



Inpainting larger missing regions via progressive guidance decoding network

Xiucheng Dong^{1,3} · Jinyang Jiang² · Shuang Hou³ · Chencheng Yang³

Received: 13 July 2022 / Accepted: 1 January 2023 / Published online: 20 January 2023
© The Author(s) 2023

Abstract

For images corrupted for various reasons, the size of the corrupted area is often arbitrary and it has been a challenge to inpainting the larger missing areas. Though popular multistage networks ease the inpainting difficulty by repairing damaged image from coarse to fine, their common drawback is that the result of each stage is easily misguided by the wrong content generated in the previous stage. To address this problem, we propose a novel progressive guidance decoding network. First, multiple parallel decoding branches fill and refine the missing regions by top-down passing the reconstructed priors. This inpainting way of progressive guidance avoids adverse effects of inappropriate premises, since the decoding branches can learn what priors can be utilized. And convolution layers of decoder with different locations would pass down the different priors. The joint guidance of features and gradient priors helps the inpainting result contains the correct structure and rich details. The second fold of progressive guidance is achieved by our fusing strategy, combining ghost convolution and the designed cascaded efficient channel attention (CECA) to fuse and reweight the features from different branches. CECA explores the dependencies among adjant and non-adjant channels more effectively than popular ones. Finally, we merges the different-scale feature maps reconstructed by the last decoding branch and mapping them to the image space, which further improves the semantic plausibility of the restoration results. Extensive experiments verify the effectiveness of our method in both subjective and objective evaluation.

Keywords Cascaded efficient channel attention (CECA) · Efficient multiscale fusion (EMF) · Image inpainting · Progressive decoding network

Introduction

Broken images would hinder the correct representation and transmission of information, and also increase the difficulty of image recognition, tracking, and localization tasks, etc. Thus, image inpainting is integrated into our lives and important computer vision tasks, such as repairing missing features, removing unwanted objects from photos, and removing subtitle. In the natural case, the damaged area of an image is arbitrary, including shape and size. For the smaller and sparse missing regions, repairing is relatively simple. For the larger

and complex missing region, repairing is difficult due to the drastic reduction of surrounding known information.

Image inpainting aims to recover missing pixels of an corrupted image so as to generate a visually realistic image. Recently, some multistage inpainting networks, such as two-stage networks [1–7] and progressive recurrent networks [8,9], experience multiple encoder–decoders to progressively refer missing contents, which mitigate the difficulty of directly predicting correct missing contents. For two-stage networks, they first reconstruct constraints in the first network, including blurry images [2,3], edges [4,5], and structures [1,6]. Then, the completed constraints are fed into the next network as additional clues. For progressive recurrent networks [8,9], they gradually shrink the missing holes by repeating multiple encoding–decoding stages, which requires a large number of parameters and hard to control the number of cycles. These multistage inpainting networks all take result inferred by the previous stage as input and further predict the remaining missing pixels. Therefore, the errors

✉ Jinyang Jiang
1085220997@qq.com

¹ Sichuan University jinjiang College, Meishan, China

² Guang'an Vocational Technical College, Guang'an, China

³ School of Electrical Engineering and Electronic Information, Xihua University, Chengdu, China

predicted in the previous stage easily influence the inpainting of the next stage, resulting in distorted structures and texture artifacts in the restoration images. And the larger the missing region, the more likely it is occurs. These methods have several drawbacks: (1) They only utilize only feature priors [2,3,8,9] or gradient priors [1,4–6] to guide the inpainting process, which does not consider the global contents and local details together. (2) Priors are only used as input of the next stage, and the deeper layer would forget these guidance information. (3) All priors are used to directly guide the inpainting process, whether or not helpful. Although papers [10,11] design new architecture to address the problems of multistage networks, not all of above-mentioned drawbacks are resolved. MADF [10] progressively fill and refine the missing contents through multilayer prior guidance, but it only allows feature priors to be passed between the recovery decoder and the refinement decoders. Paper [11] avoids structural repair alone and points out that texture and structural information interact with each other during restoration. However, structural and texture information directly guide each other in multiple layers, and misinformation can easily affect the whole reconstruction process.

Global and local information is equally important to understand an image. Existing approaches, such as the multi-column network (MC-Net) [6] and ACGAN [12], aim to map an image to multiscale features by adopting parallel encoding branches with various receptive fields. However, fewer studies have explored how to effectively fuse and reconstruct multiscale features into the image space. This is the fourth issue to be solved: (4) mapping only the last feature maps to the image space tends to cause semantic ambiguity in restoration images.

To completely address these problems, we propose a new end-to-end training model and design a progressive guidance decoding network, whose multiple parallel decoding branches can achieve coarse-to-fine restoration by progressively passing down reconstructed priors. The reconstructed maps of the previous decoding branch are passed to the next decoding branch, providing priors for the reconstruction process of the next decoding branch.

Our designed network improves the model performance in three aspects. First, feature priors and gradient priors are conveyed in multiple layers, rather than being the only input of the next branch. The feature priors represent the global semantics and the gradient priors reveal sharpness regions, including edges and details, which help to enhance the prominent information of the repaired image. Second, we adopt ghost convolution [13] and designed cascaded effective channel attention (CECA) to further process stacked features from different decoding branches, enhancing the contributing parts of the stacked features and suppress the inappropriate contents. The specific operation is introduced in the section “[Progressive guidance decoding network](#)”. Finally, our

paper not only utilizes the attention-based multiscale perceptual res2block (AMPR) [14] to extract multiscale features, but also designs efficient multiscale fusion (EMF) to map different-level features into the final image. Coarse-scale feature maps have large receptive fields, focusing more on global semantic information. Fine-level feature maps have small receptive fields, focusing more on local detail information.

Our network effectively avoids the some drawbacks of multistage networks and improves the stability in the inpainting process. Experimental results on the publicly available datasets Places2 [15] and CelebA-HQ [16] demonstrate the effectiveness of the proposed model, especially when dealing with large and complex holes. Our designed decoding work mainly contains three contributions:

(1) The progressive guidance decoding network consists of several parallel decoding branches. The previous decoding branch provides multilayer feature priors and gradient priors for the inpainting process of the next decoding branch. The feature priors contain the global semantics and the gradient priors contain the high-frequency information, such as details and contours.

(2) We newly designed Cascaded Efficient Attention Mechanism (CECA) to reweight fused feature maps or gradient maps, which enhances the contributing parts and suppresses the inappropriate parts.

(3) Our proposed EMF maps different-level features into the final completed image. The inclusion of coarse-scale features can enhance the global semantic information of the restoration image.

Related work

Deep convolutional networks have shown strong potential in computer vision tasks, and learning-based methods have shown promising inpainting performance as well. In this section, we divide the recent works related to our method into three categories and describe them, including single-stage inpainting, multistage inpainting, and multistream inpainting.

Single-stage inpainting

Pathak et al. [17] proposed the Context Encoder (CE), where we assume that the semantics of the holes can be learned by a series of convolutional layers and reasonable losses (pixel-level reconstruction loss and adversarial loss). Although the inpainting results have some defects, such as obvious artifacts and ambiguity, they lay a foundation for subsequent research work. Based on CE, Global & Local [18] use two discriminators, namely the global discriminator and local discriminator, to ensure the global and local semantic consistency with the surrounding areas, respectively. PEN-Net [19] leverages

