Multimedia Tools and Applications (2021) 80:7687–7710
https://doi.org/10.1007/s11042-020-10098-y

Blind detection of glow-based facial forgery

Zhiqing Guo[1] · Lipin Hu[1] · Ming Xia[1,2] · Gaobo Yang[1]

Received: 19 May 2020 / Revised: 27 August 2020 / Accepted: 19 October 2020 /
Published online: 30 October 2020

Abstract
With the rapid development of artificial intelligence technologies, various generative models can synthesize fake face images with photo-realistic effects. Glow, a generative flow using invertible 1×1 convolution, is a state-of-the-art technique for efficient synthesis of face images with high resolution and fidelity. However, facial forgeries bring serious challenges to morality, ethics and public confidence. Especially, facial forgeries might change the semantic content conveyed by a face image. A Convolutional Neural Network (CNN) based model, namely SCnet, is proposed to expose the Glow-based facial forgery. Specifically, an image sharpening operator is embedded in the convolutional layer as the pre-processing layer of the network to highlight the traces left by Glow. Then, SCnet is specifically designed to automatically learn high-level forensics features from pre-processing results. Moreover, a fake face dataset is built by exploiting the CelebA face image dataset and the Glow-based forgery technique. A series of experiments are conducted to prove the effectiveness of the proposed approach. Experimental results show that the proposed approach achieves a classification accuracy up to 95.92% under various post-processing operations.

Keywords Convolutional neural networks · Fake face datasets · Facial forgery · Passive image forensics

1 Introduction

1.1 Face Forgery

Face images contain rich and intuitive personal identity information such as gender, race, emotion, age and health status. As a widely-accepted biological modality, face image has been used in many applications such as automatic border control and online payment by face-scanning. However, face images have vulnerability and weak privacy, which implies

✉ Gaobo Yang
yanggaobo@hnu.edu.cn

[1] School of Information Science and Engineering, Hunan University, Changsha, 410082, China

[2] College of Electrical and Information Engineering, Southwest Minzu University, Chengdu 610041, China

that they are easy to be tampered and forged [2]. There are many specifically-designed face image editing tools such as PicTreat, FaceYou, FaceSwap and FaceForge. Especially, with the rapid development of Artificial Intelligence (AI) technologies, many generative models including Generative Adversarial Networks (GANs) [13], Generative Flow Models [19] and Variational Autoencoders (VAEs) [20] were presented in past years to generate fake face images. They achieve much better visual qualities than the classical Computer Graphics (CG) based methods such as Face2Face [37]. Figure 1 shows the most recent progresses in generating fake face images. These techniques promote the development of industrial applications such as film production, affective interaction, and virtual/augmented reality. However, they might also be used for malicious purposes. In June 2019, the Associated Press reported that spy used AI to create fake LinkedIn photo to fool targets via Phantom LinkedIn Profile [12]. Katie Jones, whose photo is believed by AI experts to be almost certainly created by GAN, does not exist. Apparently, this brings serious crisis to social security and public confidence. U.S. lawmakers held their first hearing devoted primarily to the threat of artificially generated imagery.

Face image tampering can be divided into two categories: face identity tampering (FIT) and face expression tampering (FET) [33]. FIT refers to face replacement via FaceSwap [21], or generating fake face images of entirely imaginary people [16, 18]. FET refers to generating face images with specific expressions [9], or transferring facial expression from the source actor to the target [37]. Face2Face animated the facial expressions of the target video by a source actor and re-rendered the manipulated video in a photo-realistic fashion [37]. In recent years, a few generative models, which include ExprGAN [9], StarGAN [6], GANimation [31] and NeuralTextures [36], were proposed for photo-realistic FET. Besides, Glow [19], which is a common DeepFake forgery, was also proposed by extending existing NICE [8] and RealNVP [10] flows. By exact latent-variable inference and efficient texture synthesis, Glow synthesizes hyper-realistic faces images, in which facial expression

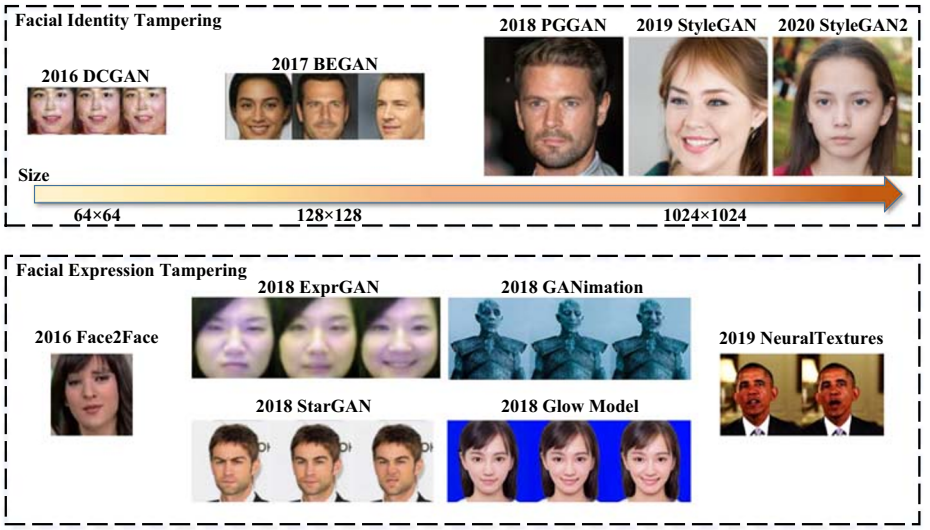

Fig. 1 Research progress in generating fake face images. In the facial identity tampering, from left to right, the spatial resolutions of fake face images are 64×64, 128×128 and 1024×1024, respectively. In the facial expression tampering, we show the expression images generated by different forgery methods

is intensity-adjustable. Facial expression is a high-level facial attribute, which is also an important form of non-verbal communication. We can perceive true emotions from facial expressions existed in face images. If facial expressions, especially those of politicians or public figures, are maliciously tampered, it might bring serious public opinions. Therefore, at the era of face-scanning everywhere and the beginning of AI to "lie", fake facial expression detection is becoming an urgent issue to be solved in the community of image forensics.

1.2 Fake face detection

Image blind detection is to identify the authenticity of digital images without the reference of the original image or any pre-embedded information [24]. In the community of fake face image detection, blind detection is to expose fake face images.

Existing blind detection works are mainly proposed to expose FIT. Nhu et al. [29] added a fully connected layer to VGG-Face [30] and fine-tuned the network to detect fake face images generated by DCGAN [32] and PGGAN [17]. Dang et al. proposed a customized Convolutional Neural Networks (CNN), namely CGface [7], to detect fake face images generated by PGGAN and BEGAN [5]. However, CNN tends to learn the content features of images instead of subtle tampering traces. If the content features are suppressed before the image is fed into the CNN model, the detection performance will be improved. Thus, Mo et al. [28] combined the high pass filter with the CNN model to detection fake face images generated by PGGAN, which achieves a detection accuracy up to 99.4%. Zhou et al. [39] built a new dataset of tampered face images with two online face swapping Apps including SwapMe.[1] and FaceSwap[2] Then, a two-stream network was presented for face tampering detection by capturing tampering artifacts and local noise residuals. In addition, Li et al. [25] proposed a novel method to focus on the blending boundary artifacts between the altered face and the existing background image to expose the FIT. However, it is difficult to achieve FET detection only by focusing on the boundary artifacts of human face contour.

