Localization of Inpainting Forgery With Feature Enhancement Network

Yushu Zhang, *Member, IEEE*, Zhibin Fu, Shuren Qi, Mingfu Xue, *Senior Member, IEEE*, Zhongyun Hua, *Member, IEEE*, and Yong Xiang, *Senior Member, IEEE*

Abstract—Inpainting the given region of an image is a typical requirement in computer vision. Conventional inpainting, through exemplar-based or diffusion-based strategies, can create realistic inpainted images at a very low cost. Also, such easy-to-use manipulation poses new security threats. Therefore, the detection of inpainting has attracted considerable attention from researchers. However, the existing methods are typically not suitable for the general detection of various inpainting algorithms. Motivated by this, in this work, an efficient feature enhancement network is proposed to locate the inpainted regions in the digital image. First, we design an artifact enhancement block to effectively capture the traces left by diffusion or exemplar-based inpainting. Then, the VGGNet is used as a feature extractor to describe advanced and low-resolution features. Finally, to take full advantage of enhanced features, we concatenate the features obtained by the feature extractor and the up-sampling operations. Extensive experimental evaluations, covering benchmarking, ablation, robustness, generalization, and efficiency studies, confirm the usefulness of the proposed method. This is especially true on the conventional inpainting dataset, our method obtains an average F1 score 7.63% higher than the second-best method. Theoretical and numerical analyses support the effectiveness of our feature enhancement network in representing the artifacts in inpainted images, exhibiting better potential for real-world forensics than various state-of-the-art strategies.

Index Terms—Image inpainting, forgery localization, feature enhancement, feature concatenation

1 Introduction

Currently, image processing is becoming easier due to the advanced photo editing software, which allows users to manipulate images without professional knowledge. Unfortunately, attackers might take advantage of the editing tools to maliciously create fake media with the potential in misleading the public. This compels us to develop fake media detection methods that can verify the authenticity and integrity of media.

- *Yushu Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China. E-mail: yushu@nuaa.edu.cn.*
- *Zhibin Fu, Shuren Qi, and Mingfu Xue are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. E-mail: {binzhif, shurenqi, mingfu.xue}@nuaa.edu.cn.*
- *Zhongyun Hua is with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China. E-mail: huazhongyun@hit.edu.cn.*
- *Yong Xiang is with Deakin Blockchain Innovation Laboratory, School of Information Technology, Deakin University, Burwood, VIC 3125, Australia. E-mail: yong.xiang@deakin.edu.au.*

Manuscript received 25 June 2022; revised 21 November 2022; accepted 21 November 2022. Date of publication 28 November 2022; date of current version 13 May 2023.
This work was supported in part by the National Key R&D Program of China under Grant 2021YFB3100400, in part by the National Natural Science Foundation of China under Grant 62072237, in part by the State Key Laboratory of Information Security under Grant 2022-MS-02, and in part by the Basic Research Program of Jiangsu Province under Grant BK20201290.
(Corresponding author: Shuren Qi.)
Recommended for acceptance by P. D'Urso.
Digital Object Identifier no. 10.1109/TBDATA.2022.3225194

As a powerful technique, image inpainting can reconstruct the missing regions in a visually plausible way. It could be grouped into two categories: conventional inpainting [1], [2], [3], [4], [5], [6], [7], [8] and deep learning-based (DL-based) inpainting [9], [10], [11], [12], [13]. The former is to fill the target hole with the appropriate background content of the same image, while the latter can generate realistic visual content through neural networks. Additionally, many conventional inpainting methods that combine exemplar and diffusion-based techniques have been designed [14], [15]. In fact, the development of conventional inpainting algorithms is longer than that based on GAN, and the algorithm implementation is relatively mature. Common image processing software such as Photoshop, GMIC, and OpenCV all involve conventional repair algorithms. Therefore, it is still very practical to detect traditional inpainted images. However, those approaches can also be exploited to generate forged images by adversaries. In practical scenarios for inpainting, the object or key information (such as data, time, or number) can be removed to mislead observers, as shown in Fig. 1. Hence, it is of practical significance to detect forgery regions created via inpainting tools.

Over the past decades, a variety of forgery detection/localization approaches [16], [17], [18], [19] have been proposed. Among them, many methods were built on the analysis of the abnormal features, such as color inconsistencies [16], JPEG compression artifacts [17], color filter array [18], and photo response non-uniformity [19]. However, these methods did not consider the prior knowledge of image inpainting, thus it is difficult to gain a satisfactory performance in the inpainting detection.

In general, image inpainting is performed by exemplar-based or diffusion-based strategy. More specifically,

2332-7790 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

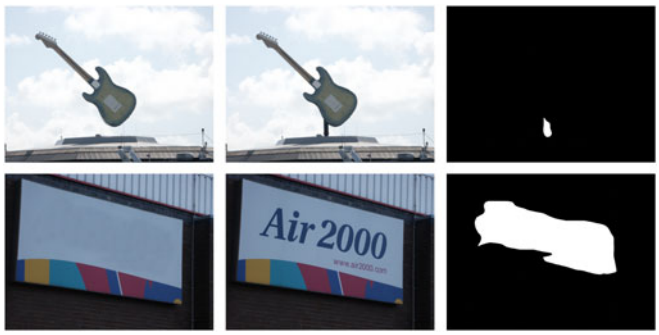


Fig. 1. Two examples of inpainted images: inpainted (left), original (middle), and ground-truth (right).

the exemplar-based approach tries to find the best candidate exemplar for covering the damaged region. As for the diffusion-based approach, it propagates the image content from the boundary to the missing area by modeling the diffusion process. Therefore, the forgery detection for inpainting can be inspired by the principles of these inpainting algorithms. For the detection of exemplar-based inpainting, some existing approaches [20], [21], [22], [23] focus on searching similar blocks within the given image to expose inpainted patches. In the case of detecting diffusion-based inpainting, Li et al. [24] pioneered a localization method based on the local variances of the changes in inpainted and original regions. Since deep learning has demonstrated superior performance in many applications [25], [26], [27], [28], [29], Zhu et al. [30] utilized an encoder-decoder network to localize inpainted regions in an image. As filtering is widely used in various fields of images [31], [32], [33], [34], [35], Li et al. [36] used a high-pass full convolutional network to locate the tampered regions by deep-learning inpainting. Nevertheless, to the best of our knowledge, the *universal detection/localization* for such two types of image inpainting is still lacking.

In this work, we propose an efficient feature enhancement network for conventional inpainting localization. More specifically, it consists of three blocks: artifact enhancement, feature extraction, and forgery output. The first block focuses on exposing the anomalous artifacts of the inpainted region. To this end, we design a high-pass filtering module with four spatial rich model (SRM) kernels [31] and a Laplacian kernel, which captures inconsistencies in the domain of noise residuals. Moreover, we fuse such residual features with color features to capture tampering artifacts. Next, the VGGNet [37] with 13 convolutional layers and 4 max-pooling layers is employed as the second block to effectively extract anomalous features from the inpainted image. Besides, to take further advantage of the features from the enhancement block, we introduce the up-sampling structure of U-Net [38], which combines high and low-resolution features for outputting more precise forgery regions. Finally, extensive experiments exhibit state-of-the-art performance in the localization of conventional inpainting. For this paper, the major contributions are as follows:

- The existing methods are not sufficient to cope with the universal detection of various conventional inpainting methods. To this end, we built a network for the *general localization of conventional image inpainting*.
- We design an *artifact enhancement block* that is capable of capturing inpainting artifacts and providing high-quality features for the feature extractor and forgery output block.
- On the basis of artifact enhancement, we further introduce a *feature fusion strategy* to improve the quality of up-sampling.

This paper is structured as follows. In Section 2, we briefly introduce the development status of conventional inpainting techniques as well as inpainting detection. Section 3 presents our method in detail. In Section 4, we conduct several experiments to evaluate the localization performance of the presented approach. Finally, Section 5 gives a conclusion and highlights the future work.

2 Related Works

2.1 Conventional Inpainting Methods

The conventional inpainting technique aims to recover the damaged region with the spatial information of the undamaged region. It basically includes exemplar-based and diffusion-based, where the exemplar-based inpainting is effective in reconstructing large regions, and the diffusion-based inpainting performs well in achieving local intensity consistent.