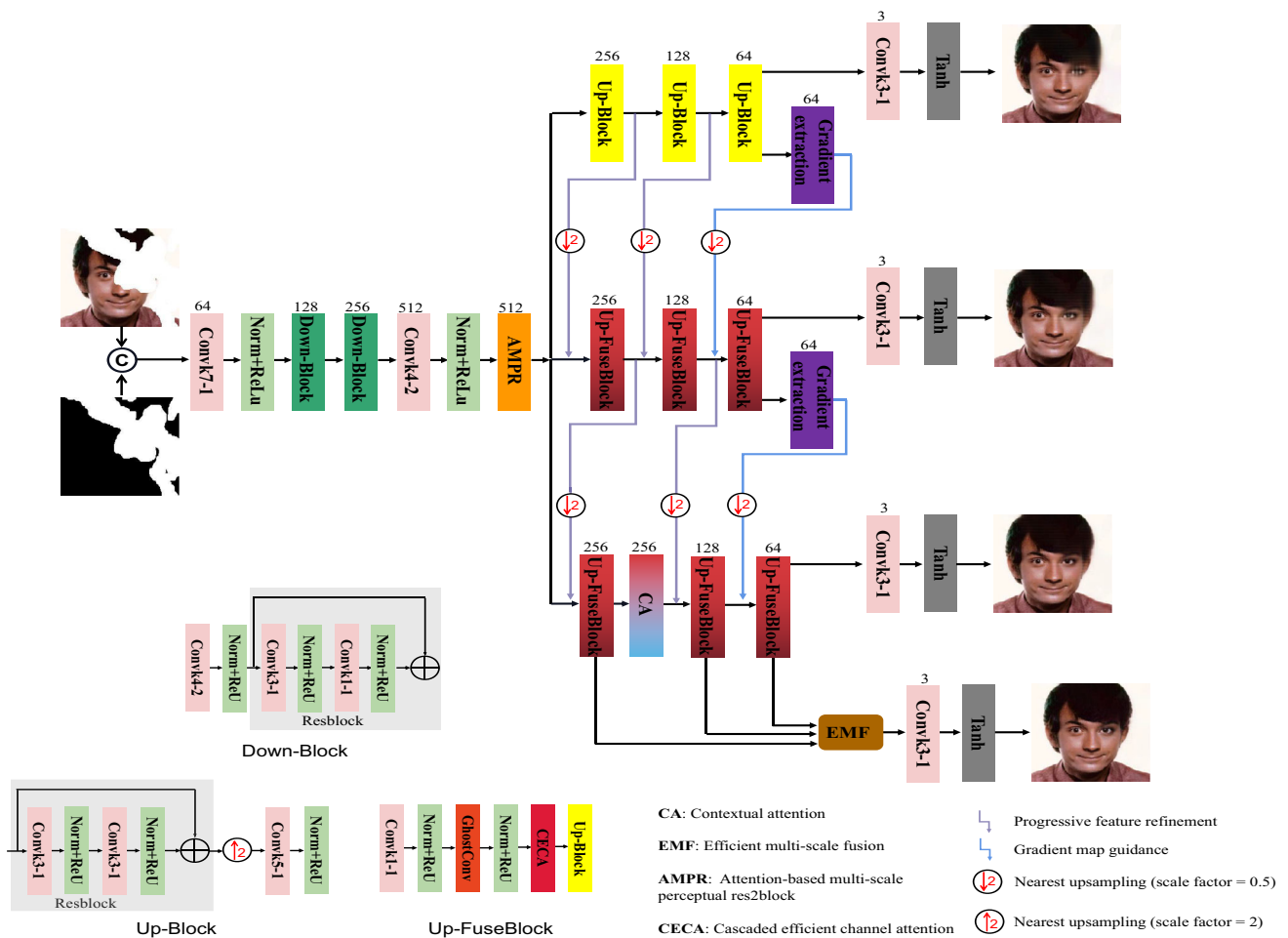


Fig. 1 The overall pipeline of the generator. It adopts a typical encoder–decoder structure. In the encoder, Down-Blocks and AMPR, for example, are used to understand the semantic information. In the decoder, feature maps and gradient maps from the preceding branch are passed to the next branch for further refinement and guidance. In our paper, the gradient maps are extracted by depthwise convolution

(Depthconv) with fixed sobel kernels. In addition, Up-FuseBlocks are utilized to effectively fuse maps from different branches. Finally, the EMF is used to merge the multiscale feature maps, so that the reconstruction image contain multiscale information. “Norm” represents the Instance Normalization. “Conv” denotes the convolution. The details of “k *-*”. * denotes the kernel size, and ** denotes the stride

U-Net with pyramid attention to progressively learn missing regions from the feature-level maps to image-level ones, which ensures both visual and semantic coherence. SGENet [20] iteratively updates the structural priors and the inpainted image in an interplay framework. It utilizes semantic segmentation map as guidance in each scale of inpainting. However, semantic segmentation may has poor boundary segmentation, resulting in incorrect object edges in the restoration images.

Multistage inpainting

Recently, multistage inpainting networks [1–5,7–9,21] have became the mainstay of solving the ill-posed inpainting issue. The coarse-to-fine network proposed by Yu et al. [2,3] 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 the attention mechanism. Some other methods utilize the inherent priors of an image, such as edges [4,5], structures [7], and segmentation maps [1,22], to provide structure priors for image completion process. Usually, the first network is used to fill the missing structural clues. Then, the complete clues are integrated in the next network, as one of inputs, to guide the synthesis of the missing contents. FRRN [8] progressively infers the valid pixels by iteratively adopting the full residual block with step loss. However, both input and inferred outputs are represented in the image space, resulting in expensive computation and less practical. RFR [9] gradually shrinks the large holes of feature maps from the boundary to the center through multiple recurrences. It repeatedly estimates the hole boundaries and sends the repaired ones to the next recur-

rence. Finally, the inferred feature maps per recurrence are merged to generate the output. The drawbacks of multistage inpainting network have discussed in the section “Introduction”.

Multistream inpainting

Unlike multistage networks that utilize several cascade networks to repair missing areas, some methods [6, 10–12, 23, 24] adopt multiple parallel networks to complete restoration, and here, we categorize them as multistream networks. Papers [6, 12, 23] realize that reasonable feature representation is important for image inpainting and use parallel branches to extract multiscale features. Multicolumn [6] adopts a multiscale encoder including three branches, and different branches transform the image into features with various receptive fields. ACFGAN [12] designs coarse-and-fine structures. The coarse path learns the semantic content with a larger receptive field by utilizing the cascaded dilated convolutions, and the fine path can extract more details with a smaller receptive field. JPGNet [24] utilizes parallel predictive filtering and generative network to preserve local structure and fill numerous missing pixels, respectively. It not suitable for restoring large missing areas, since the accurate predictive filtering relies on a large number of neighboring pixels. Paper [11] designs 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. MADF [10] employs a series of parallel refinement decoders with designed Pointwise Normalization (PN) to progressively refine the missing contents through multilayer prior guidance. Each layer takes the feature maps from the next upsampling layer of the previous decoder as priors. The drawbacks of paper [10, 11] have discussed in the section “Introduction”.

Proposed methods

Our proposed end-to-end architecture consists of generator and discriminator. The generator, as shown in Fig. 1, consists of an encoding network and an progressive decoding network including multiple parallel decoding branches. The encoding network can learn multiscale features through our designed AMPR. In the progressive decoding network, the reconstructed features and gradients from preceding decoding branch are utilized to guide the reconstruction of next decoding branch, which progressively fill and refine the masked regions.

Our discriminator follows the two PatchGANs as in [7], predicting the authenticity of all image patches with different sizes instead of the whole image. Spectral normalization is

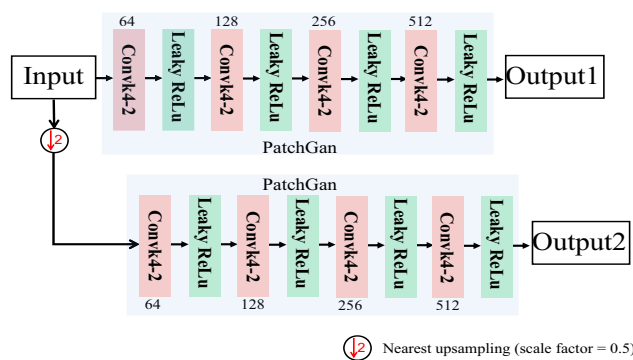


Fig. 2 The overall pipeline of the discriminator and two PatchGANs have the same structure. “Conv” denotes the convolution. The details of “k *-*”. * denote the kernel size, and ** denotes the stride. Spectral normalization is used in the discriminator

used in the discriminator to stabilize the training as well. The overall pipeline of the discriminator is shown in Fig. 2.

In this section, we first describe the encoding network and the details of the progressive decoding network in the sections “Encoding network” and “L1 loss”, respectively. Then, the corresponding loss function is presented in the section “Loss function”.

Encoding network

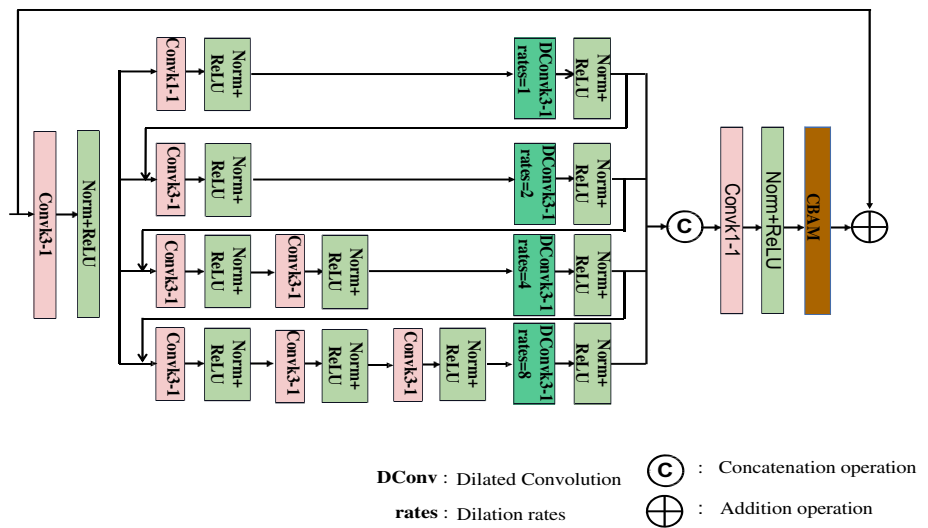
The encoding network aims to compress the 256×256 broken image into multilevel feature maps, which allows the computer to understand the semantic content. Damaged images always contain multiscale objects, so suitable feature representations play a vital role in understanding the relationship between missing and known regions.