For the forensics of FET, Rössler et al. [34] built a FaceForensics dataset with about 500K FET images by using the CG-based Face2Face. By fine-tuning an existing CNN model, a preliminary detection was conducted on this dataset. Afchar et al. [1] designed a compact deep learning model, namely MesoNet, for facial video forgery detection, which achieved detection accuracies of 98% for Deepfake videos and 95% for Face2Face videos on the FaceForensics dataset. Figure 1 compares the final FET effects among Face2Face and five generative models. We can observe that Glow and NeuralTextures achieves the best photo-realistic FET results. Thus, the forensics for Glow-based or NeuralTextures-based FET will be more challenging. In this paper, we will focus on the detection of Glow model. Unluckily, there is still no work reported for the blind forensics of Glow and other generative models based FET. The reasons behind this are two-fold: First, since tampering detection usually falls behind image tampering, FET detection, which is a countermeasure to FET, is an issue to be addressed in the AI era, especially when these generative models based FET techniques are the most recent. Second, the open FaceForensics dataset is built by only exploiting Face2Face, we are still lacking a more universal dataset, which should contain fake face images by the latest generative models, for FET forensics. Actually, existing

[1] https://itunes.apple.com/us/app/swapme-by-faciometrics.
[2] https://github.com/MarekKowalski/FaceSwap.

works for face image forgery detection are still in the stage of preliminary exploration. They provide only binary classification about the trustworthiness of face images without considering complex conditions for practical forensics scenarios.

1.3 Our Contribution

To address the above issues, we propose a blind detection approach for the Glow-based FET. As the latest generative model, Glow is exploited to build a fake face dataset, which is referred to the Glow-based Fake Face (GFF) dataset, from the CelebA face image dataset [27]. Then, a forensics model based on Sharpening operation and CNN, namely SCnet, is designed for face forensics. The main works and contributions are summarized as follows.

- The GFF dataset, which contains 321,378 face images, is built by exploiting the Glow model and the CelebA face image dataset. Different from the FaceForensics dataset, the GFF dataset is based on the state-of-the-art generative model, which will be available for researchers soon via GitHub.[3]
- The CNN-based SCnet model is proposed for face forensics. This is the first attempt towards the blind detection of fake facial expression generated by advanced generative models. Specifically, an image sharpening operator is embedded in the convolutional layer as pre-processing layer to highlight subtle traces left by Glow, and the SCnet model is designed to automatically learn high-level forensics features.
- We simulate more complex scenarios for face forensics as real as possible. A series of experiments are conducted to prove the effectiveness of the proposed approach, which is evaluated by multi-class classification tasks. Compared with Meso-4 [1], the proposed SCnet model achieves higher detection accuracies and better generalization capabilities.

The rest paper is organized as follows. Section 2 describes the building of the GFF dataset; Section 3 proposes the SCnet model for face forensics; Section 4 reports experimental results and analysis; Section 5 concludes this paper.

2 GFF dataset generation

The CelebA dataset is a large-scale face attributes dataset with large diversities including 10,177 number of identities and more than 200K face images. Glow is a generative flow for photo-realistic facial expression synthesis, which can change face attributes to different expressions. It embeds a series of steps of flow into a multi-scale architecture, where each step of flow consists of actnorm, invertible 1×1 convolution, and coupling layer. This architecture has a depth of flow K, and number of levels L [19], see Fig. 2.

Some reference have proved that different GAN models will leave their unique manipulation traces [38]. In addition, reference [14] marks various style images generated by StarGAN as the same label for training, thus realizing the detection of StarGAN. That is, no matter the face image is changed to any style by the same GAN model, the images of different styles will leave same manipulation traces belonging to the model. Therefore, we use Glow model to generate a type of image (i.e. change the original expression into a smiling face) to expose the Glow model by extracting unique manipulation traces. In addition, we

[3] https://github.com/EricGzq/GFF-Dataset.

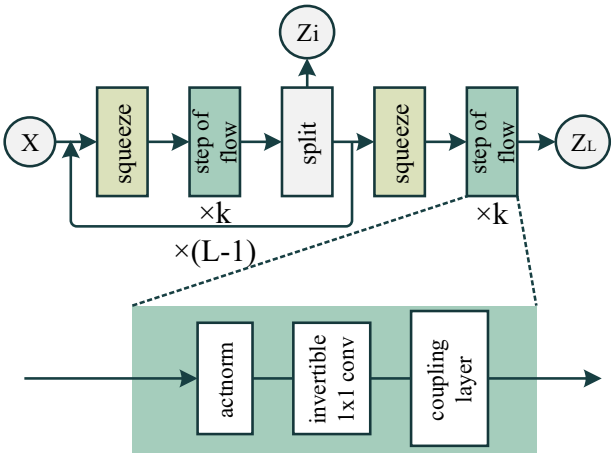

Fig. 2 Glow model. Each step of flow consists of an actnorm step, an invertible 1 × 1 convolution and an affine transformation

put the smiling face image in CelebA dataset into the real face categories in GFF dataset to prevent the forensics model from being classified according to semantic information (smiling face and non-smiling face).

The construction of GFF dataset is shown in Fig. 3. Specifically, there are four steps to build the GFF dataset. Firstly, face localization is conducted for the 202,599 face images in the CelebA dataset, and those face images with failed face localization are removed.

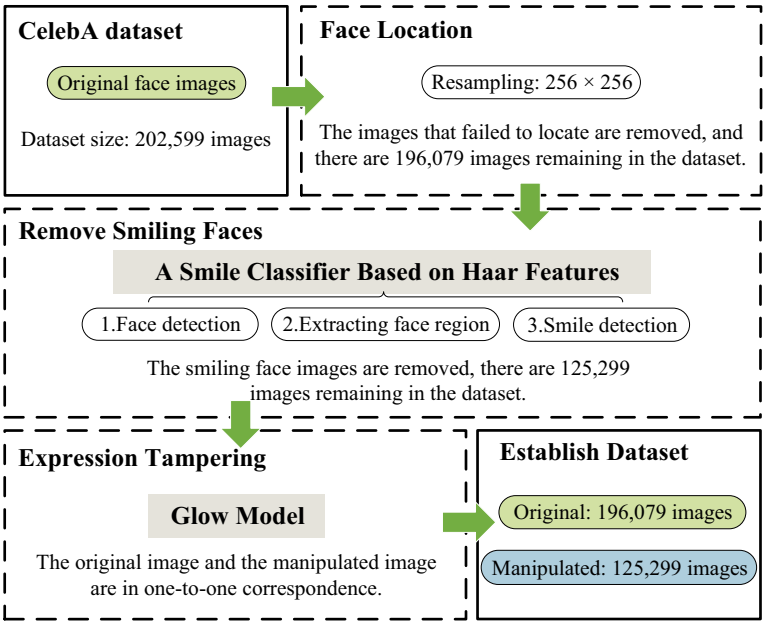

Fig. 3 Flowchart of building the GFF dataset

Fig. 4 Some face images in the GFF dataset. It can be seen that the expression image generated by Glow model does not leave any visual artifacts

Secondly, the Haar-based cascade smiling face classifier[4] is used to remove smiling face images. Thirdly, the rest face images after the above two steps are input into the Glow model for facial manipulation. Fourthly, considering the diversity of the GFF dataset, the removed smiling face images by the second step, which are original face images, are put back into the GFF dataset. Thus, the GFF dataset, which contains 321,378 face images, is built for the face forensics. Each image has 256×256 pixels. We randomly select some face images from the GFF dataset, which are shown in Fig. 4.

