



CTNet: hybrid architecture based on CNN and transformer for image inpainting detection

Fengjun Xiao¹ · Zhuxi Zhang² · Ye Yao²

Received: 31 March 2023 / Accepted: 4 September 2023 / Published online: 19 September 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Digital image inpainting technology has increasingly gained popularity as a result of the development of image processing and machine vision. However, digital image inpainting can be used not only to repair damaged photographs, but also to remove specific people or distort the semantic content of images. To address the issue of image inpainting forgeries, a hybrid CNN-Transformer Network (CTNet), which is composed of the hybrid CNN-Transformer encoder, the feature enhancement module, and the decoder module, is proposed for image inpainting detection and localization. Different from existing inpainting detection methods that rely on hand-crafted attention mechanisms, the hybrid CNN-Transformer encoder employs CNN as a feature extractor to build feature maps tokenized as the input patches of the Transformer encoder. The hybrid structure exploits the innate global self-attention mechanisms of Transformer and can effectively capture the long-term dependency of the image. Since inpainting traces mainly exist in the high-frequency components of digital images, the feature enhancement module performs feature extraction in the frequency domain. The decoder regularizes the upsampling process of the predicted masks with the assistance of high-frequency features. We investigate the generalization capacity of our CTNet on datasets generated by ten commonly used inpainting methods. The experimental results show that the proposed model can detect a variety of unknown inpainting operations after being trained on the datasets generated by a single inpainting method.

Keywords Image inpainting detection · Deep neural network · Hybrid CNN-Transformer encoder · High-pass filter

1 Introduction

Image inpainting is the process of recovering an image from a degraded version according to the residual image information. Digital image inpainting technology emerged in the early twentieth century, and was initially used to restore ancient artworks, cultural relics, and paintings. With the explosive growth of digital information, image inpainting technology is now flexibly used on various occasions, including image retouching, image denoise [1], watermark erasing [2], film special effects, and background beautification.

In recent years, digital image inpainting technology has become a research hotspot [3–10], which can be broadly classified into traditional image inpainting technology and deep learning-based image inpainting technology. Traditional digital image inpainting techniques [3–6] include diffusion-based methods and exemplar-based methods. Traditional digital image inpainting techniques are limited by the use of local diffusion information and can not create complex textures or semantic components [11]. To learn semantic features of the image, deep learning-based image inpainting technologies [7–10] train inpainting models on large-scale datasets and achieve impressive results across a wide range of application scenarios [12]. Benefiting from the use of adversarial networks, these methods can adapt to different resolutions and generate missing objects automatically.

However, the image inpainting technique acts as a two-edged sword. One can either use image inpainting to restore the integrity of the image or tamper with the image by filling in the gaps left by object removal, as shown in Fig. 1. With the development of image editing equipment, the tampering

✉ Ye Yao
yaoye@hdu.edu.cn

¹ Zhejiang Informatization Development Institute, Hangzhou Dianzi University, Xiasha Higher Education Zone, Hangzhou 310018, Zhejiang, China

² School of Cyberspace, Hangzhou Dianzi University, Xiasha Higher Education Zone, Hangzhou 310018, Zhejiang, China

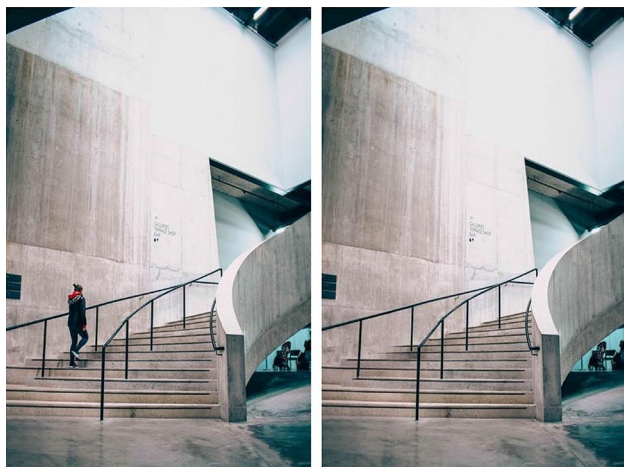


Fig. 1 Examples of image inpainting. Left is the original image, and the right is the inpainted image

traces hidden in inpainted images become more difficult to be identified by human eyes. If image inpainting technologies are maliciously used in news media, judicial forensics, or to mislead the public, they will cause great harm to people's daily life and bring about social and political instability. To distinguish the authenticity of images, the in-depth research on digital image inpainting detection has attracted the attention of many researchers.

Early image inpainting detection methods mainly identify inpainting regions by calculating the similarity features between image blocks [13–17]. As convolutional neural networks become more prominent in computer vision, more researchers have turned to inpainting detection models incorporating convolutional neural networks [18–24]. These deep learning-based image inpainting detection models show apparent advantages in extracting image inpainting traces and reducing false alarm rates.

Although deep learning-based approaches have yielded positive results in image inpainting detection, they often have issues such as the lack of generalizability and relying on specific features and data distribution [25]. Current inpainting forensic technologies generally use CNN as the feature extractor, which tends to learn content features from the image, and it is easy to ignore tiny artifacts incurred during the inpainting process. At the same time, the convolution and downsampling operations in the forensic network may bring about information loss in feature representation, and the bilinear upsampling can also blur the precise mask predicted by the network.

In view of the above shortcomings, this paper constructs a hybrid CNN-Transformer network, which consists of the hybrid CNN-Transformer encoder, the feature enhancement module, and the decoder module, for image inpainting detection and localization. The key idea of the hybrid

CNN-Transformer encoder is to capture long-range dependencies of images through self-attention mechanism of Transformer. The feature enhancement module utilizes hierarchically combined layers to extract deep frequency features, which the decoder incorporates into the upsampling process of the predicted mask as extra supervision to improve the generalization capability. We optimize the CTNet framework using a hybrid loss function consisting of pixel-level and image-level loss, reducing the impact of class imbalance in the training dataset. Lastly, we provide a comparative analysis to investigate both the performance and generalization of our model.

The main contributions of our work include the followings:

- 1) We introduce an encoder with a hybrid CNN-transformer architecture for image inpainting detection, which makes up for the defect of Transformer exclusively focusing on modeling the global context. The hybrid encoder takes full advantage of the characteristics of CNN and transformer to extract local and global inpainting features.
- 2) The high-frequency features are applied to supervise the upsampling process of the extracted feature map output by the hybrid encoder, which results in better accuracy in detecting inpainted regions.
- 3) We tackle the challenge of constructing an image inpainting detection method that attains good generalizability for a variety of unseen inpainting operations and, thus, is not limited to a specific inpainting method.

Experimental results show that our model achieves state-of-the-art image inpainting detection performance on the test set generated by ten commonly used inpainting methods. The comparative experiments demonstrate the effectiveness and generalization ability of our approach.

The rest of the paper is organized as follows. Section 2 briefly reviews several works related to image inpainting approaches, inpainting detection, and Transformer. Section 3 presents the details of our method. Experimental results are discussed in Section 4, including the experimental results of our model and comparison with state-of-art methods. Section 5 makes a summary of our work.