Fundamentally, the exemplar-based approach works by searching for the best matching patches in the undamaged region and copying them to the target location. The inpainting algorithm in [2] was designed to fill the missing area by the mean of searching for the undamaged patches with the least mean-squared-error distance. Subsequently, various improved methods of [2] were developed to improve the priority calculation and optimize patch searching. For example, the method in [3] defined a patch priority order based on structural sparsity to speed up calculations. Liu et al. [4] proposed a exemplar-based method by multi-scale graph cuts. To speed up the inpainting operation, the random patch search method was adopted in [6] for finding the best patch. In addition, an inpainting method that adopts a novel non-local texture similarity measure and nonlinear filtering to select several candidate patches was presented in [8].

There are two types of diffusion-based inpainting algorithms. One of them reconstructs the damaged area by minimizing the high-order partial differential equation or variational repair function, and the other propagates the pixel intensity continuously into the damaged area along the isophote direction. Inspired by the ideas of classical fluid mechanics, the diffusion-based method of [1] propagated isophote lines continuously from the exterior into the region to be inpainted. Later, a novel method based on two fourth-order partial differential equations [5] was proposed to repair the image. Li et al. [7] diffused the target region by computing the distance and direction between the damaged pixels and its neighborhood pixels. In summary, these methods ensure the local intensity smoothness and are applicable to the completion of lines, curves, and small areas in the image.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

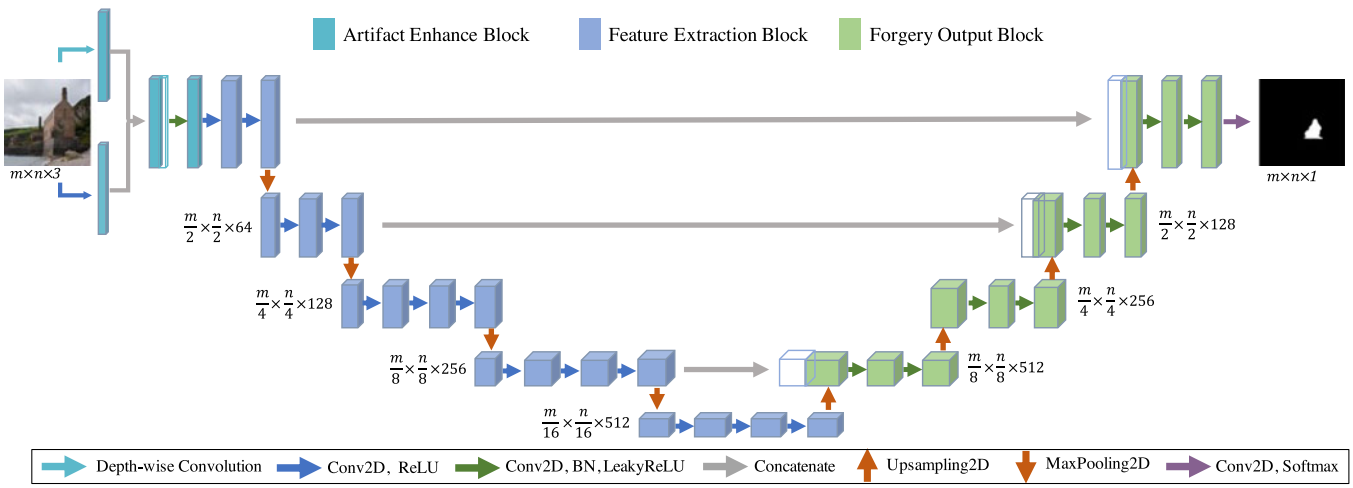


Fig. 2. The architecture of the proposed network. It consists of three different blocks: artifact enhancement block, feature extraction block, and forgery output block. The first block is equipped with 5 filters for guiding the network to learn inpainting artifacts in the frequency domain. Despite this shallow representation, the feature extraction block is then performed for learning more representative features. On the basis of artifact enhancement, the feature fusion strategy is performed in the forgery output block to generate more refined localization results.

2.2 Detection of Conventional Inpainting

Hitherto, there are several research paths in the field of inpainting detection. For the detection of exemplar-based inpainting, many existing techniques are based on the fact that the inpainted patches are copied from the same image. Thus, the common pipeline of those algorithms mainly contains two major processes: suspicious region detection and forgery region identification. In [20], the inpainting process can be briefly explained as follows: i) the zero-connectivity labeling was applied on patch pairs to yield matching degree features for all patches in the suspicious region, ii) the fuzzy memberships of patch matching degrees were computed to describe the uncertainty in detecting tampered region, and iii) the tampered regions were identified by a cut set. Unfortunately, this method requires manually selecting suspicious regions in advance and takes a long time to search similar regions. To overcome these drawbacks, a new method based on multi-region relation was proposed in [21] for identifying tampered regions from suspicious areas. Moreover, the two-stage searching algorithm based on the weight transformation was also applied in [21] to speed up the calculation. This method had only a limited improvement in computational time yet. In [22], the average sum of the absolute difference between the inpainted image and the reserved JPEG compressed image under different quality factors was used to detect the inpainted image. Furthermore, Liang et al. [23] utilized central pixel mapping to accelerate the search of suspicious regions and integrated the greatest zero-connectivity component labeling and fragment splicing detection to predict tampered areas. However, these two methods still exhibit high false alarms. Recently, the approach of [30] constructed a convolutional neural network (CNN) to predict the inpainting probability for each pixel, and hence locate the inpainted patches. Although this approach effectively reduces the false alarm, its performance for the localization of conventional inpainting is unsatisfactory.

As a pioneering attempt for diffusion-based inpainting detection, Li et al. [24] proposed a method by analyzing the local variance of image Laplacian along the isophote direction. This method can quickly find the region of diffusion-based inpainting, but it also suffers from a high false positive rate and is unsuitable for exemplar-based localization.

In practice, an adversary may employ both diffusion-based and exemplar-based algorithms to achieve realistic fake images. Consequently, the above schemes that only consider the detection/localization of a single type cannot achieve satisfactory results in such real-world scenarios. For this reason, we propose an end-to-end strategy for the general localization of diffusion-based and exemplar-based inpainting regions.

3 The Proposed Network

3.1 Network Overview

We propose a network for inpainting localization via a feature enhancement strategy. Our approach is characterized by stronger generality and robustness. Unlike regular CNN-based forgery detection methods taking the tampered images as the input of the feature extractor, our network enhances more subtle manipulation traces within the image before the feature extraction.

Fig. 2 depicts the overview of the proposed network with details about each block. The architecture consists of three different blocks: artifact enhancement block, feature extraction block, and forgery output block. The first block is designed to capture inpainting traces by high-pass filters and a series of convolution operations. In the feature extraction block, the VGGNet [37] is performed to learn more manipulation features. As for the forgery output block, the up-sampling structure in [38] is employed to output more precise inpainted regions. In what follows, we elaborate on the details of our proposed concept blocks. In Table 1, we list core notations for this paper.

3.2 Artifact Enhancement Block

Different from the regular CNN-based forgery detection, which tends to learn content-dependent features, we pay close attention to capturing artifacts of the inpainting

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Notations and Definitions

Notation	Definition
(i,j)/ (u,v)	The spatial/ frequency domain coordinates
x	The input of the network layer
y	The output of the network layer
h	The activation function
w	The weight of the convolution kernel
b	The bias item of the convolution
f	The image function
F	The Fourier coefficient
H, W	The height and width of the image
$\oplus$	The concatenating operation

operation. Therefore, we hope that the proposed method can suppress image content to expose inpainting traces in the first block. Inspired by [36], [39], the inpainting feature enhancement in our work is achieved through the feature fusion of image noise residuals obtained by high-pass filters and RGB features. Besides focusing on the spatial domain, some methods are devoted to analyzing the artifacts of forgery in the frequency domain.

We design a high-pass filtering module with 5 filter kernels in the artifact enhancement block, where 4 SRM kernels are selected from [31], and the last one is Laplacian kernel. See Section 4 for the analysis of filtering kernels. Here, SRM kernels can effectively expose the noise inconsistencies in exemplar-based complex texture regions and the Laplacian kernel has the capability of exposing diffusion-based inpainting noise inconsistencies. A depth-wise convolution with the kernel size of 3 and the stride of 1 is applied to independently perform calculations on each channel of the input layer. In our model, we set the high-pass filter kernel as the depth separable convolutional initial kernel. Hence, the filtering module will perform 5 depth separable convolution operations and each convolution configures a different kernel. We take the RGB channels as the input of the high-pass filtering module to get 15 noise feature maps. At the same time, we perform a regular convolution with 3 filters on the RGB image. Subsequently, the color features and noise residuals are fused to obtain an output of 18 channels. Finally, we configure a 3×3 convolution with a stride of 1 for the fused feature map, generating 32 channels.