Glow is the state-of-the-art generative model for facial forgery. Most face images produced by Glow have perfect visual qualities. However, Glow still lacks sufficient robustness. There are also some face images with undesirable artifacts, which can be perceived by human eyes. Please note that no matter whether the face images produced by Glow are desirable or not, there exist subtle or noticeable manipulation traces, which might be exploited for fake face detection. Moreover, due to the sample diversity of the CelebA dataset, the GFF dataset is also diverse. Figure 5 shows the diversity in the dataset. The forensics models are trained and tested on the GFF dataset, which is avoided from image classification in terms of image content and expression semantics. Thus, the proposed SCnet model focuses better on the manipulation traces left by Glow model.

3 SCnet model for FET forensics

CNN has been increasingly used in image forensic applications, which achieve superior performances when detecting image manipulations including JPEG recompression, median filtering, image resizing and contrast enhancement. The CNN models for image forensics learn deep features from manipulation traces, instead of image contents. If CNN can be forced to directly learn the manipulation traces extracted from the image instead of the image itself, we can more effectively improve the detection performance. Thus, the design of CNN models for image forensics can be roughly divided into two categories: (1) Stacking

[4] https://github.com/liuxiaolong19920720/Laughter-detection-python.

Fig. 5 The diversity in the GFF dataset. In the above two types of training data, face images have various expressions and styles, so that the detection model is classified according to subtle manipulation traces left by Glow instead of semantic information

existing CNN modules for a specific image forensics task [29] [7]; (2) Introducing constrained convolution layer [3] or high-pass filter as pre-processing to enforce CNN learn deep features from manipulation traces.

Most prior face forensics works are designed by stacking traditional CNN modules. These efforts do not specifically restrict CNN to learn manipulation traces directly. In this paper, a CNN model with image sharpening as pre-processing, which is referred to as the SCnet model, is proposed for the detection of the Glow model. By automatically learning high-level forensics features, the SCnet model can robustly detect fake face images with various post-processing operations. Figure 6 is the architecture of the proposed SCnet model. There are three conceptual blocks, namely pre-processing block, hierarchical feature extraction block and classification block, which are detailed as follows.

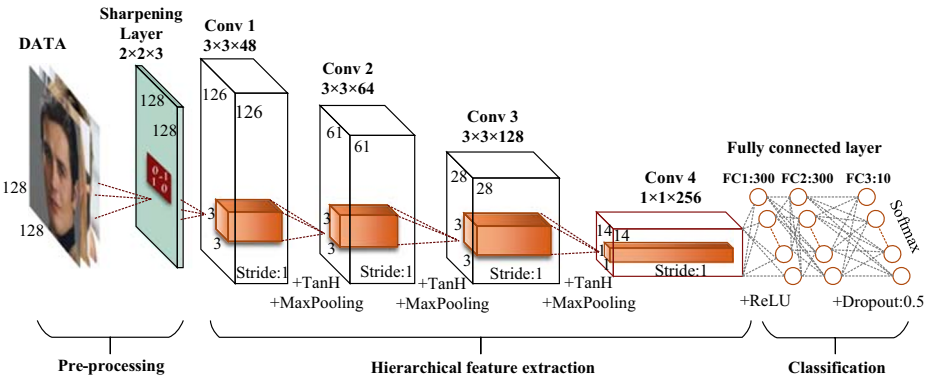

Fig. 6 Proposed SCnet architecture. Conv: Convolutional Layer; TanH: Hyperbolic Tangent Activation Function; ReLU: Rectified Linear Unit; FC: Fully Connected Layer

3.1 Sharpening convolutional Layer

The artifacts left by the Glow model are usually very subtle. Before fedding training data into a CNN model for feature learning, the manipulation traces should be highlighted via appropriate pre-processing. Thus, the CNN model learns better features for artifacts representation, which will improve forensics performances such as detection accuracy and robustness.

Facial expression is represented by face image details such as wrinkles. For a face image, wrinkles usually exist in faces due to expression changes. Since the Glow model learns facial expressions from a vast number of face images with natural expressions, the face images generated by it have seemingly natural details such as wrinkles. Compared with other FET methods, Glow model presents richer expression details. If these details can be highlighted, CNN can learn the forensics clues from it more easily. In addition, we observe that though Glow enhances face image details for better FET, it leaves some subtle traces in the global image, such as isolated points or lines. Motivated by the fact that high frequency components reflect image details and abnormal pixels, image sharpening is exploited as pre-processing to expose the artifacts by enhancing high frequency components.

There are many derivative-based operators for image sharpening. Among them, the Roberts cross operator is a first-order edge detector. Its basic idea is to approximate the gradient of an image through discrete differentiation, which is achieved by a pair of 2×2 convolution kernels G_x and G_y that differentiate diagonally adjacent pixels' values. Specifically, G_x and G_y are defined as follows.

$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}, G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \tag{1}$$

Please note that one kernel is simply the other rotated by 90^o. The Roberts cross operator can highlight regions of high spatial frequency, which correspond to details (such as isolated points or lines). Since facial expression details or isolated points in face images are multi-directional, the first-order difference in either diagonal or back-diagonal direction is sufficient to expose manipulation traces. To simplify computation, only the first-order difference in diagonal direction is used to define the image sharpening operator for pre-processing. That is, the convolution kernel R is given by

$$R = \begin{bmatrix} 0 & -1 \\ +1 & 0 \end{bmatrix} \tag{2}$$

To achieve the end-to-end model training, we add a convolutional layer with the kernel size of 2×2 in front of the network, and fix the parameters of the convolution kernel with R (please note that the parameters are not updated during back propagation).

Figure 7 shows the effectiveness of image sharpening as pre-processing. We depict the RGB images and the corresponding pre-processing results in various situation. The first column to the sixth column correspond to 6 types of images, i.e., the original image, the manipulated image by Glow, and with further post-processing operations, respectively. There are four kinds of post-processing operations, which are JPEG compression (JP, QF=30), Gaussian Blur (GB, 5×5), Mean Filtering (ME, 5×5) and Median Filtering (MED, 5×5), respectively. Moreover, some representative blocks, which are marked with red boxes, are selected from them. These blocks are enlarged to be shown in (c1)-(c6) and (f1)-(f6), respectively. From Fig. 7, we can observe that the images with the same operations exhibit

