



A transformer–CNN for deep image inpainting forensics

Xinshan Zhu^{1,2} · Junyan Lu¹ · Honghao Ren¹ · Hongquan Wang¹ · Biao Sun¹

Accepted: 7 July 2022 / Published online: 4 August 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

As an advanced image editing technology, image inpainting leaves very weak traces in the tampered image, causing serious security issues, particularly those based on deep learning. In this paper, we propose the global–local feature fusion network (GLFFNet) to locate the image regions tampered by inpainting based on deep learning. GLFFNet consists of a two-stream encoder and a decoder. In the two-stream encoder, a spatial self-attention stream (SSAS) and a noise feature extraction stream (NFES) are designed. By a transformer network, the SSAS extracts global features regarding deep inpainting manipulations. The NFES is constructed by the residual blocks, which are used to learn manipulation features from noise maps produced by filtering the input image. Through a feature fusion layer, the features output by the encoder is fused and then fed into the decoder, where the up-sampling and convolutional operations are employed to derive the confidential map for inpainting manipulation. The proposed network is trained by the designed two-stage loss function. Experimental results show that GLFFNet achieves a high location accuracy for deep inpainting manipulations and effectively resists JPEG compression and additive noise attacks.

Keywords Forensics · Inpainting · Transformer · Convolutional neural networks

1 Introduction

Digital images, as the primary carriers of information, are becoming increasingly important. However, with the popularization of image acquisition equipment and the rapid development of image editing software, the proliferation of digital image forgeries in recent years reduces the credibility of images, which has tremendously negative impacts on society and individuals. Therefore, image forensics technologies have received increasing attention. Various forensics

schemes have been proposed to detect common image processing operations [1–3] and malicious tampering operations [4–6].

Image inpainting is a technique used to repair damaged or missing regions based on the known content of the input image in a visually plausible manner. Conventional image inpainting approaches can mainly be split into two categories: diffusion-based [7–10] and patch-based [11–14]. Conventional image inpainting methods can achieve good results when the missing regions to be inpainted are small and when the image structure and texture are relatively simple. However, they fail to fill the missing regions with consistent and reasonable content, due to the lack of understanding and perception regarding the high-level image semantics. Recently, an increasing number of researchers are attempting to use deep learning-based methods to obtain higher-quality inpainting results. Many deep image inpainting methods have been proposed, including generative adversarial network (GAN)-based methods [15–18], convolutional neural networks (CNNs)-based methods [19,20], and transformer-based methods [21].

To detect image inpainting, the conventional methods depend on hand-crafted features, e.g., the features based on the similarities between image patches [22–24]. Such meth-

✉ Biao Sun
sunbiao@tju.edu.cn

Xinshan Zhu
xszhu@tju.edu.cn

Junyan Lu
lgy9826@tju.edu.cn

Honghao Ren
rhh@tju.edu.cn

Hongquan Wang
1015203049@tju.edu.cn

¹ School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

² State Key Laboratory of Digital Publishing Technology, Beijing 100871, China

Fig. 1 An example for image tampering by image inpainting: an original image (left) and the inpainted image (right)



ods have some common shortcomings, such as the high computational cost of feature extraction and the high false-alarm rates in uniform image regions. At present, some attempts have been taken to develop CNN-based methods for image inpainting forensics [25–27]. By automatic feature extraction, the significant performance advantage is provided over conventional methods. However, the convolution operations for feature extraction are performed through sliding windows. This only allows us to extract local manipulation features and thus restricts the forensic performance inevitably.

Since the pixels in inpainted regions are generated by inpainting manipulations with the known pixels, the global relationships between pixels caused are important clues for inpainting forensics. Transformer [28] was first proposed in the domain of natural language processing and has also achieved excellent performance on various computer vision tasks [29–31]. Due to the use of self-attention mechanism, transformer can effectively model the long-distance relationships between pixels and obtain the global feature representation.

Based on the above considerations, we establish a new end-to-end network for image inpainting forensics in this paper, called global–local feature fusion network (GLFFNet). The main contributions of this work are fourfold as below:

1. We develop the forensic network following the encoder–decoder network structure to directly classify each pixel of an image. In the network, the two-stream encoder combining transformer and CNN serves as the basic feature extraction module due to its excellent feature learning ability. In the decoder, by performing up-sampling and convolution operations on the features from the encoder, a binary image of the same size as the input image is generated to indicate the location of the tampered regions.
2. We build the two-stream encoder by incorporating a spatial self-attention stream (SSAS) and a noise feature extraction stream (NFES). The SSAS has large

and diverse receptive fields and is thus able to effectively model long-range dependencies between pixels. The NFES is used to extract local noise features from the filtered noise maps. The generated features are further fused by a convolutional layer.

3. We design a two-stage loss (TSL) function to evaluate the quality of not only the final output of the decoder but also the feature maps produced by the two-stream encoder. By the use of TSL, the two-stream encoder is more effectively guided to learn valid features for inpainting forensics.
4. To train and test our method, we set up datasets for image inpainting forensics using four state-of-the-art image inpainting methods. Extensive experiments show that the proposed GLFFNet consistently outperforms the state-of-the-art inpainting forensics methods.

The remainder of this paper is organized as follows. Section 2 summarizes the related works on image inpainting forensics. Next, in Sect. 3, the architecture and details of GLFFNet are carefully described. Then, a series of tests are performed to evaluate the presented GLFFNet in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Related works

Image inpainting forensics is a challenging problem because inpainted regions are perceptually identical to untouched regions. There has been less research on the problem so far. The existing inpainting forensics methods can be divided into two categories: conventional and deep learning based.

2.1 Conventional forensics Methods

Most of conventional methods are based on the premise that the generated image patches are very similar to the reference

patches. Early, Wu et al. [32] designed a fuzzy membership function of patch similarity measured by the zero-connected length (ZCL) to identify inpainted regions. The disadvantage of this approach is that the suspicious region needs to be selected manually in advance. Bacchuwar et al. [33] proposed a forensics method based on jump block matching. The method reduces the computational cost, but suffers from a high false-alarm rate in uniform image area.

In [34], a two-stage search method based on a weight transformation was proposed to accelerate the search for suspicious patches and filter false-alarm patches using region relations. Although both the false-alarm performance and computational efficiency are improved by the method relative to previous approaches, the detection accuracy is restrained due to the use of the approximately similar patch search. Zhang et al. [35] used the center pixel mapping method to search similar patches and fragment splicing operations to remove false-alarm patches, thereby further improved the detection effect of inpainting.

Zhao et al. [36] detected tampered regions by calculating and segmenting the average sum of the absolute difference images between a tampered image and its JPEG-compressed versions. However, this is only applicable in the context of JPEG compression. Liu et al. [37] designed marginal density and neighboring joint density features for forensics to the combinations of inpainting, compression, filtering, and resampling operations. However, this method has shortcomings similar to those of [36].

Conventional forensics methods depending on patch similarity have several apparent shortcomings. First, they often suffer from the high computational cost since the necessary patch search process is very time-consuming, especially for large images. Second, a high false-alarm rate is inevitably caused in uniform image regions (such as sky and grass) where many image patches are very similar. Finally, the patch similarity is easily destroyed by some common post-processing operations, like JPEG compression, causing the weak robustness.

2.2 Deep learning-based methods

Deep learning-based approaches have been studied on image forensics, including median filter detection [38], copy–move forensics [39], JPEG compression forensics [40], generic image manipulation detection [41], face tamper detection [42], and video forgery detection [43]. As for image inpainting, Zhu et al. [26] proposed an encoder–decoder network based on a CNN to locate the region tampered by patch-based inpainting. Furthermore, Li et al. [25] designed a residual network (ResNet) with a high-pass filtering layer for the forensic task. The above two networks only take use of local features extracted by cascaded convolution operations.

In [44], the image forensics was accomplished for patch-based image inpainting by integrating a long-short term memory (LSTM) network and CNN. The LSTM network is in favor of eliminating the influence of false-alarm patches. Liu et al. [27] proposed progressive spatio-channel correlation network (PSCC-Net) to realize both image manipulation detection and localization. Recently, Wu et al. [45] proposed image inpainting detection network (IID-Net), where the feature extraction block was automatically designed by neural structure search (NAS) algorithm. However, the detection performance is unsatisfactory for the inpainted regions of small size.

