Complex & Intelligent Systems (2024) 10:289–303
https://doi.org/10.1007/s40747-023-01158-5

ORIGINAL ARTICLE

Image inpainting via progressive decoder and gradient guidance

Shuang Hou[2] · Xiucheng Dong[1,2] · Chencheng Yang[2] · Chao Wang[2] · Hongda Guo[2] · Fan Zhang[2]

Received: 6 January 2023 / Accepted: 3 June 2023 / Published online: 20 July 2023
© The Author(s) 2023

Abstract
Very recently, with the widespread research of deep learning, its achievements are increasingly evident in image inpainting tasks. However, many existing multi-stage methods fail to effectively inpainting the larger missing areas, their common drawback is that the result of each stage is easily misguided by the wrong content generated in the previous stage. To solve this issue, in this paper, a novel one-stage generative adversarial network based on the progressive decoding architecture and gradient guidance. Firstly, gradient priors are extracted at the encoder stage to be passed to the decoding branch, and multiscale attention fusion group is used to help the network understand the image features. Secondly, multiple parallel decoding branches fill and refine the missing regions by top-down passing the reconstructed priors. This progressively guided repair avoids the detrimental effects of inappropriate priors. The joint guidance of features and gradient priors helps the restoration results contain the correct structure and rich details. And the progressive guidance is achieved by our fusion strategy, combining reimage convolution and design channel coordinate attention to fuse and reweight the features of different branches. Finally, we use the multiscale fusion to merge the feature maps at different scales reconstructed by the last decoding branch and map them to the image space, which further improves the semantic plausibility of the restoration results. Experiments on multiple datasets show that the qualitative and quantitative results of our computationally efficient model are competitive with those of state-of-the-art methods.

Keywords Multiscale attention fusion group · Channel coordinate attention · Multiscale fusion · Image inpainting

Introduction

Image inpainting is to fill in the missing parts of incomplete images by synthesizing visually realistic and semantically correct replacement content. Its goal is to predict appropriate pixels of the missing areas. It serves a wide range of applications, such as photoediting, decapping, and removing unwanted objects from photos. Well-repaired areas should have reasonable semantic structures and visually realistic textures. Image inpainting appear in different scenarios, such as deleting objects during image editing [1, 2], scratch or text deletion [3, 4], and image inpainting [5, 6].

The deep encoder-decoder network with adversarial loss [7] has become popular for solving image inpainting problems. Pathak et al. [8] proposed a context encoder, which was the first work incorporating generative adversarial loss in an encoder-decoder architecture to repair broken pixels. However, the result always contains many distorted structure or texture artifacts. To alleviate this problem, Chen et al. [9] improved the discriminator by letting local discriminators identify similar patches in different images. Some other networks [10–14] utilize reasonable constraints, such as edges [12, 14] and structures [11, 15], to mitigate the difficulty of directly predicting correct missing contents. Generally, two-stage networks first reconstruct these constraints in the first stage. The completed constraints are then fed into the second stage as additional clues, helping to generate the final refined image. However, these networks are not trained end-to-end, so errors predicted in the first-stage easily influence the training of the second stage. And they use only feature priors [10–13] or gradient priors [11, 12, 14, 15] to guide the repair process, without considering both global content and local details. The priors of the above papers are used to directly guide the repair process, whether helpful or not. Although the papers [10, 16] designed new architectures to address multilevel networks, not all of the above drawbacks were addressed. MADF [16] guides progressive filling and refine-

✉ Xiucheng Dong
dxc136@163.com

1 Sichuan University Jinjiang College, Meishan, China

2 School of Electrical Engineering and Electronic Information, Xihua University, Chengdu, China

ment of missing content through multiple layers of priors, but it only allows passing feature priors between restoration decoders and refinement decoders. Paper [10] avoids structural restoration alone and states that texture and structural information interact with each other in the restoration process. However, structural and texture information directly guide each other in multiple layers, and misinformation can easily affect the whole reconstruction process.

Local and global information are equally important to understand an image. Approaches such as the multi-column network [15] and ACGAN [17] aim to map an image to multiscale features by adopting parallel encoding branches with various receptive fields. However, few studies have explored how to effectively fuse and reconstruct multiscale features in the image space. Paper [18] proposes a cross-scale feature extraction method to fill the urgent need for multi-scale detection. However, mapping only the last feature maps to the image space tends to cause semantic ambiguity in restoration images.

To completely solve these problems, we propose a new end-to-end training model and design a gradient-guided progressive decoding network with multiple parallel decoding branches that can achieve coarse-to-fine recovery by progressively passing down the reconstructed individual structured priors. The reconstructed mapping of the previous decoding branch is passed to the next decoding branch for the reconstruction process of the next decoding branch.

Our designed network improves the model performance in two aspects. First, we design a progressive-guided decoding network consisting of multiple parallel decoder branches, using the decoder to refine the features from the previous decoder and the gradient features from the gradient transition module to guide the repair. We pass a powerful prior knowledge gradient map to the decoding network via the gradient transfer module, which contains the edges and details of the image, helping to improve the salient details of the inpainting image. We use ghost convolution [19] and an attention mechanism to better fuse feature maps from different sources. Ghost convolution uses fewer parameters to learn features. The attention mechanism helps the convolutional neural network to decide which part of the image to focus on when generating text descriptions. For the attention mechanism, we designed channel coordinate attention (CCA) based on coordinate attention [20] to decompose channel attention into two parallel 1D feature encodings. The repair results are progressively refined by adding supervised information increments such as reconstruction, perceptual, and gradient loss to the decoder. Our framework relies on no explicit intermediate cues, and learns which intermediate information is most reliable and useful through end-to-end training.

Second, our model leverages learned intermediate signals from multiscale feature spaces. In this paper to better extract fused and reconstructed multi-scale features, we use an multiscale attention fusion group (MAFG) and MF to extract multiscale features and fuse feature maps from different decoder layers in a final output map containing global semantics and local details. The MAFG has several branches, each assigned a different filter size and expansion rate. MF integrates feature maps from three resolutions in the last decoding branch.

Experiments on publicly available datasets, including Places2 [21] and CelebA [22], demonstrate that the proposed framework can generate semantically plausible and richly detailed content in various scenes, even when missing regions are large and complex. Experimental results are shown in Tables 3 and 4. Our approach significantly outperforms current state-of-the-art methods, both quantitatively and qualitatively.

Our main contributions are as follows:

(1) A progressive decoder network with three decoder branches uses semantic features to guide the repair network, and a gradient map to guide the repair of high-frequency information;
(2) An MAFG extracts multiscale features in the encoder network, and an MF reconstructs feature maps of different sensory fields to form a final output map with full semantic and detailed content;
(3) Our proposed parallel multi-branch decoder architecture is shown to be effective in improving repair performance in multiple dataset tests.

Related work

Image restoration has been studied for more than 20 years, and convolution neural network–based approaches, especially those using generative adversarial networks, have enabled image restoration to take a large step forward. We summarize some work related to our approach below.

Single-stage inpainting

Pathak et al. [8] proposed the context encoder (CE), which utilizes the semantics of the damaged region and the associated loss function to jointly improve the restoration process. Although this approach had some drawbacks, such as obvious artifacts and blurring after restoration, it laid the foundation for subsequent research work. Based on the CE model, Iizuka et al. [23] used global and local discriminators to jointly train the network to ensure that global and local semantics were the same as the surrounding regions. Shift-Net based on U-Net architecture (Shift-Net) [24] recovers lost blocks with good accuracy in terms of structure and fine texture. Unlike the single encoder approach, Wang et al. [15] proposed a three-branch encoder structure, where each branch accepts feature

maps with different sensory fields. Shi et al. [25] utilized the attention transfer cross-layer mechanism to solve the information loss between codec network layers. Paper [14] uses parallel encoder-decoder to model the structure-constrained texture synthesis and texture-guided structure reconstruction in a coupled manner by parallel networks, which repairs structures and textures at the same time.

