Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-13028-2

A multi-level feature integration network for image inpainting

Tao Chen[1] · Xin Zhang[1,2,3] · Bernd Hamann[4] · Dongjing Wang[1] · Hua Zhang[1]

Received: 10 September 2021 / Revised: 15 December 2021 / Accepted: 28 March 2022

Abstract
Deep learning-based methods have shown great potential in image inpainting, especially when dealing with large missing regions. However, the inpainted results often suffer from blurring, and improper textures can be created without an understanding of semantic information. In order to extract more features from the known regions, we propose a multi-level feature integration (MFI) network for image inpainting. We complete hole regions by two generators. For each generator, we use the MFI network to fill the hole region with multi-level skip connections. With multi-level feature integration, the network gains more knowledge about the global semantic structures and local fine details. Moreover, instead of a deconvolution layer or an interpolation algorithm, we adopt a sub-pixel layer to up-sample feature maps and produce more coherent results. We use PatchGAN to support the refinement generator network to produce more discriminative detail. Our experiments done with the Paris StreetView, CelebA-HQ and Places2 datasets demonstrate the effectiveness of our MFI network for producing visually pleasing results with semantically ordered textures.

Keywords Image inpainting · Multi-level · Feature integration · Skip connection

1 Introduction

Image inpainting is an important research topic in the fields of computer vision and image processing. Image inpainting has various applications, such as image restoration, photo editing, image encoding, transmission, etc. Specifically, image inpainting aims to generate convincing content by filling missing pixels according to the contextual information and feature distribution of the image.

Broadly speaking, existing image inpainting methods can be divided into two categories, traditional methods and deep learning based methods. Traditional methods [4, 7, 8, 18, 22, 38, 47] synthesize new pixels with explicit image features and priors. For example, the

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No.LQ21F020015 and No.LQ20F020015.

✉ Xin Zhang
zhangxin@hdu.edu.cn

Extended author information available on the last page of the article.

diffusion based methods [1, 31] grow the known neighbor information inward the missing region with smooth assumption. The basic idea of example based methods [5, 7, 47, 48] search for similar image patches from the source region and use them to fill the missing regions. Moreover, low rank priors [9, 22, 38] and sparsity priors [15, 46] are also used for solving image inpainting problem. For images with simple texture and small missing regions, traditional methods can realistically complete the image with high visual quality. However, for images with complex scenes and non-repeatable textures, traditional methods cannot inpaint the missing region with semantically reasonable contents.

Deep learning-based methods [12, 25, 28, 42] generally combine generator networks with discriminator networks to learn high-level features and the distributions of image dataset, and predict the missing regions through generator which has strong mapping capabilities. Those methods can generate semantic objects and complex textures, but blurry boundaries inconsistent with surrounding areas and disordered structures still remain in generated images. Inspired by attention mechanism successfully used in natural language processing, contextual attention models [17, 20, 41, 45] are proposed to capture semantic relevance in images. Specifically, the attention mechanism models the relationship between unknown patches and spatially global patches to generate more details at the cost of high complexity.

The generator networks in the existing methods are based on encoder-decoder architecture. The encoder part maps the image to high-level latent feature space by several convolutional layers with down-sampling at the same time, while the decoder part is responsible for reconstructing the encoded features to image space with hole region synthesized. Some methods [17, 26, 45] strengthen the generator by adding long skip connection between encoder layers and decoder layers which producing the same resolution features. This structure is known as U-Net [27]. However, these models can not make the best of the semantic information in the known regions. The local and global coherence is important for the visual quality of the generated image. To this end, we propose the **M**ulti-level **F**eature **I**ntegration (MFI) network for generating plausible content in image holes. Our work is inspired by papers [3, 6] which prove that having both long and short skip connections is beneficial for feature extraction.

We adopt a rough-to-fine architecture [41] and synthesize the content for the hole region in two stages. Specifically, the first stage completes the missing regions with global structures and rough textures. The coarse result of this first stage is the input of the second stage. In this way, the second stage has a larger receptive field and can generate more fine details. To better extract and integrate the contextual information from different levels, we apply both adaptive long skip connections and multi-level short skip connections to the generator network. Especially, with the long skip connections, the features extracted by the encoder layers are adaptively integrated into the decoder features, which can improve fine details in generated images and also stabilize the training process. We use the multi-level short skip connection in both encoder blocks and decoder blocks to integrate low-level features and high-level features for better coherence. Moreover, we propose to use a sub-pixel layer [29] in the decoder part to restore spatial information by incorporating features skipped at various levels, which improves the accuracy and stability of the network. Both the first and second stages use the proposed MFI network as the generator network, which integrates multi-level contextual information and is effective in generating semantically consistent structures and well-ordered textures. We combine reconstruction loss, structure loss and adversary loss with PatchGAN to optimize our inpainting model. We conducted extensive experiments for comparisons, an ablation study and a user study using different datasets. The results

demonstrate the superiority of our proposed network. The major contributions presented in this paper are:

- We propose a **M**ulti-level **F**eature **I**ntegration (MFI) network to complete the missing regions in images. MFI network can produce sharp structures, semantically ordered textures and consistent boundaries.
- We design the multi-level skip connections including adaptive long skip connection and multi-level short connection to integrate different level features to achieve local and global coherence.
- We use sub-pixel layer for up-sampling instead of a deconvolution layer or interpolation algorithm to alleviate checkerboard and fuzzy artifacts.
- Experiments on multiple datasets demonstrate the effectiveness of our proposed method.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the detailed architecture and key components of the proposed image inpainting method. Section 4 presents experimental results and analysis, including qualitative and quantitative comparisons, a user study and an ablation study. Section 5 draws conclusions and points out possibilities for future work. In the Appendix Section, we present the detailed architecture and all parameters used in the proposed model.

2 Related work

2.1 Deep learning-based image inpainting

Deep learning based image inpainting methods mainly use convolutional network and generative adversarial strategy for completing images with holes. Context encoders method [25] is the pioneer work for incorporating encoder-decoder generative network with discriminative network for image inpainting. This method is able to generate semantic contents, but also produce perceptual discontinuities. Iizuka et al. [12] extended this idea and introduced the globally and locally consistent discriminators. Unfortunately, this approach has limitations on complex structural textures and needs blending post-process to keep boundary coherence. Yang et al. [40] proposed a joint optimization model of content and texture networks to synthesize sharp structures and fine details. However, the texture constraints in this method needs to search the most similar patch for each unknown patch, which is time-consuming. Moreover, two-stage architectures are explored with different kinds of priors, such as edges [24], smooth images [26] and segmentation map [30]. These additional networks predicting the prior information can guide the image generative network to generate more semantically plausible details. But they are also probable to introduce noise information if the prediction accuracy is not high enough. Yu et al. [41] proposed a rough-to-fine architecture by cascading two generative networks, which can enlarge the receptive field and produce more refined texture details. How to make full use of low-level semantic information is also of concern, such as multi-level generative network by liu et al. [19] and densely connected generative networks by Shen et al. [28].

