours	BigGANs+CASIA2.0	97.95%
Ours	CASIA2.0	**99.25%**
Rao et al. [15]	CASIA2.0	97.83%
Alahmadi et al. [27]	CASIA2.0	97.50%
Wang et al. [6]	CASIA2.0	95.60%

BigGANs dataset and CASIA2.0 dataset is 97.95%. And if we train our model only on CASIA2.0 dataset just like [15], [27] and [6], the detection accuracy is 99.25%, and the accuracy reported in [15], [27], and [6] were 97.83%, 97.50% and 95.60%, respectively. Furthermore, CASIA1.0 dataset can be considered as a simplified version of CASIA2.0 dataset, if our model trained and tested on CASIA1.0 dataset, the detection accuracy can to be 100%, and the accuracy reported in [15], [27] and [36] were 98.04%, 97.00% and 96.81%, respectively. Therefore, our method works well in detecting tampered images compared with previous work.

D. GENERALIZABILITY

1) GENERALIZABILITY ON TAMPERED IMAGES

We tested the generalizability of our model in the COVERAGE dataset and the GIRP dataset. Specifically, we first evaluate our approach on two datasets separately and then perform cross evaluation on the two datasets (one dataset as training and other as testing). Since COVERAGE dataset and GPIR dataset have limited samples, they are too small for retraining our deep neural network. We used 50% images randomly selected from COVERAGE tampered images or GPIR tampered images to fine-tune the original trained model, and the remaining 50% of the images and another dataset for test. We only performed 150-step parameter updates operation using a learning rate of 0.0001, and with a minibatch size of 8. The experimental results are shown in Table 3.

TABLE 3. The generalizability on COVERAGE dataset and GPIR dataset.

Training dataset	Testing dataset	Accuracy
COVERAGE	COVERAGE	100%
GPIR	GPIR	100%
COVERAGE	GPIR	100%
GPIR	COVERAGE	100%

Experiments show that our model can be quickly and easily transferred to other small, novel tampered datasets.

2) GENERALIZABILITY ON GANs GENERALIZED IMAGES

The new GANs model for image generation is growing fast. It is important to have a general way to detect the GANs generated images.

Our method has generalized detection capabilities for a variety of GANs models. We selected those GANs models with good quality of generated images to test the performance of our model, and use rule (9) to calculate the accuracy. Note that the model we used to evaluate the generalization performance is the one that has been trained in the experiment in Table 1 with Cr and Cb components, Scharr operator, truncation value of 192, depthwise separable convolution, and also trained on the BigGANs dataset and CASIA 2.0 dataset. The experimental results are shown in Table 4.

Experiments show that our model has a strong generalization ability to detect images generated by GANs. This is a remarkable observation, that means the images generated by various GANs models have their inherent universal characteristics, and our model also learned these characteristics well.

TABLE 4. The generalizability on the images generated by other various GANs on the BigGANs trained model, the datasets was described above.

Dataset	Number of Images	Accuracy
StyleGAN+LSUN	10000+10000	99.93%
PGGAN	10000	99.8%
Selected PGGAN	169	98.81%
SNGAN(128+256)	3575+3575	100%

V. CONCLUSION

In this paper, we propose a general model that can detect both tampered images and GANs generated images. First, we converted the RGB image to be detected into YCrCb color space and extracted the image edge information of the Cr component and Cb component. Then, we convert image edge features into GLCM in order to do image scaling with without losing the image tempering information. Finally, GLCM is fed into our designed deep neural network based on depthwise separable convolution for training and detection. The edge feature extraction method and the deep neural network model we designed can identify tampered images and GANs generated images with a high macro average of F1 score of 0.9865. Also, our model achieves good performance in just detecting tampered images compare with previous work. Besides, our model can detect images generated from scratch by different GANs models at the same time with high accuracy. We think the reason is that the images generated by GANs will leave trace on the edges of the objects, and our model learned this mark well.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[2] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: https://arxiv.org/abs/1809.11096

[3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4401–4410.

[4] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 5907–5915.

[5] W. Wang, J. Dong, and T. Tan, "Effective image splicing detection based on image chroma," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1257–1260.