Multistage inpainting

Multistage structures [9–13, 26, 27] have become the mainstay in solving the image inpainting task. The coarse-to-fine network of Yu et al. [10] divides image inpainting into two steps: filling the holes roughly through the coarse network, and refining the repaired blurry image by the refinement network with an attention mechanism. Jiang et al. [26] proposed a network that repairs structures and enriches textures by parallel branching, introducing vertical guidance mechanisms between parallel branches to ensure that each branch utilizes only the desired features of the other branch and outputs high-quality content. JPGNet [28] uses parallel preprocessing active filtering and generative networks to preserve local structure and fill a large number of missing pixels, respectively. It is not suitable for recovering large missing regions because accurate predictive filtering relies on a large number of neighboring pixels. The paper [10] models structure-constrained texture synthesis and texture-guided structure reconstruction through parallel networks, and restores structure and texture in a coupled manner through parallel networks. MADF [16] employs a series of refinement decoders with designed pointwise normalization (PN) to progressively refine the features. Each layer takes the two sources of feature maps as inputs, from the previous layer of the current decoder and the next upsampling layer of the previous decoder, respectively.

Proposed method

Our end-to-end architecture consists of a two-part molecular network: (1) a generator to predict pixels in the missing part; and (2) a discriminator for adversarial training of the network. The network structure is shown in Fig. 1. The generator consists of an encoder network, gradient transfer module, MAFG, and branch progressive decoder network with multiple parallel decoding branches. The gradient transfer module is used in the encoder stage to extract the gradient mapping from the encoder network and pass it to the decoder network. Multiscale features are learned by the MAFG. In the decoder network, the gradient information from the encoder and the reconstructed features from the decoder branch of the previous layer are used to guide the image completion process, gradually filling and refining the mask region.

Our discriminator uses two PatchGANs [13] to predict the veracity of all image blocks. For stable training, spectral normalization is used in the discriminator, whose structure is shown in Fig. 2. And Fig. 3 are structure of the fusion modules used in the network.

Finally, we describe the encoding and the progressive decoding networks, introduce the MF, and give the corresponding loss functions.

Encoding network

The encoding network inputs a damaged image to a feature map, which facilitates the understanding of the semantic content of the image. Damaged regions usually contain multiple structures, so finding the appropriate features is crucial to inpainting the missing regions. We extract the gradient map of the image containing global and local features [29] in the decoding stage through a gradient transfer module based on the U-Net architecture [30] to guide the repair of the decoder.

MAFG in encoder

Robust feature representations help repair the network to produce semantically accurate content and crisp details [31]. We argue that robust feature representations are embodied in two aspects: features with multiscale receptive fields and precise spatial locations. The former usually contains global and local information, which helps the network understand the missing semantic content and enrich the texture details. The latter is essential to the visual system. For these purposes, we propose an MAFG at the bottleneck, which consists of three attention modules, the attention modules as shown in Fig. 4b. These modules will be discussed in detail in "Attention module". The MAFG as shown in Fig. 4a. The process is

$$x_{a1} = F_{a1}(x_{in}) \tag{1}$$

$$x_{a2} = F_{a2}(x_{a1}) + x_{a1} \tag{2}$$

$$x_{a3} = F_{a3}(x_{a2}) + x_{a2} \tag{3}$$

$$x_{\text{out}} = F_{\text{r}}[F_{\text{r}}(x_{a1} + x_{a2}) + x_{a3}] + x_{\text{in}}, \tag{4}$$

where $x_{\text{in}}, x_{\text{out}}, x_{\text{an}}, F_{\text{an}}$, and F_{r} are the input, resultant output, output of the n-th attention module, and two conv layers with pixel-shuffle layers, respectively. We use this module for the last layer of the decoding network [32].

Attention module

The attention module (Fig. 4b) aims to capture scale-diverse features and retain relatively precise spatial location information, reducing the number of channels with 1×1 convolutions and using strided convolutions to augment

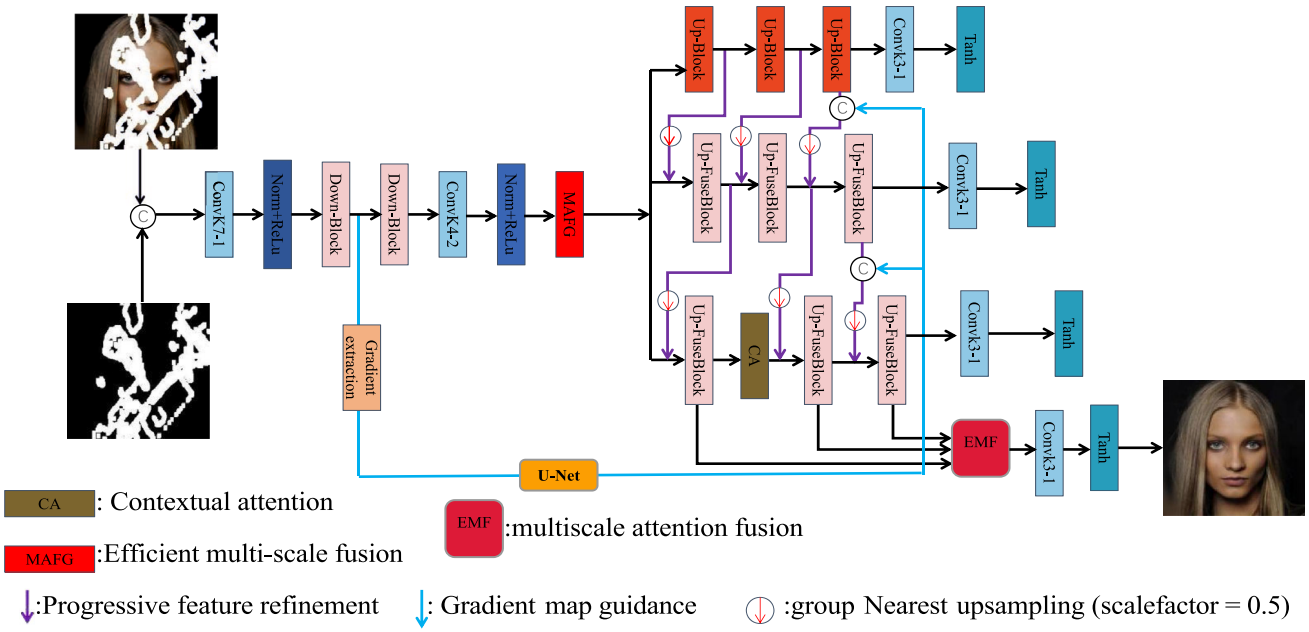


Fig. 1 Pipeline of generator, which adopts a typical encoder-decoder structure. In the encoder, down-blocks and MAFG, for example, are used to understand the semantic information. In the decoder, feature maps and gradient maps from the preceding branch are passed to the next branch for further refinement and guidance. Gradient maps are extracted by depthwise convolution (Depthconv) with fixed Scharr kernels. EMF is used to merge multiscale feature maps so that the reconstructed image contains multiscale information. "Norm" and "Conv" denote instance normalization and convolution, respectively. In k ∗-∗∗, ∗ denotes kernel size and ∗∗ denotes stride