In addition, the attention mechanism has been studied in the context of image generation. Various attention-related modules are designed to improve the visual quality of generated images, such as the variants [33, 36, 39] of contextual attention [41], self-attention [32], coherent semantic attention [17], learnable bidirectional attention maps [37] and the attention transfer network [45]. The attention module uses known features to fill in the unknown regions, which is effective for generating visually pleasing images but has high computational complexity. Novel feature normalization methods [34, 43, 50] differentiate the valid

and hole regions when calculating feature statistics. However, region normalization tends to produce blurred results [43].

The above image inpainting methods make impressive progress for image inpainting with large area holes, and can produce more plausible content than traditional methods. However, due to the lack of deep semantic understanding, the generated images mostly have fuzzy boundaries and distorted structures. Some methods try to use post-processing or spatial attention mechanism for further improvement, but the complexity also increases. Therefore, we propose a novel multi-level feature integration method, which is presented in detail in Section 3.

2.2 Skip connections

Skip connections are commonly used in network structures. On the one hand, they integrate the features from shallow layers into features from deeper layers, and support the network maintain more contextual semantic information. On the other hand, they make the training process stable by passing the gradients from layers to layers and alleviate degradation phenomenon. In the following, we introduce long skip connections and short skip connections, which are relevant to our work.

Long skip connection The early deep learning-based image inpainting methods [12, 25, 40, 41] adopt the encoder-decoder structure for the generator network without skip connections. Thus, abundant low-level features extracted by the shallow layers are lost when the network goes deeper. U-Net [27] is a popular network in image semantic segmentation field. It adds long skip connections between the corresponding encoder layers and decoder layers. The features from encoder layers are concatenated to the features from decoder layers in channel dimension. The image inpainting methods [17, 26, 39] which are recently proposed prefer to adopt U-Net structure to design their generative networks, and they are able to synthesize contents with more contextual details. However, the concatenation of two feature maps from encoder part and decoder part results in redundancy of information and parameters.

Short skip connection is proposed in ResNet [10]. It acts as an identity mapping in residual blocks. Short skip connections is capable of stabilizing gradient updates and speeding up the convergence of deep architectures. It also enables feature re-usability as the long skip connection. The inpainting methods [17, 26] adopt residual blocks with short skip connections in the generative network to strengthen their models.

3 Approach

We use a rough-to-fine architecture [41] to stabilize the training process and increase the receptive field of the model. As shown in Fig. 1, we first obtain a coarsely inpainted image through the use of the rough generator network G_a. Then the texture details are enriched through the fine generator network G_b. We propose a multi-level feature integration (MFI) network and use it in both networks, G_a and G_b. Finally, the inpainted results and ground truth are used as input to a patch discriminator network for classifying them as real or fake, which enforces the generators to synthesize more high-frequency information. Before introducing the architecture in detail, we define our notations in Table 1.

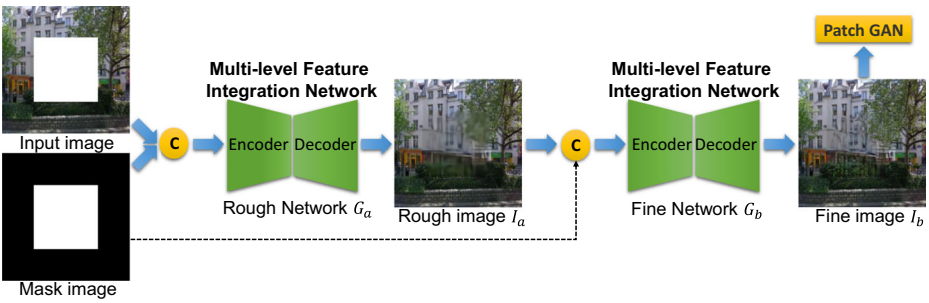

Fig. 1 Architecture of the proposed method. The corrupted image with the mask image are fed into a rough generator G_a, which uses the proposed MFI network. The generated rough image I_a along with the mask image is sent to a fine generator G_b, and produce the inpainted image I_b. I_b and the ground truth image are then sent to patch discriminator to be classified as real or fake

3.1 Rough–to-fine architecture

As shown in Fig. 1, the generating model in our architecture includes two MFI generators G_a and G_b. The network G_a takes the concatenation of the input image I_{in} with holes and the mask image I_m as input, and generates a rough prediction I_a,

$$I_a = G_a(f_{cat}(I_{in}, I_m)). \tag{1}$$

As the network G_a has a limited perspective field, the predicted image I_a is blurry and contains insufficient high-frequency details. Thus, the inpainted image with the mask image is subsequently used as input for the fine generator network G_b. Based on the initial prediction

Table 1 Notations – Used acronyms and symbols

Notation	Description
MFI	Multi-level Feature Integration
G_a	rough generator network
G_b	fine generator network
I_a	rough image generated by network G_a
I_b	fine image generated by network G_b
I_g	ground truth
I_{in}	input image with hole regions
I_m	mask image with 1 indicating hole region and 0 indicating valid region
f_{conv}	convolutional operation
f_{down}	convolutional operation with down-sampling
$f_{conv1\times1}$	convolutional operation with 1×1 kernel
f_{cat}	concatenate along channel dimension
f_{ps}	pixel shuffle operation
$\phi^l_{description}$	features from $l-th$ layer in encoder part
$\varphi^L_{description}$	features from $L-th$ layer in decoder part
$L_{re}, L_{MS\_SSIM}, L_d, L$	loss function

image I_a, the network G_b is capable of capturing more global information and synthesizing refined details for image I_b,

$$I_b = G_b(f_{cat}(I_a \cdot I_m + I_{in} \cdot (1 - I_m), I_m)). \tag{2}$$

Furthermore, the completed image I_b along with the ground truth image I_g is passed through the discriminator network, which enforces the generating model to produce more discriminative information.

3.2 The multi-level feature integration network

We propose the MFI network as the structure of the generators G_a and generator G_b. The structure of the MFI network is shown in Fig. 2. Specifically, adaptive long skip connections and multi-level short connections are applied to the encoder-decoder generator network. In the encoder part, we use five down-sampling modules to capture contextual features at different scales, and we map the input to high-dimensional latent feature space. Accordingly, five up-sampling modules are cascaded to decode the high-level features back to the pixel domain, and the hole region is gradually completed.