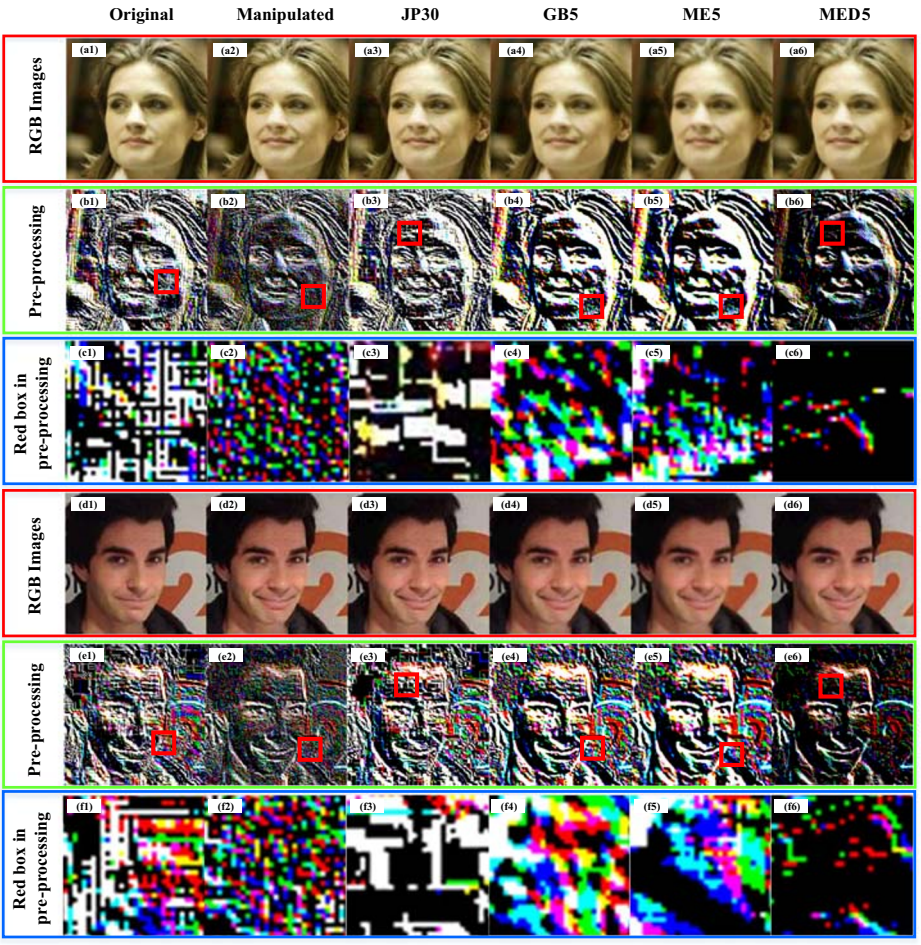

Fig. 7 Comparison of the pre-processing results under the different post-processing operations. The red border illustrates the RGB images with various operations. The green border illustrates the pre-processing results corresponding to RGB images. The blue border illustrates the detailed information in the pre-processing results. JP30: JPEG compression with the quality factor of 30; GB5: gaussian blur with a kernel size of 5×5; ME5: mean filtering with a kernel size of 5×5; MED5: median filtering with a kernel size of 5×5

their unique texture characteristics after pre-processing. For example, the original images after image sharpening (c1 and f1) have regular texture distribution, but the tampered images after image sharpening (c2 and f2) contain a large number of dot and strip textures, which also proves that Glow model will leave its manipulation traces in the global image. For the tampered images with further JP (c3 and f3), there are plenty of blocky textures. For the tampered images with further GB (c4 and f4) or ME (c5 and f5), they seem to be similar, the later (c5 and f5) have more stripe textures than the former (c4 and f4). For the tampered images with further MED (c6, f6), they also contain a few dot and strip textures but their background is much simpler. Since these images show different texture features after image sharpening, which facilitates the SCnet model to learn more discriminative features for classification.

3.2 SCnet model

After the pre-processing layer, image data are fed into the hierarchical feature extraction block, which is formed by stacking four convolutional layers, to obtain high-level forensics features. The convolutional operation is defined as follows.

$$f_j^{(n)} = \sum_{i=1}^{i} f_i^{(n-1)} * \omega_{ij}^{(n)} + b_j^{(n)} \tag{3}$$

where $f_j^{(n)}$ is the j^{th} feature map in the n^{th} layer, $f_i^{(n-1)} * \omega_{ij}^{(n)}$ is the convolution between the feature map of the i^{th} channel in the $(n-1)^{th}$ layer and the i^{th} channel of the j^{th} convolutional kernel in the n^{th} layer, and $b_j^{(n)}$ is the bias term of the j^{th} convolutional kernel in the n^{th} layer.

The initial coefficients of the convolutional kernels and fully connected layers are randomly set. The weights are updated by alternating iteration of forward propagation and back propagation of the input image data [22]. Stochastic Gradient Descent (SGD) is used for training. The rules for iterative updating are

$$\bigtriangledown \omega_{ij}^{(n)} = \varepsilon \frac{\partial E}{\partial \omega_{ij}^{(n-1)}} - M \cdot \bigtriangledown \omega_{ij}^{(n-1)} + D \cdot \varepsilon \cdot \omega_{ij}^{(n-1)} \tag{4}$$

$$\omega_{ij}^{(n)} = \omega_{ij}^{(n-1)} - \bigtriangledown \omega_{ij}^{(n)} \tag{5}$$

where $\bigtriangledown$ is the gradient, $\omega_{ij}^{(n)}$ is the weight of the i^{th} channel of the j^{th} convolutional kernel in the n^{th} layer, E is the loss function, and ε is the learning rate. Moreover, we introduce the momentum M and the decay D to accelerate model training [23].

The model needs to minimize the average loss E between the true label and the network output to make it converge, which completes the model training [11]. The average loss E is

$$E = -\frac{1}{x} \sum_{i=1}^{x} \sum_{k=1}^{n} L_i^{(k)} log(y_i^{(k)}) \tag{6}$$

where $L_i^{(k)}$ is the true label of the i^{th} image in k^{th} class, $y_i^{(k)}$ is the network output, x is the number of training sample, and n is the number of neurons in the output layer.

Four convolutional layers with increasing number of kernels are stacked for hierarchical feature extraction. Each layer learns a new set of feature maps from the previous layer. The size of the receptive field for the first three convolutional layers is 3×3, and the size of the receptive field in Conv 4 is 1×1. The number of kernels for the 4 convolutional layers is 48, 64, 128, and 256, respectively. The reason behind this is that the next convolutional layer needs to increase the number of feature maps to fully extract features from the previous layer. The influence of the number of the convolutional kernels will be verified in Section 4.1.2. The stride in each convolutional layer is 1, and the first three convolutional layers will be followed by a set of hyperbolic tangent activation functions (TanH) and max pooling functions. To reduce the dimension of feature maps and the probability of overfitting, the pooling function adopts an overlapping kernel with a size of 3×3 and a stride of 2. The deep features learned by the first three convolutional layers are obtained from learning local spatial correlation in the receptive field. The convolutional kernel of size 1×1 learns the linear combination of features located in the same location but different channels. For the Conv 4 layer, cross-channel information integration is achieved to better represent the relation between the previously learned feature maps. The ReLU activation function is

used to further increase nonlinearity. The final classification module consists of 3 fully connected layers. The first two fully connected layers contain 300 neurons, respectively. The last fully connected layer contains 10 neurons, which correspond to the original image and 9 possible tampering operations.

4 Experiment results

In this section, a series of experiments are conducted on the GFF dataset to prove the effectiveness of the proposed SCnet model under various detection conditions.

Datasets Digital images are usually compressed for storage and transmission. Moreover, some post-processing operations including JP, GB, ME and MED might be used to hide manipulation traces, which will mislead the judgment of the forensics model. In fact, we can not predict which post-processing operations that the candidate image has undergone. Thus, it is necessary to simulate some scenes that may occur in real forensics scenarios to improve the robustness of the forensics model in complex Internet environments. Some common post-processing operations are used to simulate the practical internet environment, as shown in Table 1.