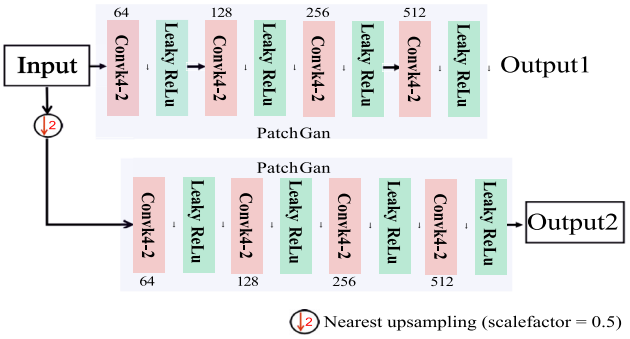


Fig. 2 Pipeline of generator that adopts a typical encoder–decoder structure. "Conv" denotes convolution. In k ∗-∗∗, ∗ denotes kernel size and ∗∗ denotes stride. Spectral normalization is used in discriminator

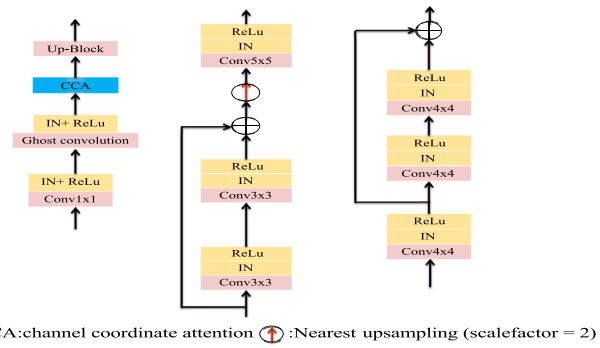
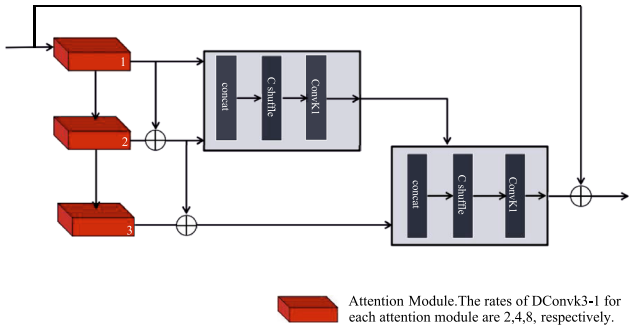


Fig. 3 From left to right, the Up-Block, the Up-FuseBlock and the Down-Block are shown in Figure. "IN" and "Conv" denote instance normalization and convolution, respectively. In k ∗-∗∗, ∗ denotes kernel size and ∗∗ denotes stride

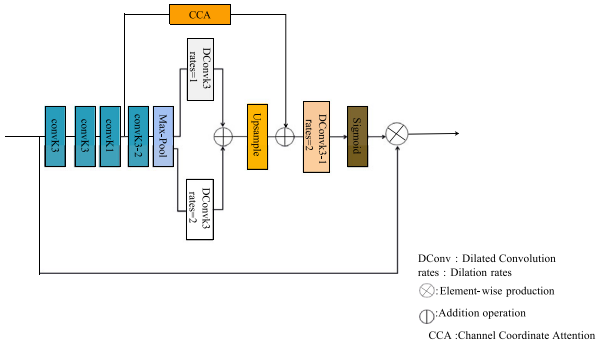
the receptive field. The spatial and channel sizes are later increased to match the input size. Subsequently, dilated convolutions with different filter sizes are introduced, and we element-wise add all output features of dilated convolutions to minimize gridding effects. Features from different sources are treated equally if we stack them directly in channel dim, which is not consistent with human vision. Thus, an attention mechanism is utilized to tell us which features need special attention. Here, we introduce another CCA to refine our input features. It is merged into the attention module to improve network performance. Lastly, a sigmoid operation is applied to obtain a similar effect to a channel attention mechanism. We apply Conv layers with different filter sizes between attention modules to extract multiscale features that are followed by pixel-shuffle layers.

Progressive decoding network

Some image inpainting models [15, 33] use a complex multi-branch encoding network, but a simple single-branch structure is used in the decoding network. This is not effective for reconstruction with a large range of missing images. We propose a progressive decoding network, as shown in



(a)



(b)

Fig. 4 Structure of attention module

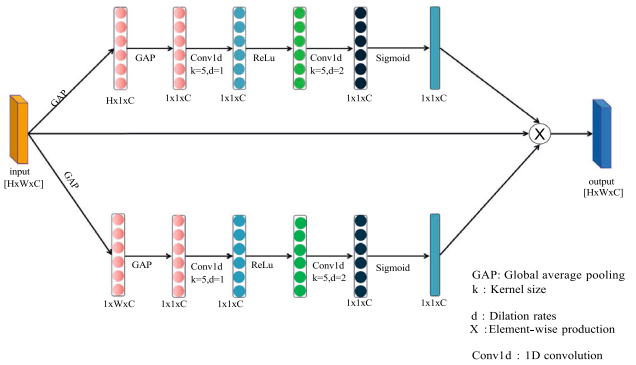


Fig. 5 Implementation of CCA

Fig. 1, which consists of three parallel decoding branches. Progressive refinement repair consists of a feature map- and gradient map-guided repair. A contextual attention mechanism in the first upsampling block of the last decoding branch improves the performance of the model. Contextual attention mechanisms can establish long-term relationships between complete and missing features, resulting in realistic and clear textures.

We define the notation used below. $D_{k,l}$ denotes the l-level upsampling block (Up-Block or Up-FuseBlock) of the k-th decoding branch, $f_{k,l}$ denotes the corresponding output feature maps, f_{e} denotes the compressed feature maps from the encoding network, and f_{out} denotes the output feature maps of the whole network, where k and l take values 1, 2, 3.

Progressive feature refinement

The recovered features $f_{k,l}$ of branch $D_{k,l}$ are fed to the next branch $D_{k+1,l-1}$ and fused with $f_{k+1,l-1}$. The reconstructed features can provide feature priors for subsequent restoration and further optimize its effect.

We use two strategies to fuse features from different branches. We splice them together in the channel dimension, and use ghost convolution [19] to learn the features. Ghost convolution has three steps. Intrinsic features are obtained using ordinary convolution with fewer kernels. Based on a set of generated features, a series of linear operations are used to obtain feature graphs of another set of features, including redundant graphs. The two batch feature graphs are stacked in the channel dimension to get the final output.

We add CCA (Fig. 5) to the fusion process. An attention mechanism that focuses on more critical information for the current task reduces attention to other information, preventing erroneous or invalid features from entering the upsampling block. coordinate attention (CA) [20] captures long-range dependencies along one spatial direction and retains accurate location information along the other direction. The generated feature maps are individually encoded to form orientation-aware and position-sensitive feature maps, which can be applied to the input feature maps to enhance the representation of the target of interest. However, CA does not explore the feature relationships between adjacent channels with a small perceptual field. We correct this by introducing one-dimensional convolution to realize CCA. We exploit two 1D global pooling operations to aggregate the input features along the vertical and horizontal directions into two direction-aware feature maps with embedded direction-specific information, and separately encode them into attention maps that capture long-range dependencies of the input feature map along one spatial direction. The first of two 1D convolutions, with a dilate rate of 1, builds connections among adjacent channels. The second 1D dilated convolution, with a dilate rate of 2, explores long-term connections among nonadjacent features. Both attention maps are applied to the input feature map via multiplication to emphasize the representations of interest.