2 Related work

2.1 Image inpainting

Many methods have been proposed for image inpainting, including traditional methods based on diffusion or patches [26–34], and those based on deep learning [35–43]. Bertalmio *et al.* [26] proposed the first diffusion-based approach in 2000, which smoothly propagated the image information

along the isophote to fill the missing region. Telea *et al.* [31] proposed an inpainting algorithm based on propagating an image smoothness estimator along the image gradient, which was simple to implement. Bertalmio *et al.* [32] involved Navier–Stokes equations for an incompressible fluid, which had the benefit of well-developed numerical results. Herling *et al.* [33] presented an approach based on high-quality image inpainting and enabled the realization of Diminished Reality applications. Huang *et al.* [34] proposed an algorithm for automatically guiding patch-based image completion leveraging mid-level structural cues.

Traditional image inpainting methods use the internal statistical information of the image, such as the edges of damaged parts or similar image patches, which lack global semantic information and can not produce non-continuous contents. To solve the shortcomings of traditional methods, image inpainting technology based on deep learning has gradually become a research hotspot. Deep learning-based techniques for image inpainting combine global semantic and texture information, significantly improving inpainting effects. Yu *et al.* [39] presented a novel generative network with a contextual attention layer, which can not only synthesize image structures but also use neighboring images features. Yan *et al.* [40] introduced a shift-connection to U-Net, which exhibited fast speed with fine details by deep feature rearrangement. Nazeri *et al.* [41] developed a two-stage adversarial model that comprised of an edge generator and an image completion network which can be used as an interactive image editing tool. To tackle the challenge of visually plausible results caused by deep learning-based methods, Wu *et al.* [42] suggested an end-to-end generative model through combining a local binary pattern. Considering the impact of the corrupted regions of the image on normalization, Yu *et al.* [43] proposed a spatial region-wise normalization that divided pixels into small regions and computed the mean and variance for normalization.

2.2 Inpainting detection

Many inpainting detection approaches have been developed to prevent the malicious use of inpainting operations. Wu *et al.* [13] proposed a blind detection method based on zero-connectivity features and fuzzy membership. However, the semi-automatic detection method requires manual selection of suspicious areas, which requires a large amount of computation. Bacchuwar *et al.* [14] improved the algorithm of [13] by converting the image to Y-Cb-Cr format and using only the luminance component Y for detection, which speeds up the computation. Aiming at the problem that the uniform region in the image background will interfere with the detection process, Chang *et al.* [15] applied a multi-region relation technique to identify suspicious image patches from homogeneous regions.

Zhu *et al.* [16] built a convolutional neural network to detect patch-based inpainting operations. The authors constructed a class label matrix for each pixel of the image in the process of training the encoder–decoder network. Li *et al.* [18] employed a fully convolutional network based on high-pass-filtered image residuals, which enhanced the difference between the inpainted and untouched region. Considering the spatial and channel correlations of feature maps, Xiao *et al.* [44] introduced a squeezed excitation block, which is applied in the feature extraction and upsampling stage to pay more attention on the spatial location and channel dependence. Zhang *et al.* [19] used a modified U-shaped feature pyramid network (FPN) to extract multi-scale inpainting features. Wang *et al.* [20] used MASK R-CNN [45] combined with FPN to extract features, which can detect images tampered by conventional inpainting methods and images modified by deep learning-based methods. Li *et al.* [21] proposed a method for generating a universal training dataset, which imitates the noise pattern discrepancies between the real and synthesized contents to train universal detectors. Chen *et al.* [22] proposed a multi-view feature learning with multi-view supervision network that contained novel elements designed for learning semantic-agnostic features. Wu *et al.* [23] introduced MT-Net, a unified deep neural architecture that can handle images of arbitrary sizes and many known forgery types. MT-Net detects forged pixels by identifying local anomalous features, thus it also performs well in image inpainting detection. The forensic model proposed by Wu *et al.* [24] consists of three parts, namely feature enhancement, feature extraction, and decision block. Their model is designed with the assistance of the NAS algorithm and the embedded attention modules to optimize the latent high-level features. Yang *et al.* [46] provides a near original image augmentation strategy to push the inpainted images closer to the original images. The authors add hard samples into the training set and as a result help improve the accuracy of their model.

However, many state-of-the-art image inpainting detection methods have problems such as heavy computation and time-consuming pre-processing, most of which also suffer from poor generalization ability. To address these issues, we propose a pixel-level inpainting detection method based on a hybrid CNN-Transformer encoder, which reduces computation and enhances the generalization ability to unseen inpainting methods.

2.3 Transformer

Transformer [47] is a classic NLP model proposed by the Google team in 2017 and has been widely used in NLP and machine translation. Based on the encoder-decoder architecture, Transformer ultimately uses self-attention mechanism which replaces the temporal structure of the recurrent

neural network. Generally, Transformer-based models are trained on large text corpus and fine-tuned for specific tasks to achieve better computational efficiency and accuracy. Due to its remarkable ability to supporting parallel processing of sequences, the Transformer architecture has proven its potential in reducing training time significantly. Consequently, this technique has since become the state-of-the-art approach in many NLP tasks.

Inspired by Transformer, the Vision Transformer (ViT) proposed by Dosovitskiy *et al.* [48] breaks the isolation between NLP and computer vision (CV), and successfully applies Transformer to image block sequences for further image classification tasks. The pioneering work of ViT model uses a self-attention mechanism to capture global features from shallow networks, which solves the problem of CNN’s difficulty in capturing and storing long-range dependency information. When pre-trained on large-scale datasets and transferred to multiple mid-sized or small image recognition datasets (ImageNet [49], CIFAR-100 [50], VTAB [51], etc.), ViT demonstrates superior transferability on downstream tasks.

There are many follow-up studies [52–56] extending ViT. For example, Touvron *et al.* [52] introduced the pyramid structure into the Transformer, which performed better on dense prediction tasks. Wang *et al.* [53] proposed a teacher–student distillation training strategy for ViT, and added a distillation token as supplementary information for the classification token. The above studies reveal

the effectiveness of Transformer in computer vision tasks; thus, we propose an image inpainting detection method that combines Transformer and CNN encoder.