In brief, we perform a list of high-pass filtering and convolution operations to gain a result of 32 channels.

3.3 Feature Extraction Block

After the fusion of local noise residuals and color features, the inpainting traces are effectively enhanced, which is beneficial to guide the feature extraction block to accurately identify inpainting regions. To further improve the feature quality, we take VGGNet [37] which has made great success in the large-scale computer vision field as the feature extractor. The VGGNet employs multiple convolutions of smaller kernel sizes (3×3). This can not only reduce the number of parameters but also improve the fitting ability of the network. The calculation process of 3×3 convolution is given below

$$y = h\left(\sum_{m=-1}^{1}\sum_{n=-1}^{1} w_{m,n}x_{i+m,j+n} + b\right), \tag{1}$$

where $x_{i,j}$ stands for the pixel at the coordinate (i,j) of input, and $w_{m,n}$ is the weight of convolution kernel at the coordinate (m,n). Moreover, the max pooling with 2×2 kernel size is calculated using the following formula

$$y_{i,j} = \max\left(x_{i,j}^{l-1}, x_{i,j+1}^{l-1}, x_{i+1,j}^{l-1}, x_{i+1,j+1}^{l-1}\right), \tag{2}$$

where $y_{i,j}^{l}$ represents the value at the coordinate (i,j) of the l-th network layer.

TABLE 2
Specifications of the Feature Extraction Block

Stage	Layer	Kernel	Stride	Input depth	Output depth
1	Convolution	3	1	32	64
	Convolution	3	1	64	64
	Max-pooling	2	2	64	64
2	Convolution	3	1	64	128
	Convolution	3	1	128	128
	Max-pooling	2	2	128	128
3	Convolution	3	1	128	256
	Convolution	3	1	256	256
	Convolution	3	1	256	256
	Max-pooling	2	2	256	256
4	Convolution	3	1	256	512
	Convolution	3	1	512	512
	Convolution	3	1	512	512
	Max-pooling	2	2	512	512
5	Convolution	3	1	512	512
	Convolution	3	1	512	512
	Convolution	3	1	512	512

To learn higher-level representative features, the feature extraction block contains a total of 17 layers: 13 3×3 convolutions and 4 max-pooling operations. Each convolutional layer is followed by a rectified linear unit (ReLU) [40]. The feature extraction block can be divided into five stages. Specifically, the first two stages are both composed of two successive convolutional layers and a maximum pooling layer; the third and fourth stages are both composed of three convolutional layers and a maximum pooling layer; the fifth stage is configured with three consecutive convolutional layers. Additionally, the number of output feature maps in stage 1 is 64; the number of output channels in each stage except the last stage is doubled to a maximum of 512 channels; the last stage has the same number of output channels as stage 4. More detailed information is depicted in Table 2.

3.4 Forgery Output Block

Through feature extraction, we continuously down-sample the input to filter out features that contain redundant information. Since the pooling operations reduce the size of the feature maps to 1/16 of the input image, the feature

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

TABLE 3
Specifications of the Forgery Output Block

Step	Layer	Kernel	Stride	Input depth	Output depth	Concatenate
1	Up-sampling	2		512	512	✓
	Convolution	3	1	1024	512	
	Convolution	3	1	512	512	
2	Up-sampling	2		512	512	
	Convolution	3	1	512	256	
	Convolution	3	1	256	256	
3	Up-sampling	2		256	256	✓
	Convolution	3	1	384	128	
	Convolution	3	1	128	128	
4	Up-sampling	2		128	128	✓
	Convolution	3	1	192	64	
	Convolution	3	1	64	64	
	Convolution	3	1	64	2	

maps need to be restored to the same spatial resolution as the input image for further pixel classification. Here, if our forgery output block is built only on a series of up-sampling layers, the accuracy of the prediction will be restricted by the feature map from the second block. This is because the up-sampling operation can not generate more semantic information and may even lose the details about the inpainted regions.

As we know, the low-level features usually refer to details in the image, such as edge, corner, color, and gradients, which can be obtained by convolutional layer, SIFT [41], or HOG [42]. As for the high-level features, they are built on low-level features to detect objects and shapes in the image. Thus, the fusion of low-level and high-level features can enrich more detailed semantic information of high-level feature maps. It also can take full advantage of the enhanced features to effectively and precisely predict inpainted regions. For this, we introduce the decoder structure of [38], which integrates low-level and high-level features to improve the spatial resolution of feature maps. The forgery output block is shown in Fig. 2. The feature concatenation between the two blocks occurs in steps 1, 3, and 4 from bottom to top. We express this feature concatenation as follows:

$$y^l = {x_1}^{l-1} \oplus {x_2}^{l-1}, \tag{3}$$

where ${x_1}^l$ is the feature maps of the l-th layer in the forgery output block, ${x_2}^l$ denotes the corresponding feature maps of the l-th layer in the feature extraction block. In each step, two successive convolutional layers are employed to extract useful features, since there is a lot of redundant information in low-level feature maps.

After four times of up-sampling processing, an output result with a depth of 2 and the same size as the input image is obtained. Eventually, each pixel of the result is binary classified to identify the inpainted regions by the Softmax function. More detailed settings about up-sampling and convolutional layers in the forgery output block are demonstrated in Table 3. The training procedure of the model is shown in Algorithm 1.

Algorithm 1. The Training Algorithm

Input: Training data D ; training epochs M; learning rates α.
Output: Model f

for $epoch = 1$ *to* M **do**
  for $minibatch(x_i, y_i) \subset D$ **do**
    $i_\psi \leftarrow I(x_i)$ ▷ Artifact Enhancement
    $p_\psi \leftarrow P(x_i)$ ▷ Feature Extraction
    $q_\psi \leftarrow Q(x_i, h_\psi)$ ▷ Forgery Output
    $l_\psi \leftarrow \nabla_\psi[Loss(i_\psi, p_\psi, q_\psi, y_i)]$
  end for
  $k_\psi \leftarrow k_{\psi-1} - \alpha \cdot l_\psi$ ▷Update k_ψ
end for

4 Experimental Evaluation

In this section, we conduct a series of experiments to evaluate the performance of the proposed method. We first create the synthetic datasets for training and testing. Then, the performance of our method and four related inpainting detection methods are compared on the testing datasets. In addition, we evaluate the robustness of the proposed method against JPEG compression, rotation, and scaling.

4.1 Experimental Setup

Training Datasets. To train the model, we randomly select 50000 different images with fixed size 256×256 from the Places [43] database to generate synthetic inpainted images with corresponding ground-truth masks. The Places contains 10 million images, including more than 400 different types of scene environments. We first generate missing regions in these images, where these regions are located in the center of images, and the tampered areas occupied 10% to 12% of the whole image. Considering the randomness of the data, the shapes of the missing regions are random, including rectangles, circles, irregular shapes, etc. Several examples are presented in Fig. 5. Then, the exemplar-based inpainting method [6] is introduced to repair the missing regions for generating the training dataset. The inpainted images are divided into two subsets: 48000 instances with

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

Fig. 3. The sample images for the missing regions (marked in white) with the regular and irregular shapes.

corresponding ground-truth masks are exploited for training our CNN and the left 2000 are served as validation. Additionally, to obtain better results, we perform other image processing operations, which include resizing and flipping. At last, the images and labels are fed to the model for training.

Testing Datasets. In order to test the localization performance of the proposed model, we create additional testing datasets from four databases. We choose 300 images randomly from Places [43], ImageNet [44], and CelebA [45] databases respectively to generate inpainted images. Note that there is no intersection between the testing datasets and training datasets. There are a total of 900 inpainted images with corresponding ground-truth masks in the testing datasets. We crop these images from the center to generate 256×256 images and apply different painting methods to repair these images separately to generate different datasets. To better test the performance, the tampered locations and shapes in the images are random. The testing images possess four different tampering ratios: 5 %, 10 %, 15 %, and 20 %. The following two datasets are used for performance evaluation:

- *Exemplar-Based Inpainting Dataset.* The dataset contains 900 inpainted images with corresponding tampering masks generated by the exemplar-based inpainting algorithm [6].
- *Diffusion-Based Inpainting Dataset.* In this dataset, we apply the diffusion-based inpainting method [1] to generate testing images.