In the subsequent experiments, the original image is abbreviated as Raw Pic, and the fake face image is called Glow Pic. For the Glow Pics in the GFF dataset, they are processed with the post-processing operations and parameters summarized in Table 1. Thus, ten datasets are obtained for our experiments. The post-processing operations in the form of A+B+C means A the first, B the second and C the last. For example, Glow Pic + ME3 + Resizing means the Glow Pic is performed with a 3×3 mean filtering and 50% image resizing. Table 2 summarizes the details of the dataset used in the experiment.

Experimental Setting In the experiment, we use one Nvidia GeForce GTX 1080 Ti GPU to train the model, which is implemented via the Caffe framework [15]. All image samples are firstly converted into the LMDB format for the use in Caffe. When converting to the LMDB format, the images with a size of 256×256 in the GFF dataset are resized into the size of 128×128, which is equivalent to a 50% image down-sampling.

4.1 Ablation study of the SCnet model

4.1.1 Ablation study for pre-processing layer

In this subsections, we will discuss the impacts of the pre-processing layer. For experiment, there are about 1070k images for training, which include 170k real face images and 900k different types of fake face images (see Table 2). When training the model, SGD is used to

Table 1 List of parameters for post-processing

Operation type	Post-processing operation	Parameters
Compression	JPEG compression (JP)	QF = 30, 60
Spatial filtering	Gaussian Blurring (GB)	K_{size} = 3, 5
	Mean Filtering (ME)	K_{size} = 3, 5
	Median Filtering (MED)	K_{size} = 3, 5

Table 2 List of datasets for experiments

Classification		Dataset Size	Classification		Dataset Size
Raw	Raw Pic + Resizing	Train:170,780 Test:15,299	GB5	Glow Pic + GB5 + Resizing	Train:100,000 Test:15,299
Glow	Glow Pic + Resizing	Train:100,000 Test:15,299	ME3	Glow Pic + ME3 + Resizing	Train:100,000 Test:15,299
JP30	Glow Pic + JP30 + Resizing	Train:100,000 Test:15,299	ME5	Glow Pic + ME5 + Resizing	Train:100,000 Test:15,299
JP60	Glow Pic + JP60 + Resizing	Train:100,000 Test:15,299	MED3	Glow Pic + MED3 + Resizing	Train:100,000 Test:15,299
GB3	Glow Pic + GB3 + Resizing	Train:100,000 Test:15,299	MED5	Glow Pic + MED5 + Resizing	Train:100,000 Test:15,299

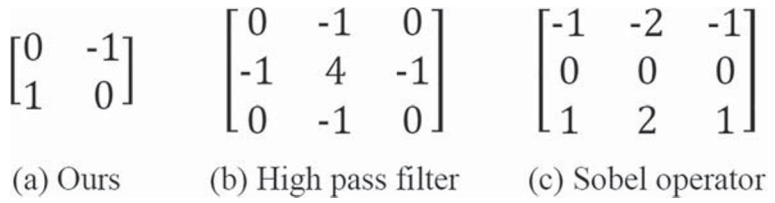

Fig. 8 Comparison of different pre-processing methods

iteratively optimize the model, where the momentum $M = 0.95$ and the decay $D = 0.005$. The model adopts a discrete learning rate strategy, which is defined as follows.

$$\varepsilon = \varepsilon_b \times \gamma^{\lfloor (\frac{I}{N}) \rfloor} \tag{7}$$

where $\varepsilon_b = 0.001$ is the basic learning rate, $\gamma = 0.5$, $\lfloor \rfloor$ is rounding down, $N = 1000$ is the fixed step size, and I denotes current iteration. Since the batch size is 64, we train the CNN model in each epoch for 16,731 iterations.

To evaluate the performance gains achieved by the pre-processing layer, about 153k test images are used for experiments. It should be noted that these test images have never appeared in the training set. The pre-processing method of SCnet model is replaced with several operators (see Fig. 8) to compare the performance of different methods. We use the accuracy rate commonly used in multi-classification tasks to evaluate the performance of the model. The experimental results are summarized in Table 3. Our sharpening method achieves an accuracy rate of 93.77%, which is more effective than both the sobel operator (68.95%) and high-pass filter (91.03%) [28].

4.1.2 Ablation study for CNN model

The design of the CNN model has direct and decisive influence on detection accuracy. Next, we will further conduct several experiments to discuss the impacts of the other components of the SCnet model, which include: (1) the 1×1 convolutional layer; (2) the activation functions; (3) the pooling layers; (4) the convolutional kernel size; (5) the number of the convolutional kernels; (6) the stride for convolution. Specifically, the experimental setups are as follows:

- *1×1 Convolutional Layer:* Its main purpose is cross-channel interaction and information integration. It also perform dimension increase on the number of feature map channels [26]. To verify the effect of the 1×1 convolutional kernel in the Conv 4 layer, it is replaced with 3×3 convolutional kernels. Comparisons are made between the 1×1 convolutional kernel and the 3×3 convolutional kernels.
- *The Activation Function:* The activation function is to increase the nonlinearity of the network, so that the SCnet model can approximate any nonlinear function. In recent

Table 3 Experimental results of different pre-processing method

Pre-processing method	Accuracy rate
Sobel operator	68.95%
High pass filter	91.03%
Ours	93.77%

years, many activation functions have appeared such as ReLU, TanH and Sigmoid. ReLU and TanH, which are two most common activation functions, are selected for comparisons. Specifically, we design the following three schemes: (1) all activation functions adopt ReLU; (2) all activation functions adopt TanH; (3) activation functions adopt the combination of TanH and ReLU. Among them, since the model that adopts ReLU as activation functions can not work well for training, we only provide the statistical results of the rest two schemes.

- *The Pooling Layer:* This layer is introduced to remove redundant information while keeping main features extracted from the previous layer. It also reduces the parameters and prevents over-fitting. There are two widely-used pooling strategies: max pooling and average pooling. For the SCnet model, we use the max pooling strategy for all the pooling layers. Thus, it is replaced with the average pooling strategy to train the SCnet model.
- *The Convolutional Kernel Size:* The size of the convolutional kernel is either 1, 3, 5 or 7. However, the 3×3 convolutional kernel is the smallest that captures well information in a receptive field. It is claimed that using 3×3 convolutional kernel can improve the performance of the CNN model [35]. Here, we try to use the 5×5 convolutional kernel for the first convolutional layer, and then train the model.
- *The number of the Convolutional Kernels:* With the deepening of the CNN network, the dimension of feature maps will decrease. A deeper convolutional layer extracts more representative features. The next convolutional layer needs to increase the number of feature maps to fully extract the features of the previous layer. For the SCnet model, the number of the convolutional kernels determines the number of feature maps. Here, the number of convolutional kernels in Conv 3 and Conv 4 is reduced by 50% (i.e., Conv 3=64 and Conv 4=128).
- *The Stride in Convolutional Layer:* The stride represents the step size of the convolutional kernel sliding in the horizontal and vertical directions. It determines the dimension of feature maps. A small stride of the convolutional kernels extracts more abundant features than a large stride. The stride of convolutional kernel in SCnet model is set to 1, which is replaced with 2 for comparative experiments. Specifically, three schemes are designed: (1)the stride in Conv 1 is replaced by 2; (2)the stride in Conv 1 and Conv 2 are replaced by 2; (3)the stride in Conv 1, Conv 2 and Conv 3 are replaced by 2.