Receptive field block (RFB) [25] combines the unique characteristics of Inception [26] and ASPP [27]. Inspired by it, we have designed AMPR (Fig. 3) in paper [14], which takes both the size and dilation rate of the convolution kernel into account. In addition, parallel branches are connected to each other using a skip connection like Res2net [28], which not only extracts multiscale features at granular levels, but also obtains an accurate object position. Finally, we use the attention mechanism CBAM [29] to fuse multiscale features from different branches. We apply this block in the last layer of the encoding network as in [30] to extract multiscale features.

Progressive guidance decoding network

One-shot decoding network is insufficient to reconstruct satisfactory images, especially for large holes. To improve the quality of reconstructed results, we design a progressive decoding network (Fig. 1), including a sequence of decoding branches used for progressive reconstruction and refinement of missing contents. It mainly depends on feature and gra-

Fig. 3 The structure of AMPR. In convolution, the default value of the rates is 1



gradient guidance. In addition, contextual attention [2] builds a long-term relationship between the complete features and missing features. Thus, we use it in the first upsampling block of the last decoding path to produce realistic and clear textures.

For easier reading, the notations used in the following text are described here. We utilize $D_{k,l}$ to denote the l -level upsampling block (Up-Block or Up-FuseBlock) of the k th decoding branch, $f_{k,l}$ and $g_{k,l}$ to denote the corresponding output feature maps and gradient maps, respectively, f_e to denote the compressed feature maps from the encoding network, and f_{out} to denote the output feature maps of the whole network. In this paper, we set both k and l to 1, 2, 3.

Progressive feature and gradient guidance

In this section, we will introduce how to realize feature guidance and gradient guidance, respectively.

Feature guidance: the recovered features $f_{k,l}$ of the branch $D_{k,l}$ are fed to the next branch $D_{k+1,l-1}$ and fused with the $f_{k+1,l-1}$. In this way, the reconstructed features can provide feature priors for the subsequent restorations and being further refined.

Gradient guidance: the gradient information reveals the difference between the adjacent pixels. This means that flat areas of an image have small gradient values, while sharp areas, including edges and details, have large gradient values. Thus, gradient prior can enhance the edges and details of the recovery image. Our proposed gradient guidance has two parts. One is the gradient map guidance (shown in Fig. 1). Rich gradient information of the last block from the preceding branch $g_{k-1,l}$ is passed to the next decoding branch step by step, which helps the model concentrate more on the reconstruction of sharp regions. And the other is gradient

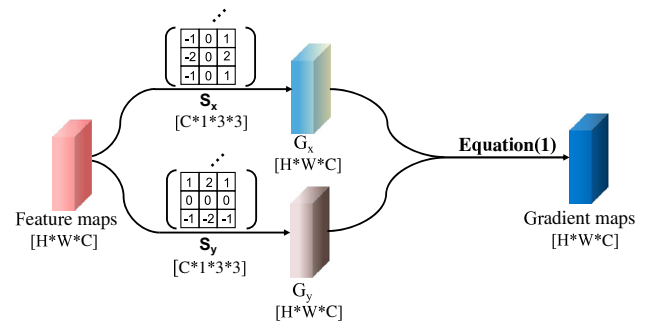


Fig. 4 The implementation of gradient extraction

loss restriction (the section “**Gradient loss**”) for each recovery image.

The process of gradient extraction is shown in Fig. 4. First, we utilize the sobel operator [31] with different orientations as the fixed convolution kernels S_x and S_y . The shape of sobel kernels is $[C, 1, 3, 3]$, representing output channels, input channels, kernel size, and kernel size, respectively. Second, we use the fixed sobel kernels S_x and S_y to convolve the feature map by the `nn.functional.conv2d` function, in which the groups are set to C . This step generates the horizontal and vertical gradient maps G_x and G_y . Finally, each value of gradient maps is calculated by (1)

$$G = \sqrt{G_x^2 + G_y^2}. \tag{1}$$

Fusing strategies

When fusing the feature maps and gradient maps from two branches, we propose two strategies to fuse features more effectively and efficiently. First, the ghost convolution, proposed by Han et al. [13], achieves ordinary convolution in

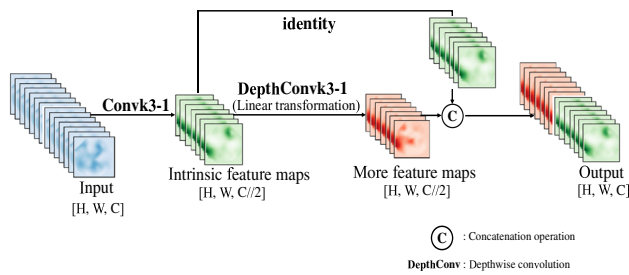


Fig. 5 The implementation of ghost convolution

a more cost-efficient manner. Thus, we replace the convolution of learning features with the ghost convolution. As shown in Fig. 5, the process of the ghost convolution is split into three steps. First, an ordinary convolution with fewer kernels is used to obtain intrinsic features. Based on a generated batch, a series of cheap linear operations are then adopted to obtain more feature maps, including redundant maps. Finally, these two batches are stacked in the channel dimension to learn the data distributions. If both the input and output channels of a convolution layer are set to 256, the parameters of the $k \times k$ ordinary convolution are $256 \times 256 \times k \times k + 256 = 65,536k^2 + 256$, while the parameters required for the ghost convolution are only $(256 \times 128 \times k \times k + 128) + (128 \times k \times k + 128) = 32,896k^2 + 256$. This value difference indicates that the combination of reduced-dimensional convolution and the ghost convolution is more practical.

The complete procedure of the progressive feature refinement can be seen in Fig. 1. Starting from the second decoding branch, the feature map f_e , $f_{k,l-1}$ or $g_{k,l-1}$ is first stacked with the reconstructed maps $f_{k-1,l}$ of the preceding branch in the channel dimension. Then, these stacked features go through a 1×1 compress convolution following the ghost convolution and CECA in order. Finally, the recalibrated features are further reconstructed by Up-FuseBlock.

We then add extra protection: an attention mechanism, which screens for high-value information from massive ones. When fusing features, we can utilize the attention mechanism to highlight critical features and prevents invalid features from entering the upsampling block, which further avoids the instability existing in multistage networks. The ECA [32] replaces the 2D convolution or full-connected layer used in the previous channel attention approaches with the 1D convolution, avoiding the channel compression and showing the potential ability to achieve cross-channel interaction in a cost-efficient way. However, ECA only explores the feature relationship between adjacent channels and has smaller receptive fields. To overcome this limitation, we introduce the 1D dilated convolution and extend ECA to a novel attention module called Cascaded ECA (CECA). As shown in Fig. 6, CECA consists of two consecutive 1D convolutions.

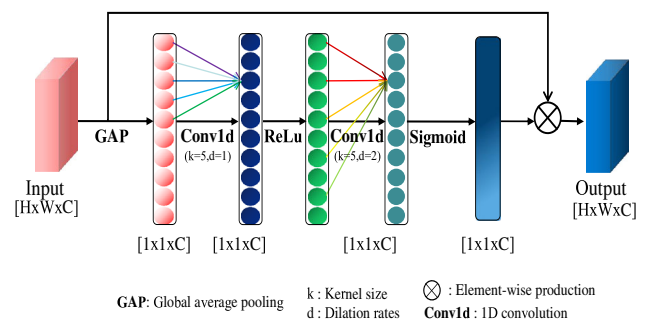


Fig. 6 The implementation of CECA. In Conv1D, the colorful arrows denote the different kernel weights

The first 1D convolution, with a dilated rate of 1, builds the connections among adjacent channels; the second 1D dilated convolution, with the dilate rate of 2, explores the long-term connections among non-adjacent features. In addition, cascaded design can ensure adequate cross-channel interactions while avoiding the inefficient capture of all channel dependencies at once as in the traditional channel attention.