Implementation Details. The proposed network for localization is implemented in the TensorFlow deep learning framework [46]. In all of our experiments, the Adam optimizer [47] with 1×10^{-4} initial learning rate is set to calculate network parameters. The learning rate will decrease every epoch by 8%. Moreover, in all convolutional layers, the kernel weights are initialized with variance scaling initializer. We train the whole network for 30 epochs and set the batch size as 8 to improve the training speed. The model achieves its optimal localization when the verification loss value converges to the minimum. After training, we save the best parameters of the model for testing. We carry out all the experiments on an Nvidia RTX 3080 GPU server.

Performance Metrics. Since image forgery localization is a pixel-level binary classification problem, we evaluated the performance of the proposed and existing methods by using Intersection over Union (IoU), F1-score, precision, and recall.

Comparative Methods. We compared the proposed method with four existing inpainting localization methods.

- LDI [24]. A method was proposed to discriminate the tampered regions altered by diffusion-based inpainting techniques.
- Patch-CNN [30]. A method was proposed to locate the inpainted regions altered by exemplar-based inpainting techniques.
- HP-FCN [36]. A method was proposed for the localization of deep inpainting techniques by using a high-pass fully convolutional network.
- IML-PS [48]. A method was proposed to accomplish the task of tampering localization by focusing on the detection of commonly used editing tools and operations in Photoshop.

4.2 Study of Inpainted Regions

In this part, we discuss the differences between inpainted and pristine regions, we convert the gray-scale image into the frequency domain through a fast Fourier transform

$$F(u,v)=\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}f(i,j)e^{-2j\pi(ui/H+vj/W)}, \tag{4}$$

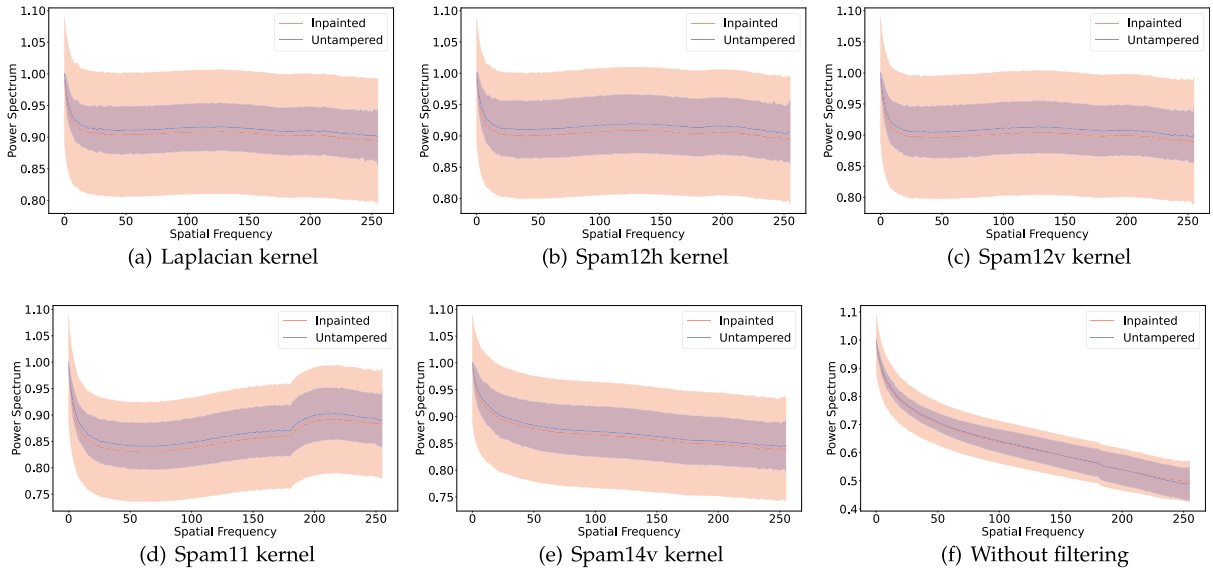
where $f(i,j)$ denotes the value in spatial domain with coordinate (i,j), $i\in[0,H-1]$ and $j\in[0,W-1]$, and $F(u,v)$ refers to the Fourier coefficient of $f(i,j)$. Then, we calculate the 2D power spectrum of the frequency coefficient

$$P(u,v)=|F(u,v)|^2. \tag{5}$$

At last, we derive the azimuthally averaged 1D power spectrum [49] from the Fourier power spectrum for the convenience of observation. The radial average power spectrum (RAPS) is a direction-independent average power, which provides a very convenient method to interpret the energy spectrum.

To analyze the difference between the inpainted image and the original image, we randomly select 250 images from the Places database [43] to create inpainted images. These images were operated with the approaches [6] and [1], respectively, where the tampered areas are irregular and the tampering ratio is 20%. Meanwhile, the corresponding 250 authentic images are exploited for comparison. We calculate the RAPS of the two groups of images respectively and then compute the statistic measures (mean and standard deviation) of RAPS. We begin our analysis from Figs. 4 and 5. In Figs. 4f and 5f, the statistics of RAPS for inpainted and pristine images without high-pass filtering are shown, and the others (Figs. 3a–e and 4a-e) show the statistics of RAPS for the same images with high-pass filtering. Notice that, the color area in the figure refers to the standard deviation, which reflects the degree of dispersion between individuals in the group, and the line in the color area represents the mean value. We can observe that the inpainted image and the original image have similar RAPS when there is no high-pass filtering, while the two groups of images with filtering operation have different RAPS. We selected 4 SRM kernels from [31], and 1 Laplacian kernel according to the magnitude of statistical difference in RAPS. From Figs. 4a and 5a, it can be found that the images processed by the Laplacian kernel are significantly

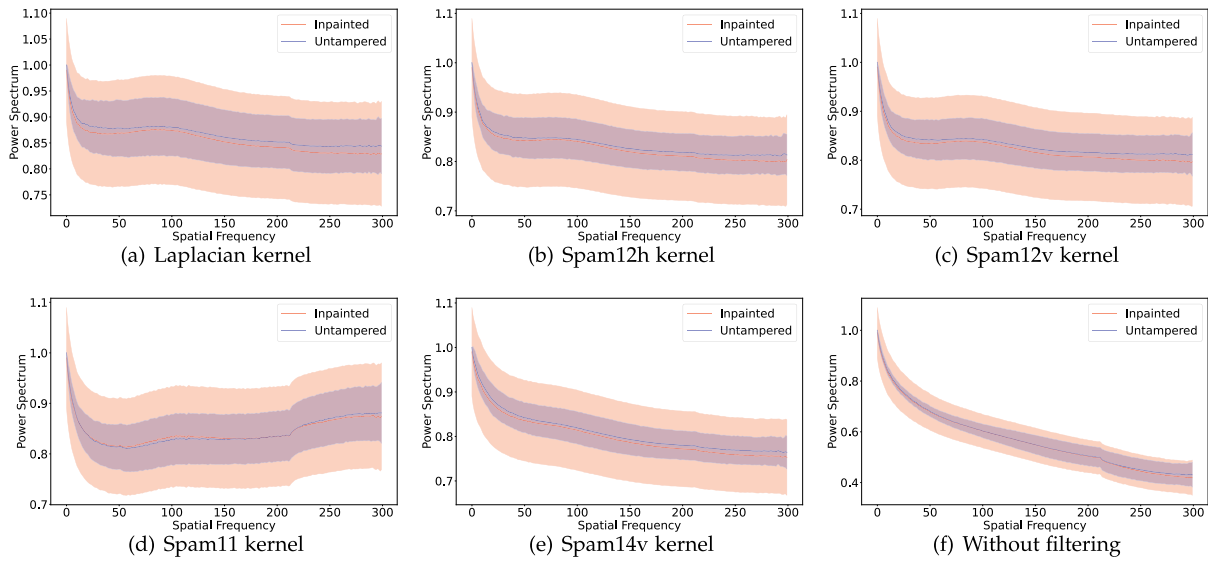
Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.



(a) Laplacian kernel (b) Spam12h kernel (c) Spam12v kernel

(d) Spam11 kernel (e) Spam14v kernel (f) Without filtering

Fig. 4. The statistics of RAPS for inpainted/untampered images with/without high-pass filtering of different kernels, where the inpainted images are generated by the exemplar-based inpainting method [6].