There are two main differences between our MFI network with the U-Net [27] and the existing encoder-decoder generative networks [12, 17, 25, 28, 41, 42, 45]: First, for the long skip connection proposed in U-Net architecture, instead of simply adding or concatenating features from encoder layers and decoder layers, we include a feature integration block that incorporates encoder and decoder features with learnable weights along the channel dimension. The features capturing more information have higher assigned weights, while the less important features receive less attention. Second, for the components of the encoder and decoder, we propose a multi-level module, which consists of two branches for extracting high-level and low-level features, respectively. By contrast, most existing encoder-decoder image inpainting networks employ one-branch convolutional modules. The multi-level module makes our MFI network capable of capturing richer features, which improves the visual quality of generated images.

3.2.1 Adaptive long skip connection

As shown in Fig. 2, we add adaptive long skip connections between encoder and decoder layers. Compared to the decoder part, there is more low-level information, like edges, colors and contours, in the feature maps extracted by the encoder layers. This low-level

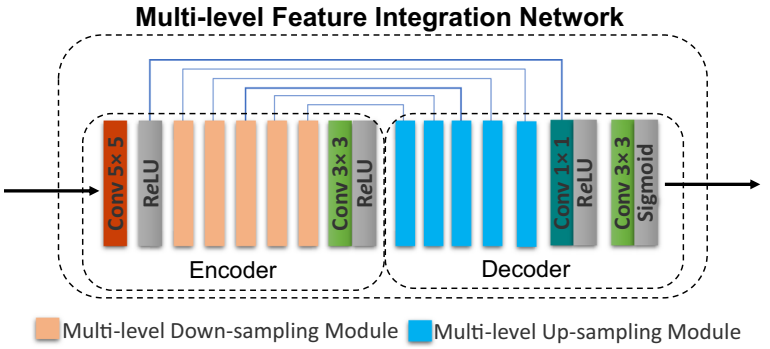

Fig. 2 Structure of Multi-level Feature Integration Network

information can help synthesize boundaries and detailed textures. Without these long skip connections, the low-level information will be lost gradually when the network progresses more deeply. Features extracted by the decoder lie in high-level and high-dimensional feature space. Thus, we design the adaptive long skip connection to concatenate corresponding feature maps from the encoder and decoder layers. In contrast to the vanilla U-Net [27], we use a 1×1 convolution layer to adaptively integrate features from two different layers. Each channel of the integrated feature is the sum of features from the encoder layer and features the from decoder layer along the channel dimension with learnable weights that adapt to the goals. The details and formulations are described in Section 3.2.3. The long skip connection integrates low-level features into the decoder part to enrich multi-level semantic information. With the adaptive long skip connections, the network converges faster and generates more plausible detail information.

3.2.2 The multi-level down-sampling module

For the encoder, we have designed a multi-level residual module for extracting high-level features with down-sampling. The structure is shown in Fig. 3. The input features from the immediately previous layer pass through two branches. The lower branch obtains the high-level features by cascading a down-sampling convolution block and a no-stride convolution block. The upper branch is used for short skip connection. Different from the identity map used in the original ResNet [10], we use a down-sampling convolution block for a short skip connection to make the integrated features from two branches have the same resolution. As the upper branches have only one convolution block, we consider its output as low-level features with respect to the high-level features obtained by two convolution blocks in the lower branch. Formally, we use ϕ_{in}^{l} to denote the input feature map from the l_{th} layer; f_{down} refers to the down-sampling convolution operation, which is implemented by setting the stride parameter in the convolution layer as 2. f_{conv} refers to the convolution layer with a stride of one. The low-level feature $\phi_{low-level}^{l+1}$ in the upper branch is calculated as

$$\phi_{low-level}^{l+1} = f_{down}(\phi_{in}^{l}). \tag{3}$$

The high-level feature $\phi_{high-level}^{l+1}$ in the lower branch is obtained by first down-sampling f_{down} and then performing a convolution f_{conv}, i.e.,

$$\phi_{high-level}^{l+1} = f_{conv}(f_{down}(\phi_{in}^{l})). \tag{4}$$

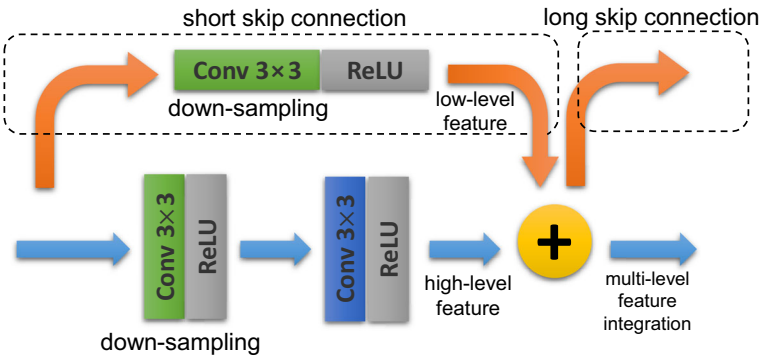

Fig. 3 Multi-level down-sampling module in encoder part. The input feature map is sent to two branches: upper branch as short skip connection extracts low-level features and lower branch extracts high-level features

Finally, the output feature of this layer $\phi^{l}_{high-level}$ is obtained by adding the low-level and high-level features. Further, information is transmitted to the decoder through the long skip connection, i.e.,

$$\phi^{l+1}_{en} = \phi^{l+1}_{low-level} + \phi^{l+1}_{high-level}. \tag{5}$$

In our multi-level down-sampling module, both low-level and high-level features are fused to extract more semantic information.

3.2.3 The multi-level up-sampling module

The structure of the multi-level up-sampling module is shown in Fig. 4. In this module, the input feature map φ^{L}_{in} from the immediately previous layer, the L_{th} layer, is used as input to a two-branch residual block, which is similar to the multi-level down-sampling module. The difference is that there is no down-sampling operation in this residual block. The convolution layers in this block apply 3×3 kernels with a stride of one. The output φ^{L+1}_{de} is calculated as

$$\begin{cases} \varphi^{L+1}_{low-level} = f_{conv}(\varphi^{L}_{in}) \\ \varphi^{L+1}_{high-level} = f_{conv}(f_{conv}(\varphi^{L}_{in})) \\ \varphi^{L+1}_{de} = \varphi^{L+1}_{low-level} + \varphi^{L+1}_{high-level}. \end{cases} \tag{6}$$

We integrate the feature ϕ^{l+1}_{en} from the encoder layer with the feature φ^{L+1}_{de} from the decoder layer via an adaptive long skip connection, i.e.,

$$\varphi^{L+1}_{fusion} = f_{conv1\times1}(f_{cat}(\phi^{l+1}_{en}, \varphi^{L+1}_{de})), \tag{7}$$

where $f_{conv1\times1}$ refers to the convolution layer with kernel size 1×1. This convolution layer works on the channel dimension integration. Assume that the size of ϕ^{l+1}_{en} and φ^{L+1}_{de} is $N \times C \times H \times W$, where N is the batch size, C represents the channel number and H and W indicate spatial size (height and width). After concatenating feature ϕ^{l+1}_{en} and ϕ^{l+1}_{en} in the channel dimension, the feature size becomes $N \times 2C \times H \times W$. The 1×1 convolution layer