Some research efforts have been made to solve the forensic problem by object detection technologies. Wang et al. [46] first developed the Faster R-CNN [47] with ResNet-101 [48] as the backbone network to locate the inpainted image regions. The work in [49] used Mask R-CNN [50] instead of Faster R-CNN, and improved the RPN network to learn multi-scale features for the multi-task inpainting forensics. However, the main issue is that the shape of the inpainted regions cannot be recognized by such methods.

In principle, the existing CNN-based inpainting forensics methods extract local manipulation features by convolutional operations, but the long-distance relationships between pixels are neglected. This evidently restricts the performance of image inpainting forensics. Facing the issue, we develop an end-to-end forensics network combining transformer with a CNN. To the best of our knowledge, this is the first attempt to investigate transformer for forensically determining the presence or location of image manipulations.

3 Method

In this section, the global–local feature fusion network, abbreviated as GLFFNet, is presented. GLFFNet is built based on the encoder–decoder network structure, since the encoder–decoder networks are widely used for a number of pixel-wise image classification tasks and produce good results. The architecture of GLFFNet is illustrated in Fig. 2. As shown in Fig. 2, given a color image, we first encode it into feature maps by a two-stream encoder, and then send the obtained features into the decoder to generate the final forensic results. We elaborate the designed network structure and loss function in the following.

3.1 Two-stream encoder

The designed encoder contains two branches: a spatial self-attention stream (SSAS) and a noise feature extraction stream (NFES). SSAS learns the global manipulation features from inpainted color images by well-designed transformer. Such features represent the texture difference between the

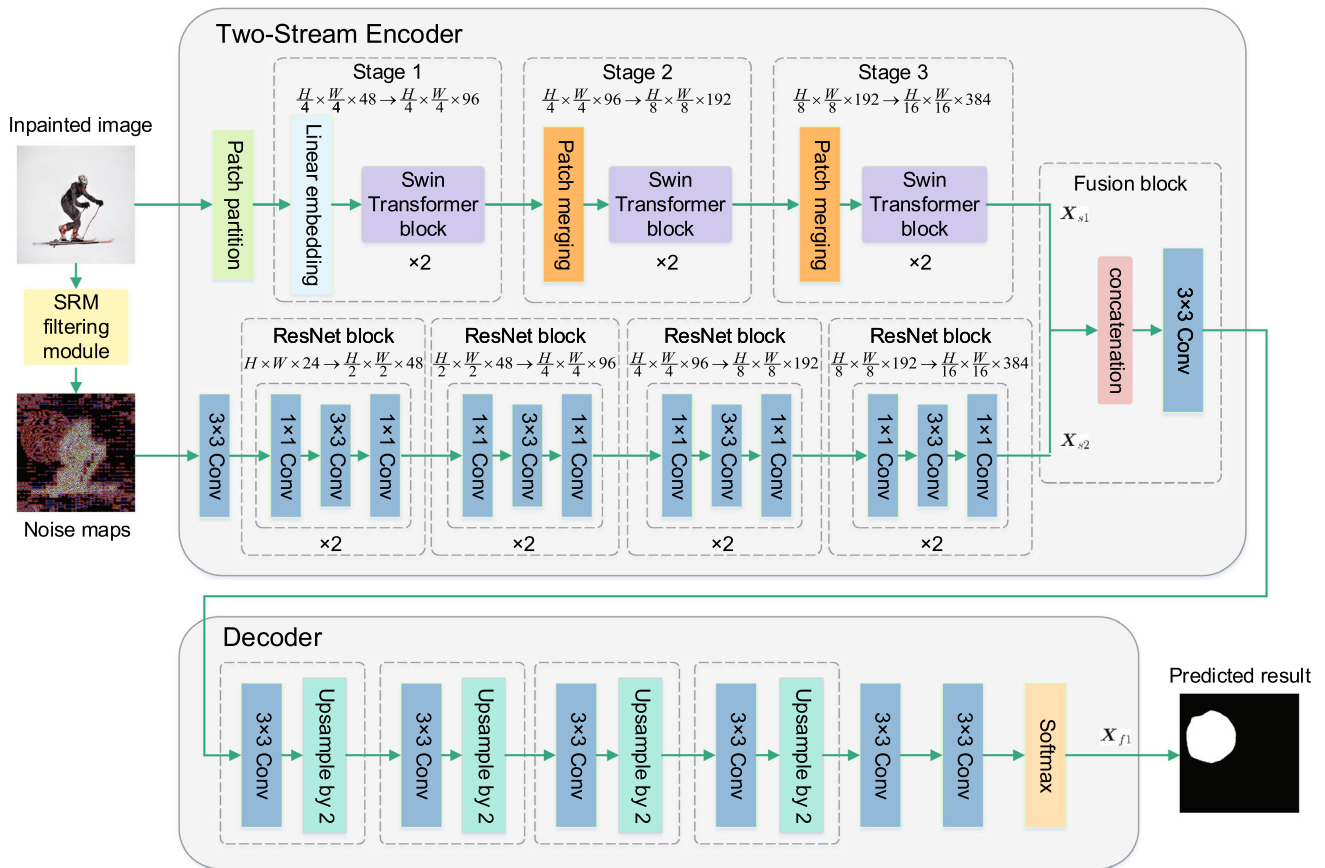


Fig. 2 The architecture of the global-local feature fusion network (GLFFNet)

inpainted regions and the untouched image regions. NFES is formed by a set of residual blocks, which accepts the noise maps generated by the spatial rich model (SRM) for the extraction of local manipulation features. The use of SRM helps eliminate the semantic information and expose the manipulation traces. Then, a cross-domain feature fusion is performed on the extracted features to obtain the comprehensive and valid manipulation features.

3.1.1 SSAS

There are three stages within SSAS, and each stage contains two successive swin transformer blocks [30] with the structure in Fig. 3. As shown in Fig. 3, a regular transformer is first employed, which consists of a window-based multi-head self-attention (W-MSA) module, followed by a 2-layer multi-layer perceptron (MLP) with GELU nonlinearity in between. A layer with layer norm (LN) is arranged before W-MSA module and MLP, and after each of them a residual connection is performed. Let $\hat{\mathbf{Z}}_n$ and \mathbf{Z}_n , respectively, denote the outputs of the W-MSA module and MLP for block n . We can write

$$\hat{\mathbf{Z}}_n = \text{W-MSA}(\text{LN}(\mathbf{Z}_{n-1})) + \mathbf{Z}_{n-1} \quad (1)$$

and

$$\mathbf{Z}_n = \text{MLP}(\text{LN}(\hat{\mathbf{Z}}_n)) + \hat{\mathbf{Z}}_n \quad (2)$$

In W-MSA module, MSA is performed within local windows which are obtained by partitioning an image in a non-overlapping manner. This causes the lack of adequate information interactions across windows. The issue is overcome by the posterior block in Fig. 3, called swin transformer, which is built in the same style as a regular transformer, but replacing W-MSA module by shifted window-based multi-head self-attention (SW-MSA) module. Therefore, for block $n + 1$, we have

$$\hat{\mathbf{Z}}_{n+1} = \text{SW-MSA}(\text{LN}(\mathbf{Z}_n)) + \mathbf{Z}_n \quad (3)$$

and

$$\mathbf{Z}_{n+1} = \text{MLP}(\text{LN}(\hat{\mathbf{Z}}_{n+1})) + \hat{\mathbf{Z}}_{n+1} \quad (4)$$

Using W-MSA and SW-MSA alternately in SSAS, global manipulation features can be learned, meanwhile, the computational complex can be maintained [30].

The implementation details for SSAS are given as follows. First, patch partition is carried out on the input image of

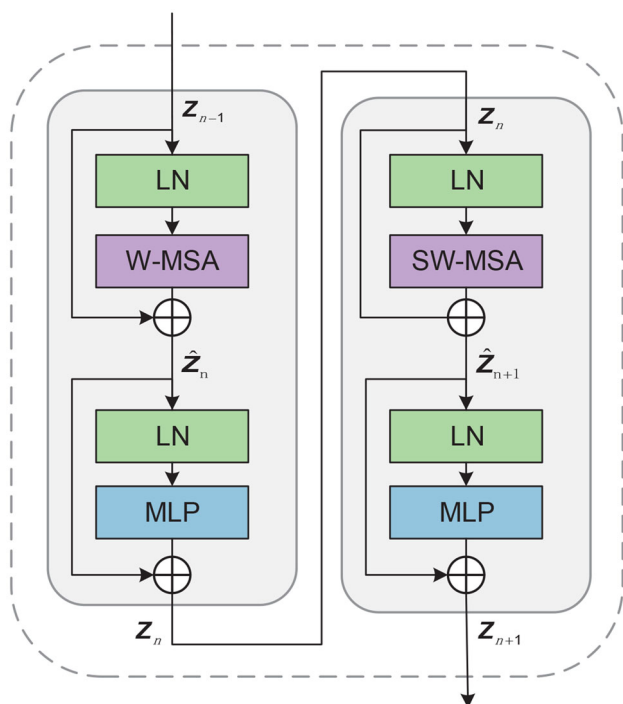


Fig. 3 The structure of the swin transformer blocks

size $H \times W$, producing non-overlapping image blocks of size $M \times M$. Each block is treated as a “token,” and the corresponding feature map of size $1 \times 1 \times 3M^2$ is obtained by concatenating its raw pixel RGB values. In our experiments, $M = 4$ is taken.

