Image Fine-grained Inpainting

Zheng Hui, Jie Li, Xiumei Wang, and Xinbo Gao*
School of Electronic Engineering, Xidian University
Xi'an, China
zheng_hui@aliyun.com, leejie@mail.xidian.edu.cn, wangxm@xidian.edu.cn, xbgao@mail.xidian.edu.cn

Abstract

Image inpainting techniques have shown promising improvement with the assistance of generative adversarial networks (GANs) recently. However, most of them often suffered from completed results with unreasonable structure or blurriness. To mitigate this problem, in this paper, we present a one-stage model that utilizes dense combinations of dilated convolutions to obtain larger and more effective receptive fields. Benefited from the property of this network, we can more easily recover large regions in an incomplete image. To better train this efficient generator, except for frequently-used VGG feature matching loss, we design a novel self-guided regression loss for concentrating on uncertain areas and enhancing the semantic details. Besides, we devise a geometrical alignment constraint item to compensate for the pixel-based distance between prediction features and ground-truth ones. We also employ a discriminator with local and global branches to ensure local-global contents consistency. To further improve the quality of generated images, discriminator feature matching on the local branch is introduced, which dynamically minimizes the similarity of intermediate features between synthetic and ground-truth patches. Extensive experiments on several public datasets demonstrate that our approach outperforms current state-of-the-art methods. Code is available at https://github.com/Zheng222/DMFN.

1. Introduction

Image inpainting (a.k.a. image completion) aims to synthesize proper contents in missing regions of an image, which can be used in many applications. For instance, it allows removing unwanted objects in image editing tasks, while filling the contents that are visually realistic and semantically correct. Early approaches to image inpainting are mostly based on patches of low-level features. PatchMatch [2], a typical method, iteratively searches optimal patches to fill in the holes. It can produce plausible results when painting image background or repetitive textures. However, it cannot generate pleasing results for cases where completing regions include complex scenes, faces, and objects, which is due to PatchMatch cannot synthesize new image contents, and missing patches cannot be found in remaining regions for challenging cases.

With the rapid development of deep convolutional neural networks (CNN) and generative adversarial networks (GAN) [6], image inpainting approaches have achieved remarkable success. Pathak *et al*. proposed context-encoder [20], which employs a deep generative model to predict missing parts of the scene from their surroundings using reconstruction and adversarial losses. Yang *et al*. [29] introduced style transfer into image inpainting to improve textural quality that propagates the high-frequency textures from the boundary to the hole. Li *et al*. [17] presented semantic parsing in the generation to restrict synthesized semantically valid contents for the missing facial key parts from random noise. To be able to complete large regions, Iizuka *et al*. [9] adopted stacked dilated convolutions in their image completion network to obtain lager spatial support and reached realistic results with the assistance of a globally and locally consistent adversarial training approach. Shortly afterward, Yu *et al*. [30] extended this insight and developed a novel contextual attention layer, which uses the features of known patches as convolutional kernels to compute the correlation between the foreground and background patches. More specifically, they calculated attention score for each pixel and then performed transposed convolution on attention score to reconstruct missing patches with known patches. It might be failing when the relationship between unknown and known patches is not close (*e.g*. masking all of the critical components of a facial image). Wang *et al*. [26] proposed a generative multi-column convolutional neural network (GMCNN) that uses varied receptive fields in branches by adopting different sizes of convolution kernels (*i.e*. 3×3, 5×5, and 7×7) in a parallel manner. This method produces advanced performance but suffers from substantial model parameters (12.562M)

*corresponding author

caused by large convolution kernels. In terms of image quality (more photo-realistic, fewer artifacts), it is still room for improvement.

The goals pursued by image inpainting are ensuring produced images with global semantic structure and finely detailed textures. Additionally, completed image should be approaching the ground truth as much as possible, especially for building and face images. Previous techniques more focus on solving how to yield holistically reasonable and photo-realistic images. This problem has been mitigated by GAN [6] or its improved version WGAN-GP [7] that is frequently utilized in image inpainting methods [20, 9, 29, 17, 30, 24, 28, 26, 33, 31]. However, concerning fine-grained details, there is still much room to enhance. Besides, these existing methods haven't taken into account the consistency between outputs and targets, *i.e*., semantic structures should be as much similar as possible for facial images and building images.