Efficient multiscale fusion

To enable the restored image to include different-scale information, we integrate different-resolution maps in the last decoding branch. Generally, feature maps with high resolution often contain rich texture information that is crucial to the subjective perception. And feature maps with low resolution contain global semantic information. As illustrated in Fig. 7, our fusing strategy is named EMF. Inspired by CRM [33], our EMF mainly contains two attention mechanisms: $Attn_l$ and $Attn_{co}$. One is to explore the dependencies between features in a individual map, and the other aims to collectively preserve important features in all different-scale maps.

Feature map $f_{3,l}$ produced by the 1 blocks of the final decoding path ($l = 1, 2, 3$) has different resolutions (64×64 , 128×128 , and 256×256). These different-scale feature maps are upsampled to the same resolution before fusing. We first employ our designed CECA on $f_{3,l}$ and obtain the corresponding channel attention vectors $Attn_l$, which highlight the contributing channels and suppress the useless ones. The common attention vector $Attn_{co}$ is generated by adding all unique $Attn_l$ along the channel dimension and entering the Softmax function. Large values in the vector indicate that the all cross-scale features from this channel are beneficial for final reconstruction.

The obtained $Attn_l$ and $Attn_{co}$ are applied on the input feature maps $f_{3,l}$ in a channel-wise multiplication operation, which generates the recalibrated maps $f_{3,l}^a$ and $f_{3,l}^{co}$, respectively. Subsequently, each discriminative feature map $f_{3,l}^*$ is produced by adding $f_{3,l}^a$ and $f_{3,l}^{co}$, which contain both

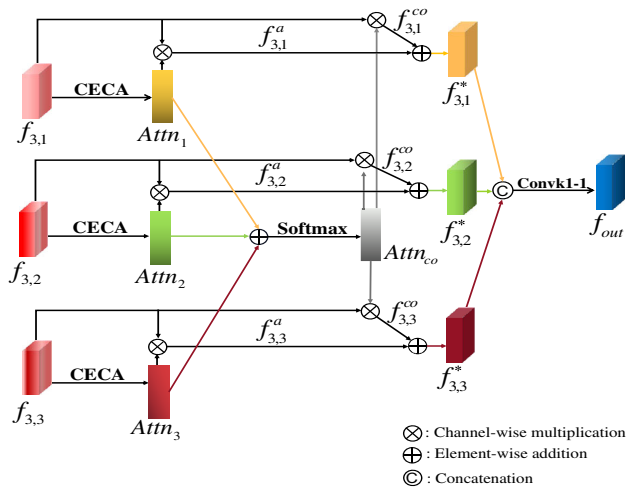


Fig. 7 The implementation of EMF

self-unique and co-critical information. Finally, 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 experiment described in the section “Experiment” can demonstrate the superiority of our fusion module over CRM. The procedure [33] can be defined as

$$f_{3,l}^a = Attn_l \otimes f_{3,l} = CECA(f_{3,l}) * f_{3,l} \tag{2}$$

$$f_{3,l}^{co} = Attn_{co} \otimes f_{3,l} = Softmax \left(\sum_{l=1}^3 Attn_l \right) * f_{3,l} \tag{3}$$

$$f_{3,l}^* = f_{3,l}^a + f_{3,l}^{co} \tag{4}$$

$$f_{out} = Convk1 - 1(Concatenate(f_{3,1}^*, f_{3,2}^*, f_{3,3}^*)). \tag{5}$$

Loss function

First, we utilize the different losses as in paper [7] to converge our model, including L1loss, perceptual loss [34], adversarial loss [35], and style loss [34]. In addition, we add the gradient loss for restricting the gradient space of the reconstructed image and the pyramid loss [14] to supervise the intermediate features. During the training process, the last branch is applied on all losses, while other branches are constrained by L1loss and gradient loss.

Pyramid loss

We perform pyramid perceptual loss on the $f_{3,1}$ and $f_{3,2}$ of the last decoding branch, which refines the predictions for missing regions at each scale so as to reconstruct final images in the right direction. We first use the activation layers $\{relu3_1\}$ and $\{relu2_1\}$ of VGG19 [36] to extract feature maps with two resolutions from an real image, and then cal-

culate the loss $L_{pyramid}$ between the extracted real features and the features predicted by our corresponding generation layer ($f_{3,1}$ and $f_{3,2}$). The loss calculation is given in (6)

$$L_{pyramid} = \sum_{i=2}^3 \|\Phi_i(I_{gt}) - f_{3,i}\|_1. \tag{6}$$

Here, I_{gt} denotes the real image. Φ_i denotes the i th selected activation layer of the pretrained VGG network. $f_{3,l}$ denotes the l -level feature maps of the last decoding branch. Note that the size of $f_{3,l}$ is the same as $\Phi_i(I_{gt})$.

L1 loss

Given the reconstructed features $f_{1,3}$, $f_{2,3}$, $f_{3,3}$ and f_{out} , we first transform them into an image space and use the notation $I_k(k = 1, 2, 3)$ and I_{out} to represent each, respectively. In addition to constraining the output image I_{out} , the L1 loss also constrains the generated images I_k from each branch. The L1 loss is defined as the mean absolute error between each completion image I_k and ground truth I_{gt} . The calculation is seen in (7) and (8)

$$L_1^k = \sum_{k=1}^3 \|I_k - I_{gt}\|_1 \tag{7}$$

$$L_1^{out} = \|I_{out} - I_{gt}\|_1. \tag{8}$$

Perceptual loss

The perceptual loss and style loss are based on the pretrained VGG network, which forces the generated semantic structures and rich textures to be similar to the ground truth. Let $\Phi_i(x)$ 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

$$L_{per}^{out} = E \left[\sum_{i=1}^5 \|\Phi_i(I_{out}) - \Phi_i(I_{gt})\|_1 \right]. \tag{9}$$

Style loss

The style loss is shown in (10). We employ a corresponding Gram matrix (G_j) on each selected feature map generated by VGG19, and use L1 loss to calculate the error

$$L_{sty}^{out} = E_j \left[\sum_{i=1}^5 \|G_j^{\Phi_i}(I_{out}) - G_j^{\Phi_i}(I_{gt})\|_1 \right]. \tag{10}$$

Adversarial loss

The generator aims to generate as realistic an image as possible to fool the discriminator. The goal of discriminator is to endeavor to judge whether the input is from the ground truth or not. This process of the game will make the generator output increasingly high-quality. PatchGAN and spectral norm are always used to train the inpainting model, which solves the problem of model collapse and stabilizes the converge process of the discriminator

$$L_{adv}^{out} = E[\log(1 - D(G(I_{in}, M)))] + E[\log(D(I_{gt}))]. \quad (11)$$

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

Gradient loss

Gradient loss is the second part of gradient guidance, which is applied on the final recovery image of the each branch. Under the gradient constraint, both the reconstructed intermediate images I_k ($k = 1, 2, 3$) and final image I_{out} contain rich and correct details and edges, which benefits the gradient maps guidance in the decoding network (section “Fusing strategies”), as well. As shown in (12)–(13), we formulate the gradient loss by narrowing the distance between the gradient maps extracted from the recovery image and ones from the ground truth. The implementation of gradient extraction is shown in Fig. 6

$$L_{gra}^k = \sum_{i=1}^3 \|Gra(I_k) - Gra(I_{gt})\|_1 \quad (12)$$

$$L_{gra}^{out} = \|Gra(I_{out}) - Gra(I_{gt})\|_1, \quad (13)$$

where Gra represents the process of gradient extraction.

Overall loss

The total loss is defined in (14). For the weight setting of most losses, we refer to the paper [7]. And for the L_{gra}^{out} , L_1^k and L_{gra}^k , we define their weights by experiment (Section “Discussion on the weights of L_{gra}^{out} , L_1^k and L_{gra}^k ”)

$$L_{total} = 0.01L_{pyramid} + L_1^k + 5L_1^{out} + 0.1L_{per}^{out} + 180L_{sty}^{out} + L_{adv}^{out} + L_{gra}^k + 3L_{gra}^{out}. \quad (14)$$

Experiment

In this section, we start by providing the detailed experimental settings. Then, we compare our proposed models with previous state-of-the-art algorithms through objective quantitative experiments and subjective qualitative experiments.

Experimental settings

Datasets

We train and evaluate our model on two well-known public datasets with different characteristics.

Places 2 [15] is a collection that contains 365 nature scenes and over millions images. Our model is trained on the standard training splits, including 1.8 million images, and evaluated on 10 k images chosen from validation splits.

CelebA-HQ [16] is a dataset that focuses on high-quality human faces with the size of $512 * 512$. 27 K images are selected for training, and the remaining images are used for evaluation.

Mask dataset [37] is a dataset that provides 12 k irregular mask images. These masks can be classified into different categories based on the hole-to-image area ratios (e.g., (0.1, 0.2] and (0.2, 0.3]).