Next, through a linear embedding layer, the original feature of a token is projected to an arbitrary dimension C , e.g., $C = 96$ in our simulations, and then input into two successive swin transformer blocks. The process is referred to as “Stage 1.” The number of tokens is $\frac{H}{4} \times \frac{W}{4}$ and maintained in “Stage 1.”

After Stage 1, a patch merging layer is adopted, where the feature maps of adjacent 2×2 tokens are concatenated in the channel dimension and then projected to a dimension $2C$. This reduces the number of tokens by a factor of 0.25, causing the decrease in the inference time. Then, swin transformer blocks are arranged for feature transformation and the network is made to be deeper. The patch merging layer and the swin transformer blocks together form “Stage 2.” At last, “Stage 3” is the repetition of Stage 2, which produces $4C$ feature maps of size $\frac{H}{16} \times \frac{W}{16}$.

Figure 2 shows the basic structure of SSAS. Notice that information exchanging can be conducted across tokens due to the use of Transformed blocks. Therefore, SSAS can effectively represent long-range dependent relationships between pixels as the global manipulation features. This is the main reason why we use SSAS for feature extraction. Our simulations show that SSAS brings about significant performance improvement on the forensic task.

3.1.2 NFES

NFES is designed to extract manipulation features from the derived input other than the target image. The reasons are explained as follows. Image inpainting methods generally leave very weak traces in an image, particularly those based on deep learning. Meanwhile, CNNs tend to learn features related with an image’s content. As a result, it is difficult to extract valid features directly from the color image for inpainting forensics. To remove the interference from image content and expose inpainting artifacts, the input image is first filtered by SRM-based filters [51] to obtain the noise maps.

Following the suggestions in [52], we select three SRM kernels from [51] and derive three noise maps accordingly. As shown in Fig. 4, the selected kernels are all designed for high-pass filters and each kernel has a size of 5×5 . With the filters, the high-frequency noise is emphasized rather than the image content, and thus the inpainting artifacts can be sufficiently exposed. Figure 5 demonstrates the effect of the utilized SRM filters. We can see that the inpainting traces become much clearer after filtering the sample images.

After the filtering module, a feature extraction network based on a residual CNN [48] is established, as shown in Fig. 2. First, the noise maps are sent into a convolutional layer with 24 kernels of size 3×3 to derive more diverse features. Then, four ResNet blocks are adopted to effectively learn the inpainting features. Each ResNet block contains two “bottleneck” units. In each bottleneck unit, three consecutive convolutional layers with kernel sizes of 1×1 , 3×3 , and 1×1 , and a residual skip connection are placed. Moreover, after each convolutional layer, a batch normalization layer and a rectified linear unit (ReLU) layer are applied. The feature channel depth is resized by the factors of 0.5, 1, and 4 through the three convolutional layers in the first bottleneck unit, and 0.25, 1, and 4 in the second unit. The channel reduction in each unit allows the sequent convolutional layer to extract the inpainting features more quickly and efficiently. Additionally, in the second unit of each ResNet block the convolution operation with a stride of 2 is applied for feature aggregation and spatial resolution reduction. Overall, NFES has a 3-channel input and a 384-channel output with a spatial resolution being $\frac{1}{16}$ of the input. This can obtain output feature maps of the same size as the SSAS while maintaining a similar amount of computation.

3.1.3 Feature fusion

Currently, feature fusion is commonly accomplished by one of a cascade, an addition, and a multiplication of multiple features. We compare these methods, and finally cascade the outputs of SSAS and NFES, as shown in Fig. 2.

Fig. 4 The SRM filtering module (top) and the selected SRM kernels (bottom)

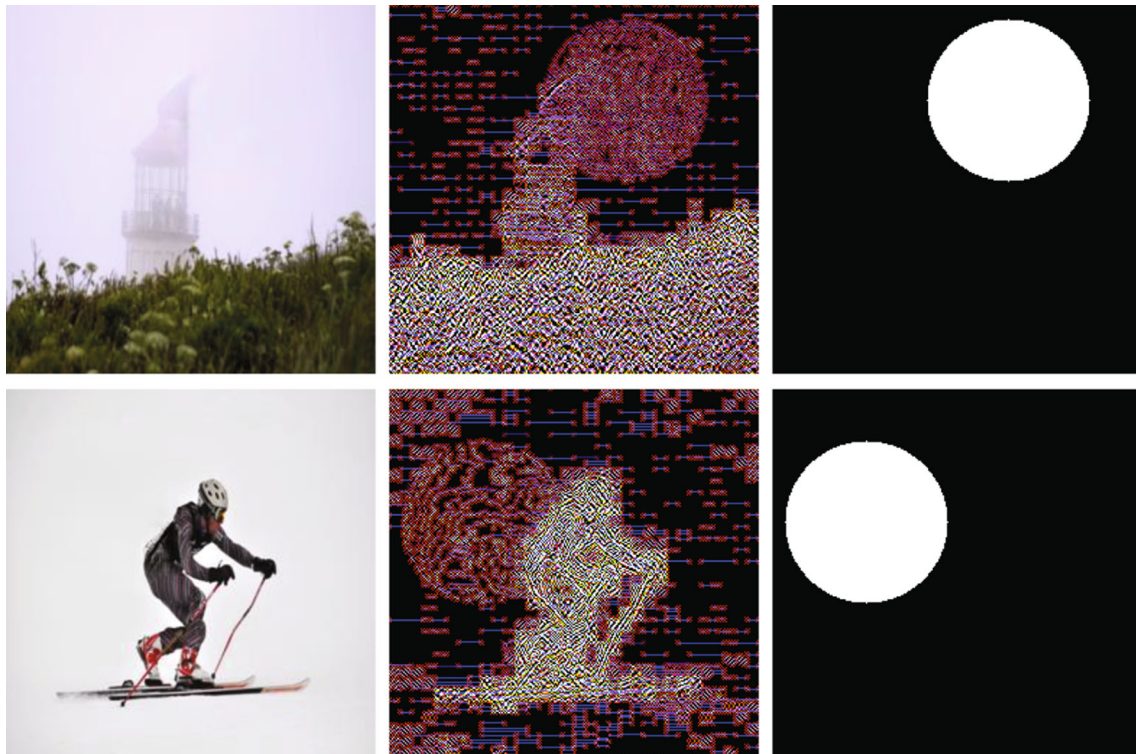
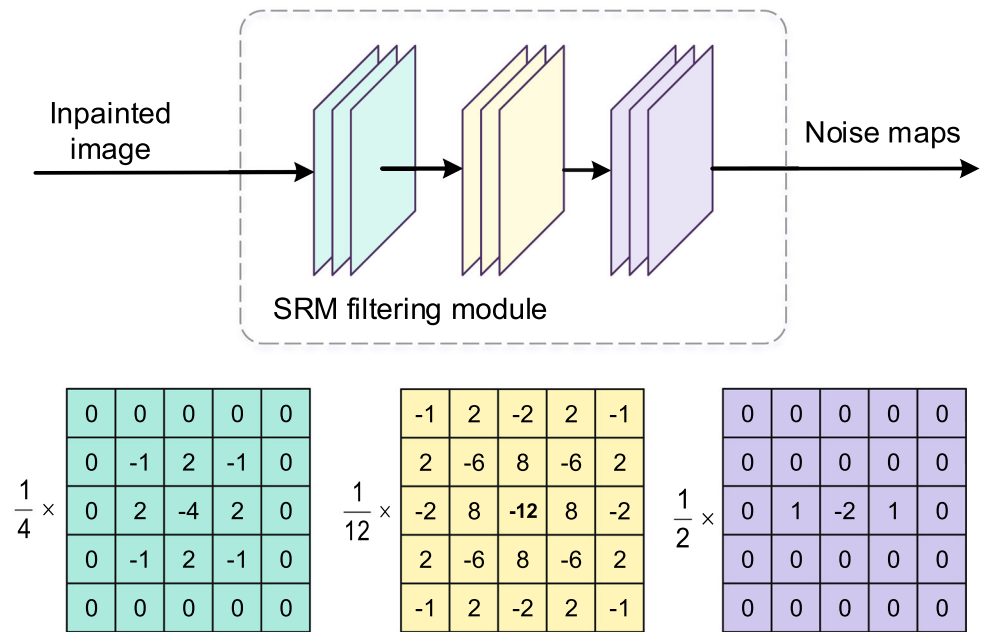


Fig. 5 Filtering effect of the selected SRM filters on two sample images: the tampered images (left), the noise maps (middle), and the real masks (right)

Further, a convolutional layer with kernels of size 3×3 is used to refine the fused features. Meanwhile, the number of channels is reduced to 384 to eliminate redundant features. By the feature fusion, comprehensive feature representation can be obtained.