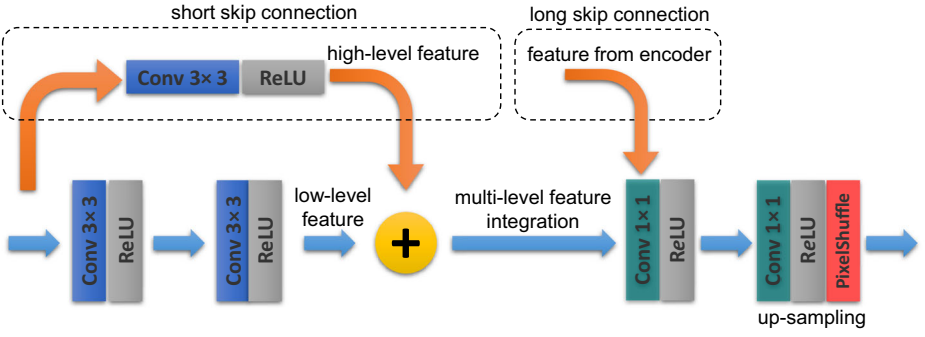

Fig. 4 Multi-level up-sampling module in decoder part. The former part is a two-branch residual block with short skip connection. The latter part uses a 1×1 convolution layer to learn how to optimally integrate the multi-level features from previous layer and the features from encoder. Then, features are up-sampled via a pixel shuffle layer

adaptively integrates the $2C$ channels of features into C channels of features, i.e., φ_{fusion}^{L+1}. The 1×1 convolution layer has C kernels, and each kernel is of size $2C \times 1 \times 1$. The i-th channel of φ_{fusion}^{L+1} is the adaptive sum of all $2C$ channels of features

$$\varphi_{fusion\_C^i} = \sum_{m=1}^{m=C} k_m \times \phi_{en\_C^m} + \sum_{n=1}^{n=C} k_n \times \varphi_{de\_C^n}, \tag{8}$$

where $\varphi_{fusion\_C^i}$ is the i-th channel feature of φ_{fusion}^{L+1}, $\phi_{en\_C^m}$ is the m-th channel feature of ϕ_{en}^{l+1} and $\varphi_{de\_C^n}$ is the n-th channel feature of φ_{fusion}^{L+1}. k_m with $m = 1, 2..., C$ and k_n with $n = C+1, C+2, ...2C$ are the parameters in one kernel of the convolution layer $f_{conv1\times1}$. On the one hand, the integration process can learn how to optimally fuse these features channel-wisely by learnable weights. On the other hand, it can remove redundant information adaptively by lowering the number of channels.

Finally, the fused feature φ_{fusion}^{L+1} is up-sampled with the same ratio as done in the corresponding encoder layer. Instead of using a common deconvolution operation or traditional interpolation operation like bi-cubic interpolation, we use the efficient sub-pixel convolution layer used in the PixelShuffle method [29] for image super-resolution to up-sample the feature φ_{fusion}^{L+1}, i.e.,

$$\varphi_{up}^{L+1} = f_{ps}(f_{conv1\times1}(\varphi_{fusion}^{L+1})), \tag{9}$$

where f_{ps} is a periodic shuffle operation that re-arranges the feature elements. In pixel shuffling operation, the spatial size increases by decreasing the channel numbers. Therefore, $f_{conv1\times1}$ is employed to make channel number consistent. With the use of learnable up-sampling filters, the sub-pixel convolution is adaptive to different feature maps, and it is therefore more effective for synthesizing semantic and well-ordered information. By contrast, the common interpolation algorithms have fixed weights for a local patch and often blur high-frequency information. Moreover, we have conducted an ablation study and present it in Section 4.5 to compare images generated with and without sub-pixel convolution layers. As the results in Fig. 10 show, with the sub-pixel convolution layer our model can generate highly detailed and fine structures, with desirable visible edges in regions of abrupt and high contrast. The generated textures are also of high quality.

3.3 Patch GAN

The traditional discriminator returns a singular value between zero and one for each input image, which cannot fully reflect the local characteristics of images. Therefore, we use Patch GAN [14] to extract local high-frequency features. The discriminator in Patch GAN generates an $N \times N$ matrix X for each input image. Each element X_{ij} represents a local patch in the input image, and the discriminator judges each patch as real or fake. The implementation of our discriminator is shown in Fig. 5. We use five convolutional layers $C1 - C5$ for feature extraction. The first four layers $C1 - C4$ use $LeakyRelu$ as activation function, and layers $C2 - C4$ use the BatchNorm [13] layer to stabilize the training process. The discriminator maps the image to feature space by multiple down-sampling steps, with a stride value of two in the convolutional layers. Finally, we calculate the discriminative loss in feature space directly, which is efficient and makes the training process more stable. Further, it enables the refinement network to synthesize more meaningful high-frequency detail. The adversarial loss is discussed in detail in Section 3.4.

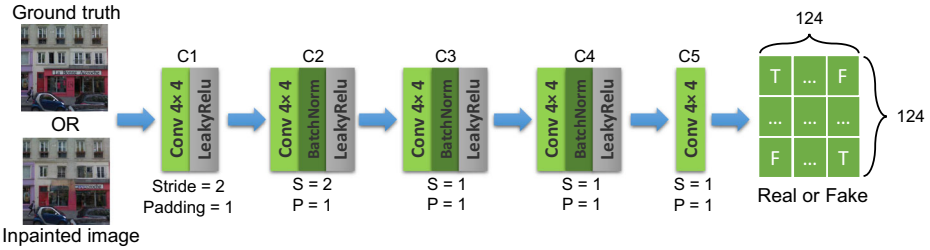

Fig. 5 Structure of the patch discriminator network

3.4 Loss function

***L_1* loss** We understand L_1 loss as reconstruction loss L_{re} to ensure pixel-wise consistency of a rough image I_a and a fine image I_b, with a ground truth image I_g, i.e.,

$$L_{re} = E[||I_a - I_g||_1 + ||I_b - I_g||_1]. \tag{10}$$

Multi-scale SSIM Loss In order to increase structural similarity between inpainted images and the ground truth, we incorporate multi-scale structural similarity (MS_SSIM) loss [35]. The SSIM metric measures similarities for brightness, contrast and structure between two images. MS_SSIM loss L_{ms_ssim} is calculated as

$$L_{ms\_ssim}(I_b, I_g) = 1 - \frac{1}{N}\sum_{n=1}^{N} SSIM(D_n(I_b, I_g)), \tag{11}$$

where $D_n(\cdot)$ is the average pooling operation for down-sampling the inpainted image I_b and the ground truth I_g; 2^{n-1} is the down-sampling ratio. We calculate the MS_SSIM loss for five scales. Based on the MS_SSIM loss, both local and global structure similarity constraints are used to guide the network to produce visually more acceptable images.