Training settings

During training, we train the model with the batch size 8 using the Adam [38] optimizer, and the corresponding parameters (β_1 , β_2) and learning rate are set to 0.9, 0.999, and 0.0002, respectively. All experiments are conducted on an RTX 3090 GPU (24G), and the training process of CelebA-HQ and Places2 takes about 2.5 days and 9 days, respectively. All the masks and images for training and testing are the size $256 * 256$. Our model does not require any post-processing.

To make a fair comparison, we train our model with the same mask type as used in pretrained state-of-the-art models (the section “State-of-the-art algorithms for comparison”), including the center square mask, random rectangular mask, and irregular mask. The center square mask means that all images are masked with a $128 * 128$ square bounding box in the center position. The random rectangular mask represents that the image images are randomly covered with a blank rectangular, and the size between $64 * 64$ and $128 * 128$. The irregular mask originates from the mask dataset provided by Nvidia [37].

State-of-the-art algorithms for comparison

We compare our model with eight recent state-of-the-art ones: Multicolumn network (MC-Net) (ECCV 2018) [37], EdgeConnect (EC) (ICCVW 2019) [5], PEN-Net (CVPR

Table 1 Quantitative results comparison between our proposed model and MC-Net. We test the incomplete image with center holes on the CelebA-HQ dataset

DATASET	MODEL	Central holes		
		SSIM	PSNR	LPIPS
CelebA-HQ	MC-Net [6]	0.916	26.63	0.0454
	Ours	0.920	27.12	0.0433

2019) [19], Gated Conv (GC) (ICCV 2019) [3], StructureFlow (SF) (ICCV 2019) [7], RFR (CVPR 2020) [9], ACFGAN (Neurocomputing 2020) [12], and MADF (TIP 2021) [10]. To be fair, all the comparison algorithms are evaluated by their officially released pretrained model as much as possible. However, for EC [5], SF [7], RFR [9], and MADF [10], they do not validate their performance on the CelebA-HQ dataset, so we retrain them using the default parameters provided in their source code. For the Places2 dataset, the pretrained model of MC-Net [37] and RFR [9] is not available, so we only compare them on the CelebA-HQ dataset with our model owing to the restriction of the training time.

Based on center square holes, we compare our model with MC-Net [6]. Based on random rectangular holes, we compare it with PEN-Net [19] and ACFGAN [12]. When filling the irregular holes, we compare it with EC [5], SF [7], GC [3], RFR [9], and MADF [10].

Quantitative evaluation

As in previous image inpainting works, we measure the models' inpainting performance in various scenes using the PSNR, SSIM, and LPIPS [39] indexes. PSNR measures the L2 distance between the real image and repaired image at the pixel level. SSIM measures structural similarity between two sources by calculating the mean, standard deviation, and covariance, which reflect human perceptions more precisely compared with PSNR. LPIPS first uses Alexnet to extract the features from the real image and generated image, respectively, and then calculates their feature distance. Tables 1, 2, 3, and 4 show the evaluation results over CelebA-HQ and Places2 dataset. For PSNR and SSIM, the higher the better, while, for LPIPS, the lower the better.

Evaluation results with the regular holes

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

the comparison models in various scenes, indicating that our model can fill the larger holes effectively.

Evaluation results with the irregular holes

Tables 3, 4, and 5 show performance comparison between different methods under irregular holes. For the CelebA-HQ dataset (Table 3), when the mask ratio increases, our model outperforms state-of-the-art models except for MADF [10] in terms of PSNR and SSIM, but our LPIPS is best among all methods. We posit that this is because the MADF is not trained on adversarial loss, and the PSNR is substantially higher than ours. For the Places2 dataset (Table 4), the advantage of our model emerges as the mask ratio increases, and LPIPS index is comparable to MADF. In addition, we compare the parameters between ours and MADF (shown in Table 5). Although MADF performs well on some objective metrics, our model has fewer parameters than MADF and more practical. Taking performance and cost into account, our model is a better choice for restoring damaged image.

Qualitative evaluation

Quantitative metrics cannot fully reflect humans' subjective feelings, so qualitative evaluation is introduced as an another judgment criterion. Image inpainting technique is not only used to complete missing features, but also to edit images. In this section, we demonstrate the superiority of our approach in both two applications.

Completing missing features

Completing missing features aims to recover the real structure and texture in the missing areas, as similar as possible to the groundtruth. Figures 8, 9, and 10 show the repairing results of our method and other methods on the people faces (CelebA-HQ dataset). Figures 11 and 12 show the repairing results on the nature scenes (Places2 dataset).

When filling center holes (Fig. 8) in the CelebA-HQ dataset, the faces generated by our model are more similar to those of the real image compared to MC-Net. When filling random holes (Fig. 9), PEN-Net and ACFGAN generate unreasonable structures. Filling irregular holes is more challenging than filling regular holes. As shown in Fig. 10, the repaired regions of EC, SF, and GC always contain unreasonable structures and texture artifacts. For example, in the first row, the left eye generated by the EC and SF is asymmetric with the original right eye. In the third row, the skin produced by GC has obvious texture artifacts. RFR is able to generate plausible structures, but the results still contain low-quality regions, for example, the ears of the last row. The MADF shows the strong potential of generating reasonable structures and clear textures without artifacts, but it

Table 2 Quantitative results of the different methods with random rectangular holes on CelebA-HQ dataset and Places2 dataset

DATASET	MODEL	Central holes			Random holes		
		SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS
CelebA-HQ	PEN-Net (CVPR 2019) [19]	0.900	25.88	0.0615	0.929	27.09	0.0610
	ACFGAN (Neurocomputing 2020) [12]	0.906	26.16	0.0497	0.931	27.09	0.0413
	Ours	0.913	26.49	0.0469	0.938	27.80	0.0392
Places2	PEN-Net (CVPR 2019) [19]	0.732	20.63	0.1949	0.818	22.46	0.1314
	ACFGAN (Neurocomputing 2020) [12]	0.759	20.73	0.1457	0.837	22.62	0.0994
	Ours	0.765	21.20	0.1380	0.831	22.87	0.1108

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

DATASET	MODEL	0–0.2			0.2–0.4			0.4–0.6		
		SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS
CelebA-HQ	GC (ICCV 2019) [5]	0.982	33.80	0.0212	0.938	27.12	0.0693	0.846	22.58	0.1261
	EC (ICCVW 2019) [3]	0.985	34.85	0.0146	0.944	27.79	0.0480	0.842	22.72	0.1097
	SF (ICCV 2019) [7]	0.983	34.05	0.0173	0.941	27.42	0.0530	0.843	22.81	0.1261
	RFR (CVPR 2020) [9]	0.985	35.17	0.0133	0.947	27.97	0.0443	0.861	23.30	0.0979
	MADF (TIP 2021) [10]	0.987	35.63	0.0170	0.954	28.59	0.0580	0.877	23.92	0.1299
	Ours	0.985	34.55	0.0143	0.949	27.41	0.0442	0.870	23.61	0.0954

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

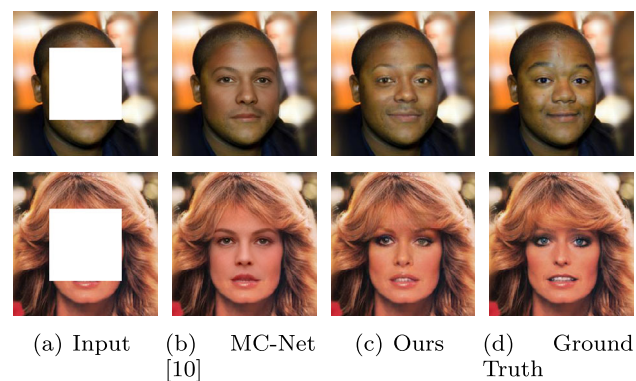
DATASET	MODEL	0–0.2			0.2–0.4			0.4–0.6		
		SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS
Places2	EC (ICCVW 2019) [38]	0.969	31.41	0.0270	0.880	24.08	0.1071	0.707	19.72	0.2250
	GC (ICCV 2019) [3]	0.969	30.36	0.0546	0.875	23.65	0.1406	0.711	19.41	0.1921
	SF (ICCV 2019) [7]	0.959	29.58	0.0359	0.870	23.24	0.1008	0.731	19.75	0.1916
	MADF (TIP 2021) [10]	0.969	31.50	0.0243	0.879	23.98	0.0840	0.728	20.13	0.1751
	Ours	0.968	31.04	0.0277	0.882	24.27	0.0878	0.731	20.48	0.1790

Table 5 Parameters of the our model and MADF

MODEL	Parameters
MADF(TIP 2021) [10]	347.6 M
Ours	103 M

is prone to generating smooth details and edges. Compared with the results of these baselines, our results have distinct and more coherent structures as well as fine textures, even though the holes are relatively large and complex. In addition, for the partial occlusion face recognition (e.g., mask occlusion) task, a common solution is to repair the occluded faces before recognizing them. Therefore, the performance of face restoration indirectly determines the accuracy of face recognition.