(a) Laplacian kernel (b) Spam12h kernel (c) Spam12v kernel

(d) Spam11 kernel (e) Spam14v kernel (f) Without filtering

Fig. 5. The statistics of RAPS for inpainted/untampered images with/without high-pass filtering of different kernels, where the inpainted images are generated by the diffusion-based inpainting method [1].

different in RAPS. This shows that the Laplacian kernel has the ability to expose exemplar-based inconsistencies and diffusion-based inpainting inconsistencies. At the same time, it suggests that high-pass filtering is helpful to expose inpainting artifacts. The kernels of high-pass filtering are shown in Fig. 6.

Furthermore, we also experimentally validate the design of kernels. It can be seen from Table 4 that the greater the difference between the kernels in RAPS, the more helpful it is to improve the model performance. The performance improvement of the fusion of five kernels is the best. This may be due to the fusion of kernels increasing the nonlinear

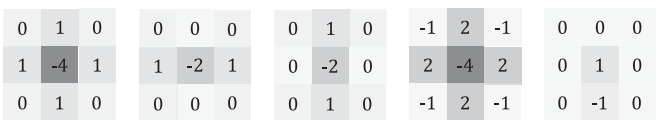


Fig. 6. The five filter kernels used to enhance inpainting features. There are the Laplacian, Spam12h, Spam12v, Spam11, and Spam14v kernel from left to right.

TABLE 4
F1-Score (%) for Different Kernel Settings

Kernel Settings	Exemplar-based dataset	Diffusion-based dataset
	F1	F1
Laplacian	93.84	86.08
Spam12h	94.56	83.75
Spam12v	94.92	84.79
Spam11	95.21	82.67
Spam14v	92.95	81.78
No kernel	91.92	80.07
Fusion	96.95	**89.49**
Learnable kernels	**97.92**	85.27

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

TABLE 5
Localization Results (%) for the Methods With/Without Artifact Enhancement Block

Methods	With enhancement				Without enhancement			
	F1	IoU	Precision	Recall	F1	IoU	Precision	Recall
Patch-CNN [30]	88.28	80.13	80.82	**98.58**	86.23	77.08	78.10	**97.90**
HP-FCN [36]	93.86	89.62	93.17	94.80	93.14	88.02	92.66	93.94
IML-PS [48]	96.02	90.55	96.01	96.17	95.24	91.50	94.69	96.06
Proposed	**96.95**	**94.40**	**96.79**	97.20	**95.87**	**92.35**	**95.42**	96.33

fitting power of the model. However, increasing the number of kernels will also greatly increase the training time. Thoughtfully, we finally chose these filter kernels to train our model. In addition, we also explore the impact of trainable filter kernels on the performance of model. From Table 4, one can see that although learnable kernels can gain model performance on the exemplar-based dataset, they are not satisfactory on the diffusion-based dataset. Since we pay more attention to the generalization ability of the model to unknown inpainting algorithms, the filter kernels are set as non-learnable.

4.3 Ablation Analysis

The key design of the network architecture has a great influence on the overall performance. Before conducting comparative experiments, we perform an ablation analysis to validate the effectiveness of our design choices. We examine the impact of different designs through three sets of experiments, i.e., i) the use of artifact enhancement block, ii) the type of feature extractor, iii) the concatenation of features from various steps, and iv) the training loss functions. In these experiments, we utilize the same datasets for training and testing. The average F1-score, IoU, recall, and precision are reported over the testing datasets.

Effect of Artifact Enhancement Block. In this part, we assess the performance gains achieved through using the artifact enhancement block as the first block. To this goal, we first remove the artifact enhancement block from our model and then retrain the model. Experimental results obtained on the testing instances are shown in Table 5. It can be found that the performance of our model without enhanced artifacts decreases compared with the model with enhanced artifacts. Obviously, the artifact enhancement block can effectively improve the performance of our model. Moreover, in order to verify whether this block works for another model, we adopt it to the models in Patch-CNN, HP-FCN, and IML-PS. From Table 5, one can know that the designed artifact enhancement block can effectively improve the prediction performance of all methods.

TABLE 6
Localization Results (%) for Different Feature Extraction Networks

Feature extractor	F1	IoU	Precision	Recall
MobileNets [51]	90.16	84.65	92.77	90.56
ResNet [50]	94.96	90.74	94.24	95.83
VGGNet [37]	**96.95**	**94.40**	**96.79**	**97.20**

Feature Extractor. In this subsection, we discuss the impact of three different backbone networks, ResNet [50], MobileNets [51], and VGGNet [37], as feature extractor. To this end, we remove the fully connected layers of the three networks and then fine-tune these models to configure the artifact enhancement and forgery output blocks. From the results in Table 6, we can observe that the performance of VGGNet performs better than others.

Feature Concatenation. The forgery output block fuses the high-level feature maps with the low-level feature maps obtained by the feature extractor. We now explore the effect of different concatenate layers on the network performance. Note that the feature concatenation is performed after up-sampling, and the forgery output block has a total of 4 steps for up-sampling and feature concatenation. Table 7 illustrates the experimental results of feature fusion at different stages of the forgery output block. The results suggest that the model achieves the best performance when the feature fusion occurs in the first, third, and fourth steps. The artifact enhancement block can provide more representative features for the feature extractor and forgery output.

Loss Functions. The loss function can guide the network to optimize the parameters. Here, we investigate the impact of different loss functions on model performance. From the results of Table 8, we can find that the standard cross entropy loss achieves the best F1-score, precision, and IoU. The weighted cross entropy loss obtains the

TABLE 7
Localization Results (%) for Feature Concatenation in Different Steps

Step 1	Step 2	Step 3	Step 4	F1	IoU	Precision	Recall
				94.87	92.04	95.77	94.51
✓				94.01	91.65	94.07	94.82
	✓			95.57	92.26	96.07	95.49
		✓		94.90	91.81	94.86	94.56
			✓	96.01	93.75	95.84	96.85
✓	✓			95.57	92.05	96.83	94.60
✓	✓	✓		96.24	93.19	96.26	96.37
✓	✓		✓	95.82	92.50	97.22	94.65
✓	✓	✓	✓	95.84	92.56	96.73	95.23
✓		✓		95.78	92.37	95.80	96.21
✓		✓	✓	**96.95**	**94.40**	**96.79**	**97.20**
✓			✓	95.62	92.33	95.57	95.90
	✓	✓		95.74	92.38	95.68	96.03
	✓	✓	✓	95.53	92.06	95.80	95.59
	✓		✓	95.98	92.70	95.37	96.74
		✓	✓	96.12	93.47	96.04	96.85