3 Proposed method

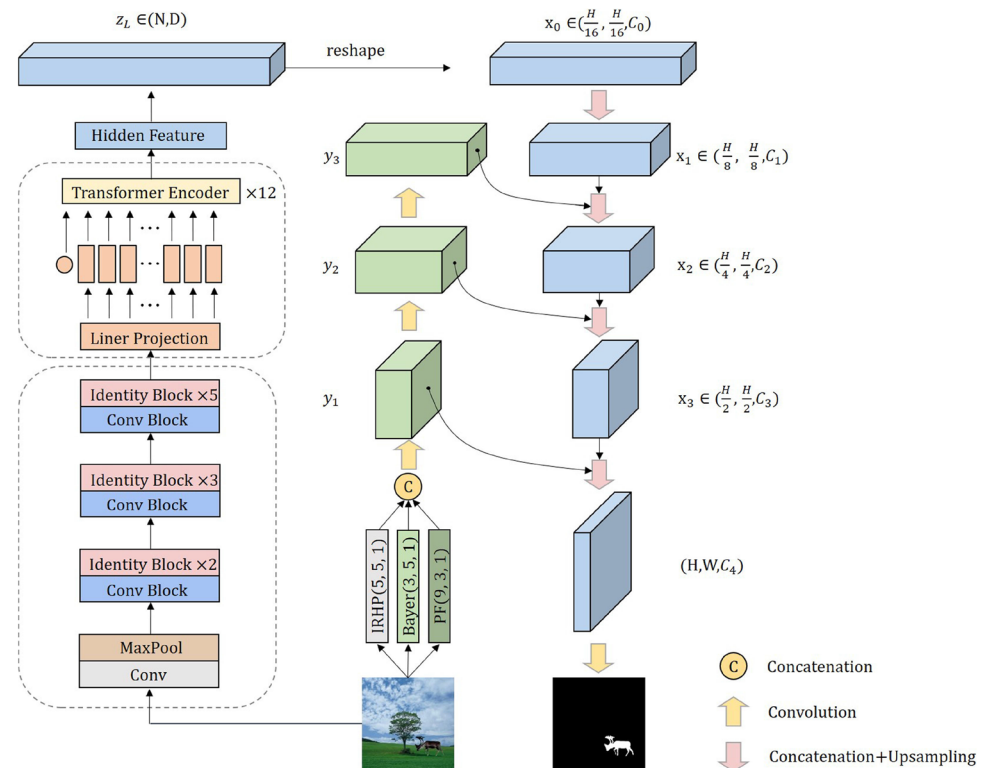
In this study, we propose an image inpainting detection method based on the CNN-Transformer hybrid structure (abbreviated as CTNet). On the one hand, it effectively resolves the issue that a CNN network must transmit information onto the subsequent layers and cannot capture the long-term dependence relationship. On the other hand, the decoder upsamples the features encoded by the Transformer and incorporates residual noise as an auxiliary information into the upsampling process to improve the localization accuracy. The overview of the CTNet architecture is presented in Fig. 2. We elaborate three main blocks of the CTNet, i.e., the hybrid CNN-Transformer encoder, the feature enhancement module and the decoder module.

3.1 Hybrid CNN-Transformer encoder module

3.1.1 Convolution block

The original ViT encoder aggregates global features and ignores image details at low resolutions when used for classification tasks. This property is not conducive to

Fig. 2 The architecture of the proposed CTNet



reconstructing the full resolution image during the upsampling process, resulting in rough edges output by the decoder. Consequently, we first encode the image into high-level feature representations using a CNN, which are then reshaped into a 2D sequence as the input of the ViT encoder.

The CNN part of the hybrid encoder is shown in Fig. 3. Given an RGB image $x \in \mathbb{R}^{H \times W \times C}$ with 3 channels and its corresponding binary mask y , we first feed image x into the ResNet50 [57] network pre-trained on the ImageNet dataset to extract the output feature representations of the fourth layer.

The stage 0 in Fig. 3 is the first layer of the ResNet 50 network, which performs convolution, regularization, activation functions, and max pooling on the input image. The stage 1–stage 3 in Fig. 3 are the second–fourth layers of the ResNet 50 network, each of them contains two residual blocks with skip connection, namely Conv Block and Identity Block. The Conv Block is applied to change the dimension of the feature maps, while the Identity Block corresponds to the case where the input has the same dimension as the output. The utilization of the residual module solves the problem of gradient vanishing or exploding when training deep neural networks, and it can also preserve and ensure data integrity to a certain extent.

The output feature representations of the first–fourth layers of ResNet50 are: $F_0 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 64}$, $F_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 256}$, $F_2 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 512}$, $F_3 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 1024}$. We employ F_3 as the input of the Transformer encoder.

3.1.2 Transformer Encoder Block

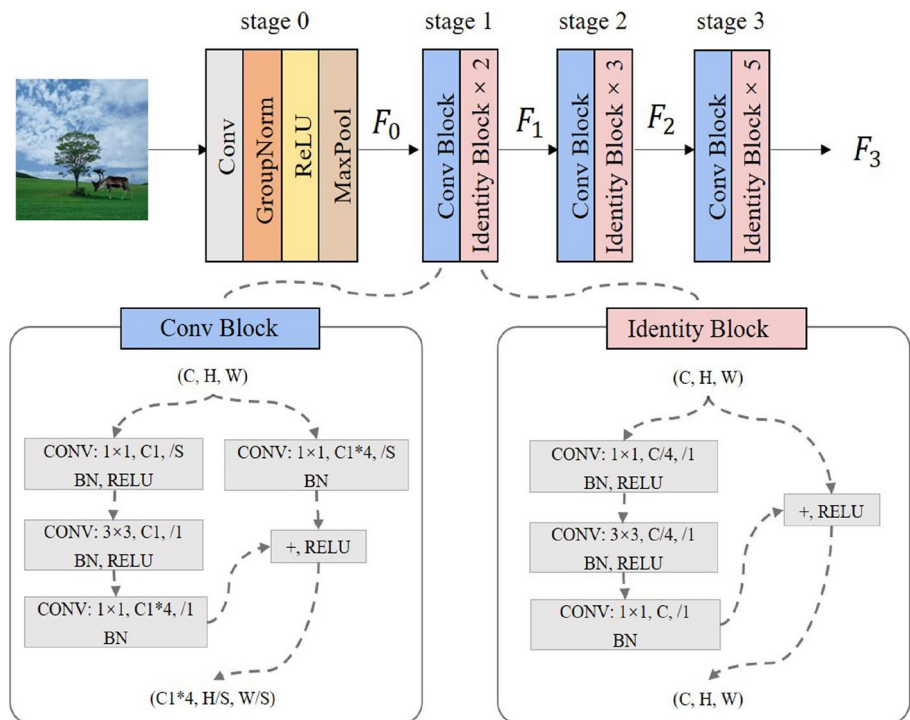
The Transformer encoder employed in ViT first splits the input image $x \in \mathbb{R}^{H \times W \times C}$ into N non-overlapping patches of the size $P \times P$, obtaining a flattened 2D sequence: $\{x_p^i \in \mathbb{R}^{P^2 \cdot c} \mid i = 1, 2 \dots N\}$, where sequence length $N = \frac{H \times W}{P^2}$. Then, the encoder uses a trainable linear projection to map the serialized x_p into the D -dimensional embedding space. To obtain the 1×1 patch sequence z_0 , each block is randomly initialized and embedded through the convolution layer with a kernel size of $P \times P$:

$$z_0 = [x_{cls}; x_p^1 \mathbf{E}; x_p^2 \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}_{pos}, \tag{1}$$

where z_0 is the 1×1 patch sequence input to the L -layer Transformer encoder after linear mapping and position embedding, x_{cls} is a class token that gathers information from all the patches, $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ denotes the trainable linear projection, and $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \cdot D}$ denotes the position embedding.