The Places2 dataset contains various scenes, both indoors and outdoors. As shown in Fig. 11, compared to those of other

**Fig. 8** Completing results on CelebA-HQ dataset with 128 × 128 center holes

models, our inpainting results have fewer noticeable mask artifacts and less bad textures. As shown in Fig. 12, when

Fig. 9 Completing results on the CelebA-HQ dataset with random regular holes of a size between $64 * 64$ and $128 * 128$

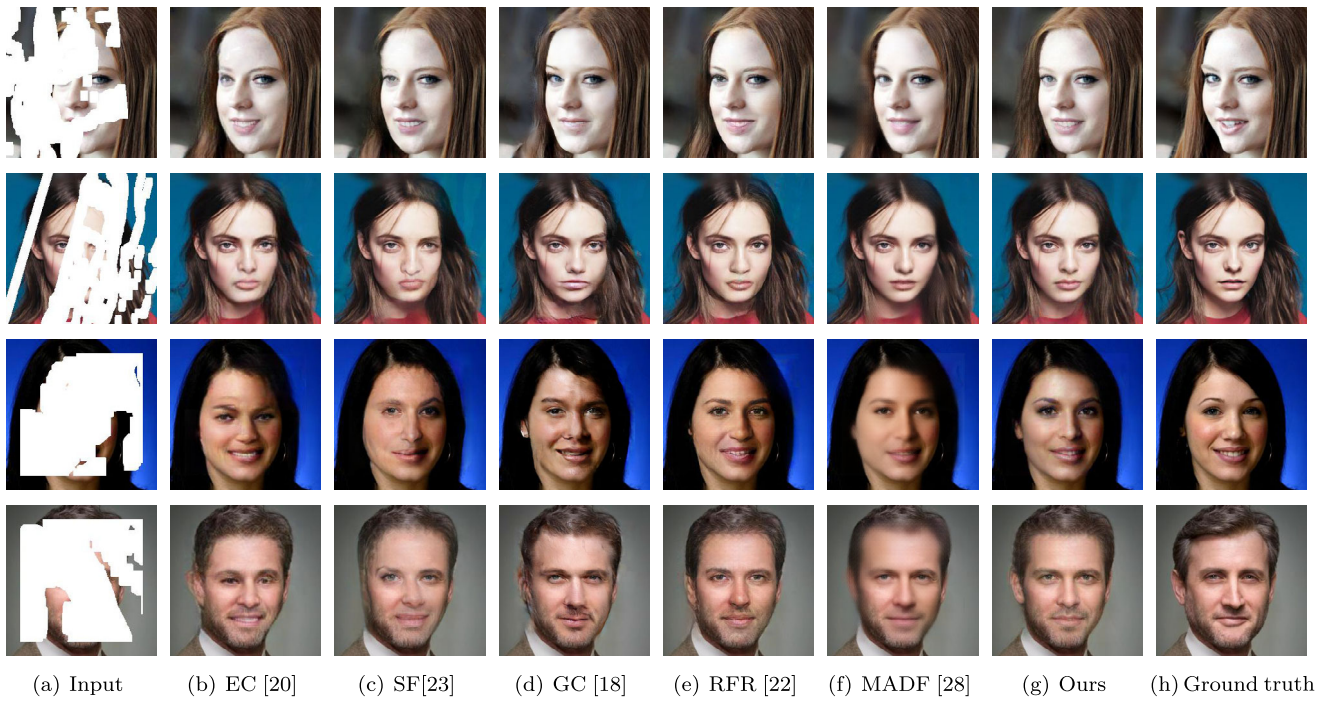
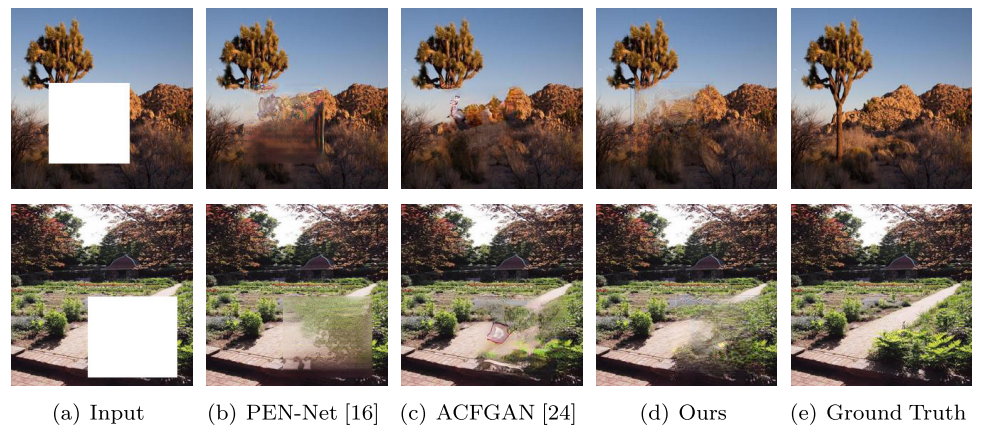


Fig. 10 Completing results on the CelebA-HQ dataset with irregular holes

Fig. 11 Completing results on the Places2 dataset with random rectangular holes of a size between $64 * 64$ and $128 * 128$