Gradient guidance

Gradient information shows the difference between adjacent pixels. Flat areas of the image have smaller gradient values, and sharp areas (edges and details) have larger values. Thus, the gradient information reflects the sharpness of each

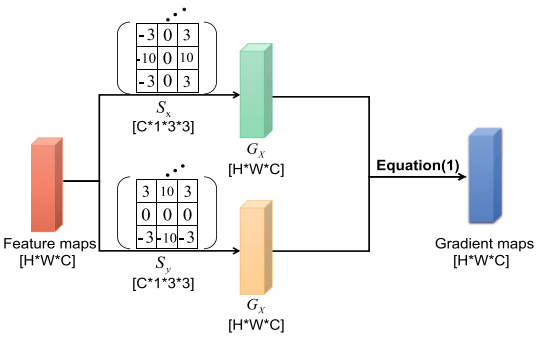


Fig. 6 Implementation of gradient extraction

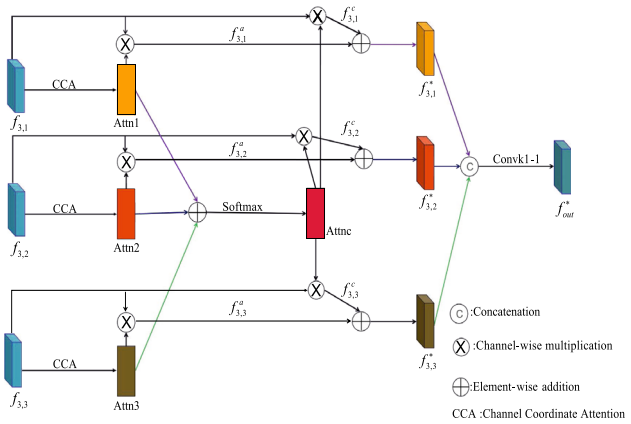


Fig. 7 Implementation of MF

region of the image, and is important to enhanced image edge and detail recovery. Our proposed gradient guidance has two components: gradient mapping guidance (Fig. 1) and gradient loss constraint, which use gradient information to supervise the generation process and improve the consistency of the generated image on the boundary.

In the encoder network stage, the more forward network layers contain richer structural information, so we extract the gradient map of the image in the first layer of the decoder, as shown in Fig. 6. We use Scharr operators with different orientations as fixed convolution kernels S_x and S_y. We use the fixed Sobel kernels S_x and S_y to convolve the feature map by the nn.functional. This step generates horizontal and vertical gradient maps G_x and G_y. Each value of gradient maps is calculated as

$$G = \sqrt{G_x^2 + G_y^2}. \tag{5}$$

A simple image inpainting process is performed on the extracted gradient map using a gradient transfer module. The restored rich gradient information is incorporated into progressive feature-guided restoration, and passed to the last layer of each branch of the decoder, helping the model to infer accurate neighborhoods and focus on the reconstruction of sharp regions.

Multiscale fusion

To make the inpainting image as similar as possible to the real image, it is crucial that the local and global information of the reconstruction image is consistent with the real image. High-resolution feature maps typically contain rich texture information that is essential for subjective perception. Low-resolution feature maps contain global information, such as the semantics of the context. Our multiscale fusion module mines and combines more appropriate global and local information [34], as shown in Fig. 7. To make the recovered images contain information at different scales, we integrate feature maps from three resolutions in the last decoding branch.

We take one feature map ($f_{3,l}$), with resolutions of $64*64$, $128*128$, and $256*256$, produced by the lth layer of the final decoding path ($l = 1, 2, 3$). These unresolved feature maps are upsampled to the same resolution before fusion. As shown in Fig. 8, we employ CCA on $f_{3,l}$ to obtain the corresponding channel attention vectors (Attn_l), which highlight the contributing channels and suppress useless ones. The common attention vector (Attn_c) is first generated by adding all unique $Attn_l$ along the channel dimension. Large values in the vector indicate that all cross-scale features from this channel are beneficial for final reconstruction. Then Attn_c is fed into a softmax function, which limits the value to the range [0–1].

The obtained $Attn_l$ and Attn_c are applied on the input feature maps $f_{3,l}$ in channel-wise multiplication to generate recalibrated maps $f_{3,l}^a$ and $f_{3,l}^c$, respectively. Subsequently, each discriminative feature map ($f_{3,l}^*$) is produced by adding $f_{3,l}^a$ and $f_{3,l}^c$, which contain both self-unique and co-critical information. All enhanced maps ($f_{3,l}^*$) are concatenated and fused by a $1*1$ convolution layer to create the final reconstruction maps (F_{out}). The procedure can be defined as:

$$f_{3,l}^a = \text{Attn}_l \otimes f_{3,l} = \text{CECA}(f_{3,l}) * f_{3,l} \tag{6}$$

$$f_{3,l}^c = \text{Attn}_c \otimes f_{3,l} = \text{Softmax}\left(\sum_{l=1}^{3} \text{Attn}_l\right) * f_{3,l} \tag{7}$$

$$f_{3,l}^* = f_{3,l}^a + f_{3,l}^c \tag{8}$$

$$f_{out} = \text{Convk1} - 1(\text{Concatenate}(f_{3,1}^*, f_{3,2}^*, f_{3,3}^*)). \tag{9}$$

Loss function

We use L1 loss, perceptual loss [35], pyramid loss, style loss [35], adversarial loss [7] and gradient loss to make the model converge faster. During training, the last branch of the decoding network is bounded by all the loss functions, and

(a) Input (b) MC-Net [15] (c) Ours (d) Grund Truth

Fig. 8 Qualitative results of CelebA-HQ dataset under $128 * 128$ center holes

the other branches are bounded by L1 loss and gradient loss. We describe these loss functions below.

L1 loss

We convert reconstructed features $f_{1,3}$, $f_{1,3}$, $f_{1,3}$, and f_{out} to image space, with notation I_k $(k = 1, 2, 3)$ and I_{out}, respectively. In addition to constraining the output image (I_{out}), L1 loss constrains the generated image of each decoder branch. L1 loss is the mean absolute error between each completion image (I_k) and ground truth (I_{gt}), where $N_{I_{\text{gt}}}$ is the number of elements in (I_{gt}). L_1^k and L_1^{out} represent the L1 loss of reconstructed features and repaired images, respectively. These are calculated as

$$L_1^k = E\left[\sum_{i=1}^{3} \|I_k - I_{\text{gt}}\|_1\right] \tag{10}$$

$$L_1^{out} = E[\|I_{\text{out}} - I_{\text{gt}}]\|_1]. \tag{11}$$

Pyramid loss

We use $f_{3,1}$ and $f_{3,2}$ of the last decoding branch to calculate the pyramidal perceptual loss. Pyramid loss has two advantages: (1) optimizing the estimation of features missing regions and allowing better learning of the correlation between known and unknown image blocks in the contextual attention module; (2) refining the prediction of missing regions at each scale to better reconstruct the final image. We use activation layers relu3_1 and relu2_1 of VGG19 [36] to extract feature maps with two resolutions from a real image, and calculate L_{pyramid} between the extracted real features and those predicted by the corresponding generation layer (($f_{3,1}$ and $f_{3,2}$)),

$$L_{\text{pyramid}} = \sum_{i=1}^{2} \|\Phi_i(I_{\text{gt}}) - f_{3,i}\|_1, \tag{12}$$

where I_{gt} is the real image, Φ_i is the i-th selected activation layer of the pretrained VGG network, and $f_{3,i}$ is the corresponding feature map of the last decoding branch. Note that $f_{3,i}$ is the same size as $\Phi_i(I_{\text{gt}})$.