The Transformer encoder consists of L layers of Multi-head Self-Attention (MSA) and Multi-Layer Perceptron (MLP) modules, as shown in Fig. 4. For the l -th layer of the encoder, we denote its input as $z_{\ell-1}$ and the output as z_{ℓ} , and apply Layer Normalization (LN) before each MSA

Fig. 3 The CNN part of the hybrid encoder, (C, H, W) represents the input dimension. C and C1 are the number of convolution kernels, and S is the stride. In stage 1, C = C1, stage 2, 3, 4, C = 2 × C1



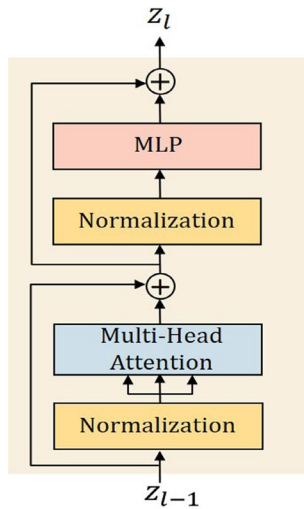


Fig. 4 One layer of the Transformer encoder. Each consists of MSA and MLP blocks, with LN applied before every block, and residual connection applied after every block

and MLP block. The residual connection is applied after the MSA and MLP blocks as follows:

$$\begin{aligned} \mathbf{z}'_{\ell} &= \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \\ \mathbf{z}_{\ell} &= \text{MLP}(\text{LN}(\mathbf{z}'_{\ell})) + \mathbf{z}'_{\ell}. \end{aligned} \tag{2}$$

In this paper, we utilize the output feature map $F_3 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 1024}$ of the convolution module as the input of the Transformer encoder. We first perform linear projection on F_3 to convert it into a D ($D = 768$) dimensional sequence. Next, patch embedding is applied to get 1×1 patches and then input to the L ($L = 12$) layer Transformer encoder for feature extraction. The final output feature map of hybrid encoder is $\mathbf{z}_L \in (\mathbb{N}, D)$.

3.2 Feature enhancement module

The key to detecting inpainting areas in image detection is to obtain the tampering traces. For the inpainting traces mainly exist in the high-frequency components of digital images, the first step in inpainting detection is to suppress contents of input images and highlight inpainting traces. A common practice is to use high-pass filters to obtain high-frequency residuals, such as PF (pre-filtering) filter [18], Bayer filter [58], IRHP (Improved Random High-Pass) filter [59].

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 5 The initialized convolution kernels for the PF filter

As shown in Fig. 5, the PF filter proposed by Li *et al.* [18] employs a learnable pre-filtering module to extract the residual noise of the image. The PF filter convolves each channel of the three-channel image with a 3×3 high-pass filter kernel with a stride of 1. Then, the obtained noise residuals are concatenated as the input of the subsequent network.

Bayar *et al.* [58] proposed a constrained convolution structure to suppress image content and can adaptively learn low-level forensic features. The authors add additional constraints to the first convolutional layer filter of the deep neural network to capture the changes in the dependencies among neighboring pixels caused by inpainting operation. The constraints of the first convolution layer are as follows:

$$\begin{cases} \mathbf{w}_k^{(1)}(0, 0) = -1 \\ \sum_{m,n \neq 0} \mathbf{w}_k^{(1)}(m, n) = 1, \end{cases} \tag{3}$$

where $\mathbf{w}_k^{(1)}(m, n)$ represents the weight of the k -th convolution kernel at position (m, n) in the first convolutional layer, and $\mathbf{w}_k^{(1)}(0, 0)$ is the center position of the convolution kernel. The weights at the center of the convolution kernel are initialized to -1, and then the weights on the surrounding positions are normalized so that their sum equals 1. During the training phase, the constrained convolutional layer updates the weights through backpropagation after each iteration, enabling the network to extract image manipulation features adaptively.

IRHP [59] is an improved random high-pass filter initialization method that reduces computational costs while maintaining data flow stability at kernel's input and output. During the weight initialization of the CNN's first layer, a group of random high-pass filters are generated, and the data flow is kept stable at the input and output of the convolution kernel. Precisely, a simple high-pass filter template of size 5×5 is depicted in Fig. 6:

where $w_{i,i=1,2,\dots,N}$ are scalar random variables following the simple uniform distribution, C is an unknown constant. To initialize the convolution kernel as a high-pass filter, the mathematical expectation of $w_{i,i=1,2,\dots,N}$ should be equal to $-\frac{C}{N}$, which compensates for the unknown constant C . Accordingly, the interval of the uniform distribution for the sampling of w_i is $U\left(-\frac{2C}{N}, 0\right)$ or $U\left(0, -\frac{2C}{N}\right)$, depending on

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \\ w_6 & w_7 & w_8 & w_9 & w_{10} \\ w_{11} & w_{12} & C & w_{13} & w_{14} \\ w_{15} & w_{16} & w_{17} & w_{18} & w_{19} \\ w_{20} & w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ x_6 & x_7 & x_8 & x_9 & x_{10} \\ x_{11} & x_{12} & x_{25} & x_{13} & x_{14} \\ x_{15} & x_{16} & x_{17} & x_{18} & x_{19} \\ x_{20} & x_{21} & x_{22} & x_{24} & x_{25} \end{bmatrix}$$

Fig. 6 Left is the initialization weight of the IRHP convolution kernel, and the right is the input feature matrix of the CNN

the value of C . From this distribution, we can calculate the variance of w_i as:

$$\text{Var}(w_i) = \frac{C^2}{3N^2}. \quad (4)$$

With the assumption of $x_{i,i=1,2,\dots,N}$ are mutually correlated random variables and $w_{i,i=1,2,\dots,N}$ are random variables with a non-zero expectation, the authors proved that the variance of w_i should be equal to $\frac{1}{N}$. Combining the above two equations, we can derive the value of C as:

$$\text{Var}(w_i) = \frac{C^2}{3N^2} = \frac{1}{N} \Rightarrow C = \pm\sqrt{3N}. \quad (5)$$

In the case of a 5×5 filter, $3N = 3 \times (5 \times 5 - 1) = 72$, $C = \pm\sqrt{72} \approx \pm 8.4853$. Because the distribution of w_i is $U\left(-\frac{2C}{N}, 0\right)$ or $U\left(0, -\frac{2C}{N}\right)$, the uniform distribution interval of w_i is $U(-0.7071, 0)$ or $U(0, 0.7071)$ (Fig. 7). Fig. 8 shows a randomly initialized IRHP convolution kernel:

Extensive experiments are conducted to find proper combinations of different high-pass filters. We find that the combination of PF filter, Bayer filter, and IRHP filter performs best on the test dataset. Therefore, we hierarchically combine PF filter, Bayer filter and IRHP filter as three sets of filters to extract image noise residual. We first input the three-channel image $x \in \mathbb{R}^{H \times W \times C}$ into the PF filter, Bayer filter and IRHP filter, respectively. Then, the enhanced features are concatenated along the channel to obtain $y_c \in (H, W, 15)$. Next, y_c is passed through three standard convolutional layers. Each convolutional layer includes convolution, batch normalization, and activation functions. The outputs of each layer $y_1 \in \left(\frac{H}{2}, \frac{W}{2}, C_3\right)$, $y_2 \in \left(\frac{H}{4}, \frac{W}{4}, C_2\right)$, $y_3 \in \left(\frac{H}{8}, \frac{W}{8}, C_1\right)$ are used to supervise the upsampling process of the subsequent decoder network.