Adversarial Loss We use a least squares GAN [23] to improve the visual quality of generated images, where the objective function L_d of minimizing least squares errors is defined as

$$L_d = \begin{cases} \min\limits_D \frac{1}{2} E_{I_g \sim P_r}[D(I_g) - 1]^2 + \frac{1}{2} E_{I_{in} \sim P_f}[D(G(I_{in}))]^2 \\ \min\limits_D \frac{1}{2} E_{I_{in} \sim P_f}[D(G(I_{in})) - 1]^2 \end{cases}, \tag{12}$$

where $D(\cdot)$ denotes the discriminator network and $G(\cdot)$ represents the generator network. P_f defines the distribution of an input image I_{in} with holes, and P_r is the distribution of the real image I_g. The constant 1 labels the real image, and 0 labels are used for the generated image. The least squares discriminative loss improves training stability and performance of the generator.

Overall Loss Our loss function L is defined as

$$L = \lambda_r L_{re} + \lambda_{ms\_ssim} L_{ms\_ssim} + \lambda_d L_d, \tag{13}$$

where λ_r, λ_{ms_ssim} andλ_d are adaptive weights for reconstruction loss, structural loss and adversarial loss, respectively.

4 Experiments

We conduct extensive experiments to verify the effectiveness of our proposed method, including qualitative comparisons in Section 4.2, quantitative comparisons in Section 4.3, user study in Section 4.4 and ablation study in Section 4.5.

4.1 Basic settings

4.1.1 Datasets

We evaluate our method and the comparison methods on three commonly used public datasets: Paris StreetView dataset [44], CelebA-HQ dataset [21], and Places2 dataset [49].

- Paris StreetView dataset comes from Google StreetView. The dataset focuses on buildings in Paris and contains structural information such as stacked buildings, doors and windows. We divide the dataset according to the original split strategy: 14,900 images for training and 100 images for testing.
- CelebA-HQ dataset contains 30,000 high-resolution face images. We sequentially divide the dataset into 2000 testing images and 28,000 training images according to the split strategy from paper [19].
- Places2 dataset [49] is a scene images dataset containing 10 million images for 365 scene categories. Each scene has 5,000 images for training and 100 images for testing. We use images from four scenes, including tree farm, valley, mountain path and mountain, for training and testing.

4.1.2 Training settings

All models are tested and trained on Ubuntu 18.04 operating system, which is powered by a server with two Intel(R) Xeon(R) Silver 4108 @ 1.80GHz CPU and four NVIDIA GeForce RTX 2080Ti 11GB GPUs. The version of Python is 3.6.7. The batch size of our experiment is 22. For both training images and testing images, the resolution is 256×256. We use resize and random cropping transform for training dataset. Concerning hyper-parameters, we set loss function weights as $\lambda_r = 4$, $\lambda_d = 2$, $\lambda_{ms\_ssim} = 1$; we set the learning rate as $lr = 0.0002$, based on experiments. It takes 82,000 iterations, 150,000 iterations and 100,000 iterations, respectively, to train our model for the Paris StreetView, CelebA-HQ and Places2 datasets. For the comparison methods, we use the same parameter settings as in their papers. Note that we present the detailed architecture and all parameters settings the proposed MFI network in the Appendix Section.

4.1.3 Comparison methods

We compare our method with the following four methods.

- PM: Patch Match, a traditional exemplar-based method proposed by Barnes et al. [2]. The continuity of the image and the surrounding patch are used to vote for image completion.
- GL: Global and Local, proposed by Lizuka et al. [12]. Globally and locally consistent discriminators are proposed to increase the local consistency and global consistency of image completion.
- CA: Contextual Attention, proposed by Yu et al. [41]. The features of the non-missing patches are used as convolution kernels to generate the missing patches to refine the fuzzy inpainting results.

– ML: Multi-Level generative network, proposed by Liu et al. [19]. A three branches generative network is built to capture features of various levels while reducing training time.

In terms of model size, there are 6.1 million, 3.6 million, 15.1 million and 15.0 million learnable parameters for the GL, CA and ML methods and our method, respectively. The average times for inpainting one image of resolution 256 × 256 are 0.182 second, 0.041 second, 0.037 second and 0.012 second for the GL, CA and ML methods and our method. Method GL adopts Poisson image blending as a post-processing step, which improves the consistency but also increases the inpainting time. Method CA has a non-local attention layer, which is beneficial to generate sharp and clear textures, but very time-consuming. By contrast, our method is high-efficiency regarding the inpainting time.

4.2 Qualitative comparison

We conduct experiments with two different masking strategies, including rectangle mask and irregular mask.

Firstly, we train our model and the comparison models with rectangle masks on Paris StreetView dataset. Visual testing results are shown in Fig. 6. The size for missing region is 128 × 128, accounts for 25% pixels. From Fig. 6b and c, GL method [12] and ML method [19] can generate semantic content, but the inpainted regions are very blurry and lack details. These two methods are based on generative network, but they use only one generator network and have no skip connections, which makes the network less powerful. As presented in Fig. 6d, PM method [2] can produce sharp restoration in some simple and regular pattern scenes by using similar neighbor patches. However, in complex scenes, like trees in the first and fourth images, windows in the second and third images, the results of PM method suffer from disordered textures and unreasonable contents. Observed from Fig. 6d, CA method [41] can synthesize plausible result at the semantic level, but it also generate disordered textures and colors. For example, the repetition tree texture in the first image, the green color propagated to the building wall in the fourth image and the inconsistent boundary in the fifth image. Compared with these methods, our model achieves better visual effect. As shown in Fig. 6e, our model generates sharp and semantically reasonable textures and consistent boundaries. This benefits from the adaptive long and short skip connections in the proposed multi-level feature integration network, and the rough-to-fine architecture also helps extract more semantic features.

Secondly, we use the irregular mask dataset provided in the paper [16] to train and test our method on CelebA-HQ dataset and Places2 dataset. The mask dataset contains more than 50 thousand mask images for training, and has six different missing pixel ratio ranges for 12 thousand test mask images. The training masks and testing masks are randomly assigned to the training and testing images. From the inpainted results shown in Fig. 7, we can see that our method can generate plausible content with irregular masks on different datasets.