The reason we only adopt concatenation and convolution operation is that the two streams have already provided sufficient and distinct features. In contrast, complex structure may jeopardize the overall generalization ability of our method and degrade the performance.

3.2 Decoder

The decoder generates a binary image, called the predicted mask, where the tampered regions are white. To this goal, the decoder is established by convolutional layers, up-sampling layers, and the Softmax layer. The architecture of the decoder is demonstrated in Fig. 2, and described in detail as follows.

The first layer is a convolutional layer with kernels of size 3×3 and stride of 1, and the feature channel depth is made to decrease by half. Behind this layer, an up-sampling layer is employed to enlarge the feature resolution by a factor of 2. The up-sampling operation is realized by the bilinear interpolation. The above two layers are repeatedly applied four times to produce the feature maps with the same resolution as the target image and the channel depth being 24. Next, through two consecutive convolutional layers with kernels of size 3×3 and stride of 1, the feature channel depth is further reduced to 8 and 2, respectively. Finally, the learned features are fed into the Softmax layer, where the 2-way Softmax function is performed to derive the confidential map \mathbf{Y} regarding the pixel-wise inpainting detection. Consequently, the predicted binary mask $\hat{\mathbf{Z}}$ is derived according to the rule whether the elements of the confidential map are larger than 0.5 or not.

3.3 Loss function

During training stage, a loss function is needed to quantify the quality of a particular set of the parameter settings for all network weights. In this study, we design the two-stage loss (TSL) for training our network, where the quality for the final output and the results of the intermediate stage are jointly accessed.

Ideally, the final confidential map \mathbf{Y} output by our network should equal to the ground truth mask \mathbf{Z} . Their difference is indicated by the loss \mathcal{L}_f , which is obtained by

$$\mathcal{L}_f = L_{\text{BCE}}(\mathbf{Y}, \mathbf{Z}), \tag{5}$$

where $L_{\text{BCE}}(\cdot)$ denotes the binary cross-entropy loss (BCEL) function [26].

The loss \mathcal{L}_f is particularly important, since the proposed network is driven to produce the final output being in accordance with the real mask with the use of \mathcal{L}_f . However, the prediction performance is also closely related with the intermediate features \mathbf{X}_{s1} and \mathbf{X}_{s2} produced by SSAS and NFES, respectively. Moreover, the two streams have very different structures, causing the difference on the convergence rate. Therefore, to attain a better training model, it is necessary to introduce the loss \mathcal{L}_{si} regarding \mathbf{X}_{si} , $i \in \{1, 2\}$.

The loss \mathcal{L}_{si} is computed between \mathbf{X}_{si} and \mathbf{Z} to measure how well the intermediate features agree with the ground truth mask. For the purpose, a 3×3 convolutional layer

Algorithm 1 Training Algorithm of GLFFNet

Input: $\mathcal{F}, \theta_{\mathcal{F}}$ are the proposed forensic network and the corresponding parameters;

$\mathcal{D} = \{(\mathbf{X}_k, \mathbf{Z}_k) | k = 1, 2, \dots, N\}$ is the training dataset, and \mathbf{X}_k and \mathbf{Z}_k stand for the k -th sample image and the corresponding ground truth mask;

Output: Optimized forensic network \mathcal{F}

- 1: Initialize the network \mathcal{F}
- 2: **for** each mini-batch $\mathcal{D}_b \in \mathcal{D}$ **do**
- 3: Perform forward inference to obtain the features $\mathbf{X}_{s1}, \mathbf{X}_{s2}$ and the final output \mathbf{Y} for each image
- 4: Transform \mathbf{X}_{s1} and \mathbf{X}_{s2} into \mathbf{X}'_{s1} and \mathbf{X}'_{s2} by a convolutional layer followed by a Softmax layer
- 5: Down-sample the real mask \mathbf{Z} for each image by a factor of $\frac{1}{16}$ to derive \mathbf{Z}'
- 6: Calculate the losses $\mathcal{L}_{s1}, \mathcal{L}_{s2}, \mathcal{L}_f$, and \mathcal{L} by (5) to (7)
- 7: Update the model parameters $\theta_{\mathcal{F}}$ according to the overall loss \mathcal{L}
- 8: **end for**

followed by a Softmax layer is applied to \mathbf{X}_{si} , resulting in \mathbf{X}'_{si} with the channel depth being 1. The real mask \mathbf{Z} is down-sampled by a factor of $\frac{1}{16}$ with the nearest-neighbor algorithm, deriving the low-resolution version \mathbf{Z}' . Then, applying the function $L_{\text{BCE}}(\cdot)$ again, the loss \mathcal{L}_{si} is expressed as

$$\mathcal{L}_{si} = L_{\text{BCE}}(\mathbf{X}'_{si}, \mathbf{Z}'). \tag{6}$$

Apparently, with the use of the losses \mathcal{L}_{s1} and \mathcal{L}_{s2} , GLFFNet can be guided to more effectively learn inpainting features for forensics. Moreover, the issue is alleviated that one of the encoder streams tends to be optimized during the training phase.

The overall loss \mathcal{L} is obtained by combining the above three losses as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{s1} + \lambda_2 \mathcal{L}_{s2} + \mathcal{L}_f, \tag{7}$$

where λ_i , $i \in \{1, 2\}$, is a hyperparameter indicating the importance of the loss \mathcal{L}_{si} . In our simulations, the hyperparameters are both set to 1. The pseudocode of the training procedure is shown in Algorithm 1.

The parameters of neural networks are commonly optimized by on-line gradient descent. Therefore, the convergence rate of our network is closely related to the gradients of the overall loss with respect to the network parameters. We use vectors \mathbf{W}_1 and \mathbf{W}_2 to denote the parameters involved in SSAS and NFES, respectively. Then, the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_i}$ can be derived as $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_i} = \lambda_1 \frac{\partial \mathcal{L}_{si}}{\partial \mathbf{W}_i} + \frac{\partial \mathcal{L}_f}{\partial \mathbf{W}_i}$, $i \in \{1, 2\}$. Clearly, by choosing the hyperparameters λ_1 and λ_2 , the gradients $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_1}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_2}$ can be changed, and thus the convergence rates of the SSAS and NFES λ_1 and λ_2 can be adjusted.

4 Experiment

To evaluate the performance of the method developed in this paper, we built an image inpainting forensics dataset, and trained and tested our GLFFNet on the dataset. Intersection over Union (IoU) and F1 score were used as forensic performance metrics, and comparison experiments were conducted with typical image inpainting forensics methods. Finally, we conducted an ablation study to verify the main components of our proposed network.

4.1 Training and testing datasets

We randomly selected 21,600 images from the Place365 dataset [53] to generate training and testing images of size 256×256 . The real mask is randomly generated for each image in shape, size, and location [54]. Three types of inpainted regions were included: circular, rectangular, and irregular regions. The area of the inpainted region was randomly set to one of 1%, 5%, 10%, and 15% of the whole image, and the inpainted region was randomly located on the image plane. Meanwhile, we randomly employed four typical deep learning-based image forensics methods, including DeepfillV2 [16], PIC [18], LaMa [19], and ICT [21], to tamper a given image. Finally, we divided all the inpainted images into three parts: 18,000 images for training, 1,800 images for validation, and 1,800 images for testing.