[6] W. Wang, J. Dong, and T. Tan, "Image tampering detection based on stationary distribution of Markov chain," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2101–2104.

[7] G. Muhammad, M. H. Al-Hammadi, M. Hussain, A. M. Mirza, and G. Bebis, "Copy move image forgery detection method using steerable pyramid transform and texture descriptor," in *Proc. Eurocon*, pp. 1586–1592, Jul. 2013.

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

[9] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[10] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1053–1061.

[11] H. Mo, B. Chen, and W. Luo, "Fake faces identification via convolutional neural network," in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, 2018, pp. 43–47.

[12] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 38–45.

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

[14] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," 2015, *arXiv:1506.03365*. [Online]. Available: https://arxiv.org/abs/1506.03365

[15] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[16] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: https://arxiv.org/abs/1710.10196

[17] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*. [Online]. Available: https://arxiv.org/abs/1802.05957

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1251–1258.

[20] 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," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[21] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, Jul. 2013, pp. 422–426.

[22] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[23] Y. Mirsky, T. Mahler, I. Shelef, and Y. Elovici, "CT-GAN: Malicious tampering of 3D medical imagery using deep learning," 2019, *arXiv:1901.03597*. [Online]. Available: https://arxiv.org/abs/1901.03597

[24] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[26] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: https://arxiv.org/abs/1805.08318

[27] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing image forgery detection based on DCT and local binary pattern," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 253–256.

[28] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[29] S.-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. A. Efros, "Detecting photoshopped faces by scripting photoshop," 2019, *arXiv:1906.05856*. [Online]. Available: https://arxiv.org/abs/1906.05856

[30] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 472–480.

[31] L. Nataraj, T. M. Mohammed, B. S. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, "Detecting GAN generated fake images using co-occurrence matrices," 2019, *arXiv:1903.06836*. [Online]. Available: https://arxiv.org/abs/1903.06836

[32] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of GAN-generated fake images over social networks," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 384–389.

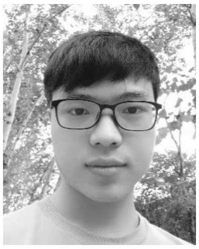
[33] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy–move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.

[34] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "COVERAGE—A novel database for copy-move forgery detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 161–165.

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

[36] S. Agarwal and S. Chand, "Image forgery detection using co-occurrence-based texture operator in frequency domain," in *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Singapore: Springer, 2018, pp. 117–122.

KEJUN ZHANG was born in Baishan, Jilin, China, in 1972. He received the Ph.D. degree in computer science and technology from the University of Science and Technology Beijing, in 2006. Since 2011, he has been an Associate Professor and a Master's Tutor with the Computer Science and Technology Department. He is currently the Deputy Director of the Postgraduate Department, Beijing Institute of Electronic Science and Technology. He holds two patents and four software copyrights. His research interests include intelligent computing, natural language processing, machine learning, access control and audit, content security, and cloud computing security. He is a council member of the Beijing Association for Artificial Intelligence and received three provincial and ministerial level awards.



YU LIANG was born in Loudi, Hunan, China, in 1995. He received the B.S. degree from Xi'an University, in 2017. He is currently pursuing the M.S. degree in deep learning and digital image forensics with Xidian University. His research interests include deep learning, machine learning, and digital image forensics.

JIANYI ZHANG received the Ph.D. degree in computer security from Beijing University of Posts and Telecommunications (BUPT). From 2009 to 2012, he was a Security Researcher with Huawei Digital Technologies, Beijing, China. Since 2012, he has been with the Faculty of Computer Science, Beijing Electronic Science and Technology Institute (BESTI). His research interests include the internet security, data security, and privacy.

ZHIQIANG WANG was born in China, in 1985. He received the Ph.D. degree in information security from Xidian University. He is currently an Assistant Professor with the Department of Computer Science and Technology. His research interests include system security and network security.

XINXIN LI was born in Handan, Hebei, China, in 1996. She received the B.S. degree from the Hebei Normal University of Science and Technology, in 2017. She is currently pursuing the M.S. degree in computer vision with Xidian University. Her research interests include image processing, image illumination estimation, augmented reality, and deep learning.