Perceptual loss

Perceptual loss and style loss are based on the pretrained VGG network, and force the generated semantic structures and rich textures to be similar to the ground truth. Let Φ_x represent the features of the i-th activation layer in a VGG19 [36] network when given the image x. We use activation layers relu1_1, relu2_1, relu3_1, relu4_1, and relu5_1 for our loss calculation,

$$L_{\text{per}}^{out} = E\left[\sum_{i=1}^{5} \|\Phi_i(I_{\text{out}}) - \Phi_i(I_{\text{gt}})\|_1\right]. \tag{13}$$

Style loss

To ensure that the restored image has the same style as the original, we introduce style loss,

$$L_{\text{sty}}^{out} = E_j\left[\sum_{i=1}^{5} \|G_j^{\Phi_i}(I_{\text{out}}) - G_j^{\Phi_i}(I_{\text{gt}})\|_1\right]. \tag{14}$$

We use the Gram matrix (G_j) on each selected feature map generated by VGG19, and calculate the error using L1 loss.

Adversarial loss

We use an adversarial loss to supervise the training process of the repair model, and PatchGAN and spectral parametric regularization to train the network model, which solves the problem of model collapse and stabilizes the convergence of the discriminator. The adversarial loss function of the discriminator D is defined as

$$L_{\text{adv}}^{out} = E[\log(1 - D(G(I_{\text{in}}, M)))] + E[\log(D(I_{\text{gt}}))], \tag{15}$$

where I_{in} is the input image covered with mask M; the value 1 denotes missing pixels; and G and D are the generator and discriminator, respectively.

Gradient loss

The second part of gradient guidance, gradient loss, is applied on every decoder branch. Both the reconstructed intermediate images $(I_k$ $(k = 1, 2, 3))$ and final image (I_{out}) contain rich and correct details and edges, which benefit gradient map guidance in the decoding network. We formulate the gradient loss by narrowing the distance between the gradient maps extracted from the recovery image and ones from the ground truth,

$$L_{\text{gra}}^{k} = E\left[\sum_{i=1}^{3} \|\text{Ge}(I_k) - \text{Ge}(I_{\text{gt}})\|_1\right] \tag{16}$$

$$L_{\text{gra}}^{out} = E[\|\text{Ge}(I_{\text{out}}) - \text{Ge}(I_{\text{gt}})\|_1], \tag{17}$$

where Ge represents gradient extraction. The implementation of gradient extraction is shown in Fig. 6.

Overall loss

The total loss is

$$L_{\text{total}} = 0.02L_{\text{pyramid}} + L_1^k + 5L_1^{out} + 0.1L_{\text{per}}^{out} + 180L_{\text{sty}}^{out} + L_{\text{adv}}^{out} + 3L_{\text{gra}}^{out} + L_{\text{gra}}^{k}. \tag{18}$$

Experiments

We provide basic network settings. To objectively and comprehensively evaluate the performance of image recovery models, we qualitatively and quantitatively compare the image recovery results of various state-of-the-art models.

Experimental settings

Datasets

We trained and evaluated our model on two well-known public datasets with different characteristics.

Places2 [21] contains 365 natural scenes and over one billion images. Our model was trained on a standard training segmentation set with 1.8 million images, and evaluated on 10,000 selected training images. The scenes in the dataset include city, forest, beach, indoor, etc. This dataset can be used to train the image restoration model so that it can be better adapted to the image problems in real applications.

CelebA-HQ [22] focuses on high-quality human faces, with size 512 * 512. The dataset consists of a large number of high-quality face images and is widely used for training and evaluating image restoration algorithms. We selected 27,000 images for training, and the remaining images for evaluation.

The Mask dataset [37] provides 12,000 irregular mask images, which can be categorized based on hole-to-image area ratios (e.g., (0.1, 0.2], (0.2, 0.3]).

Training settings

We trained the model with batch size 8 using the Adam [38] optimizer and the corresponding parameters ($\beta 1$, $\beta 2$), with the learning rate set to 0.9, 0.999, and 0.0002. Experiments were conducted on an RTX 3090 GPU (24 G), and the training of CelebA-HQ and Places2 took about 2.5 days and 9 days, respectively. All the masks and images for training and testing had size $256 * 256$. Our model requires no post-processing.

For a fair comparison, we trained our model with the same mask type as used in pretrained state-of-the-art models, including center square mask, random rectangular, and irregular masks. With the center square mask, all images are masked with a $128 * 128$ square bounding box in the center position. With the random rectangular mask, images are randomly covered with a blank rectangle with size between $64 * 64$ and $128 * 128$. The irregular mask originates from a mask dataset provided by Nvidia [12].

State-of-the-art algorithms for comparison

We compared our model with the multi-column network(MC-Net) [15], EdgeConnect (EC) [12], PEN-Net [39], GatedConv (GC) [26], StructureFlow (SF) [13], RFR [27], ACFGAN [33], MADF [16] and PG-Net [26]. To the extent possible, comparison algorithms were evaluated by their officially released pretrained models. We trained PG-Net [26] using the official code and default parameters. The performance of EC [12], RFR [27] and SF [13] has not been validated on the CelebA-HQ dataset, so we retrained them using the default parameters provided in their source code. The pretrained models of MC-Net [15]and RFR [27] are not available for the Places2 dataset, so we only compared them with our model on the CelebA-HQ dataset due to time limitations.

Based on center square holes, we compared with MC-Net [15]. Based on random rectangular holes, we compare with PEN-Net [39] and ACFGAN [33]. When filling the irregular holes, we compare with EC [12], SF [13], GC [26], RFR [27], MADF [16] and PG-Net [26]. Our network models all achieved better results

Quantitative evaluation

We measured the inpainting performance of the model in various scenes using PSNR [40], SSIM [40] and LPIPS [41] indexes. PSNR measures the L2 distance between the real and repaired images at the pixel level used in image processing mainly to quantify the reconstruction quality evaluation of images and videos affected by lossy compression. The higher its value means the more similar the two images are, and vice versa, the greater the difference. SSIM measures structure similarity by the mean, standard deviation, and covariance, which reflect human perceptions more precisely than PSNR. The closer its value is to 1 means the more similar the two images are, and vice versa, the greater the difference. LPIPS uses AlexNet to extract features from the real and generated images, and calculates their feature distance and its lower value indicates that the two images are more similar, and

Table 1 Quantitative result comparison, testing incomplete image with center holes on CelebA-HQ dataset

Dataset	Model	Central holes		
		SSIM	PSNR	LPIPS
CelebA-HQ	MC-Net [37]	0.916	26.63	0.0454
	Ours	**0.926**	**27.59**	**0.0387**

Bold data represents the best-performing value

vice versa, the greater the difference. Tables 1, 2, 3 and 4 show the evaluation results on the CelebA-HQ and Places2 datasets.

Evaluation results with regular holes

As shown in Table 1, our model outperforms MC-Net on all metrics, and consists of a multiscale encoder and single decoder. This demonstrates that multiple decoding branches are beneficial for the inpainting task. As shown in Table 2, for random holes, our models produce better results than PEN-Net, and comparable results to AFGAN on the Places2 dataset. For larger center holes, our model performs better than comparison models in various scenes, indicating that it can effectively fill larger holes.

Evaluation results with irregular holes

We compared the performance of different methods with irregular holes. For the CelebA-HQ dataset (Table 3), our model outperforms SOTAs in terms of PSNR, SSIM and LPIPS. For the Places2 dataset (Table 4), the advantage of our model emerges as the mask ratio increases. In addition, we compare the parameters between oursPG-Nnet and MADF (shown in Table 5). Although the network model of MADF and PG-Nnet has fewer parameters than ours, our network model performs better in some objective indicators. Considering the performance and cost, our model is a better choice for restoring damaged images.