3.3 Decoder module

For the feature representation \mathbf{z}_L output by the encoder, the proposed decoder first reshapes it utilizing two convolution layers, and converts the hidden feature $\mathbf{z}_L \in (N, D)$ to $\mathbf{x}_0 \in \left(\frac{H}{16}, \frac{W}{16}, C_0\right)$. Then, we adopt the simple progressive

$$\begin{bmatrix} -0.2313 & -0.0666 & -0.1706 & -0.5250 & -0.5772 \\ -0.5182 & -0.0018 & -0.5467 & -0.2407 & -0.0559 \\ -0.4711 & -0.5040 & 8.4853 & -0.5513 & -0.3093 \\ -0.4237 & -0.1851 & -0.6255 & -0.5770 & -0.5398 \\ -0.2109 & -0.6927 & -0.6765 & -0.1684 & -0.0787 \end{bmatrix}$$

Fig. 7 Example of a 5×5 IRHP convolution kernel. The center value $C=8.4853$, $w_{i,i=1,2,\dots,N}$ is a pseudo-random initialization parameter following the uniform distribution $U(-0.7071, 0)$

upsampling approach to export the prediction mask with with same resolution as the original image. Each upsampling block consists of three stages. The length and width of the original feature map are doubled by bilinear interpolation, and concatenated with three features output by the feature enhancement module along the channel, respectively. Then, two standard convolution layers are used to reduce the number of channels.

Specifically, the feature maps $\mathbf{x}_1 \in \left(\frac{H}{8}, \frac{W}{8}, C_1\right)$, $\mathbf{x}_2 \in \left(\frac{H}{4}, \frac{W}{4}, C_2\right)$, $\mathbf{x}_3 \in \left(\frac{H}{2}, \frac{W}{2}, C_3\right)$ are concatenated with three features $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ output by the feature enhancement module to regularize the upsampling operation of the predicted mask. Finally, the feature map obtained by the concatenation of \mathbf{x}_3 and \mathbf{y}_1 goes through a convolution and softmax layer, outputting the predicted mask with the same size as the input image.

3.4 Loss function

We consider a pixel-scale loss for enhancing the model's sensitivity for detecting pixel-level inpainting, and an image-scale loss to reduce the false alarm rate. Typically, the missing regions of the inpainted images tend to be much smaller than the original regions, and merely using cross-entropy loss results in a low true-positive rate for the model to correctly identify tampered regions. Focal loss [60] is applied to make up for the defect of cross-entropy loss. Focal loss is a variant of cross-entropy loss:

$$L_{\text{Focal}}(y, \hat{y}) = - \sum \alpha(1 - \hat{y})^\gamma * y \log(\hat{y}) - \sum (1 - \alpha)\hat{y}^\gamma * (1 - y) \log(1 - \hat{y}), \quad (6)$$

where y is the ground-truth mask of the digital image, \hat{y} is the predicted mask output by the proposed CTNet, $\alpha, \gamma \in (0, 1)$ are hyperparameters. In addition to the pixel-scale loss, we also employ an image-level binary cross-entropy loss [22] L_{Cls} to reduce the false alarm rate:

$$L_{\text{Cls}}(y, \hat{y}) = -y * \log G(\hat{y}) + (1 - y) * \log(1 - G(\hat{y})), \quad (7)$$

where $G(\hat{y})$ is the output value of the predicted image mask \hat{y} after global max pooling, $G(\hat{y})$ should tend to be zero when the input image is untampered. In summary, we use a combination of pixel-level loss and image-level loss:

$$L(y, \hat{y}) = \lambda_1 L_{\text{Focal}}(y, \hat{y}) + \lambda_2 L_{\text{Cls}}(y, \hat{y}), \quad (8)$$

where $\lambda_1, \lambda_2 \in (0, 1)$ are regularization hyperparameters.

4 Experiments and results

4.1 Datasets

Following the training and test set configuration of [24], the training set we chose contains a total of 48k inpainted images sampled from Dresden [61] and Places [62] datasets. The test set contains 10k images sampled from Dresden, Places, CelebA and ImageNet datasets to conduct various experiments. The training images are modified by the inpainting method proposed by [7]. The test images are inpainted by ten representative inpainting approaches, among which four are conventional methods, namely TE [31], NS [32], PM [33] and SG [34], and the remaining six methods are based on deep learning, which include GC [7], CA [39], SH [40], EC [41], LB [42], and RN [43]. Each of the above inpainting methods modifies 1k original images, and we obtain a total of 10k inpainting images and corresponding masks of the test set. The inpainting shapes include basic rectangles, circles, ellipses, etc.

To enhance the model robustness, we adopt following data augmentation methods for the training images:

- **Resize.** The resolution of all images in the training set is adjusted to 256×256.
- **Rotation.** Half of the training samples are flipped horizontally, and the others are flipped vertically.
- **Normalization.** We perform normalization on three channels of the input RGB image, mapping pixel values to a range between zero and one.

4.2 Implementation details

The experimental platform in this research is shown in Table 1.

We deploy the stochastic gradient descent(SGD) as the optimizer and set the learning rate, momentum, and weight decay as 0.001, 0.9, and $1e-6$ to update the CTNet. During the training process, the CTNet deploys ResNet-50 and ViT as backbone networks of the hybrid encoder, both of which are pre-trained on ImageNet dataset. The average F1 score and AUC (Area Under the Receiver operator characteristic

curve) are selected as indicators to evaluate our model's performance.

4.3 Benchmark experiments

A series of experiments are conducted to evaluate the detection performance of the CTNet comprehensively. In benchmark experiments, we compare CTNet with four state-of-the-art image inpainting detection methods on the test set. The involved detection algorithms include MVSS-Net [22], ManTra-Net [23], HP-FCN [18] and IID-Net [24]. The benchmark experiments in this paper include testing directly using the pre-trained model officially released by the original paper, and retraining on our training dataset [24] before testing with 10 unseen inpainting methods. The training strategies and parameter settings are consistent with the original paper.

Table 2 shows the comparison results using AUC and pixel-level F1 scores as critics. In the "Retrain" column, "GC" indicates that the models are retrained on the training set synthesized by GC [7], while "-" means the pre-trained models are directly applied to evaluate the detection performance on the test set.

We can observe that our proposed CTNet has achieved the best detection results on the test set synthesized by 10 unseen inpainting methods. The average AUC value and F1 score have reached 99.11% and 89.71%, leading the second-placed IID-Net model by 0.7% and 2.14%. The advantage of CTNet is more prominent with F1 score, indicating its performance on heterogeneous images is very stable.

In benchmark experiments, it is not difficult to find that the pre-trained HP-FCN and MVSS-Net models have poor performance when directly used for testing. At the same time, their average AUCs are greater than 50%, and the binary classification results are higher than random predictions. However, their average F1 scores are lower than 1%, indicating that these methods are only designed for specific image inpainting processes, which causes problems such as inability to detect unexpected image inpainting methods and poor generalization ability.

We also give some examples of the detection results in Fig. 8. It can be seen that the proposed algorithm obtains the most accurate results comparing with other deep learning-based digital image inpainting detection algorithms. Because our CTNet adopts a hybrid structural encoder that simultaneously considers the global information and high-resolution representation of the image, it has a stronger inpainting feature extraction ability than the CNN encoder used by our competitors. In the meantime, CTNet also shows great adaptability to unseen inpainting algorithms.