4.3 Quantitative comparisons

We evaluate quantitatively our method and the comparison methods on Paris StreetView dataset with center rectangle mask, in terms of five metrics, i.e., L_1 error, L_2 error, PSNR (peak signal-to-noise ratio), SSIM (structure similarity index) and FID (Fréchet inception distance) [11]. The first four metrics are calculated in RGB space, and indicate the pixel domain accuracy between the inpainted image and the ground truth. The metric FID calculates the perceptual distance in high level feature space. We feed the inpainted image

Fig. 6 Qualitative comparisons on Paris StreetView dataset with center rectangle mask. Images from top to bottom are: (**a**) Input (**b**) GL (**c**) ML (**d**) PM (**e**) CA (**f**) ours (**g**) Ground Truth

and the ground truth to the pre-trained Inception-V3 model to extract features, and then conduct FID calculation. For image inpainting task with large missing areas, for example, object removal, the task is to generate reasonable content with high visual perception quality, not pixel-wise restoration with the ground truth. Thus, as the L_1 error, L_2 error,

Fig. 7 Visual results with irregular masks. The first two rows are the test results for CelebA-HQ dataset and the last two rows are the test results for the Places2 dataset

PSNR and SSIM metrics can reflect texture distortions to some extent, the FID metric is more consistent with human visual perception.

Table 2 lists the evaluation results, and shows that our method performs best in terms of FID metric, indicating that our method can generate superior visual effects compared to other methods. The quantitative evaluation results are also consistent with the qualitative

Table 2 Numerical comparison for Paris StreetView dataset

Method	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$	FID^-
PM [2]	5.37	2.74	0.84	23.21	55.60
GL [12]	4.36	1.71	0.86	25.26	79.82
CA [41]	5.13	2.53	0.85	23.83	59.96
ML [19]	4.37	1.72	0.87	25.18	82.72
Ours_MFI	**4.33**	**1.67**	**0.87**	**25.38**	64.73
Ours	4.68	2.00	0.86	24.93	**49.94**

Best results for each metric are shown in bold numbers. The notation $^-$ stands for "lower value is the better method" and $^+$ stands for "higher value is the better method". The method called Ours_MFI network has one MFI network, while the method called Ours refers to the whole architecture with two MFI networks

Table 3 The average subjective evaluations for five inpainting methods from 60 volunteers

Method	PM	GL	CA	ML	Ours
average score	6.92	5.11	6.48	5.00	**7.15**

results in Section 4.2. In terms of L_1 error, L_2 error, PSNR and SSIM metrics, our method with one MFI (Multi-level Feature Integration) network achieves the highest values, as presented in the second row from the bottom in Table 2. The evaluation results demonstrate that our proposed MFI network with adaptive long and short skip connections is effective in capturing multi-level semantic information and achieves high reconstruct accuracy in pixel domain. But for better visual quality, we suggest the two-stage architecture. By incorporating MFI network with rough-to-fine architecture, our proposed method can generate fine-detailed and semantically reasonable textures.

4.4 User study

In order to better evaluate the perceptual visual qualities of image inpainting, we conduct the user study experiment with 60 volunteers. We use the 100 test images from the Paris StreetView dataset and ask the volunteers to grade for 20 random sets of images based on their subjective visual perceptions. Each set has five inpainted images randomly from methods PM [2], GL [12], ML [19], CA [41] and our method. The grade ranges from 1 to 10, and grade 10 is for ground truth with the best visual qualities. The average subjective scores of five methods are shown in Table 3. Our method gets highest subjective evaluation 7.15. Method PM gets the second place as they use the known patches to fill the hole which is much sharper than methods GL and ML. The smoothed curve for the average user score for each test image is shown in Fig. 8. For images with natural landscapes or buildings, our model scores far more than other methods. For the completion of some advertising signs, our method still has some room for improvement.

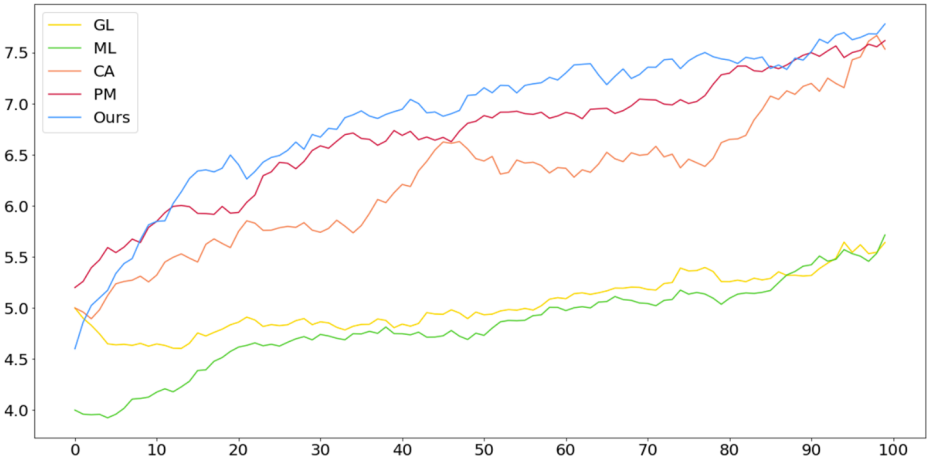

Fig. 8 The average subjective score curve after smoothing for 100 test images from Paris StreetView dataset

(a) M1 (b) Ours (c) M1 (d) Ours

Fig. 9 Visual comparisons between M1 (without the refinement network) and our integrated method

(a) M2 (b) Ours (c) M2 (d) Ours

Fig. 10 Visual comparisons between M2 (using nearest interpolation not sub-pixel upsampling layer) and our integrated method

(a) M3 (b) Ours (c) M3 (d) Ours

Fig. 11 Visual comparisons between M3 (without short skip connections) and our integrated method

Table 4 Numerical comparison for ablation study performed for Paris Streetview dataset

Method	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$	FID^-
M1(without refinement network)	**4.33**	**1.67**	**0.87**	**25.38**	64.73
M2(without sub-pixel)	4.78	2.18	0.85	24.21	50.47
M3(without short skip connection)	4.71	2.13	0.85	24.08	55.37
ours	4.68	2.00	0.86	24.93	**49.94**

Best results for each group are shown in bold numbers

4.5 Ablation study

We conduct three groups of ablation experiments to further study the effects of different parts in our method. We denote the method without the refinement network as M1, the method using nearest interpolation instead of sub-pixel upsampling layer as M2 and the method without short skip connections as M3. The visual results are shown in Figs. 9, 10 and 11, and the quantitative evaluations are presented in Table 4.

As shown in the Fig. 9a and c, method M1 with one single proposed MFI (Multi-level Feature Integration) network generates blurry completions. Our rough-to-fine architecture integrating two MFI generative networks inpaints the missing regions with fine detailed textures and sharp structures in Fig. 9b and d. Although M1 achieves better evaluations in terms of metrics L_1 error, PSNR and SSIM, it has much lower FID values than our integrated method. Figure 10 presents the comparisons between M2 and our integrated method. Observed from these visual results, we can get the conclusion that sub-pixel layers contribute to keep the semantically reasonable and ordered textures and structures. As shown in Fig. 11, results generated by M3 method show obvious fuzzy and disordered textures, such as speckled stripes on the wall in Fig. 11a and chaotic textures for the door in Fig. 11c.