Qualitative evaluation

Quantitative metrics cannot fully reflect humans' subjective feelings, so qualitative evaluation is introduced as a judgment criterion. Figures 8, 9 and 10 compare results on the CelebA-HQ dataset. Figures 11 and 12 show the visual results on the Places2 dataset.

When filling center holes (Fig. 8) in the CelebA-HQ dataset, compared with MC-Net, the face generated by this method is similar to the real image in terms of face color and eyes. When filling random holes (Fig. 9), PEN-Net produces blurry structures, incorrect colors, and artifacts with the mask shape. AFGAN generates clear but unreasonable structures. Filling irregular holes is more challenging than filling regular holes. As shown in Fig. 10, the repaired regions of EC and GConv always contain unreasonable structures and texture artifacts. For example, in the second row, the left eye generated by EC and SF is asymmetric with the original right eye. In the fourth row, the face produced by GC shows obvious texture artifacts. RFR can generate plausible structures, but the results have low-quality textures. PG-Net shows strong potential to generate reasonable structures and clear textures without artifacts but is prone to generate smooth details and edges. Compared with the results of these baselines, our results have distinct and more coherent structures and fine textures, even though the holes are relatively large and complex. We guess that when the missing hole is large, EC is difficult to repair the edges of missing areas accurately, and SF has insufficient information to understand the connection between pixels in the missing region. For RFR, although the use of recurrent feature reasoning is suitable for repairing large holes, the restored images still have some unnatural content.

The Places2 dataset contains various indoor and outdoor scenes. As shown in Fig. 11, compared to the other models, our inpainting results have fewer noticeable mask artifacts and fewer bad textures. As shown in Fig. 12, when filling irregular holes, our model produces better structures, and the results look clearer than the others. As the mask ratio increases and the information in known areas is reduced, other methods generate more apparently distorted pixels than our method, so both objective indicators and subjective vision are inferior to our model.

Ablation study

We conducted ablation experiments on the CelebA-HQ test set, using three thousand testing images with irregular masks. Qualitative and quantitative results are provided for comparison in Fig. 13, which clearly show that the photos after our network repair have no obvious texture or structure blur.

Effectiveness of feature guidance

Our progressive guidance decoding network consists of multiple parallel branches, gradually filling large holes in a guided way. The decoding network passes feature priors from the recovery decoder to the refinement decoder. We conducted additional experiments without feature guidance. As shown in Table 6, the average PSNR was 25.26 dB, SSIM was 0.894, and LPIPS was 0.0811, which are inferior to the adoption of feature guidance.

Table 2 Quantitative results of different methods with random regular holes on elebA-HQ and Places2 datasets

Dataset	Model	Central holes			Random holes		
		SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS
CelebA-HQ	PEN-Net (CVPR 2019) [39]	0.900	25.88	0.0615	0.929	27.09	0.0610
	ACFGAN (Neurocomputing 2020) [33]	0.906	26.16	0.0497	0.931	27.09	0.0413
	Ours	**0.916**	**26.81**	**0.0439**	**0.938**	**27.97**	**0.0385**
Places2	PEN-Net (CVPR 2019) [39]	0.732	20.63	0.1949	0.818	22.46	0.1314
	ACFGAN (Neurocomputing 2020) [33]	0.759	20.73	0.1457	0.837	22.62	0.0994
	Ours	**0.767**	**21.28**	**0.1463**	**0.845**	**23.28**	**0.0944**

Bold data represents the best-performing value

Table 3 Quantitative results of our model and state of-the-art models with irregular holes on CelebA-HQ dataset

Dataset	Model	0.2–0.3			0.3–0.4			0.4–0.5			0.5–0.6		
		SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS
CelebA-HQ	EC (ICCVW 2019) [12]	0.952	29.56	0.0330	0.931	26.04	0.0822	0.917	25.13	0.0730	0.812	21.69	0.1245
	GC (ICCV 2019) [26]	0.961	28.75	0.0466	0.939	26.41	0.0691	0.916	24.54	0.0948	0.824	21.60	0.1397
	SF (ICCV 2019) [13]	0.963	29.05	0.0370	0.940	26.72	0.0568	0.914	24.91	0.0787	0.820	21.81	0.1281
	RFR (CVPR 2020) [27]	0.968	29.77	0.0310	0.947	27.22	0.0482	0.923	25.45	0.0681	0.842	22.36	0.1121
	PG-Net (Appl Intell 2022) [26]	0.967	29.51	0.0331	0.948	27.18	0.0499	0.923	25.35	0.0695	0.844	22.35	0.1217
	Ours	**0.969**	**29.82**	**0.0320**	**0.950**	**27.48**	**0.0492**	**0.925**	**25.64**	**0.0680**	**0.849**	**22.71**	**0.1146**

Bold data represents the best-performing value

Table 4 Quantitative results of our model and other state-of-the-art models with irregular holes on Places2 dataset

Dataset	Model	0–0.2			0.2–0.4			0.4–0.6		
		SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS
Places2	EC (ICCVW 2019) [12]	0.969	31.41	0.0270	0.880	24.08	0.1071	0.707	19.72	0.2250
	GC (ICCV 2019) [26]	0.969	30.36	0.0546	0.875	23.65	0.1406	0.711	19.41	0.1921
	SF (ICCV 2019) [13]	0.959	29.58	0.0359	0.870	23.24	0.1008	0.731	19.75	0.1916
	MADF (TIP 2021) [16]	0.969	**31.50**	0.0243	0.879	23.98	0.0840	0.728	20.13	0.1751
	Ours	**0.970**	31.47	**0.0165**	**0.883**	**24.29**	**0.0788**	**0.733**	**20.45**	**0.1667**

Bold data represents the best-performing value

Effectiveness of multiscale fusion

We designed MF to merge the multiscale features from the last decoding branch, which helps to reconstruct a final image with correct semantic structures and details. To further demonstrate the performance of MF, we conducted experiments without it. As shown in Table 6, without MF, the average PSNR was 25.38 dB, SSIM was 0.894, and LPIPS was 0.0755, which are inferior to the adoption of MF.

Table 5 Parameters of the our model, PG-Net and MADF

Model	Parameters
PG-Net (Appl Intell 2022) [26]	352 M
MADF (TIP 2021) [16]	**347.6** M
Ours	363 M