4.2 Training details

The proposed GLFFNet with the input of size 256×256 was implemented using the PyTorch deep learning framework. GLFFNet was trained on a workstation with 3.5 GHz Intel Xeon(R) W-3223 CPU and 64 GB memory, equipped with a single NVIDIA GeForce RTX 3090 GPU. The network weights except for those in SRM kernels were initialized by Kaiming initialization method [55]. The Adam optimizer with decoupled weight decay (AdamW) [56] was adopted to iteratively update the network weights during the training process. The optimizer recovers the original formulation of weight decay regularization by decoupling the weight decay. This can accelerate training convergence and greatly improve the generalization performance of the network. The training parameters of the optimizer were set as follows. The learning rate was initialized to 1×10^{-3} , and decreased by 0.95 per epoch. The batch size was set to 64. In addition, the training data were augmented by JPEG compression with a randomly selected quality factor between 75 and 100.

For comparisons, the state-of-the-art methods proposed in [25], [27], [26] and [45] were chosen. We used the publicly available implementations of HPFCN [25] and IID-Net [45]. FCNet [26] and PSCC-Net [27] were implemented by ourselves. These methods were re-trained on our dataset strictly

following the training procedures and parameter settings presented in their papers.

4.3 Forensic performance evaluation: no attacks

The forensic performance is first evaluated on the original inpainted images. Figure 6 shows the visualization results obtained by all the tested approaches on four typical deep learning-based image inpainting methods. All of the methods are able to approximately locate the inpainted regions with different shapes and sizes. However, the methods for comparisons fail to finely determine the edges of the inpainted regions, especially the irregular regions. The forensics results of our proposed method (e.g., column 6 of each subfigure in Fig. 6) better agree with the ground truth masks (e.g., the last column of each subfigure in Fig. 6).

Table 1 summarizes IoU and F1 score for each tested method on the testing dataset. It is clear that our GLFFNet provides very accurate and consistent inpainting localization. GLFFNet achieved IoU of 88.76 and F1 score of 93.69 on the DeepfillV2 dataset, and higher metrics on other three datasets. These results are significantly better than those obtained by other methods. The performance advantage might be brought about by the combination of CNN and transformer.

In addition, we also examine the forensic effects on inpainted regions with different shapes and sizes, as shown in Fig. 7. Clearly, for the circular inpainted regions, the performance of all the tested methods degrades as the size of the inpainted regions decreases. Similar observations can be made for rectangular and irregular regions. Moreover, our method achieves the best forensic performance for all the considered cases. In particular, our method even presents IoU larger than 80% on all the datasets while the area of the inpainted region is set to 1% of the whole image. This indicates that our methods are also very effective for small inpainted regions. Generally, a small missing region is easily inpainted and less inpainting traces are left, thus causing the forensic difficulties. Among all methods, only our method and PSCC-Net achieve good results on small regions, both of which are fused by multiple streams. This shows that the feature information of multiple different receptive fields needs to be fused. Our method learns long-short range dependent features effectively by combining CNN and transformer, and thus improves the detection accuracy significantly.

4.4 Forensic performance evaluation: typical attacks

In practice, forgers might perform some post-processing operations after inpainting to evade forensic detection. Thus, we investigate the robustness of the proposed method against JPEG compression and additive white Gaussian noise

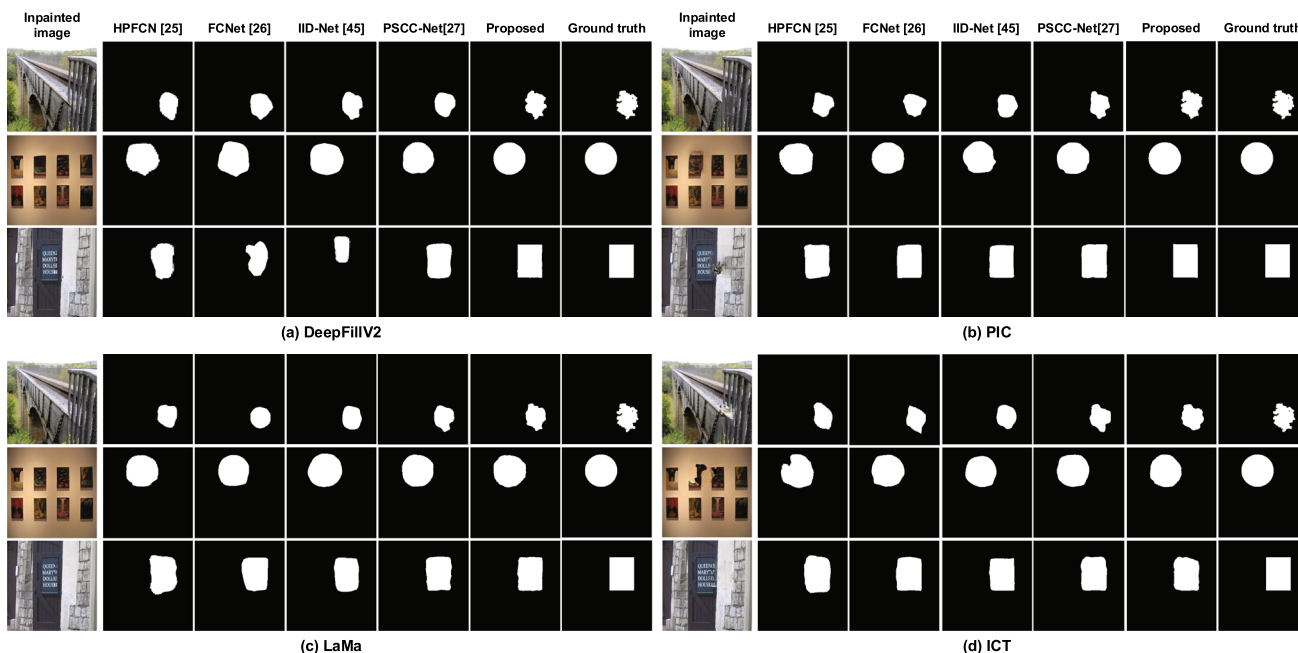


Fig. 6 Forensic results on several sample images. **a–d** Results on four different inpainting methods: DeepfillV2 [16], PIC [18], LaMa [19], and ICT [21]. The original inpainted images, forensic results obtained

by HPFCN [25], FCNet [26], IID-Net [45], PSCC-Net [27], and our results, as well as the ground truth masks are, respectively, shown in Columns 1 to 7 of each subfigure

Table 1 Average IoU and F1 score of different methods on four typical datasets without extra distortions

Inpainting method	HPFCN [25]		FCNet [26]		IID-Net [45]		PSCC-Net [27]		Proposed	
	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1
DeepfillV2	57.66	67.06	52.33	61.06	50.52	58.57	85.04	91.32	88.76	93.69
PIC	81.43	88.22	82.15	88.77	80.14	86.58	91.33	95.34	93.87	96.74
Lama	80.39	87.97	81.04	87.84	79.18	86.28	89.38	94.18	91.66	95.49
ICT	75.81	83.32	76.89	83.04	79.20	85.72	88.29	93.39	89.78	94.30

The best results are marked in bold font

(AWGN). The two manipulations are considered because they are often performed in many applications.

The testing images are first JPEG-compressed by quality factors (QF) of 95, 85, and 75, and inpainting forensics methods are performed on the compressed images while the area ratio for the inpainted region is set to 15%. The average values of IoU and F1 scores obtained by the tested methods on the DeepfillV2 and PIC datasets are listed in Table 2. It can be seen that the forensic performance degrades for all the tested methods as QF decreases, and our method still has the best forensic results in all the cases. For QF of 95, GLFFNet performs almost as well as on the original inpainted images. For QF of 75, GLFFNet obtains IoU of 79.78, 92.32, and F1 score of 88.28, 95.95 on the DeepfillV2 and PIC datasets, respectively. These results are even better than those of other methods for comparisons on the inpainted images without any alterations, although our network has a little larger performance loss compared to other methods. Therefore, our

proposed method exhibits stronger robustness against JPEG compression.

Then, the robustness is further tested under AWGN with signal-to-noise ratios (SNRs) of 50 dB, 40 dB, and 35 dB. The results on the DeepfillV2 and PIC datasets are reported in Table 3 for the area ratio 15%. As is clear, our method is still superior to other tested methods under this attack. For AWGN with SNR of 50 dB, GLFFNet performs slightly worse than that under no attacks. Even for AWGN with SNR of 35 dB, IoU of 83.43 and F1 score of 90.46 are reached by GLFFNet on the DeepfillV2. Therefore, the proposed method is insensitive to AWGN.