5 Conclusions

We have introduced an effective Multi-level Feature Integration (MFI) network for image inpainting. As components of the encoder and decoder in the MFI network, we have designed a multi-level down-sampling module and upsampling module. The two-branch structure is used to capture and integrate multi-level semantic features. Features from shallower layers are adaptively fused with features from deeper layers with learnable weights. Concerning the overall architecture, two MFI networks are cascaded to fill in hole regions in a progressive way. A joint-trained patch-discriminator guides the MFI network to synthesize higher-frequency information. Our presented experiments demonstrate that our method performs better than other methods regarding the completion of regions with "highly ordered textures" and "sharp, high-contrast structures".

The focus of our presented method is the design of a generative network, i.e., a network that is effective in capturing rich multi-level features. However, when large irregular hole regions must be filled in, performance still should be improved. First, the convolution operation has limited ability of modelling long-range dependencies. Second, valid and hole regions are not differently treated, which may introduce "blurry artifacts"; this aspect could be addressed in future research. Third, the receptive field is limited for each MFI network. The parallel dilated convolutional module can be combined with our architecture. It is our plan to incorporate advanced self-attention, a mask-aware strategy and a receptive field-aware framework with our MFI network to expand the applicability to a wider spectrum of image classes.

Appendix

The detailed architecture and used parameter values of our approach are provided in Table 5- Table 8. We use the following abbreviations in the table: Size_in stands for the spatial

size of the input in one dimension; Size_out stands for the spatial size of the output in one dimension; C_in refers to the channel number of the input; C_out is the channel number of the output; Act refers to the non-linear activation function; Norm refers to the normalization method; Conv stands for convolution layer; K stands for kernel size; S stands for stride; and P stands for padding.

Table 5 Architecture of the MFI (multi-level feature integration) network

Architecture of MFI network							
Layer	Input	Size_in	Size_out	C_in	C_out	Act	Operator
En_0	$cat(Image, Mask)$	256	256	4	32	Relu	Conv
En_1	F_En_0	256	128	32	64	Relu	MD
En_2	F_En_1	128	64	64	128	Relu	MD
En_3	F_En_2	64	32	128	256	Relu	MD
En_4	F_En_3	32	16	256	256	Relu	MD
En_5	F_En_4	16	8	512	256	Relu	MD
En_6	F_En_5	8	4	256	256	Relu	Conv
De_1	$cat(F\_En_6, F\_En_5)$	4	8	256	256	Relu	MU
De_2	$cat(F\_De_1, F\_En_4)$	8	16	256	256	Relu	MU
De_3	$cat(F\_De_2, F\_En_3)$	16	32	256	256	Relu	MU
De_4	$cat(F\_De_3, F\_En_2)$	32	64	256	128	Relu	MU
De_5	$cat(F\_De_4, F\_En_1)$	64	128	128	64	Relu	MU
De_6	$cat(De_5, En_0)$	64	128	64	32	Relu	Conv
De_7	F_De_6	128	256	32	3	Sigmoid	Conv

Conv represents convolution operation, MD represents multi-level down-sampling, and MU represents multi-level up-sampling. F_layer indicates the output feature of the layer

Table 6 Architecture of encoder module En_i

Architecture of encoder module En_i									
Layer	Operator	Input	Output	Act	K	S	P	C_in	C_out
Upper	Conv	F_En_{i-1}	F_Upper	ReLu	3	2	1	$min(2^{i+3}, 2^8)$	$min(2^{i+4}, 2^8)$
Lower	Conv	F_En_{i-1}	F_Lower0	ReLu	3	1	1	$min(2^{i+3}, 2^8)$	$min(2^{i+4}, 2^8)$
	Conv	F_Lower0	F_Lower	ReLu	3	1	1	$min(2^{i+4}, 2^8)$	$min(2^{i+4}, 2^8)$

The input is the output feature of the previous module En_{i-1}. The output is F_Upper + F_Lower. F_layer indicates the output feature of the layer

Table 7 Architecture of decoder module De_i

Architecture of decoder module De_i									
Layer	Operator	Input	Output	Act	K	S	P	C_in	C_out
Upper	Conv	F_De_{i-1}	F_Upper	ReLu	3	1	1	$min(2^{11-i}, 2^8)$	$min(2^{10-i}, 2^8)$
Lower	Conv	F_De_{i-1}	F_Lower1	ReLu	3	1	1	$min(2^{11-i}, 2^8)$	$min(2^{10-i}, 2^8)$
	Conv	F_Lower1	F_Lower	ReLu	3	1	1	$min(2^{10-i}, 2^8)$	$min(2^{10-i}, 2^8)$
Integration	Conv	Cat(F_Upper, F_Lower)	F_Integration	ReLu	1	1	0	$min(2^{11-i}, 2^9)$	$min(2^{10-i}, 2^8)$
Up-sampling	Conv	F_Integration	F_Conv	ReLu	1	1	0	$min(2^{10-i}, 2^8)$	$min(2^{12-i}, 2^{10})$
	PixelShuffle	F_Conv	F_De_i	None	None	None	None	$min(2^{12-i}, 2^{10})$	$min(2^{10-i}, 2^8)$

The Input is the output feature F_De_{i-1} of previous decoder module. The output is the feature F_De_i from up-sampling layer. F_layer indicates the output feature of the layer

Table 8 Architecture of Patch GAN

Architecture of PatchGan										
Layer	Operator	Input	Output	Act	Norm	K	S	P	In_c	Out_c
C1	Conv	Image	F_C1	LeakyReLu	None	4	2	1	3	64
C2	Conv	F_C1	F_C2	LeakyReLu	BatchNorm	4	1	1	64	128
C3	Conv	F_C2	F_C3	LeakyReLu	BatchNorm	4	1	1	128	256
C4	Conv	F_C3	F_C4	LeakyReLu	BatchNorm	4	1	1	256	512
C5	Conv	F_C4	F_C5	None	None	4	1	1	512	1

The input is the real image or fake image. Output is an 124×124 matrix X for each input image, which judges each patch as real or fake. F_layer indicates the output feature of the layer