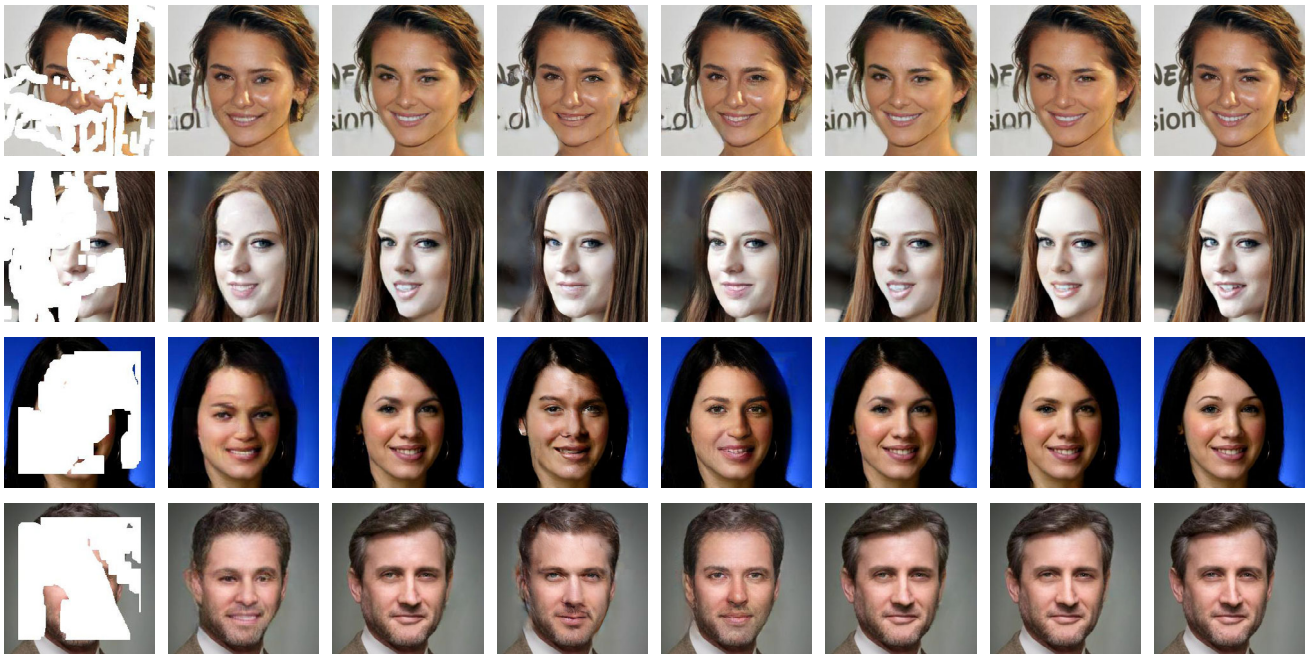
Bold indicates that the network has the least number of parameters

Effectiveness of multiscale attention fusion group

We used an MAFG to fuse and reconstruct multiscale features. We conducted some experiments on the setting of the number of attention modules, as shown in Table 7, where it can be seen that the use of three attention modules gave the best performance. To further demonstrate the performance of our proposed MAFG, we performed a comparison experiment without it, with other conditions unchanged. As shown in Table 6, our model achieved an average PSNR of 25.54 dB, SSIM of 0.895, and LPIPS of 0.0753, which is better than without MAFG.

(a) Input (b) PEN-Net [28] (c) ACFGAN [32] (d) Ours (e) Ground Truth

Fig. 9 Qualitative results of CelebA-HQ dataset under random regular holes of size between $64 * 64$ and $128 * 128$



(a) Input (b) EC [13] (c) SF [14] (d) GC [11] (e) RFR [17] (f) PG-Net [11] (g) Ours (h) Ground truth

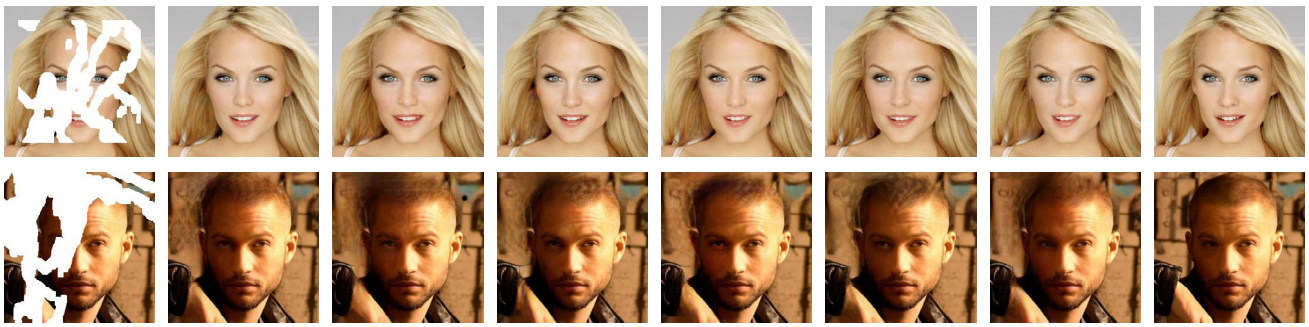
Fig. 10 Qualitative results of CelebA-HQ dataset under irregular holes

(a) Input (b) PEN-Net [28] (c) ACFGAN [32] (d) Ours (e) Grund Truth

Fig. 11 Qualitative results of Places2 dataset under random regular holes of size between $64 * 64$ and $128 * 128$

(a) Input (b) EC [13] (c) SF [14] (d) GC [11] (e) MADF [38] (f) Ours (g) Ground truth

Fig. 12 Qualitative results of Places2 dataset under irregular holes



(a) Input (b) w/o feature guidance (c) w/o MF (d) w/o MAFG (e) w/o GMG (f) w/o GG (g) Ours (h) Ground truth

Fig. 13 Visual results of ablation experiments and proposed model on CeleBA-HQ dataset; "w/o GMG" and "w/o GG" represent without gradient map guidance and without gradient guidance, respectively

Table 6 Effectiveness of feature guidance, MF, and MAFG

Dataset	Model	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	W/o feature guidance	0.894	25.26	0.0811
	W/o MF	0.884	24.86	0.0880
	W/o MAFG	0.891	25.33	0.0805
	Ours	**0.895**	**25.54**	**0.0753**

Bold data represents the best-performing value

Effectiveness of gradient guidance

We conducted two experiments to show the effectiveness and contribution of our proposed gradient guidance. The first experiment was performed without gradient map guidance but with gradient loss constraint, and the other experiment was performed without gradient map guidance and gradient loss. The results in Table 8 show that our proposed gradient guidance and gradient loss can significantly improve the network repair effect.

Discussion on the weights of L_{gra}^{out} and L_{gra}^{k}

In this section, we first fix the weights of L_{gra}^{k} to 1, and set the weight of L_{gra}^{out} to be 1, 2, 3, 4 and 5, respectively. The Table 9 shows the effect of different weights on model performance, and the model has the best inpainting performance when the loss weight is set to 3.

Table 7 MAFG's choice of number of attention modules

Dataset	Model	0.4–0.5		
		SSIM	PSNR	LPIPS
CelebA-HQ	One attention module	**0.925**	25.50	0.0689
	Two attention modules	0.924	25.54	0.0685
	Three attention modules	**0.925**	**25.64**	**0.0680**
	Four attention modules	**0.925**	25.47	0.0684

Bold data represents the best-performing value

Table 8 Effectiveness of gradient guidance

Dataset	Model	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	w/o gradient map guidance	0.892	25.24	0.0769
	W/O gradient guidance	0.892	25.15	0.0750
	Ours	**0.895**	**25.54**	**0.0753**

Bold data represents the best-performing value

Table 9 The weight discussion of L_{gra}^{out}. And the weights of L_{gra}^{k} is fixed to 1

Dataset	Weight	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	1	0.893	25.20	0.0759
	2	0.894	25.32	0.0757
	3	**0.895**	**25.54**	**0.0753**
	4	0.893	25.24	0.0791
	5	0.894	25.30	0.0785

Bold data represents the best-performing value

Table 10 The weight discussion of L_{gra}^{k}. And the weights of L_{gra}^{out} is fixed to 3

Dataset	Weight	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	1	**0.895**	**25.54**	**0.0753**
	2	0.894	25.32	0.0782
	3	0.894	25.23	0.0791

Bold data represents the best-performing value

Then, we fix the weight of L_{gra}^{out} to 3, and set the weights of L_{gra}^{k} to be 1, 2 and 3, respectively. The Table 10 shows the effect of different weights on model performance, and the model has the best inpainting performance when the loss weight is set to 1.