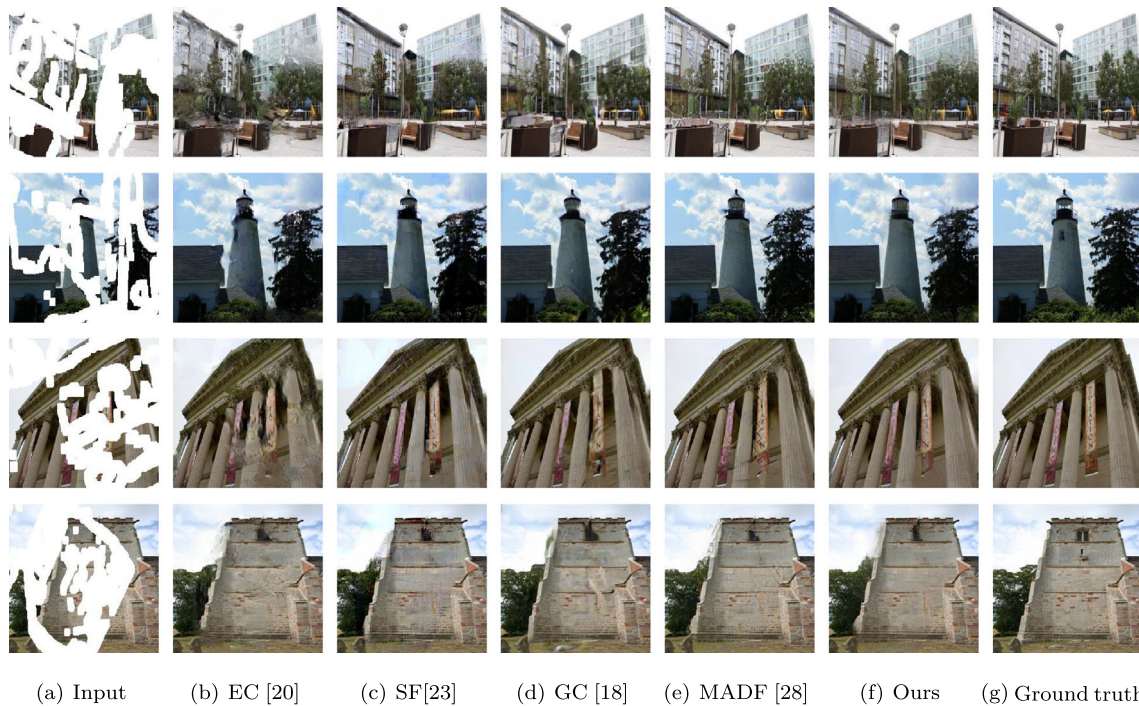


Fig. 12 Completing results on Places2 dataset with irregular holes

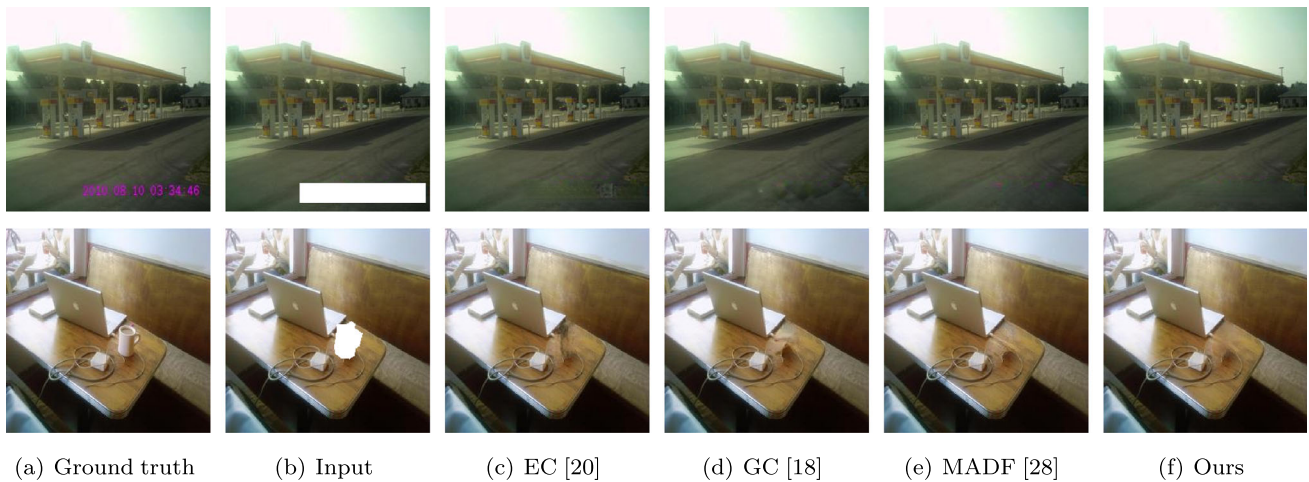


Fig. 13 Editing results of different models

filling irregular holes, our model produces better structures, and the results look clearer than those of other models.

Image editing

Image editing aims to remove unwanted objects from an image (such as objects or a captions) without leaving any trace of it. This is done by masking unwanted objects with a mask and then filling the mask with a background that matches the surrounding environment. The editing examples are shown in Fig. 13. The purpose of the first line of Fig. 13

is to remove time marker, and the second line is to remove the cup on the table. Observing the editing results, we can see that the outputs generated by our designed model have the least artifacts, are smoother, and are most consistent with the background area (zoom in for better comparison).

Ablation study

We conduct ablation experiments to demonstrate the effectiveness of our contributions. These experiments are per-

Table 6 Ablation results of each design choice

DATASET	MODEL	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	Baseline	0.889	25.15	0.0773
	Baseline+AMPR	0.891	25.21	0.0763
	Baseline[gc]+AMPR	0.891	25.20	0.0765
	Baseline[gc]+AMPR+CECA	0.893	25.28	0.0747
	Baseline[gc]+AMPR+CECA+EMF	0.895	25.34	0.0755
	Baseline[gc]+AMPR+CECA+EMF+GG	0.897	25.50	0.0751

Table 7 The effectiveness of CECA. We utilize ECA and our designed CECA to recalibrate the fused features from different branches

DATASET	MODEL	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	ECA	0.894	25.38	0.0761
	CECA	0.897	25.50	0.0751

formed on the CelebA-HQ test set with irregular masks with the mask ratio 20%–60%.

The effect of design choice

We carry out some experiments to demonstrate the effectiveness of each design choice of our network, including AMPR, ghost convolution, CECA, EMF, and gradient guidance. In Table 6, the “Baseline” is a basic architecture that contains only basic feature refinement, without the other design choices mentioned above. “GC” and “GG” denote the ghost convolution and gradient guidance, respectively. Baseline [GC] is a basic architecture that replaces an ordinary

convolution with a ghost convolution for fusion, the details of which are described in the section “Progressive feature and gradient guidance”.

From Table 6, we can see that the addition of AMPR, CECA, EMF, and gradient guidance improves the network performance. The results in the second row and third row show that the replacement of the ghost convolution drops the performance slightly, yet this design reduces a lot of parameters. To achieve a good balance between efficiency and effectiveness, we still choose the ghost convolution.

CECA vs ECA

As mentioned in the section “Progressive feature and gradient guidance”, CECA captures dependencies between feature channels more effectively compared to ECA [32]. To verify this, we utilize CECA and ECA to reweight the fused feature maps in the Up-FuseBlock, respectively. The objective results are shown in Table 7. Following the same training strategies and experimental settings, the PSNR and SSIM index of CECA are increased by 0.12 dB and 0.03, respec-

Fig. 14 Visual results of ECA, CMR, and our proposed model (From left to right) on the CelebA-HQ dataset

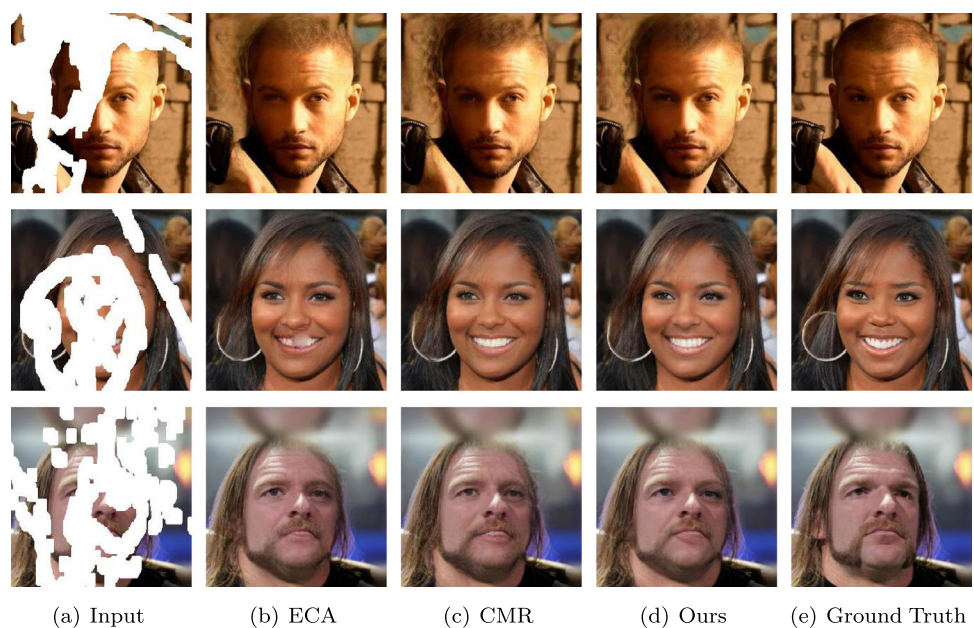


Table 8 The effectiveness of EMF. We test the model performance using CRM and EMF in the last decoding branch

DATASET	MODEL	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	CRM	0.894	25.41	0.0766
	EMF	0.897	25.50	0.0751

Table 9 The effectiveness of gradient guidance

DATASET	MODEL	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	Feature maps guidance (w/o gradient loss)	0.892	25.22	0.0774
	w/o gradient loss	0.894	25.30	0.0745
	Gradient guidance	0.897	25.50	0.0751

tively. The LPIPS index decreases by 0.01. The visual results are shown in Fig. 14, and we can see that the ECA fails to understand the semantic information of the eyes and mouth.

EMF vs. CRM

To further demonstrate the performance of our proposed EMF, we additionally conduct an experiment with replacing EMF with CRM [33]. As shown in Table 8, the CMR achieves an average PSNR of 25.41 dB, SSIM of 0.894, and LPIPS of 0.0766, which are inferior to those for the adoption of EMF. The visual results are given in Fig. 14. The last row produced by CMR shows the wrong color in the mouth, which is inconsistent with ground truth. The first row generates more blurry artifacts around the head.

The effectiveness of gradient guidance

The gradient guidance contains two parts: gradient maps guidance and gradient loss. To show the contribution of each part, we additionally conduct two experiments with feature maps' guidance and without gradient loss. Feature maps' guidance means that in the last Up-Block and Up-FuseBlock, the feature map is used instead of the gradient map for guidance.

Of note, the experiments with feature maps' guidance are performed without gradient loss, as well. First, observing the results in the first and the second row of Table 9, we see that the PSNR and SSIM indexes are significantly improved when using gradient map guidance. This is because the deepest block of the decoding network may need more gradient priors (e.g., details and edges) rather than fuzzy features. Taking the $D_{1,3}$ as an example, Fig. 14 shows some output feature maps and gradient maps. We can see that gradient maps have a stronger response in edges and details compared to feature maps.

Second, the results in the second and third row demonstrate the contribution of gradient loss. The PSNR and SSIM

indexes of the model with gradient loss are increased. This indicates that the gradient loss constrains the network to produce gradient maps with accurate high-frequency information, which better guides the detail reconstruction of next decoding branches.