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

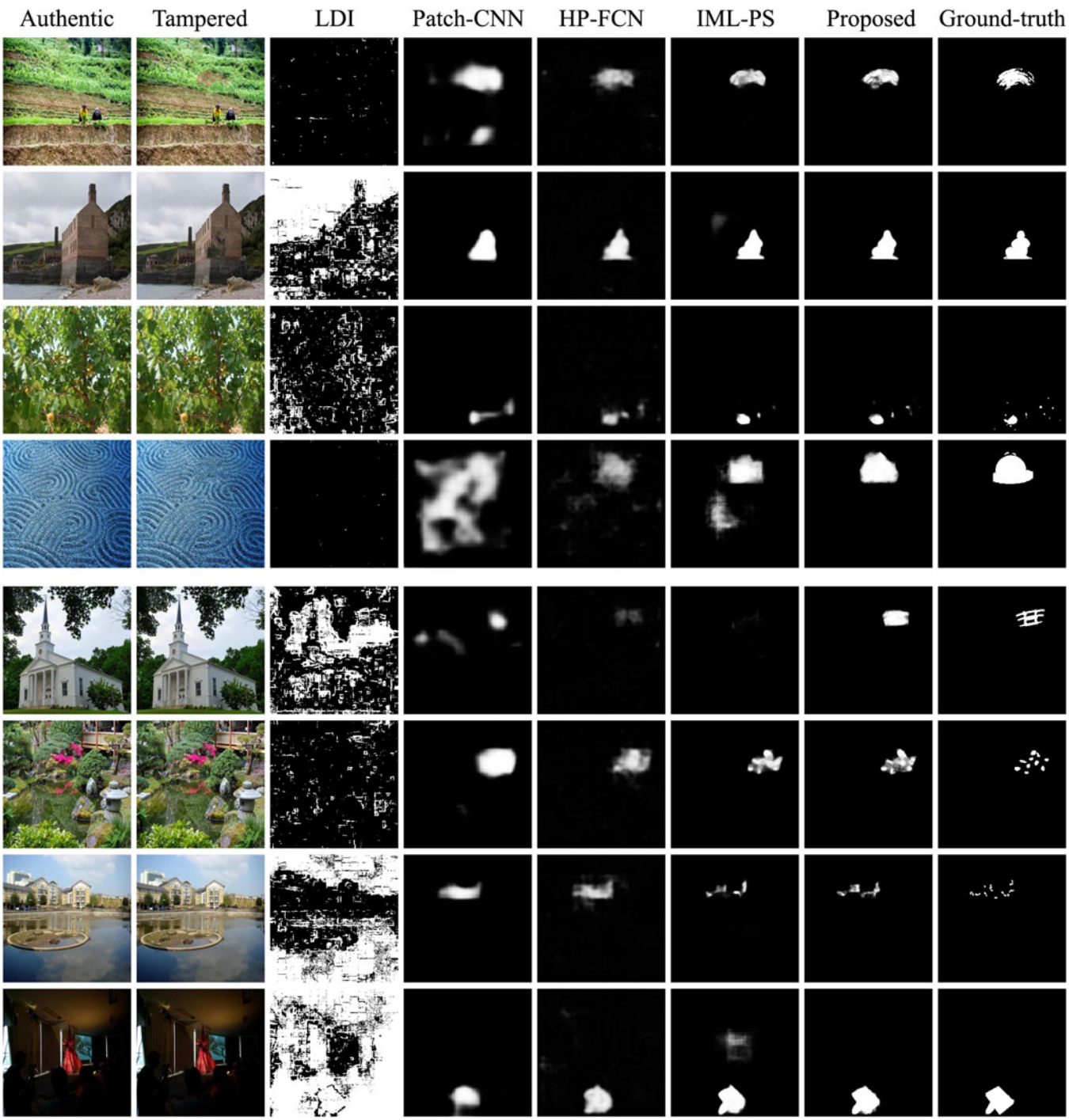


Fig. 7. Examples from the testing datasets, in which the tampered images in row #1–#4 are operated by the exemplar-based inpainting method [6] and the tampered images in row #5–#8 are operated by the diffusion-based inpainting method [1].

highest recall but gets lower precision and IoU. The focal loss achieves slightly lower performance scores than the standard cross entropy loss. Therefore, training the network with the Focal loss can achieve good localization performance overall.

4.4 Performance for Exemplar-Based Inpainting

To be fair, we retrain the models with the same training datasets. The default parameter values provided in Patch-CNN, HP-FCN, and IML-PS are used to train the model. From Table 9, one can observe that the F1-scores of LDI, Patch-CNN, HP-FCN, IML-PS, and our method are 13.71%, 86.23%, 93.14%, 95.24%, and 96.95%, respectively. Apparently, LDI, which was designed to locate the tampered regions of diffusion-based inpainting, failed for localization of exemplar-based inpainting. This phenomenon could be attributed to the fact that the exemplar-based inpainting finds similar pixels in the image to fill the missing area, which is different from the diffusion-based algorithm. Although the F1-score (95.24%) achieved by IML-PS is 1.71% lower than our approach, the performance of IML-PS

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

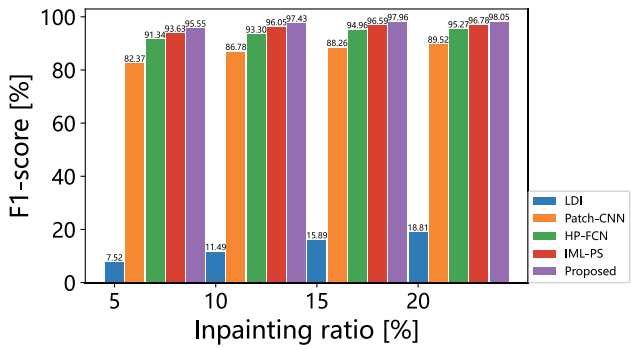


Fig. 8. The F1-scores for various inpainting ratios on the exemplar-based inpainting dataset.

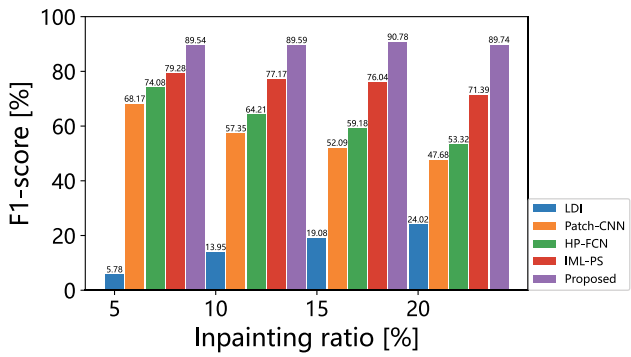


Fig. 9. The F1-scores of different methods for various inpainting ratios on the diffusion-based inpainting dataset.

subjects to a large decline on unknown inpainting datasets (see Sections 4.5 and 4.6). It can be concluded that the proposed model outperforms other comparative methods overall.

Fig. 7 exhibits several localization examples of different approaches. Intuitively, the localization results of Patch-CNN and HP-FCN are relatively rough and blurred, which cannot fulfill the accuracy requirements. The possible reason is that these methods perform up-sampling operations in the networks directly after the feature extraction, resulting in the loss of too many details. Furthermore, Patch-CNN, HP-FCN, and IML-PS tend to misjudge unmodified pixels as tampered pixels. This is probably due to the lack of an effective artifact enhancement module in their network, which makes the network unable to accurately identify the tampered pixels. Compared with these models, the proposed method can effectively discriminate the tampered regions and generate more precise localization results.

In addition, we evaluate the performance on inpainted images with tampering rates of 5%, 10%, 15%, and 20% respectively. From Fig. 8, we can observe that the average F1-scores of the proposed method achieve the best. As for the LDI, the results are still unavailable. With the reduction of the inpainted region, the proposed method has the lowest F1-score of 95.55%, which is 1.92% higher than IML-PS. The possible reason is that our network fuses the features of the feature extractor during up-sampling, which can add more details about the inpainted regions. Therefore, our network still achieves impressive and stable results concerning different inpainted areas.

TABLE 8
Localization Results (%) for Different Loss Functions

	F1	IoU	Precision	Recall
Focal loss	96.50	93.63	96.28	96.85
Standard cross entropy loss	**96.95**	**94.40**	**96.79**	97.20
Weighted cross entropy loss	95.36	91.63	92.16	**99.25**

TABLE 9
Localization Results (%) of Different Methods on the Exemplar-Based Inpainting Dataset

	F1	IoU	Precision	Recall
LDI [24]	13.71	11.92	12.40	22.20
Patch-CNN [30]	86.23	77.08	78.10	**97.90**
HP-FCN [36]	93.14	88.02	92.66	93.94
IML-PS [48]	95.24	91.50	94.69	96.06
Proposed	**96.95**	**94.40**	**96.79**	97.20

TABLE 10
Localization Results (%) of Different Methods on the Diffusion-Based Inpainting Dataset

	F1	IoU	Precision	Recall
LDI [24]	22.51	13.55	14.57	70.89
Patch-CNN [30]	55.78	42.75	66.32	59.62
HP-FCN [36]	62.93	79.44	83.88	59.44
IML-PS [48]	75.94	67.94	87.59	74.19
Proposed	**89.49**	**84.04**	**92.37**	**90.18**

4.5 Performance for Diffusion-Based Inpainting

In previous experiments, we take the exemplar-based inpainting method as an example for evaluations. In this experiment, we use the diffusion-based inpainting algorithm [1] to create the testing datasets for assessing the general detection ability against traditional inpainting forgery. Although the algorithm [1] was released earlier, it has been included in the OpenCV library as a default inpainting algorithm. The details of data generation are the same as in previous experiments. We also conduct comparative experiments with four competing methods [24], [30], [36], [48]. From Table 10, one can note that all methods are capable of identifying the diffusion-based inpainted regions. The proposed approach achieves the best F1-score, IoU, precision, and recall, which are 13.55%, 16.1%, 4.78%, and 15.99% higher than the second-best one (IML-PS), respectively. The LDI performed the worst, with an average score of only 14.54%. This is probably because the design of LDI only considers uncompressed tampered images, while most of the testing images are JPEG compressed, resulting in poor performance of LDI in the detection of compressed inpainted images. Additionally, Fig. 9 suggests that as the inpainting ratio increases from 5% to 20%, the performance of our method does not degrade and remains optimal, while the performance of other DL-based methods degrades significantly. This is due to the poor generalization ability of these methods for unlearned inpainting methods. Consequently, the proposed method can also achieve favorable performance on tampered images based on diffusion inpainting.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