4.5 Ablation analysis

We investigated the effects of the SRM filters, SSAS, NFES, two-stream encoder, and TSL through ablation experiments. For this purpose, we implemented the following variants of the full model (GLFFNet).

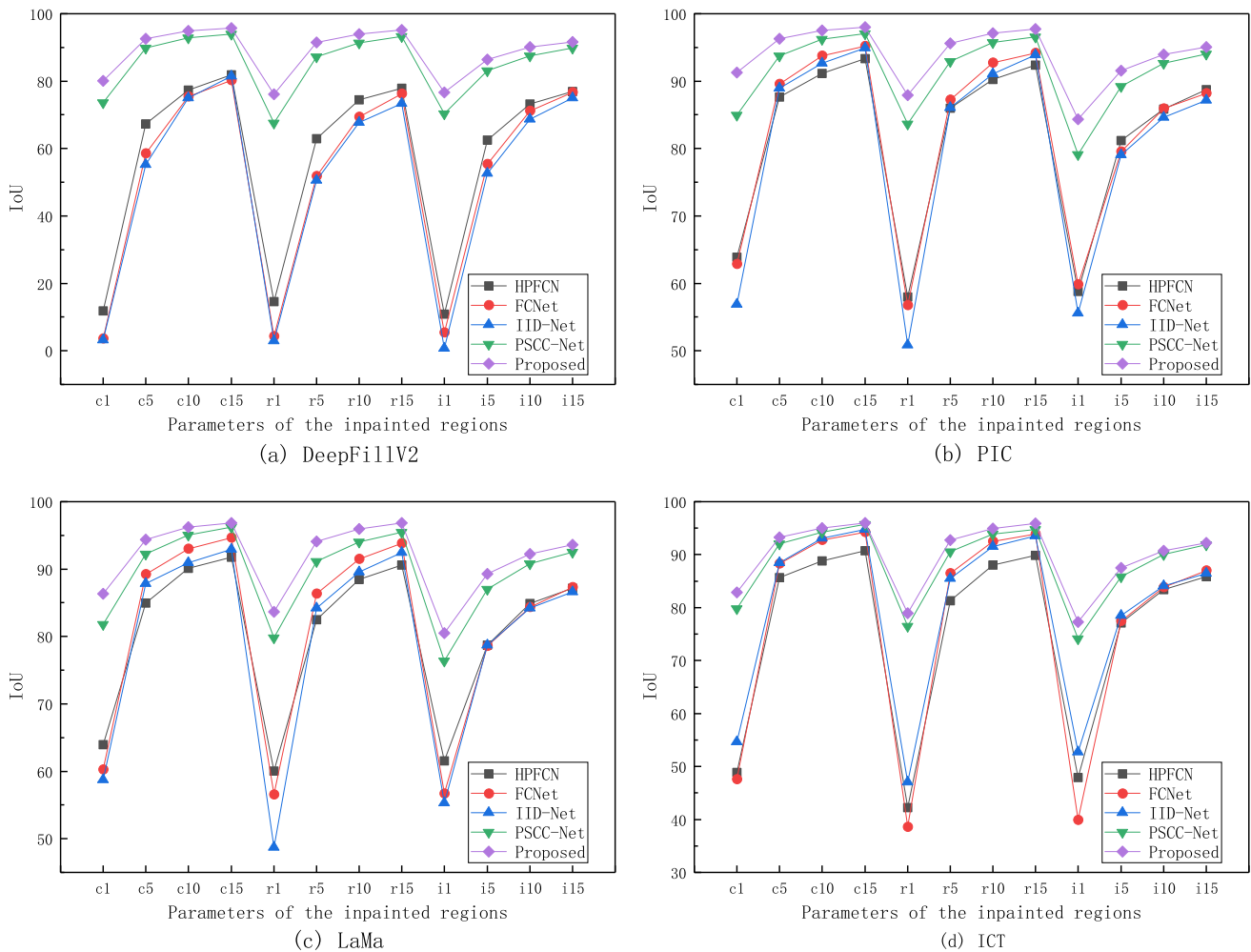


Fig. 7 Effects of parameters (shape and size) of inpainted regions on the forensic performance for different methods. **a–d** IoU curves on the testing dataset produced by: DeepfillV2 [16], PIC [18], LaMa [19], and ICT [21]. The circular, rectangular, and irregular inpainted regions

are indicated by the letters “c,” “r,” and “i” in parameters of inpainted regions, respectively. The numbers 1, 5, 10, and 15 behind these letters represent the percentage of the whole image to be inpainted

1. RCSNet: This network is built by removing SSAS and SRM filtering module in GLFFNet. In other words, the residual convolutional stream (RCS) in NFES serves as the encoder. BCEL is adopted to train RCSNet.
2. NFESNet: This network is obtained by removing SSAS in GLFFNet. That is, swin transformer (ST) blocks are not contained in NFESNet. The training is performed with the use of BCEL.
3. SSASNet: SSASNet is developed by removing NFES in GLFFNet. That is, only SSAS containing ST blocks serves as the encoder. BCEL is applied for training SSASNet.
4. DRCSNet: DRCSNet is constructed by replacing all swin transformer blocks in SSAS with ResNet blocks. As such, the encoder is composed of double RCSs (DRCS). The training of DRCSNet employs TSL.

5. DSTSNet: The network is formed by replacing NFES in GLFFNet with SSAS. That is, the encoder is composed of double swin transformer streams (DSTS). DSTSNet is trained as DRCSNet.
6. GLFFNet2: This network has the same structure as GLFFNet, but is trained using BCEL.

All these variants were trained on the training dataset for DeepfillV2 with the same training options as those of the full model. Table 4 reports the results of GLFFNet and its variants on the testing dataset for DeepfillV2 and the results under JPEG compression with QF of 75 (JPEG 75 for short).

We can see that, RCSNet with RCS as the encoder only gets IoU of 68.30 and F1 score of 74.94 under no attacks, but they become 85.24 and 91.01 for NFESNet. The significant performance gain is brought about by significant also brought about SRM filtering module. Moreover, NFESNet performs

Table 2 Average IoU and F1 score of different methods on two typical datasets under JPEG compression: the area of the tampered regions is set to 15% of the whole image

Datasets	QF(%)	HPFCN [25]		FCNet [26]		IID-Net [45]		PSCC-Net [27]		Proposed	
		IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1
DeepfillV2	95	77.10	86.77	77.58	86.95	73.34	83.45	91.82	95.70	93.14	96.43
	85	72.54	83.52	76.76	86.33	68.11	79.07	87.62	93.26	89.81	94.57
	75	68.98	80.75	75.53	85.38	64.40	75.73	78.49	87.09	79.78	88.28
PIC	95	90.70	95.09	92.35	95.98	91.56	95.54	95.42	97.65	95.85	97.87
	85	88.75	93.94	91.87	95.72	89.83	94.55	94.20	97.00	94.22	97.01
	75	86.29	92.36	91.41	95.46	86.98	92.66	90.99	95.05	92.32	95.95

The best results are marked in bold font

Table 3 Average IoU and F1 score of different methods on two typical datasets under additive white Gaussian noise: the area of the tampered regions is set to 15% of the whole image

Datasets	SNR	HPFCN [25]		FCNet [26]		IID-Net [45]		PSCC-Net [27]		Proposed	
		IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1
DeepfillV2	50 dB	76.83	86.59	77.51	86.89	76.16	85.56	90.63	94.89	93.05	96.32
	40 dB	75.31	85.55	76.84	86.38	74.86	84.49	72.52	82.21	88.34	93.55
	35 dB	72.43	83.21	76.00	85.67	71.69	81.90	48.73	62.79	83.43	90.46
PIC	50 dB	91.35	95.45	92.50	96.06	91.91	95.73	95.83	97.86	96.84	98.39
	40 dB	87.26	92.94	92.33	95.97	90.46	94.76	91.48	95.30	95.13	97.46
	35 dB	85.18	91.68	91.91	95.73	87.16	92.34	83.55	90.33	92.14	96.25

The best results are marked in bold font

Table 4 Average values of IoU and F1 score obtained by different variants of GLFFNet on the DeepfillV2 dataset

Variants	Components				No attacks		JPEG 75	
	SRM	RCS	ST	TSL	IoU	F1 score	IoU	F1 score
RCSNet		✓			68.30	74.94	44.01	50.96
NFESNet	✓	✓			85.24	91.01	43.61	52.28
SSASNet			✓		85.91	91.82	45.63	53.50
DRCSNet	✓	✓		✓	82.38	89.01	40.72	49.14
DSTNet			✓	✓	85.25	91.51	46.83	54.86
GLFFNet2	✓	✓	✓		86.51	92.08	46.74	54.84
GLFFNet (full model)	✓	✓	✓	✓	88.76	93.69	53.39	61.31

The best results are marked in bold font

better than RCSNet being subject to JPEG compression in terms of F1 score.