Discussion on the weights of L_{gra}^{out} , L_1^k and L_{gra}^k

In this section, we first fix the weights of both L_1^k and L_{gra}^k to 1, and set the weight of L_{gra}^{out} to be 1, 2, 3, 4 and 5, respectively. Table 10 shows the effect of different weights on model performance, and the model has best inpainting performance when the loss weight is set to 3.

Then, we fix the weight of L_{gra}^{out} to 3, and set the weights of branch losses L_1^k and L_{gra}^k to be the same as the weights of final losses L_1^{out} and L_{gra}^{out} , i.e., 5 and 3, respectively. Table 10 shows the experimental results, and we can see that the model performs better when the branch weights both L_1^k and L_{gra}^k are set to 1.

Conclusion

In this paper, we have extended the single decoding branch to multiple decoding branches and proposed a progressive

Table 10 The weight discussion of L_{gra}^{out} . And the weights of both L_1^k and L_{gra}^k are fixed to 1

DATASET	Weight	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	1	0.893	25.29	0.0754
	2	0.895	25.44	0.0751
	3	0.897	25.50	0.0751
	4	0.894	25.37	0.0785
	5	0.896	25.49	0.0761

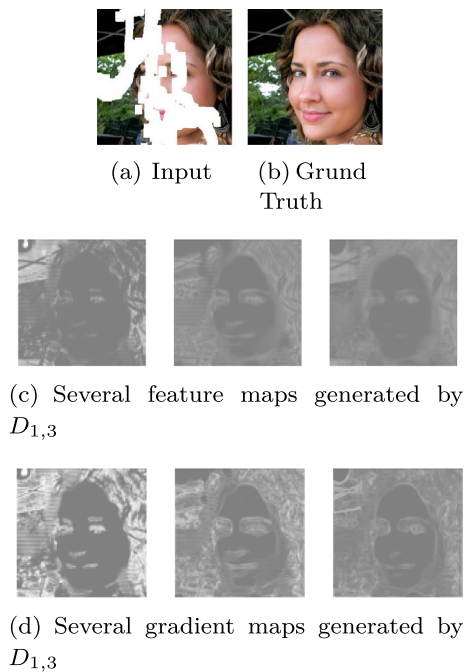


Fig. 15 Visual contrast diagram between feature maps and gradient maps

Table 11 The weights discussion of L_1^k and L_{gra}^k . And the weight of L_{gra}^{out} is fixed to 3

DATASET	Weights	0.2–0.6		
		SSIM	PSNR	LPIPS
CelebA-HQ	3 and 5, respectively	0.894	25.28	0.0799
	1 and 1, respectively	0.897	25.50	0.0751

decoding network, which progressively fills and refines missing regions. Specifically, considering characteristics of the different convolutional layers in the decoder, the recovery features (representing global semantic information) and gradients (representing local sharpness information) from the preceding branch are fed to the shallow layers and deep layer of the next branch, respectively. When fusing the intermediate maps from different branches, the ghost convolution and CECA are adopted to reduce the model parameters and avoid the forward propagation of incorrect information, respectively. In addition, our model explores multiscale information of an image by adopting AMPR in the encoder and EMF in the decoder. The proposed AMPR extracts multiscale features and improve feature representation ability. The EMF fuses the multiscale reconstructed features of the last decoding branch into final feature maps, which helps model to output the image containing global and local information in the image space. The effectiveness of the proposed framework has been validated both quantitative and qualitative experiments (Fig. 15, Table 11).

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

Acknowledgements This work is supported by the National Natural Science Foundation of China (11872069), Central Government Funds of Guiding Local Scientific and Technological Development for Sichuan Province (2021ZYD0034), the National Ministry of Education “Chunhui Plan” Scientific Research Project (Z2017076), the Chengdu Science and Technology Program (2016-YF04-00044-JH), and Natural Science Foundation of Sichuan Province [Grant No.2022NSFSC0914].

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

- Shao H, Wang Y, Fu Y, Yin Z (2020) Generative image inpainting via edge structure and color aware fusion. *Signal Process Image Commun* 87:115929
- Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Generative image inpainting with contextual attention. In: *Proceedings of the 2018 CVPR*, pp 5505–5514
- Yu J, Zhe L, Yang J, Shen X, Lu X, Huang T (2019) Free-form image inpainting with gated convolution. In: *Proceedings of the 2019 ICCV*, pp 4470–4479
- Xiong W, Yu J, Lin Z, Jiang J, Lu X, Barnes C, Luo J (2019) Foreground-aware image inpainting. In: *Proceedings of the 2019 CVPR*, pp 5833–5841
- 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*
- Wang, Y, Tao X, Qi X, Shen X, Jia J (2018) Image inpainting via generative multi-column convolutional neural networks. *Adv Neural Inf Process Syst* 331–340
- Ren Y, Yu X, Zhang R (2019) Structureflow: image inpainting via structure-aware appearance flow. In: *Proceedings of the 2019 ICCV*, pp 181–190
- Guo Z, Chen Z, Yu T, Chen J, Liu S (2019) Progressive image inpainting with full-resolution residual network. In: *Proceedings of the 27th ACM International Conference on Multimedia*, pp 2496–2504
- 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
- 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. *IEEE Transactions on Image Processing*, pp 4855–4866

11. Guo X, Yang H, Huang D (2021) Image inpainting via conditional texture and structure dual generation. In: Proceedings of the 2021 ICCV, pp 14114–14123
12. Chen M, Liu Z, Ye L, Wang Y (2020) Attentional coarse-and-fine generative adversarial networks for image inpainting. *Neurocomputing* 405:259–269
13. 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
14. Jiang J, Dong X, Fan Li, Zhang T, Qian H, Chen G (2022) Parallel Adaptive Guidance Network for Image Inpainting. *Applied Intelligence*. <https://doi.org/10.1007/s0489-022-03387-6>
15. Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2017) Places: A 10 Million Image Database for Scene Recognition. *IEEE TPAM* 40(6):1452–1464
16. Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*
17. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA (2016), Context Encoders: Feature Learning by Inpainting, in: Proceedings of the 2016 CVPR, pp. 2536-2544
18. Iizuka S, Simo-Serra E, Ishikawa H (2017) Globally and locally consistent image completion, *ACM Trans.Graphics(TOG)* 36(4) 107
19. 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
20. 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
21. Wang J, Chen S, Wu Z, Jiang Y-G (2022), FT-TDR: Frequency-guided Transformer and Top-Down Refinement Network for Blind Face Inpainting. *IEEE Transactions on Multimedia*
22. Pei Z, Jin M, Zhang Y, Ma M, Yang Y-H (2021) All-in-focus synthetic aperture imaging using generative adversarial network-based semantic inpainting. *Pattern Recognit.* 111:107669
23. Wang N, Wang W, Hu W, Fenster A, Li S (2021) Thanka Mural Inpainting Based on Multi-Scale Adaptive Partial Convolution and Stroke-Like Mask. *IEEE TIP* 30:3720–3733
24. 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
25. Liu S, Huang D, Wang Y (2018), Receptive Field Block Net for Accurate and Fast Object Detection, in: Proceedings of the 2018 ECCV, pp. 404-419
26. Christian S, Vincent V, Sergey I, Jonathon S, Zbigniew W (2016) Rethinking the inception architecture for computer vision, In :Proceedings of the 2016 CVPR, pp. 2818-2826
27. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018), Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the 2018 ECCV, pp. 801-818
28. Gao H, Chen M, Zhao K, Zhang Y, Yang H, Torr P (2019) Res2Net: A New Multi-Scale Backbone Architecture. *IEEE TPAM* 43(2):652–662
29. Woo S, Park J, Lee JY (2018) CBAM: Convolutional block attention module, in: Proceedings of the 2018 ECCV, pp. 3-19
30. Li T, Dong X, Lin H (2020) Guided Depth Map Super-Resolution Using Recumbent Y Network, *IEEE Access*, pp. 122695-122708
31. Duda RO, Hart PE (1973) in *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, pp 271–272
32. Wang Q, Wu B, Zhu P, Li P, Hu Q (2020), ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks, in: Proceedings of the 2020 CVPR, pp. 95-106
33. 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, pp. 9471-9481
34. 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
35. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D (2014) Generative adversarial nets, in: Proceedings of the 2014 NeurIPS, pp. 2672-2680
36. Simonyan K, Zisserman A (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition, in: Proceedings of the 2014 ICLR
37. Liu G, Reda FA, Shih KJ, Wang TC, Tao A, Catanzaro B (2018), Image Inpainting for Irregular Holes Using Partial Convolutions, in: Proceedings of the 2018 ECCV, pp. 85-100
38. Kingma DP, Adam JBa (2015) A method for stochastic optimization, in: Proceedings of the 2015 ICLR
39. Zhang R, Isola P, Efros AA, 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.