Table 1 Experimental environment

Item	Configuration
CPU	E5-2620 v4
GPU	GTX1080Ti
System	Ubuntu 16.04
RAM	128G
Framework	Pytorch 1.4.0

Table 2 Benchmark experimental results

Models	Retrain	Metrics	Test Dataset	Conventional Inpainting methods														Mean
				Deep learning based Inpainting methods							Conventional Inpainting methods							
				GC [7]	CA [39]	SH [40]	EC [41]	LB [42]	RN [43]	TE [31]	NS [32]	LR [4]	PM [33]	SG [34]				
MVSS-Net [22]	-	AUC		54.65	51.47	55.29	51.28	52.01	56.03	83.43	55.29	83.44	73.51	59.40	61.07			
MVSS-Net [22]	GC	AUC		93.42	89.39	97.53	77.16	95.83	94.75	93.24	95.28	99.31	97.74	95.43	93.55			
MT-Net [23]	-	AUC		73.29	82.06	93.99	89.18	92.75	85.00	97.50	99.12	98.82	93.74	96.33	91.07			
MT-Net [23]	GC	AUC		96.31	75.44	73.58	61.92	62.27	87.38	90.93	89.03	97.12	90.59	86.09	82.79			
HP-FCN [18]	-	AUC		50.18	50.22	55.84	50.02	50.67	50.01	63.30	60.53	62.59	48.02	50.26	53.79			
HP-FCN [18]	GC	AUC		96.65	87.50	98.14	74.51	96.51	96.59	92.27	97.18	99.18	98.64	99.78	94.26			
IID-Net [24]	GC	AUC		96.77	95.39	99.67	98.12	99.80	99.71	96.12	97.65	99.79	99.54	99.94	98.41			
Ours	GC	AUC		98.25	98.22	98.52	98.33	99.71	99.83	99.05	98.69	99.87	99.82	99.91	99.11			
MVSS-Net [22]	-	F1		0.81	0.82	0.84	0.81	0.81	0.83	0.93	0.84	0.93	0.97	0.93	0.87			
MVSS-Net [22]	GC	F1		61.04	46.23	70.30	66.22	49.78	40.82	84.83	81.46	85.61	66.53	75.94	66.25			
MT-Net [23]	-	F1		14.17	28.80	72.63	67.55	60.14	35.22	82.31	90.67	81.35	49.98	66.93	59.07			
MT-Net [23]	GC	F1		92.10	19.02	32.78	10.62	2.38	10.80	83.23	86.75	27.37	13.11	45.84	30.17			
HP-FCN [18]	-	F1		0.04	0.22	0.38	0.05	0.42	1.98	0.14	0.61	0.92	0.08	0.01	0.44			
HP-FCN [18]	GC	F1		76.93	35.75	81.43	8.57	55.78	56.58	41.05	44.13	50.91	24.66	73.55	51.76			
IID-Net [24]	GC	F1		83.61	81.46	94.13	87.95	96.14	94.41	82.47	85.27	87.28	75.74	94.78	87.57			
Ours	GC	F1		88.78	85.12	95.68	91.36	94.06	95.05	83.71	85.49	90.54	82.82	94.15	89.71			

The best performance are in bold

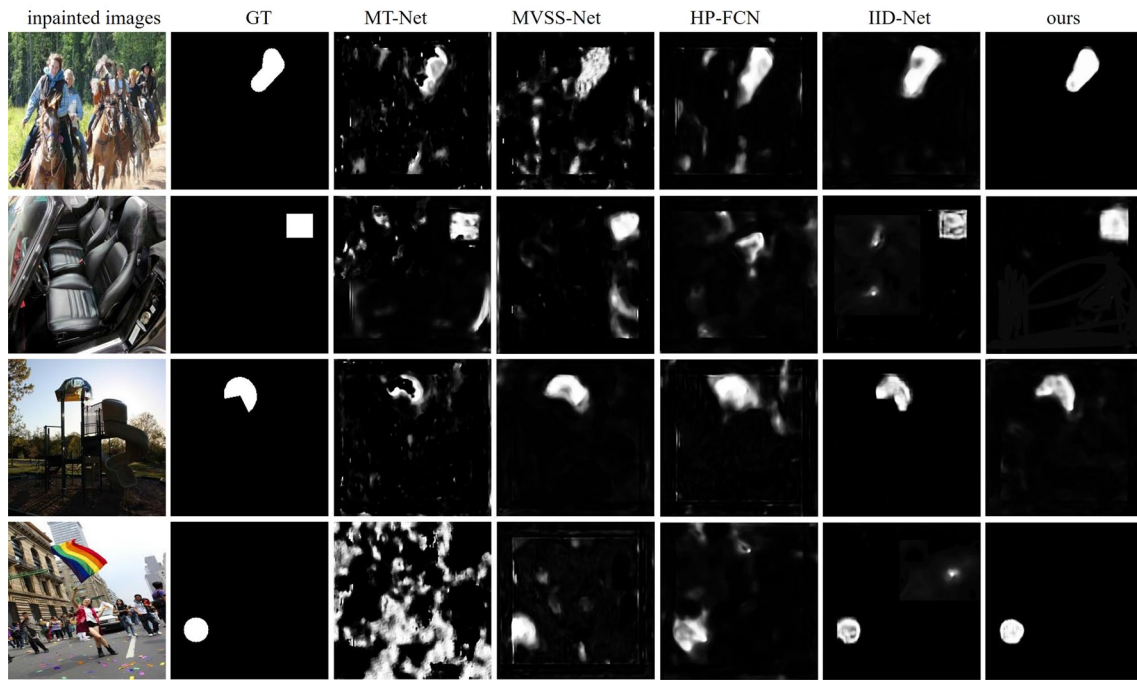


Fig. 8 Examples of the localization maps for inpainting detection

4.4 Ablation study

To verify the effectiveness of individual components on the model performance, ablation experiments are carried out in this paper. We modify some network structures in varied settings with the modules added progressively. First, we conduct experiments only utilizing one high-pass filter of the feature enhancement module. Second, we only use the CNN or Transformer module of the hybrid encoder. Third, we drop the image-level loss in the hybrid loss function. The results of the ablation experiment are shown in Table 3.

The average AUC values and F1 scores are shown in Table 3. In the part of feature enhancement module, the model utilizing a single high-pass filter cannot achieve good detection results. Experiments show that using multiple high-pass filters can effectively enhance the trace introduced by inpainting operations, significantly improving both mean AUC and F1 scores.