To complete the above experiments, the detection is performed on 10 types of datasets, which are detailed in Section 4.1. Table 4 reports the average detection accuracies for the SCnet model with different structures or parameters. Figure 9 shows the curve of detection accuracy and iteration. From Table 4, we have the following 6 observations. First, the 1×1 convolutional kernel achieves 1.32% higher detection accuracy than the 3×3 convolutional kernels, which benefits from the cross-channel interaction and information integration of the 1×1 convolutional kernel. Second, when all activation functions adopt TanH, the detection accuracy is 95.35%. The combination of TanH and ReLU as the activation functions achieves better detection accuracies, which improve about 0.54%. Third, average pooling achieves less detection accuracy than max pooling, and the detection accuracy decreases about 2.11%. Fourth, Compared with using 5×5 convolutional kernels in the first layer, using 3×3 convolutional kernels improves the accuracy about 0.48%. This implies that a bigger receptive field does not better detection accuracy, and the 3×3 convolutional kernels are sufficient for excellent feature extraction. Fifth, when the number of the convolutional kernels is 64 for Conv 3 and 128 for Conv 4, the detection accuracy is 95.26%.

Table 4 Accuracy rate for different CNN models

Models	Choices	Description of model changes	Average accuracy
Net_1	1×1 convolutional layer	1×1 convolutional kernel is replaced by 3×3 convolutional kernel.	94.57%
Net_2	Activation Function	All activation functions are replaced by TanH.	95.35%
Net_3	Pooling Layer	All pooling layers are replaced by average pooling layers.	93.78%
Net_4	Kernel Size	The convolutional kernel size in Conv 1 changes from 3 to 5.	95.41%
Net_5	Kernel Quantity	Convolutional kernel numbers of Conv 3 and Conv 4 are each reduced by 50%, i.e., Conv 3 = 64 and Conv 4 = 128.	95.26%
Net_6	Kernel Stride	The stride in Conv 1 is replaced by 2.	91.94%
Net_7		The stride in Conv 1 and Conv 2 are replaced by 2.	89.51%
Net_8		The stride in Conv 1, Conv 2 and Conv 3 are replaced by 2.	88.56%
SCnet	The model in Fig. 6	\	**95.89%**

Compared with the original SCnet model, it decreases about 0.63%. Finally, when the stride of the first convolutional layer is set to 2, the accuracy is reduced by 3.95%. Thus, the stride of the first convolutional layer has great impacts on final detection accuracy. The reason behind this is that the convolutional kernels with small stride extract richer features than the convolutional kernels with large stride. However, if the low-level features extracted by the first convolutional layer are too sparse, it will seriously affect the feature extraction by subsequent convolutional layers. Moreover, when the stride is set to 2 in the second convolutional layer (Conv 2) and the third convolutional layer (Conv 3), the detection accuracies

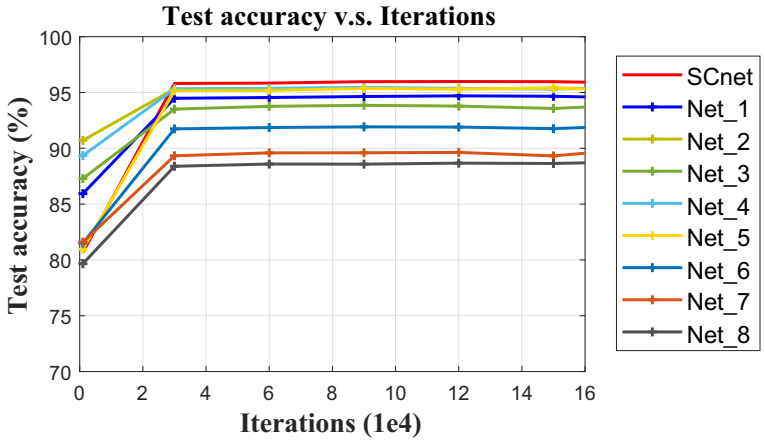

Fig. 9 Accuracy rate of 9 methods

decrease as well. Thus, when designing the CNN model for image forensics, the stride of the convolutional kernels should be set to 1 to improve detection accuracies.

Figure 10 reports the relationships between training loss, testing loss and the number of iterations. From it, we can observe that the testing loss converges to a certain value, which does not increase with the number of iterations. Thus, the proposed SCnet model has not the problem of over-fitting [4].

4.2 Multiple classification detection

To evaluate the performance gains achieved by the pre-processing layer, the SCnet model are tested with the pre-processing layer (SCnet) and without the pre-processing layer (Non-pre) for experiments. The existing Meso-4 [1] model and MISLnet [3] are also used for making comparisons with the SCnet model. We still use the 10 types of data introduced in Table 2 for experiments.

Tables 5 and 6 report the confusion matrix of Meso-4 and MISLnet for detection. Tables 7 and 8 report the confusion matrixes of the Non-pre model and SCnet, respectively. From Tables 5 and 6, Meso-4 and MISLnet achieve an average detection accuracy of 73.35% and 93.39%, respectively. From Tables 7 and 8, the Non-pre model and the SCnet achieve average detection accuracies of 81.51% and 95.92%, respectively. From the confusion matrix, it can be seen that both the MesoNet and the Non-pre model have poor performance in some operations, such as GB3, GB5, ME3, etc. The reason behind this is that the images after different post-processing operations are quite similar in RGB space. If the CNN model is directly used for detection, it is very difficult to extract discriminative features from

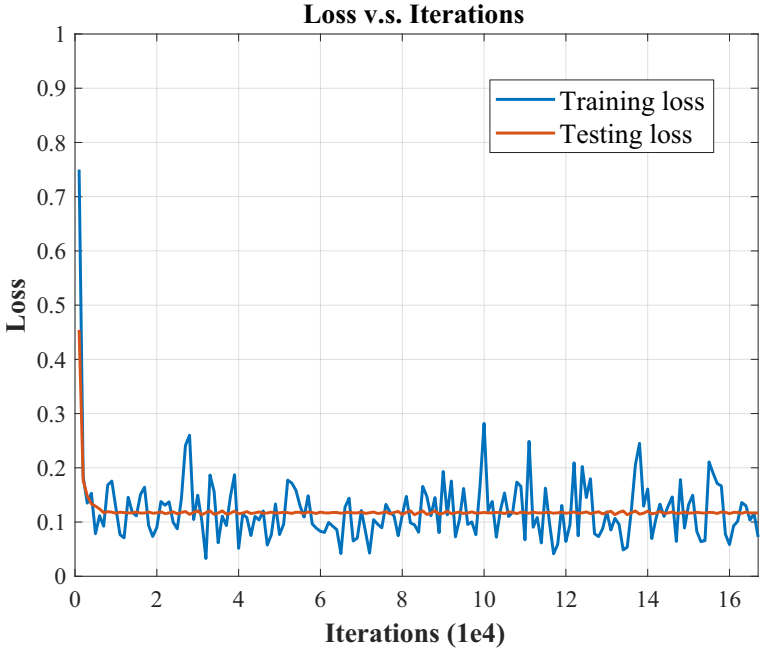

Fig. 10 Convergence curve of the SCnet on 1.07 million face images from the GFF dataset. Training and testing loss are recorded after every 1000 iterations

Table 5 Confusion matrix for identifying various types of operations using MESO-4