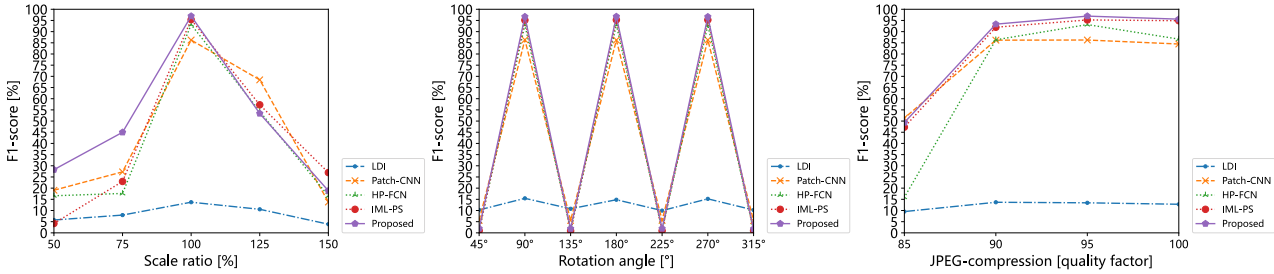


Fig. 10. The F1-score curves for different algorithms against JPEG compression, rotation, and scaling on the exemplar-based inpainting dataset.

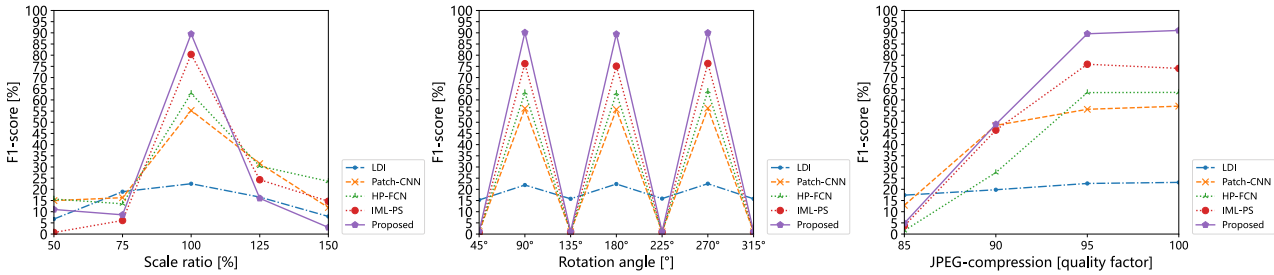


Fig. 11. The F1-score curves for different algorithms against JPEG compression, rotation, and scaling on the diffusion-based inpainting dataset.

4.6 Evaluation on Robustness

In this experiment, we study the robustness of our proposed method against JPEG compression, rotation, and scaling attacks. The corresponding post-processing operations are performed on the inpainted images to generate testing instances, and the previously trained models are employed to perform a robustness evaluation. In addition, we also assess the performance of the proposed method on unknown painting algorithms.

Scaling. To evaluate the resistance of the five methods against scaling, we carry out a scaling operation over the testing instances. The pixel area relationship is applied for sampling to enlarge the original resolution, and the cubic interpolation in the 4×4 pixel neighborhood is applied to shrink the resolution. Each image is scaled with a ratio from 50% to 150% by a step size of 25%. The average F1-scores obtained on the scaled instances are reported in Figs. 10 and 11. It can be discovered that the performance of our method obtains the best at scaling ratios of 50%, 75%, and 100% on the exemplar-based inpainting dataset. When the scaling ratio = 125% and 150%, the performance degrades severely. Furthermore, our method performs poorly on scaling robustness tests in diffusion-based inpainted images. In future work, we will improve the robustness of the method against scaling processing.

Rotation. Here, we assess the performance of the five methods on rotated images. For that, we rotate the images by 45°, 90°, 135°, 180°, 225°, 270°, and 315° respectively. Figs. 10 and 11 demonstrate that all solutions are robust against rotation attacks when rotation angle = 90°, 180°, and 270°. The proposed method still achieves the best performance on both inpainted datasets. In other cases, all methods except LDI fail to detect the rotated image. This is because other rotations involve more challenging re-interpolation and re-quantization, leading to the failure of most methods. In future work, we will try the data augmentation to effectively mitigate this situation.

JPEG compression. As a lossy compression, JPEG compression is one of the most common attacks in image detection. Note that the testing images are all in JPEG format. In this experiment, we set the quality factor (QF) range from 85 to 100 by a step size of 5. It can be observed from Fig. 10 that the localization performance will decline as the QF decreases. When the QF drops from 100 to 90, the average F1-score of our method drops by about 4% and is still the best. However, as the QF drops to 85, the performance drops a lot. The reason may be that the decrease of the QF leads to the reduction of the high-frequency components of the image, which makes it difficult for high-pass filtering to capture tampering artifacts. Moreover, from Fig. 11 we know that the proposed method performs best when QF = 100 and 95, but degrades a lot at QF = 90 and 85 on the diffusion-based images. It is worth further optimization in the future for this scenario.

TABLE 11
Localization Results (%) of Different Methods on the Photoshop Inpainting and DL-Based Inpainting Datasets

Methods	Photoshop Inpainting				DL-based Inpainting [11]				DL-based Inpainting [13]			
	F1	IoU	Precision	Recall	F1	IoU	Precision	Recall	F1	IoU	Precision	Recall
LDI [24]	16.60	8.97	33.35	12.18	17.64	**10.12**	13.46	**29.11**	**9.14**	**4.46**	**6.57**	**16.69**
Patch-CNN [30]	35.16	20.00	46.80	29.46	12.26	5.38	14.29	10.67	0.23	0.12	1.75	0.13
HP-FCN [36]	41.16	27.20	38.87	46.44	**17.89**	7.17	13.71	26.37	0.00	0.00	0.00	0.00
IML-PS [48]	15.74	9.47	37.11	10.20	2.47	1.26	1.43	11.15	0.01	0.07	1.78	0.07
Proposed	**51.74**	**39.69**	**58.61**	**56.54**	15.65	7.33	**19.53**	13.55	0.00	0.00	2.15	0.00

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

Performance on unknown inpainting algorithms. Considering that the tampered images may be generated by unknown inpainting approaches in practice, we utilize the Photoshop inpainting tool and DL-based inpainting methods [11], [13] to yield the inpainted images, where the repair tool of Photoshop is composed of a conventional inpainting algorithm and deep learning inpainting algorithms. From Table 11 we can find that the proposed method achieves the best performance on the Photoshop inpainting dataset. This means that our model has better generalization performance. Moreover, all methods fail on the DL-based inpainting dataset. This may be because the inpainting principles of DL-based inpainting methods differ significantly from those of conventional inpainting, resulting in different data distributions in the generated images.

5 Conclusion

In this work, a novel feature enhancement network has been presented for the localization of conventional inpainted images. Our method has the distinctive ability to efficiently locate both exemplar-based and diffusion-based inpainted regions. Considering that the inpainting operation inevitably leaves artifacts in the image, we designed an artifact enhancement block to capture the inpainting traces. Additionally, we found that the enhanced features can be further utilized through the concatenating operation between the feature extractor and forgery output block. Our model benefits from the artifact enhancement block and feature connection, which guide the network to learn more about inpainting features. Through a series of experiments, we evaluated the localization ability of the proposed method for conventional inpainting. The results demonstrate that the proposed method has a 1.71% higher F1-score on exemplar-based inpainting dataset and 13.55% higher F1-score on diffusion-based inpainting images than the second-best algorithm (IML-PS). Besides, the proposed method obtained an F1-score 10.58% higher than IML-PS. This shows that the proposed method has good detection capability for unknown repair algorithms.

In the future, we will further boost the robustness against challenging post-processing, especially against re-interpolation and weighting. We also intend to study a general feature enhancement block for capturing the traces of various tampering operations.

References

[1] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, pp. I–I.