To overcome the limitations of the methods as mentioned above, we present a unified generative network for image inpainting, which is denoted as *dense multi-scale fusion network* (DMFN). The *dense multi-scale fusion block* (DMFB), serving as the basic block of DMFN, is composed of four-way dilated convolutions as illustrated in Figure 2. This basic block adopts the combination and fusion of hierarchical features extracted from various convolutions with different dilation rates to obtain better multi-scale features, compared with general dilated convolution (dense v.s. sparse). For generating images with the realistic and semantic structure, we design a *self-guided regression loss* that constrains low-level features of the generated content according to the normalized discrepancy map (the difference between the output and target). *Geometrical alignment constraint* is developed for penalizing the coordinate center of estimated image high-level features away from the ground-truth. This loss can further help the processing of image fine-grained inpainting. We improve the discriminator using relativistic average GAN (RaGAN) [11]. It is noteworthy that we use global and local branches in the discriminator as in [9], where one branch focuses on the global image while the other concentrates on the local patch of the missing region. To explicitly constraint the output and ground-truth images, we utilize the hidden layers of the local branch that belongs to the whole discriminator to evaluate their discrepancy through an adversarial training process. With all these improvements, the proposed method can produce high-quality results on multiple datasets, including faces, building, and natural scene images.

Our contributions are summarized as follows:

- We propose a novel self-guided regression loss to explicitly correct the low-level features, according to the normalized error map computed by the output and ground-truth images. This function can significantly improve the semantic structure and fidelity of images.
- We present a geometrical alignment constraint to supplement the shortage of pixel-based VGG features matching loss.
- We propose a dense multi-scale fusion generator, which has the merit of strong representation ability to extract useful features. Our generative image inpainting framework achieves compelling visual results (as illustrated in Figure 1) on challenging datasets, compared with previous state-of-the-art approaches.

2. Related Work

A variety of algorithms for image inpainting have been proposed. Traditional diffusion-based methods [3, 1] propagate information from neighboring regions to the holes. They can work well for small and narrow holes, where the texture and color variance are the same. However, these methods fail to recover meaning contents in the large missing regions. Patch-based approaches, such as [5, 15], search for relevant patches from the known regions in an iterative fashion. Simakov *et al*. [22] proposed bidirectional similarity scheme to capture better and summarize non-stationary visual data. However, these methods are computationally expensive due to calculating the similarity scores of each output-target pair. To relieve this problem, PatchMatch [2] is proposed, which speeds it up by designing a faster similar patch searching algorithm.

Recently, deep learning and GAN-based algorithms have been a remarkable paradigm for image inpainting. Context Encoders (CE) [20] embed the 128×128 image with a 64×64 center hole as a low dimensional feature vector and then decode it to a 64×64 image. Iizuka *et al*. [9] proposed a high-performance completion network with both global and local discriminators that is critical in obtaining semantically and locally consistent image inpainting results. Also, the authors employ the dilated convolution layers to increase receptive fields of the output neurons. Yang *et al*. [29] use intermediate features extracted by pre-trained VGG network [23] to find hole's most similar patch outside the hole. This approach performs multi-scale neural patch synthesis in a coarse-to-fine manner, which noticeably takes a long time to fill a large image during the inference stage. For face completion, Li *et al*. [17] trained a deep generative model with a combination of reconstruction loss, global and local adversarial losses, and a semantic parsing loss specialized for face images. Contextual Attention (CA) [30] adopted two-stage network architecture where the first step produces a crude result, and the second refinement network using attention mechanism takes the coarse prediction as inputs and improves fine details. Liu *et al*. [18] introduced partial convolution that employs computational operations only on valid pixels and presented an auto-update binary mask

Ground-truth Image | Input Image | CA [30] | GMCNN [26] | DMFN (Ours)

Figure 1. Visual comparisons on CelebA-HQ. **Best viewed with zoom-in.**

to determinate whether the current pixels are valid. Substituting convolutional layers with partial convolutions can help a UNet-like architecture [10] achieve the state-of-the-art inpainting results. Yan *et al*. [28] introduced a special shift-connection to the U-Net architecture for enhancing the sharp structures and fine-detailed textures in the filled holes. This method was mainly developed on building and natural landscape images. Similar to [29, 30], Song *et al*. [24] decoupled the completion process into two stages: coarse inference and fine textures translation. Nazeri *et al*. [19] also proposed a two-stage network that comprises of an edge generator and an image completion network. Similar to this method, Li *et al*. [16] progressively incorporated edge information into the feature to output more structured image. Xiong *et al*. [27] inferred the contours of the objects in the image, then used the completed contours as a guidance to complete the image. Different from frequently-used two-stage processing [21], Sagong *et al*. [4] proposed parallel path for semantic inpainting to reduce the computational costs.

3. Proposed Method

Our proposed inpainting system is trained in an end-to-end way. Given an input image with hole $\mathbf{I}_{in}$, its corresponding binary mask $\mathbf{M}$ (value 0 for known pixels and 1 denotes unknown ones), the output $\mathbf{I}_{out}$ predicted by the network, and the ground-truth image $\mathbf{I}_{gt}$. We take the input image and mask as inputs, *i.e*., $[\mathbf{I}_{in}, \mathbf{M}]$. We now elaborate on our network as follows.

3.1. Network structure

As depicted in Figure 3, our framework consists of a generator, and a discriminator with two branches. The generator produces plausible painted results, and the discriminator conducts adversarial training.