	Predicted class									
The class		Glow	Raw	JP30	JP60	GB3	GB5	ME3	ME5	MED3	MED5
	Glow	**94.37%**	0.01%	0.01%	1.28%	0.00%	0.00%	0.00%	0.00%	4.31%	0.01%
	Raw	0.02%	**84.67%**	5.59%	0.61%	0.12%	2.07%	0.72%	6.17%	0.01%	0.01%
	JP30	0.18%	23.01%	**57.02%**	15.57%	2.46%	0.07%	1.61%	0.00%	0.07%	0.02%
	JP60	3.74%	0.52%	17.10%	**73.34%**	3.40%	0.00%	0.05%	0.00%	1.61%	0.24%
	GB3	0.35%	2.42%	4.80%	4.93%	**53.44%**	0.29%	12.89%	0.00%	1.78%	19.09%
	GB5	0.01%	12.11%	0.16%	0.05%	0.33%	**59.51%**	13.44%	14.37%	0.00%	0.03%
	ME3	0.01%	11.47%	1.79%	0.46%	8.20%	20.69%	**56.23%**	0.41%	0.05%	0.68%
	ME5	0.00%	0.34%	0.00%	0.00%	0.01%	0.97%	0.02%	**98.66%**	0.00%	0.00%
	MED3	20.56%	0.02%	0.04%	1.93%	1.01%	0.00%	0.00%	0.00%	**64.25%**	12.19%
	MED5	0.02%	0.01%	0.00%	0.03%	3.08%	0.00%	0.27%	0.00%	4.56%	**92.05%**

Table 6 Confusion matrix for identifying various types of operations using mislnet

The class	Predicted class									
	Glow	Raw	JP30	JP60	GB3	GB5	ME3	ME5	MED3	MED5
Glow	**99.98%**	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%
Raw	0.00%	**99.29%**	0.00%	0.39%	0.18%	0.00%	0.01%	0.01%	0.08%	0.04%
JP30	0.00%	0.00%	**94.75%**	5.25%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
JP60	0.22%	0.01%	0.00%	**99.76%**	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%
GB3	0.00%	0.10%	0.00%	0.04%	**98.90%**	0.00%	0.00%	0.00%	0.96%	0.00%
GB5	0.00%	0.53%	0.00%	0.00%	0.25%	**67.73%**	31.45%	0.03%	0.00%	0.00%
ME3	0.00%	0.23%	0.00%	0.00%	24.92%	0.01%	**74.84%**	0.00%	0.00%	0.00%
ME5	0.00%	0.01%	0.01%	0.00%	0.00%	0.01%	0.00%	**99.97%**	0.00%	0.00%
MED3	0.68%	0.00%	0.00%	0.02%	0.01%	0.00%	0.00%	0.00%	**99.06%**	0.24%
MED5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.39%	**99.61%**

Table 7 Confusion matrix for identifying various types of operations using non-pre model

The class	Predicted class									
	Glow	Raw	JP30	JP60	GB3	GB5	ME3	ME5	MED3	MED5
Glow	**88.11%**	0.00%	0.01%	0.03%	0.04%	0.00%	0.00%	0.00%	11.63%	0.18%
Raw	0.04%	**92.29%**	2.46%	2.86%	0.18%	0.86%	0.16%	1.10%	0.04%	0.00%
JP30	0.01%	5.76%	**76.06%**	18.11%	0.03%	0.02%	0.02%	0.00%	0.00%	0.00%
JP60	0.00%	4.68%	16.30%	**78.79%**	0.08%	0.08%	0.08%	0.00%	0.00%	0.00%
GB3	0.35%	0.90%	0.27%	0.33%	**77.97%**	1.70%	14.33%	0.04%	1.62%	2.48%
GB5	0.00%	2.59%	0.01%	0.07%	0.69%	**78.48%**	14.96%	3.09%	0.01%	0.10%
ME3	0.00%	2.37%	0.07%	0.16%	18.72%	19.45%	**58.04%**	0.42%	0.05%	0.73%
ME5	0.00%	0.37%	0.00%	0.00%	0.00%	1.42%	0.00%	**98.21%**	0.00%	0.01%
MED3	14.79%	0.01%	0.03%	0.01%	1.47%	0.01%	0.02%	0.00%	**74.92%**	8.75%
MED5	0.03%	0.07%	0.00%	0.00%	0.78%	0.07%	0.21%	0.01%	6.63%	**92.21%**

Table 8 Confusion matrix for identifying various types of operations using SCnet

The class	Predicted class									
	Glow	Raw	JP30	JP60	GB3	GB5	ME3	ME5	MED3	MED5
Glow	**99.99%**	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%
Raw	0.05%	**99.75%**	0.00%	0.03%	0.01%	0.01%	0.13%	0.01%	0.03%	0.00%
JP30	0.00%	0.00%	**97.41%**	2.59%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
JP60	0.00%	0.00%	1.64%	**98.36%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
GB3	0.15%	0.12%	0.00%	0.00%	**89.59%**	0.31%	9.76%	0.00%	0.07%	0.00%
GB5	0.00%	0.02%	0.00%	0.00%	0.03%	**94.95%**	3.21%	1.78%	0.01%	0.00%
ME3	0.01%	0.28%	0.00%	0.00%	9.60%	4.05%	**86.02%**	0.01%	0.04%	0.00%
ME5	0.00%	0.00%	0.00%	0.00%	0.00%	1.54%	0.01%	**98.45%**	0.00%	0.00%
MED3	0.00%	0.00%	0.00%	0.00%	0.08%	0.00%	0.00%	0.00%	**97.17%**	2.75%
MED5	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%	2.43%	**97.56%**

Table 9 Confusion matrix for verifying the generalization ability using SCnet

	Predicted class									
The class		Glow	Raw	JP30	JP60	GB3	GB5	ME3	ME5	MED3	MED5
	Glow	**99.99%**	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%
	Raw	0.05%	**99.86%**	0.00%	0.01%	0.06%	0.00%	0.06%	0.00%	0.00%	0.01%
	JP30	0.00%	0.01%	**97.72%**	2.27%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	JP60	0.00%	0.00%	0.92%	**99.08%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	GB3	0.18%	0.09%	0.00%	0.00%	**90.24%**	0.24%	9.19%	0.00%	0.06%	0.00%
	GB5	0.00%	0.03%	0.00%	0.00%	0.02%	**95.49%**	2.81%	1.62%	0.03%	0.00%
	ME3	0.01%	0.24%	0.00%	0.00%	9.45%	3.89%	**86.34%**	0.02%	0.05%	0.00%
	ME5	0.00%	0.02%	0.00%	0.00%	0.00%	1.40%	0.01%	**98.57%**	0.00%	0.00%
	MED3	0.00%	0.00%	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	**97.03%**	2.94%
	MED5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.19%	**97.81%**

different types of images with only subtle differences among them. MISLnet exploits constrained convolution layer to suppress content features and highlight manipulation traces, thus improving detection accuracy. However, the detection accuracy for GB5 and ME3 are still far away from satisfaction. In the SCnet, the proposed image sharpening operation can distinguish well the differences. Thus, the SCnet model achieves much better detection results on GFF dataset. Especially, the pre-processing layer improves the detection capability of the SCnet model, since it effectively highlights the manipulation traces. In addition, we also calculate the floating point operations (FLOPs) of Meso-4, MISLnet and SCnet, which are 4.22×10^7, 6.32×10^8, and 7.23×10^8, respectively. It can be seen that SCnet achieves better results than MISLnet only via smaller FLOPs.