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Ballester C, Bertalmio M, Caselles V, Sapiro G, Verdera J (2001) Filling-in by joint interpolation of vector fields and gray levels. IEEE Trans Image Process 10(8):1200–1211
2. Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (Proc. SIGGRAPH) 28(3)
3. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
4. Criminisi A, Pérez P, Toyama K (2004) Region filling and object removal by exemplar-based image inpainting. IEEE Trans Image Process 13(9):1200–1212
5. Ding D, Ram S, Rodríguez JJ (2018) Image inpainting using nonlocal texture matching and nonlinear filtering. IEEE Trans Image Process 28(4):1705–1719
6. Drozdzal M, Vorontsov E, Chartrand G, Kadoury S, Pal C (2016) The importance of skip connections in biomedical image segmentation. In: Deep learning and data labeling for medical applications. Springer, pp 179–187
7. Fan Q, Zhang L (2018) A novel patch matching algorithm for exemplar-based image inpainting. Multimed Tools Appl 77(9):10807–10821
8. Guillemot C, Le Meur O (2014) Image inpainting: Overview and recent advances. IEEE Signal Process Mag 31(1):127–144
9. Guo Q, Gao S, Zhang X, Yin Y, Zhang C (2017) Patch-based image inpainting via two-stage low rank approximation. IEEE Trans Vis Comput Graph 24(6):2023–2036
10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
11. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp 6626–6637
12. Iizuka S, Simo-Serra E, Ishikawa H (2017) Globally and Locally Consistent Image Completion. ACM Trans Graph (Proc. of SIGGRAPH 2017) 36(4):107:1–107:14
13. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448–456
14. Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1125–1134
15. Li F, Zeng T (2014) A universal variational framework for sparsity-based image inpainting. IEEE Trans Image Process 23(10):4242–4254

16. Liu G, Reda FA, Shih KJ, Wang T-C, Tao A, Catanzaro B (2018) Image inpainting for irregular holes using partial convolutions. In: The European Conference on Computer Vision (ECCV)
17. Liu H, Jiang B, Xiao Y, Yang C (2019) Coherent semantic attention for image inpainting. In: IEEE International Conference on Computer Vision (ICCV)
18. Liu J, Yang S, Fang Y, Guo Z (2018) Structure-guided image inpainting using homography transformation. IEEE Trans Multimed PP(99):1–1
19. Liu J, Jung C (2019) Facial image inpainting using multi-level generative network. In: 2019 IEEE International Conference on Multimedia and Expo (ICME). IEEE, pp 1168–1173
20. Liu X, Chen S, Song L, Woźniak M, Liu S (2021) Self-attention negative feedback network for real-time image super-resolution. Journal of King Saud University-Computer and Information Sciences
21. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV)
22. Lu H, Liu Q, Zhang M, Wang Y, Deng X (2018) Gradient-based low rank method and its application in image inpainting. Multimed Tools Appl 77(5):5969–5993
23. Mao X, Li Q, Xie H, Lau RaymondYK, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2794–2802
24. Nazeri K, Ng E, Joseph T, Qureshi FZ, Ebrahimi M (2019) Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv:1901.00212
25. Pathak D, Krähenbühl P, Donahue J, Darrell T, Efros A (2016) Context encoders: Feature learning by inpainting. In: Computer Vision and Pattern Recognition (CVPR)
26. Ren Y, Yu X, Zhang R, Li TH, Liu S, Li G (2019) Structureflow: Image inpainting via structure-aware appearance flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 181–190
27. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention
28. Shen L, Hong R, Zhang H, Zhang H, Wang M (2019) Single-shot semantic image inpainting with densely connected generative networks. In: Proceedings of the 27th ACM International Conference on Multimedia, pp 1861–1869
29. Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1874–1883
30. Song Y, Yang C, Shen Y, Wang P, Huang Q, Kuo C-CJ (2018) Spg-net: Segmentation prediction and guidance network for image inpainting. In: British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018. BMVA Press, p 97
31. Tschumperlé D (2006) Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. Int J Comput Vis 68(1):65–82
32. Wan Z, Zhang J, Chen D, Liao J (2021) High-fidelity pluralistic image completion with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 4692–4701
33. Wang N, Ma S, Li J, Zhang Y, Zhang L (2020) Multistage attention network for image inpainting. Pattern Recogn 106:107448
34. Wang N, Zhang Y, Zhang L (2021) Dynamic selection network for image inpainting. IEEE Trans Image Process 30:1784–1798
35. Wang Z, Simoncelli EP, Bovik AC (2003) Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, vol 2. IEEE, pp 1398–1402
36. Xiao Z, Li D (2021) Generative image inpainting by hybrid contextual attention network. In: International Conference on Multimedia Modeling. Springer, pp 162–173
37. Xie C, Liu S, Li C, Cheng M-M, Zuo W, Liu X, Wen S, Ding E (2019) Image inpainting with learnable bidirectional attention maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 8858–8867
38. Yaghmaee F, Peyvandi K (2020) Improving image inpainting quality by a new svd-based decomposition. Multimed Tools Appl 79(19):13795–13809
39. Yan Z, Li X, Li M, Zuo W, Shan S (2018) Shift-net: Image inpainting via deep feature rearrangement. In: Proceedings of the European conference on computer vision (ECCV), pp 1–17
40. Yang C, Lu X, Lin Z, Shechtman E, Wang O, Li H (2017) High-resolution image inpainting using multi-scale neural patch synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6721–6729

41. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5505–5514
42. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2019) Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 4471–4480
43. Yu T, Guo Z, Jin X, Wu S, Chen Z, Li W, Zhang Z, Liu S (2020) Region normalization for image inpainting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, pp 12733–12740
44. Zamir AR, Shah M (2014) Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. IEEE Trans Pattern Anal Mach Intell 36(8):1546–1558
45. Zeng Y, Fu J, Chao H, Guo B (2019) Learning pyramid-context encoder network for high-quality image inpainting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1486–1494
46. Zhang J, Zhao D, Gao W (2014) Group-based sparse representation for image restoration. IEEE Trans Image Process 23(8):3336–3351
47. Zhang L, Chang M (2021) An image inpainting method for object removal based on difference degree constraint. Multimed Tools Appl 80:1–20. https://doi.org/10.1007/s11042-020-09835-0
48. Zhang X, Hamann B, Pan X, Zhang C (2017) Superpixel-based image inpainting with simple user guidance. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, pp 3785–3789
49. Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2017) Places: A 10 million image database for scene recognition. IEEE Trans Pattern Anal Mach Intell 40(6):1452–1464
50. 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 Trans Image Process 30:4855–4866

Affiliations

Tao Chen[1] · Xin Zhang[1,2,3] · Bernd Hamann[4] · Dongjing Wang[1] · Hua Zhang[1]

Tao Chen
chenboluo@hdu.edu.cn

Bernd Hamann
hamann@cs.ucdavis.edu

Dongjing Wang
dongjing.wang@hdu.edu.cn

Hua Zhang
zhangh@hdu.edu.cn

1 School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, 310018, China

2 Hangzhou Dianzi University Shangyu Institute of Science and Engineering, Shaoxing, 312000, China

3 School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China

4 Department of Computer Science, University of California, Davis, CA 95616, USA