[2] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[3] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153–1165, May 2010.

[4] Y. Liu and V. Caselles, "Exemplar-based image inpainting using multiscale graph cuts," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1699–1711, May 2013.

[5] P. Li, S.-J. Li, Z.-A. Yao, and Z.-J. Zhang, "Two anisotropic fourth-order partial differential equations for image inpainting," *IET Image Process.*, vol. 7, no. 3, pp. 260–269, 2013.

[6] J. Herling and W. Broll, "High-quality real-time video inpainting with PixMix," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 6, pp. 866–879, Jun. 2014.

[7] K. Li, Y. Wei, Z. Yang, and W. Wei, "Image inpainting algorithm based on TV model and evolutionary algorithm," *Soft Comput.*, vol. 20, no. 3, pp. 885–893, 2016.

[8] D. Ding, S. Ram, and J. J. Rodríguez, "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1705–1719, Apr. 2019.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, 2017.

[11] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4471–4480.

[12] P. Li and Y. Chen, "Research into an image inpainting algorithm via multilevel attention progression mechanism," *Math. Problems Eng.*, vol. 2022, pp. 1–12, 2022.

[13] Q. Dong, C. Cao, and Y. Fu, "Incremental transformer structure enhanced image inpainting with masking positional encoding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 348–11 358.

[14] Y. Wei and S. Liu, "Domain-based structure-aware image inpainting," *Signal, Image Video Process.*, vol. 10, no. 5, pp. 911–919, 2016.

[15] F. Yao, "Damaged region filling by improved criminisi image inpainting algorithm for thangka," *Cluster Comput.*, vol. 22, no. 6, pp. 13 683–13 691, 2019.

[16] X. Wu and Z. Fang, "Image splicing detection using illuminant color inconsistency," in *Proc. 3rd Int. Conf. Multimedia Inf. Netw. Secur.*, 2011, pp. 600–603.

[17] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, Apr. 2012.

[18] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[19] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 809–824, Apr. 2017.

[20] Q. Wu, S.-J. Sun, W. Zhu, G.-H. Li, and D. Tu, "Detection of digital doctoring in exemplar-based inpainted images," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2008, pp. 1222–1226.

[21] I.-C. Chang, J. C. Yu, and C.-C. Chang, "A forgery detection algorithm for exemplar-based inpainting images using multi-region relation," *Image Vis. Comput.*, vol. 31, no. 1, pp. 57–71, 2013.

[22] Y. Q. Zhao, M. Liao, F. Y. Shih, and Y. Q. Shi, "Tampered region detection of inpainting JPEG images," *Optik*, vol. 124, no. 16, pp. 2487–2492, 2013.

[23] Z. Liang, G. Yang, X. Ding, and L. Li, "An efficient forgery detection algorithm for object removal by exemplar-based image inpainting," *J. Vis. Commun. Image Representation*, vol. 30, pp. 75–85, 2015.

[24] H. Li, W. Luo, and J. Huang, "Localization of diffusion-based inpainting in digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 12, pp. 3050–3064, Dec. 2017.

[25] T. Song, W. Zheng, P. Song, and Z. Cui, "Eeg emotion recognition using dynamical graph convolutional neural networks," *IEEE Trans. Affective Comput.*, vol. 11, no. 3, pp. 532–541, Jul./Sep. 2020.

[26] B. Mei, Y. Xiao, R. Li, H. Li, X. Cheng, and Y. Sun, "Image and attribute based convolutional neural network inference attacks in social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 869–879, Apr./Jun. 2020.

[27] Q. Yan et al., "Covid-19 chest ct image segmentation network by multi-scale fusion and enhancement operations," *IEEE Trans. Big Data*, vol. 7, no. 1, pp. 13–24, Mar. 2021.

[28] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep feature learning for medical image analysis with convolutional autoencoder neural network," *IEEE Trans. Big Data*, vol. 7, no. 4, pp. 750–758, Oct. 2021.

[29] Y. Yu, V. O. K. Li, and J. C. K. Lam, "Missing air pollution data recovery based on long-short term context encoder," *IEEE Trans. Big Data*, vol. 8, no. 3, pp. 711–722, Jun. 2022.

[30] X. Zhu, Y. Qian, X. Zhao, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Process.: Image Commun.*, vol. 67, pp. 90–99, 2018.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.

[31] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, 2012.
[32] Y. Wu, W. AbdAlmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9543–9552.
[33] R. Xia, Y. Chen, and B. Ren, "Improved anti-occlusion object tracking algorithm using unscented rauch-tung-striebel smoother and kernel correlation filter," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6008–6018, 2022.
[34] J. Zhang, W. Feng, T. Yuan, J. Wang, and A. K. Sangaiah, "Scstcf: Spatial-channel selection and temporal regularized correlation filters for visual tracking," *Appl. Soft Comput.*, vol. 118, 2022, Art. no. 108485.
[35] J. Zhang, J. Sun, J. Wang, Z. Li, and X. Chen, "An object tracking framework with recapture based on correlation filters and siamese networks," *Comput. Elect. Eng.*, vol. 98, 2022, Art. no. 107730.
[36] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8301–8310.
[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
[39] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1831–1839.
[40] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
[43] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.
[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[45] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.
[46] M. Abadi et al., "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operat. Syst. Des. Implementation*, 2016, pp. 265–283.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[48] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang, "Image tampering localization using a dense fully convolutional network," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2986–2999, Apr. 2021.
[49] R. Durall, M. Keuper, F.-J. Pfreundt, and J. Keuper, "Unmasking deepfakes with simple features," 2019, *arXiv:1911.00686*.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[51] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

Yushu Zhang (Member, IEEE) received the PhD degree in computer science and technology from the College of Computer Science, Chongqing University, China, Dec. 2014. He held various research positions with Southwest University, City University of Hong Kong, University of Macau, and Deakin University. He is currently a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His current research interests include multimedia security and artificial intelligence, and blockchain. He has published more than 200 refereed journal articles and conference papers in these areas. He is currently an associate editor of *Information Sciences and Signal Processing*.

Zhibin Fu received the BE degree from Fujian Normal University, Fuzhou, China, in 2019. He is currently working toward the ME degree in computer science with Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include multimedia forensics/security and machine learning.

Shuren Qi received the BA and ME degrees from Liaoning Normal University, Dalian, China, in 2017 and 2020, respectively. He is currently working toward the PhD degree in computer science with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include invariant feature extraction and visual signal representation with applications in robust pattern recognition and multimedia forensics/security.

Mingfu Xue (Senior Member, IEEE) received the PhD degree in Information and communication engineering from Southeast University, Nanjing, China, in 2014. From July 2011 to July 2012, he is a visiting PhD student in Nanyang Technological University, Singapore. He is currently an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing. He has been a technical program committee member for more than 20 international conferences. He has been the principal investigator of 11 research projects and participated in 4 other projects. He won the Best Paper Award in ICCCS2015. His research interests include artificial intelligence security, secure and private machine learning systems, and hardware security.

Zhongyun Hua (Member, IEEE) received the BS degree from Chongqing University, Chongqing, China, in 2011, and the MS and PhD degrees in software engineering from the University of Macau, Macau, China, in 2013 and 2016, respectively. He is currently an associate professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Shenzhen, China. His research interests include chaotic system, chaos-based applications, and multimedia security. He has published more than fifty papers on the subject, receiving more than 2800 citations. He currently serves as an associate editor for *International Journal of Bifurcation and Chaos*.

Yong Xiang (Senior Member, IEEE) received the PhD degree in electrical and electronic engineering from the University of Melbourne, Australia. He is a professor with the School of Information Technology, Deakin University, Australia. His research interests include information security and privacy, signal and image processing, data analytics and machine learning, Internet of Things, and blockchain. He has published 6 monographs, more than 180 refereed journal articles, and numerous conference papers in these areas. He is the senior area editor of *IEEE Signal Processing Letters* and the associate editor of *IEEE Communications Surveys and Tutorials*. He was the associate editor of *IEEE Signal Processing Letters* and *IEEE Access*, and the guest editor of *IEEE Transactions on Industrial Informatics* and *IEEE Multimedia*. He has served as honorary chair, general chair, program chair, TPC chair, symposium chair and track chair for many conferences, and was invited to give keynotes with a number of international conferences.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on October 10,2023 at 14:21:09 UTC from IEEE Xplore. Restrictions apply.