4.3 The generalization capability of the SCnet model

To test the generalization capability of the proposed SCnet model, 20K face images, which include 10K Raw Pics and 10K Glow Pics, are selected from the GFF Dataset. Following the post-processing operations in Table 2, 8 datasets are built based on 10K Glow Pics. Thus, the dataset with 100K face images are used to test the generalization capability. Note that these images have never appeared in either training set or testing set in previous experiments. The trained SCnet model is used to detect 10 types of images, which the SCnet model has never trained or tested before. Table 9 reports the confusion matrixes of the detection results. From it, the proposed SCnet model has good generalization capability.

5 Conclusion

As one of the latest generative models, Glow can synthesize photo-realistic tampering effects. Since facial expression is a high-level facial attribute and an important form of non-verbal communication, FET can change the semantic content that a face image conveys. Thus, FET might bring serious public opinions. In this paper, a CNN-based forensics model, namely SCnet, was proposed for the detection of the Glow-based FET. Specifically, the Roberts cross operator was introduced for image sharpening, which serves as pre-processing to highlight the manipulation traces left by Glow. SCnet was designed to automatically learn high-level features for blind forensics. In addition, a fake face dataset was built by exploiting the CelebA face image dataset and the Glow-based FET technique. A series of experiments were conducted to prove the effectiveness of the proposed approach. Experimental results show that the proposed approach achieves detection accuracy up to 95.92% under various post-processing conditions. Compared with Meso-4, the proposed approach improved average accuracy about 22.57%. And compared with Non-pre model, the SCnet improved average accuracy about 14.41%. For future work, we will further investigate other generative models based facial forgery for more universal forensics. With the continuous improvement of various FET techniques, it is challenging to identify various face forgery techniques by exploiting biological inconsistencies such as head poses. However, each facial forgery technique will leave unique fingerprints specific to its model, just as camera sensor fingerprints. Learning features from fingerprint traces will be a solution to universal forensics.

Acknowledgments This work is supported in part by the National Key Research & Development Plan (2018YFB1003205) and National Natural Science Foundation of China (61972143, 61972142).

References

1. Afchar D, Nozick V, Yamagishi J, Echizen I (2018) Mesonet: a compact facial video forgery detection network. Proc. IEEE Int. Workshop Inf. Forensics Security, pp 1–7
2. Bastanfard A, Bastanfard O, Takahashi H, Nakajima M (2004) Toward anthropometrics simulation of face rejuvenation and skin cosmetic. Comput Animation Virt Worlds 15(3-4):347–352
3. Bayar B, Stamm MC (2018) Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. IEEE Trans Inf Forensic Secur 13(11):2691–2706
4. Bengio Y (2012) Practical recommendations for gradient-based training of deep architectures. Neural Networks: Tricks of the Trade. Springer, pp 437–478
5. Berthelot D., Schumm T., Metz L. Began: Boundary equilibrium generative adversarial networks. [Online]. Available: 1703.10717
6. Choi Y, Choi M, Kim M, Ha J, Kim S, Choo J (2018) Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. Proceedings of CVPR, pp 8789–8797
7. Dang LM, Hassan SI, Im S, Lee J, Lee S, Moon H (2018) Deep learning based computer generated face identification using convolutional neural network. Appl Sci 8(12):2610–2628
8. Dinh L, Krueger D, Bengio Y Nice: Non-linear independent components estimation. [Online]. Available: 1410.8516
9. Ding H, Sricharan K, Chellappa R (2018) Exprgan: Facial expression editing with controllable expression intensity. Proceedings of AAAI, pp 6781–6788
10. Dinh L, Sohldickstein J, Bengio S (2017) Density estimation using Real NVP. Proceedings of ICLR, pp 1–32
11. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12:2121–2159
12. Experts: spy used AI-generated face to connect with targets via phantom LinkedIn profile. [Online]. Available: https://blackchristiannews.com/2019/06/experts-spy-used-ai-generated-face-to-connect-with-targets-via-phantom-linkedin-profile/
13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Proceedings of NIPS, pp 2672–2680
14. Guo Z, Yang G, Chen J, Sun X (2020) Fake faces detection via adaptive residual prediction network. [Online]. Available: 2005.04945
15. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. Proc. 22nd ACM Int. Conf. Multimedia, pp 675–678
16. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T Analyzing and improving the image quality of StyleGAN. [Online]. Available: 1912.04958
17. Karras T, Aila T, Laine S, Lehtinen J (2018) Progressive growing of GANs for improved quality, stability, and variation. Proceedings of ICLR, pp 1–26
18. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. Proceedings of ICCV, pp 4401–4410
19. Kingma DP, Dhariwal P (2018) Glow: Generative flow with invertible 1×1 convolutions. Proceedings of NIPS, pp 10215–10224
20. Kingma DP, Salimans T, Jozefowicz R, Chen X et al, Sutskever I, Welling M (2016) Improved variational inference with inverse autoregressive flow. Proceedings of NIPS, pp 4743–4751
21. Korshunova I, Shi W, Dambre J, Theis L (2017) Fast face-swap using convolutional neural networks. Proceedings of ICCV, pp 3677–3685
22. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
23. LeCun Y, Bottou L, Orr GB, Müller K-R (2012) Efficient backprop. Neural Networks: Tricks of the Trade. Springer, pp 9–48
24. Li H, Luo W, Qiu X, Huang J (2018) Identification of various image operations using residual-based features. IEEE Trans Circ Syst Video Technol 28(1):31–45
25. Li L, Bao J, Zhang T, Yang H, Chen D, Wen F, Guo B (2020) Face x-ray for more general face forgery detection. Proceedings of CVPR
26. Lin M, Chen Q, Yan S (2014) Network in network. Proceedings of ICLR, pp 1–10
27. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. Proceedings of ICCV, pp 3730–3738
28. Mo H, Chen B, Luo W (2018) Fake faces identification via convolutional neural network. Proc. 6th ACM Workshop on Inf. Hid. Multimedia Security, pp 43–47

29. Nhu TD, Na IS, Kim SH (2018) Forensics face detection from GANs using convolutional neural network. Proc. Int. Symp. Inf. Technol. Convergence, pp 376–379
30. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. Proceedings of BMVC, pp 1–12
31. Pumarola A, Agudo A, Martinez AM, Sanfeliu A, Moreno-Noguer F (2018) Ganimation: anatomically-aware facial animation from a single image. Proceedings of ECCV, pp 818s–833
32. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. Proceedings of ICLR, pp 1–16
33. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M FaceForensics++: learning to detect manipulated facial images. [Online]. Available: 1901.08971
34. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M FaceForensics: a large-scale video dataset for forgery detection in human faces. [Online]. Available: 1803.09179
35. Simonyan K, Zisserman A (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. Proceedings of ICLR, pp 1–14
36. Thies J, Zollhöfer M, Nießner M (2019) Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics
37. Thies J, Zollhöfer M, Stamminger M, Theobalt C, Nießner M (2016) Face2face: Real-time face capture and reenactment of RGB videos. Proceedings of CVPR, pp 2387–2395
38. Yu N, Davis L, Fritz M (2019) Attributing fake images to GANs: learning and analyzing GAN fingerprints. In: Proceedings of ICCV
39. Zhou P, Han X, Morariu VI, Davis L (2017) Two-stream neural networks for tampered face detection. Proc. CVPR. Workshops, pp 1831–1839