In the hybrid architecture encoder network part, employing only CNN or Transformer as the encoder caused the average AUC and F1 score to drop. In contrast, the hybrid encoder that combined the two improved the F1 score by 5%, proving that the hybrid encoder can effectively take high-resolution features and global context information into

Table 3 Ablation experiment results

Feature enhancement module	IRHP	✓					
	PF		✓				
	Bayer			✓			
	IRHP+PF+bayer				✓	✓	✓
Hybrid CNN-transformer encoder	Convolution block	✓	✓	✓	✓		
	Transformer encoder					✓	
	Hybrid encoder						✓
Loss function	Focal loss	✓	✓				
	Image-level loss			✓	✓		
	Hybrid loss					✓	✓
Mean AUC		91.38	90.27	93.41	98.02	97.33	99.11
Mean F1 score		84.19	82.44	83.57	85.13	84.66	89.71

The best performance are in bold

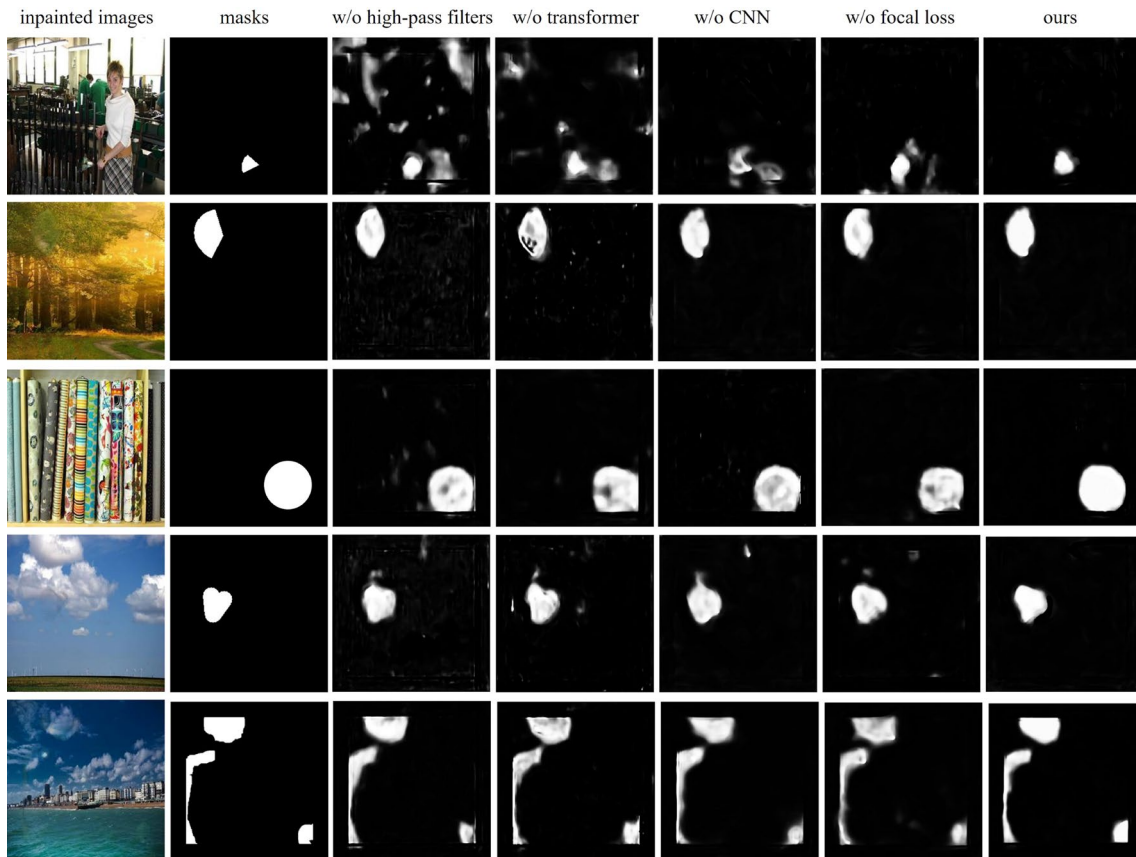


Fig. 9 Exemplar visual comparisons on test dataset, where first two columns are inpainted images and corresponding ground truth, the third to seven columns are results based on different settings

account. In the loss function part, adding image-level BCE loss on the basis of Focal Loss can effectively reduce the false alarm rate.

The qualitative results of pixel-level inpainting detection in varied setups is indicated in Fig. 9. For each row, the images from left to right are inpainted image, ground truth, without high-pass filter module, without transformer, without CNN, without focal loss and our model. From the third to seven columns of Fig. 9, we can observe that the combination of hybrid structure encoder, feature enhancement module and decoder can more precisely detect and locate inpainted regions. In summary, our model achieves a good balance between detection sensitivity and specificity.

4.5 Robustness analysis

Image inpainting operations in real life are often accompanied by post-processing operations, which may diminish model performance in real-world detection scenarios. To further verify the robustness of CTNet against image post-processing operations, we perform the following three

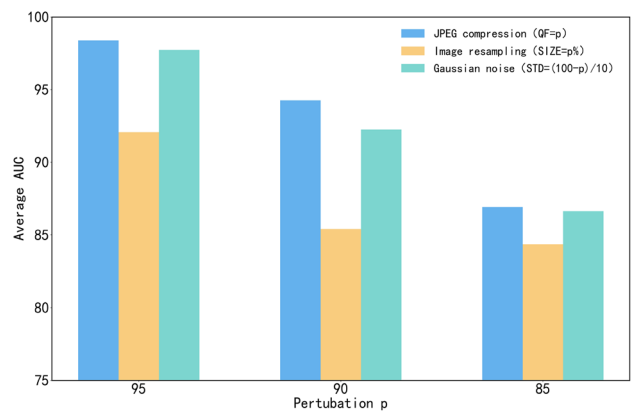


Fig. 10 Inpainting detection performance of our model under various disturbances in terms of average AUC values

perturbations on the images in the test set, and use a unified perturbation factor p for different cases:

- 1) JPEG compression. We use the unified perturbation factor p as the quality factor p to compress the test images.
- 2) Gaussian noise. To simulate the low-light conditions, we add Gaussian noise that is normally distributed with a mean of zero and a standard deviation STD equals $(100 - p)/10$.
- 3) Image resampling. We change the resolution of the digital image to $p\%$ of the original size.

Figure 10 shows the robustness analysis results of our model. The abscissa p represents the disturbance factor, and the values of p in the robustness experiments are 95, 90, and 85, respectively. As p decreases, the pixel-level inpainting detection becomes more difficult. With the decrease of perturbation factor p , the detection performance of the models dropped slightly. When the value of p is reduced from 95 to 90, we can observe that our model has better robustness to JPEG compression and Gaussian noise with small perturbations. The average AUC scores are all greater than 90 because we introduce frequency-domain features rich in high-frequency information.

As the perturbation strength increases, the detection performance of CTNet gradually decreases, but the average AUC score remains above 80. Therefore, we can speculate that the robustness of our model to image resampling perturbations will be significantly improved if image samples containing different sizes are utilized for training. The robustness test results show that the CTNet has good robustness to minor disturbances caused by image post-processing.

5 Conclusion

In this paper, we have proposed the CTNet, an image inpainting detection method based on a hybrid CNN-Transformer mechanism. Our CTNet exploits the inherent global self-attention mechanism of the Transformer encoder to obtain long-range dependencies. In addition, the proposed decoder uses the frequency-domain features output by the high-pass filter to supervise the upsampling process of the predicted mask, which improves the accuracy of locating inpainted regions. The experimental results show that the CTNet can generalize to ten commonly used inpainting methods after training on a single inpainting method. The comparative experimental results demonstrate the superiority of the detection performance of the proposed model compared to the current state-of-the-art repair detection methods.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 72374058.

Author contributions FX: Conceptualization, Supervision, Review and editing. ZZ: Writing original draft, Software, Methodology, Validation. YY: Supervision, Methodology, Review and editing.