Comparing with NFESNet, SSASNet manifests the approximately same performance on the original inpainted images. This reveals that the designed two streams SSAS and NFES effectively learn the inpainting features. Surprisingly, SSASNet is more robust to JPEG compression than NFESNet. This might be because the local inpainting features learned by RCS in NFESNet are more easily destroyed than the global ones extracted by SSASNet applying transformer blocks.

Impressively, GLFFNet2 yields IoU of 86.51 and 46.74, and F1 score of 92.08 and 54.84, respectively, for no attacks and JPEG 75. The performance of GLFFNet2 is better than that of either NFESNet or SSASNet. This indicates that the two-stream encoder combining RCS and transformer has the

performance advantage over a single-stream encoder. However, DSTNet containing double SSAS and using TSL for training is slightly inferior to the network with a single SSAS under no attacks and DRCSNet with double RCSs is even worse than NFESNet in all the cases. This further reflects that the manipulation features can be effectively refined by combining transformer and RCS other than two identical streams both learning the local or global features.

The best performance is reached by the full model, i.e., GLFFNet, with the use of all the aforementioned components. Particularly, by applying TSL, GLFFNet surpasses GLFFNet2 by about 2.2 in IoU and 1.6 in F1 score under no attacks. The performance margin becomes more larger under JPEG 75. Now, we can make a conclusion that all the

Fig. 8 Training loss curves of \mathcal{L}_{s_1} and \mathcal{L}_{s_2} for NFESNet and SSASNet, respectively: **a** $\lambda_1 = 1$ and $\lambda_2 = 1$; **b** $\lambda_1 = 10$ and $\lambda_2 = 1$

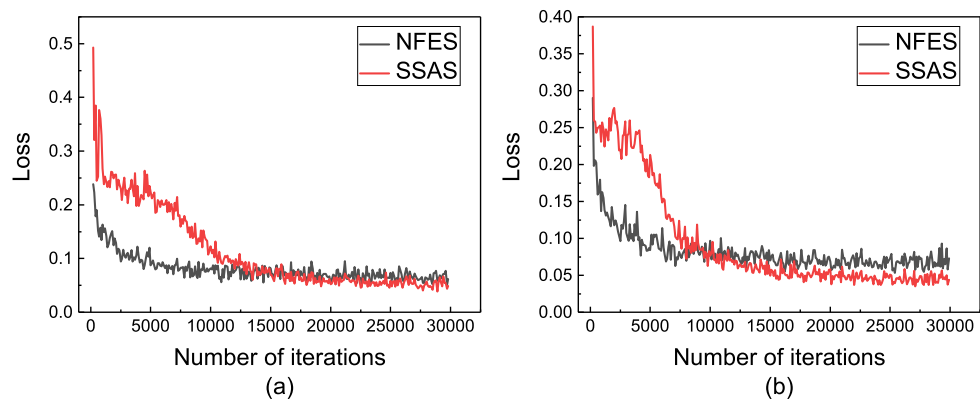


Table 5 Number of FLOPs for different variants

Variants	Input	FLOPs
NFESNet	256×256	4.4G
SSASNet	256×256	4.0G
DRCSNet	256×256	8.1G
GLFFNet (full model)	256×256	7.7G

components present the performance improvement and contribute to the overall performance.

Further, GLFFNet is trained on the DeepfillV2 dataset applying TSL with different values of the hyperparameters λ_1 and λ_2 . The training loss curves for the two loss components \mathcal{L}_{s_1} and \mathcal{L}_{s_2} in TSL are plotted in Fig. 8. It is clear that the convergence rate is different for SSAS and NFES. In Fig. 8a for the case of $\lambda_1 = 1$ and $\lambda_2 = 1$, the loss \mathcal{L}_{s_1} for SSAS degrades more slowly than that for NFES when the number of iterations is lower than 125000. Thereafter, the optimization of SSAS can be proceeded at a convergence rate close to that of NFES. The situation is changed in Fig. 8b for the case of $\lambda_1 = 10$ and $\lambda_2 = 1$, where the loss for SSAS is highlighted. SSAS converges faster than NFES after about 8k iterations, although SSAS is still inferior to NFES at the beginning. Meanwhile, the loss \mathcal{L}_{s_2} for NFES can be reached at the lower level. That is, by adjusting the hyperparameters λ_1 and λ_2 , we can control the optimization speed of SSAS and NFES, and thus the difference on the convergence rate for SSAS and NFES is able to be effectively alleviated.

Additionally, in one forward on a fixed input size 256×256 , we calculate the number of floating point operations (FLOPs) for different variants, which is shown in Table 5. From the results, SSASNet presents about 0.4G FLOPs lower than that of NFESNet. FLOPs of GLFFNet and DRCSNet are approximately twice as much as one of SSASNet, since they both contain the encoder with two streams for feature extraction. Moreover, the difference between FLOPs of GLFFNet and DRCSNet is also about 0.4G. That means that transformer can bring about the forensic performance advantage with less computation.

5 Conclusion

In this paper, a deep learning forensics approach for deep image inpainting, called GLFFNet, has been proposed. GLFFNet followed the encoder–decoder network structure in order to directly predict the pixel-wise class label regarding inpainting manipulation. The encoder was designed as a two-stream network combining transformer and a residual network. By the encoder, both the global and local manipulation features can be effectively learned and further fused to generate a comprehensive feature representation. The decoder was established by applying the up-sampling and convolution operations to gradually enlarge the resolution of the input feature maps and reduce the feature channel depth. For training GLFFNet, the TSL function was proposed taking into account the learning effect of each network stream in the encoder. GLFFNet works in a data-driven manner and thus avoids the difficulties on the design of the hand-crafted features.

The proposed GLFFNet was extensively tested on various images, and compared with state-of-the-art inpainting forensics methods. Experimental results showed that GLFFNet can learn the effective manipulation features for deep image inpainting and locate inpainted regions more accurately. Comparing with representative forensics methods, GLFFNet manifests significantly better forensic performance in terms of the location accuracy. Moreover, GLFFNet exhibited superior robustness against typical post-processing operations, i.e., JPEG compression and AWGN.

Funding This study was supported by the National Natural Science Foundation of China under Grants 61972282 and 61971303, and by the Opening Project of State Key Laboratory of Digital Publishing Technology under Grant Cndplab-2019-Z001. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Declarations

Conflicts of interest The authors declare that they have no competing interests.