Conclusion

In this paper, we extend a single decoding branch to multiple decoding branches and propose a progressive decoding network that gradually fills and refines the missing regions. Specifically, considering the characteristics of different convolutional layers in the decoder, the recovered features from the previous branch and the gradients from the encoding part are fed to the shallow and deep layers of the next branch, respectively. When fusing the intermediate mappings from different branches, reimage convolution and CCA are used to reduce the model parameters and avoid forward propagation of error information, respectively. In addition, our model explores the multi-scale information of the image by employing MAFG in the encoder and MF in the decoder. The proposed MAFG extracts the multiscale features and improves the feature representation. The MF fuses the multiscale reconstructed features from the final decoding branch into the final feature map, which helps modeling to output images containing both global and local information in the image space. The effectiveness of the proposed framework has been verified in quantitative and qualitative experiments.

In this paper, a mask representing the shape and location of the missing region is given. However, in the real world, this mask is likely to be unknown. Therefore, our future work will focus on image blind inpainting.

Acknowledgements This work was supported by the National Natural Science Foundation of China (11872069), Central Government Funds of Guiding Local Scientific and Technological Development for Sichuan Province (2021ZYD0034), National Ministry of Education "Chunhui Plan" Scientific Research Project (Z2017076), and Chengdu Science and Technology Program (2016-YF04-00044-JH).

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adap-

tation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Chang LY, Liu ZY, Hsu W (2019) VORNet: spatio-temporally consistent video inpainting for object removal. In: 2019 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW). IEEE
2. Hertz A, Fogel S, Hanocka R, Giryes R, Cohen-Or D (2019) Blind visual motif removal from a single image. arXiv preprint arXiv:1904.02756
3. Nakamura T, Zhu A, Yanai K, Uchida S (2017) Scene text eraser. In: 14th IAPR international conference on document analysis and recognition (ICDAR), pp 832–837
4. Fan Q, Zhang L (2018) A novel patch matching algorithm for exemplar-based image inpainting. Multimed Tools Appl 77:10807–10821
5. Zeng J, Fu X, Leng L, Wang C (2019) Image inpainting algorithm based on saliency map and gray entropy. Arabian J Sci Eng 44(4):3549–3558
6. Yao F (2018) Damaged region filling by improved Criminisi image inpainting algorithm for thangka. Clust Comput 22:1–9
7. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Proceedings of the 2014 NeurIPS, pp 2672–2680
8. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros A (2016) Context encoders: feature learning by inpainting. In: Proceedings of the 2016 CVPR, pp 2536–2544
9. Chen Y, Zhang H, Liu L, Chen X, Zhang Q, Yang K, Xia R, Xie J (2021) Research on image Inpainting algorithm of improved GAN based on two-discriminations networks. Appl Intell 51:3460–3474
10. Liao L, Xiao J, Wang Z, Lin C-W, Satoh S (2020) Guidance and evaluation: semantic-aware image inpainting for mixed scenes. In: Proceedings of the 2020 ECCV, pp 683–700
11. Shao H, Wang Y, Fu Y (2020) Generative image inpainting via edge structure and color aware fusion. Signal Process Image Commun 87(3):115929
12. Nazeri K, Ng E, Joseph T, Qureshi F, Ebrahimi M (2019) EdgeConnect: generative image inpainting with adversarial edge learning. In: Proceedings of the 2019 ICCVW
13. Ren Y, Yu X, Zhang R (2019) StructureFlow: image inpainting via structure-aware appearance flow. In: Proceedings of the 2019 ICCV, pp 181–190
14. Guo X, Yang H, Huang D (2021) Image inpainting via conditional texture and structure dual generation. In: International conference on computer vision
15. Wang Y, Tao X, Qi X, Shen X, Jia J (2018) Image inpainting via generative multi-column convolutional neural networks. Curran Associates Inc, Red Hook, pp 329–338
16. Zhu M, He D, Li X, Li C, Li F, Liu X, Ding E, Zhang Z (2021) Image inpainting by end-to-end cascaded refinement with mask awareness. In: IEEE transactions on image processing, pp 4855–4866
17. Chen M, Liu Z, Ye L, Wang Y (2020) Attentional coarse-and-fine generative adversarial networks for image inpainting. Neurocomputing 405:259–269
18. Shen L , Tao H, Ni Y, Wang Y, Stojanovic V (2023) Improved YOLOv3 model with feature map cropping for multi-scale road object detection. Meas Sci Technol 34(4)
19. Han K, Wang Y, Tian Q, Guo J, Xu C (2020) GhostNet: more features from cheap operations. In: Proceedings of the 2021 CVPR, pp 1580–1589
20. Hou Q, Zhou D, Feng J (2021) Coordinate attention for efficient mobile network design. In: Conference on computer vision and pattern recognition (CVPR), pp 13708–13717
21. Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2017) Places: a 10 million image database for scene recognition. In: IEEE transactions on pattern analysis and machine intelligence, pp 1452–1464
22. Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196
23. Iizuka S, Simo-Serra E, Ishikawa H (2017) Globally and locally consistent image completion. ACM Trans Graph (TOG) 36(4CD):107.1-107.14
24. Yan Z, Li X, Li M, Zuo W, Shan S (2018) Shift-net: image inpainting via deep feature rearrangement. In: Computer vision-ECCV, pp 3–19
25. Shi Y, Fan Y, Zhang N (2021) A generative image inpainting network based on the attention transfer network across layer mechanism. Optik Int J Light Electron Opt 242:167101
26. Jiang J, Dong X, Li T (2022) Parallel adaptive guidance network for image inpainting. Appl Intell. https://doi.org/10.1007/s10489-022-03387-6
27. Li J, Wang N, Zhang L, Du B, Tao D (2020) Recurrent feature reasoning for image inpainting. In: Proceedings of the 2020 CVPR, pp 7757–7765
28. Guo Q, Li X, Juefei-Xu F, Yu H, Liu Y, Wang S (2021) JPGNet: joint predictive filtering and generative network for image inpainting. In: Proceedings of the 29th ACM International conference on multimedia, pp 386–394
29. Matsui T, Ikehara M (2020) Single-image fence removal using deep convolutional neural network. In: IEEE Access, pp 38846–38854
30. Ma C, Rao Y, Cheng Y, Chen C, Lu J, Zhou J (2020) Structure-preserving super resolution with gradient guidance. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 7766–7775
31. Yuan J, Yu H (2019) Multi-scale generative model for image completion. In: Proceedings of 2019 2nd international conference on algorithms, computing and artificial intelligence (ACAI 2019), pp 21–30
32. Li T, Dong X, Lin H (2020) Guided depth map super-resolution using recumbent Y network. In: IEEE Access, pp 122695–122708
33. Chen M, Liu Z, Ye L, Wang Y (2020) Attentional coarse- and-fine generative adversarial networks for image inpainting. Neurocomputing 405:259–269
34. Ji W, Li J, Yu S, Zhang M, Piao Y, Yao S, Cheng L (2021) Calibrated RGB-D salient object detection. In: Proceedings of the 2021 CVPR, 2021, pp 9471–9481
35. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super resolution. In: Proceedings of the 2016 ECCV, pp 694–711
36. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arxiv:1409.1556
37. Liu G, Reda F, Shih K, Wang T, Tao A, Catanzaro B (2018) Image inpainting for irregular holes using partial convolutions. In: Proceedings of the 2018 ECCV, pp 85–100
38. Kingma D, Adam J (2015) A method for stochastic optimization. In: Proceedings of the 2015 ICLR

39. Zeng Y, Fu J, Chao H, Guo B (2019) Learning pyramid-context encoder network for high-quality image inpainting. In: Proceedings of the 2019 CVPR, pp 1486–1494
40. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600-612
41. Zhang R, Isola P, Efros A, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the 2018 CVPR, pp 586–595

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