Data availability The datasets used in our paper (Dresden, Places, CelebA and ImageNet) are publicly available.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Wang, H., Li, W., Hu, L., Zhang, C., He, Q.: Structural smoothness low-rank matrix recovery via outlier estimation for image denoising. *Multimedia Syst* **28**(1), 241–255 (2022)
2. Yan, W.-Q., Wang, J., Kankanhalli, M.S.: Automatic video logo detection and removal. *Multimedia Syst.* **10**, 379–391 (2005)
3. Ghorai, M., Mandal, S., Chanda, B.: A group-based image inpainting using patch refinement in mrf framework. *IEEE Trans. Image Process.* **27**(2), 556–567 (2017)
4. Guo, Q., Gao, S., Zhang, X., Yin, Y., Zhang, C.: Patch-based image inpainting via two-stage low rank approximation. *IEEE Trans. Visual Comput. Graphics* **24**(6), 2023–2036 (2017)
5. Li, H., Luo, W., Huang, J.: Localization of diffusion-based inpainting in digital images. *IEEE Trans. Inf. Forensics Secur.* **12**(12), 3050–3064 (2017)
6. Sridevi, G., Srinivas Kumar, S.: Image inpainting based on fractional-order nonlinear diffusion for image reconstruction. *Circuits Syst Signal Process.* **38**(8), 3802–3817 (2019)
7. 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*, pp. 4471–4480 (2019)
8. Wang, N., Zhang, Y., Zhang, L.: Dynamic selection network for image inpainting. *IEEE Trans. Image Process.* **30**, 1784–1798 (2021)
9. Wang, W., Zhang, J., Niu, L., Ling, H., Yang, X., Zhang, L.: Parallel multi-resolution fusion network for image inpainting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14559–14568 (2021)
10. Jiang, Y., Xu, J., Yang, B., Xu, J., Zhu, J.: Image inpainting based on generative adversarial networks. *IEEE Access* **8**, 22884–22892 (2020)
11. Dong, X., Dong, J., Sun, G., Duan, Y., Qi, L., Yu, H.: Learning-based texture synthesis and automatic inpainting using support vector machines. *IEEE Trans. Industr. Electron.* **66**(6), 4777–4787 (2018)
12. Nabi, S.T., Kumar, M., Singh, P., Aggarwal, N., Kumar, K.: A comprehensive survey of image and video forgery techniques: variants, challenges, and future directions. *Multimedia Syst.* **28**(3), 939–992 (2022)
13. 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 (2008)
14. 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 (2013)
15. 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)

16. Zhu, X., Qian, Y., Zhao, X., Sun, B., Sun, Y.: A deep learning approach to patch-based image inpainting forensics. *Signal Process Image Comm* **67**, 90–99 (2018)
17. Chu, X., Zhang, B., Tian, Z., Wei, X., Xia, H.: Do we really need explicit position encodings for vision transformers. *arXiv preprint arXiv:2102.10882* (2021)
18. 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*, pp. 8301–8310 (2019)
19. Zhang, Y., Ding, F., Kwong, S., Zhu, G.: Feature pyramid network for diffusion-based image inpainting detection. *Inf. Sci.* **572**, 29–42 (2021)
20. Wang, X., Niu, S., Wang, H.: Image inpainting detection based on multi-task deep learning network. *IETE Tech. Rev.* **38**(1), 149–157 (2021)
21. Li, A., Ke, Q., Ma, X., Weng, H., Zong, Z., Xue, F., Zhang, R.: Noise doesn't lie: Towards universal detection of deep inpainting. *arXiv preprint arXiv:2106.01532* (2021)
22. Chen, X., Dong, C., Ji, J., Cao, J., Li, X.: Image manipulation detection by multi-view multi-scale supervision. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14185–14193 (2021)
23. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9543–9552 (2019)
24. Wu, H., Zhou, J.: IID-Net: Image inpainting detection network via neural architecture search and attention. *IEEE Trans. Circuits Syst. Video Technol.* **32**(3), 1172–1185 (2021)
25. Liu, K., Li, J., Hussain Bukhari, S.S.: Overview of image inpainting and forensic technology. *Security and Communication Networks* **2022** (2022)
26. 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)
27. Chan, T.: Local inpainting models and tv inpainting. *SIAM J. Appl. Math.* **62**(3), 1019–1043 (2001)
28. Chan, T.F., Shen, J.: Nontexture inpainting by curvature-driven diffusions. *J. Vis. Commun. Image Represent.* **12**(4), 436–449 (2001)
29. Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. *IEEE Trans. Image Process.* **19**(5), 1153–1165 (2010)
30. Ruzic, T., Pizurica, A.: Context-aware patch-based image inpainting using markov random field modeling. *IEEE Trans. Image Process.* **24**(1), 444–456 (2015)
31. Telea, A.: An image inpainting technique based on the fast marching method. *J graph tools* **9**(1), 23–34 (2004)
32. Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001)
33. Herling, J., Broll, W.: High-quality real-time video inpainting with pixmix. *IEEE Trans. Visual Comput. Graphics* **20**(6), 866–879 (2014)
34. Huang, J.-B., Kang, S.B., Ahuja, N., Kopf, J.: Image completion using planar structure guidance. *ACM Trans Graphi (TOG)* **33**(4), 1–10 (2014)
35. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544 (2016)
36. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6721–6729 (2017)
37. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. *ACM Trans Graph* **36**(4), 1–14 (2017)
38. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1486–1494 (2019)
39. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5505–5514 (2018)
40. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–17 (2018)
41. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edge-connect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212* (2019)
42. Wu, H., Zhou, J., Li, Y.: Deep generative model for image inpainting with local binary pattern learning and spatial attention. *arXiv preprint arXiv:2009.01031* (2020)
43. Yu, T., Guo, Z., Jin, X., Wu, S., Chen, Z., Li, W., Zhang, Z., Liu, S.: Region normalization for image inpainting. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12733–12740 (2020)
44. Xiao, C., Li, F., Zhang, D., Huang, P., Ding, X., Sheng, V.S.: Image inpainting detection based on high-pass filter attention network. *Comput. Syst. Sci. Eng.* **43**(3), 1146–1154 (2022)
45. 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)
46. Yang, W., Cai, R., Kot, A.: Image inpainting detection via enriched attentive pattern with near original image augmentation. In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 2816–2824 (2022)
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
48. 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)
49. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255 (2009)
50. Krizhevsky, A.: Learning multiple layers of features from tiny images. The CIFAR-100 dataset <https://www.cs.toronto.edu/~kriz/cifar.html> (2009)
51. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruysen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867* (2019)
52. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: *International Conference on Machine Learning*, pp. 10347–10357 (2021)
53. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: *Proceedings*

- of the IEEE/CVF International Conference on Computer Vision, pp. 568–578 (2021)
54. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
 55. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10033–10041 (2021)
 56. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 558–567 (2021)
 57. 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)
 58. 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)
 59. Camacho, I.C.: Initialization methods of convolutional neural networks for detection of image manipulations. PhD thesis, Université Grenoble Alpes (2021)
 60. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
 61. Gloe, T., Böhme, R.: The dresden image database for benchmarking digital image forensics. In: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1584–1590 (2010)
 62. 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)

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.