References

1. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Trans. Inf. Forensics Secur.* **13**(11), 2691–2706 (2018)
2. Chen, H., Han, Q., Li, Q., Tong, X.: Digital image manipulation detection with weak feature stream. *The Vis. Comput.* 1–15 (2021)
3. Gao, H., Gao, T., Cheng, R.: Robust detection of median filtering based on data-pair histogram feature and local configuration pattern. *J. Inf. Secur. Appl.* **53**, 102506 (2020)
4. Chen, B., Qi, X., Zhou, Y., Yang, G., Zheng, Y., Xiao, B.: Image splicing localization using residual image and residual-based fully convolutional network. *J. Vis. Commun. Image Represent.* **73**, 102967 (2020)
5. Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting fake news: Image splice detection via learned self-consistency. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 101–117 (2018)
6. Yang, J., Liang, Z., Gan, Y., Zhong, J.: A novel copy-move forgery detection algorithm via two-stage filtering. *Dig. Sig. Process.* **113**, 103032 (2021)
7. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424 (2000)
8. Chan, T.F., Shen, J.: Nontexture inpainting by curvature-driven diffusions. *J. Vis. Commun. Image Represent.* **12**(4), 436–449 (2001)
9. Esedoglu, S., Shen, J.: Digital inpainting based on the Mumford-Shah-Euler image model. *Eur. J. Appl. Math.* **13**(4), 353–370 (2002)
10. Shen, J., Chan, T.F.: Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **62**(3), 1019–1043 (2002)
11. Chen, Y., Zhang, H., Liu, L., Tao, J., Zhang, Q., Yang, K., Xia, R., Xie, J.: Research on image inpainting algorithm of improved total variation minimization method. *J. Ambient Intell. Humaniz. Comput.* 1–10 (2021)
12. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* **13**(9), 1200–1212 (2004)
13. Grossauer, H.: A combined PDE and texture synthesis approach to inpainting. In: *European conference on computer vision*, pp. 214–224. Springer (2004)
14. Hays, J., Efros, A.A.: Scene completion using millions of photographs. *ACM Trans. Graph. (ToG)* **26**(3), 4–es (2007)
15. Chen, Y., Liu, L., Tao, J., Xia, R., Zhang, Q., Yang, K., Xiong, J., Chen, X.: The improved image inpainting algorithm via encoder and similarity constraint. *Vis. Comput.* **37**(7), 1691–1705 (2021)
16. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4471–4480 (2019)
17. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. In: *International Conference on Learning Representations (ICLR)* (2021)
18. Zheng, C., Cham, T.J., Cai, J.: Pluralistic image completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1438–1447 (2019)
19. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159 (2022)
20. Chen, Y., Liu, L., Phonevilay, V., Gu, K., Xia, R., Xie, J., Zhang, Q., Yang, K.: Image super-resolution reconstruction based on feature map attention mechanism. *Appl. Intell.* **51**(7), 4367–4380 (2021)
21. Wan, Z., Zhang, J., Chen, D., Liao, J.: High-fidelity pluralistic image completion with transformers. *arXiv preprint arXiv:2103.14031* (2021)
22. Chang, I.C., Yu, J.C., Chang, C.C.: A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. *Image Vis. Comput.* **31**(1), 57–71 (2013)
23. Li, H., Luo, W., Huang, J.: Localization of diffusion-based inpainting in digital images. *IEEE Trans. Inf. Forensics Secur.* **12**(12), 3050–3064 (2017)
24. Li, X.H., Zhao, Y.Q., Liao, M., Shih, F.Y., Shi, Y.Q.: Detection of tampered region for JPEG images by using mode-based first digit features. *EURASIP J. Adv. Sig. Process.* **2012**(1), 1–10 (2012)
25. Li, H., Huang, J.: Localization of deep inpainting using high-pass fully convolutional network. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8301–8310 (2019)
26. Zhu, X., Qian, Y., Zhao, X., Sun, B., Sun, Y.: A deep learning approach to patch-based image inpainting forensics. *Sig. Process.: Image Commun.* **67**, 90–99 (2018)
27. Liu, X., Liu, Y., Chen, J., Liu, X.: PSCC-Net: progressive spatio-channel correlation network for image manipulation detection and localization. *arXiv preprint arXiv:2103.10596* (2021)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *Adv. Neural Inf. Process. Syst.* 5998–6008 (2017)
29. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
30. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. In: *International Conference on Computer Vision (ICCV)* (2021)
31. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6881–6890 (2021)
32. Wu, Q., Sun, S.J., Zhu, W., Li, G.H., Tu, D.: Detection of digital doctoring in exemplar-based inpainted images. In: *2008 International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1222–1226. IEEE (2008)
33. Bacchuwar, K.S., Ramakrishnan, K., et al.: A jump patch-block match algorithm for multiple forgery detection. In: *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pp. 723–728. IEEE (2013)
34. Stamm, M.C., Liu, K.R.: Forensic detection of image manipulation using statistical intrinsic fingerprints. *IEEE Trans. Inf. Forensics Secur.* **5**(3), 492–506 (2010)
35. Zhang, D., Liang, Z., Yang, G., Li, Q., Li, L., Sun, X.: A robust forgery detection algorithm for object removal by exemplar-based image inpainting. *Multimed. Tools Appl.* **77**(10), 11823–11842 (2018)
36. Zhao, Y.Q., Liao, M., Shih, F.Y., Shi, Y.Q.: Tampered region detection of inpainting JPEG images. *Optik* **124**(16), 2487–2492 (2013)
37. Liu, Q., Sung, A.H., Zhou, B., Qiao, M.: Exposing inpainting forgery in JPEG images under recompression attacks. In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 164–169. IEEE (2016)

38. Zhang, J., Liao, Y., Zhu, X., Wang, H., Ding, J.: A deep learning approach in the discrete cosine transform domain to median filtering forensics. *IEEE Sig. Process. Lett.* **27**, 276–280 (2020)
39. Nair, G., Venkatesh, K., Sen, D., Sonkusare, R.: Identification of multiple copy-move attacks in digital images using FFT and CNN. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6. IEEE (2021)
40. Amerini, I., Uricchio, T., Ballan, L., Caldelli, R.: Localization of JPEG double compression through multi-domain convolutional neural networks. In: 2017 IEEE Conference on computer vision and pattern recognition workshops (CVPRW), pp. 1865–1871. IEEE (2017)
41. Chen, H., Han, Q., Li, Q., Tong, X.: A novel general blind detection model for image forensics based on DNN. *The Vis. Comput.* 1–16 (2021)
42. Khan, M.J., Khan, M.J., Siddiqui, A.M., Khurshid, K.: An automated and efficient convolutional architecture for disguise-invariant face recognition using noise-based data augmentation and deep transfer learning. *The Vis. Comput.* 1–15 (2021)
43. Vinolin, V., Sucharitha, M.: Dual adaptive deep convolutional neural network for video forgery detection in 3D lighting environment. *Vis. Comput.* **37**(8), 2369–2390 (2021)
44. Lu, M., Niu, S.: A detection approach using LSTM-CNN for object removal caused by exemplar-based image inpainting. *Electronics* **9**(5), 858 (2020)
45. Wu, H., Zhou, J.: IID-Net: Image inpainting detection network via neural architecture search and attention. *IEEE Trans. Circuits Syst. Video Technol.* **32**(3), 1172–85 (2021)
46. Wang, X., Wang, H., Niu, S.: An image forensic method for AI inpainting using faster R-CNN. In: International Conference on Artificial Intelligence and Security, pp. 476–487. Springer (2019)
47. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Systems* **28** (2015)
48. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
49. Wang, X., Niu, S., Wang, H.: Image inpainting detection based on multi-task deep learning network. *IETE Tech. Rev.* **38**(1), 149–157 (2021)
50. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
51. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **7**(3), 868–882 (2012)
52. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1053–1061 (2018)
53. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(6), 1452–1464 (2017)
54. Chen, Y., Liu, L., Tao, J., Chen, X., Xia, R., Zhang, Q., Xiong, J., Yang, K., Xie, J.: The image annotation algorithm using convolutional features from intermediate layer of deep learning. *Multimed. Tools Appl.* **80**(3), 4237–4261 (2021)
55. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp. 1026–1034 (2015)
56. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Xinshan Zhu received the B.E. degrees and the M.E. degree in automation control from Harbin Institute of Technology, Harbin, China, in 2000 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. His research interests include deep learning, image processing, and multimedia security.



Junyan Lu received the B.E. degree in control science and engineering from Shandong University of Science and Technology, Qingdao, China, in 2020. Currently, he is pursuing the M.E. degree in control science and engineering from Tianjin University, Tianjin, China. His research interests include deep learning, image processing, and multimedia security.



Honghao Ren received the B.E. degree in automation from China University of Geosciences (Wuhan), Wuhan, China, in 2020. Currently, he is pursuing the M.E. degree in control science and engineering from Tianjin University, Tianjin, China. His research interests include deep learning, image processing, and multimedia security.



Hongquan Wang received the B.E. degree in electrical engineering and automation and the M.E. degree in control science and engineering from Tiangong University, Tianjin, China, in 2010 and 2015, respectively. Currently, he is pursuing the Ph.D. degree in control science and engineering from Tianjin University, Tianjin, China. His research interests include deep learning, image processing, and multimedia security.



Biao Sun received the Diploma in electrical information science and technology from Central South University, Changsha, China, in 2004, and the Ph.D. degree in electrical science and technology from Huazhong University of Science and Technology, Wuhan, China, in 2013. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. From 2015 to 2016, he was a Visiting Research Fellow with the Department of Ophthalmology, Yong Loo Lin School of Medicine, National University of Singapore, Singapore. His research interests include compressed sensing, machine learning, and brain computer interface.