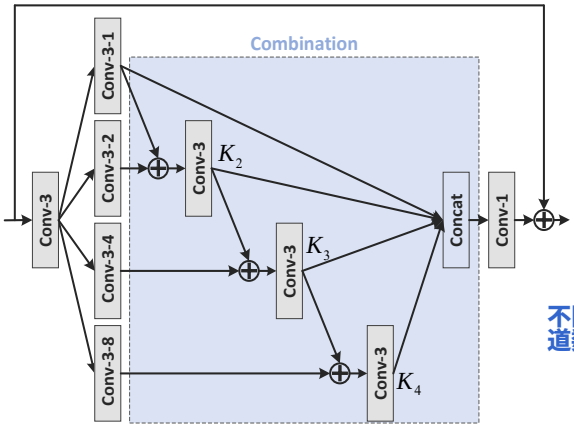


Figure 2. The architecture of the proposed dense multi-scale fusion block (DMFB). Here, "Conv-3-8" indicates 3×3 convolution layer with the dilation rate of 8 and $\oplus$ is element-wise summation. Instance normalization (IN) and ReLU activation layers followed by the first convolution, second column convolutions and concatenation layer are omitted for brevity. The last convolutional layer only connects an IN layer. The number of output channels for each convolution is set to 64 except for the last 1×1 convolution (256 channels) in DMFB.

For image inpainting task, the size of the receptive fields should be sufficiently large. The dilated convolution is popularly adopted in the previous works [9, 30] to accomplish this purpose. This way increases the area that can use as input without increasing the number of learnable weights. However, the kernel of dilated convolution is sparse, which skips many pixels during applying to compute. Large convolution kernel (*e.g*. 7×7) is applied in [26] to implement this intention. However, this solution introduces heavy

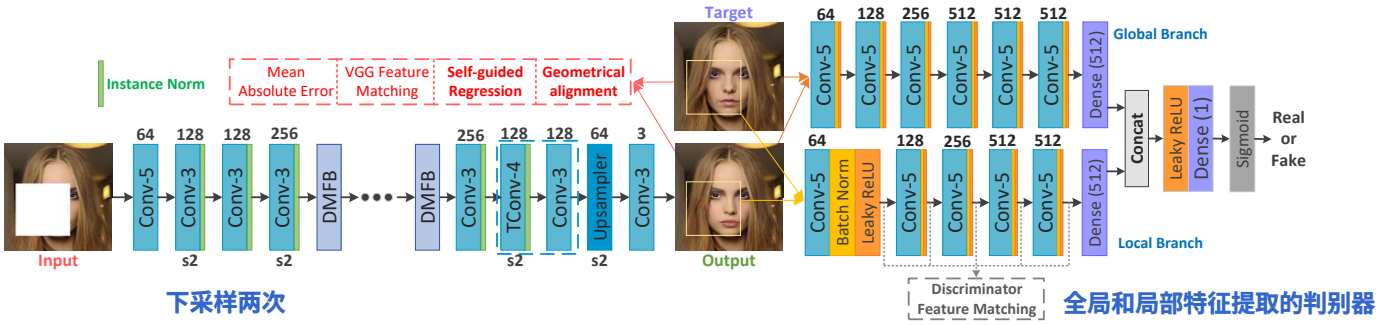


Figure 3. The framework of our method. The activation layer followed by each "convolution + norm" or convolution layer in the generator is omitted for conciseness. The activation function adopts ReLU except for the last convolution (Tanh) in the generator. Blue dotted box indicates our upsampler module (TConv-4 is 4×4 transposed convolution) and "$s2$" denotes the stride of 2.

model parameters. To enlarge the receptive fields and ensure dense convolution kernels simultaneously, we propose our dense multi-scale fusion block (DMFB, see in Figure 2) inspired by [8]. Specifically, the first convolution on the left in DMFB reduces the channels of input features to 64 for decreasing the parameters, and then these processed features are sent to four branches to extract multi-scale features, denoted as $\mathbf{x}_i$ $(i = 1, 2, 3, 4)$, by using dilated convolutions with different dilation factors. Except for $\mathbf{x}_1$, each $\mathbf{x}_i$ has a corresponding 3×3 convolution, denoted by $K_i(\cdot)$. Through a cumulative addition fashion, we can get dense multi-scale features from the combination of various sparse multi-scale features. We denote by $\mathbf{y}_i$ the output of $K_i(\cdot)$. The combination part can be formulated as

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i, & i = 1; \\ K_i(\mathbf{x}_{i-1} + \mathbf{x}_i), & i = 2; \\ K_i(\mathbf{y}_{i-1} + \mathbf{x}_i), & 2 < i \leq 4. \end{cases} \quad (1)$$

The following step is the fusion of concatenated features simply using a 1×1 convolution. In a word, this basic block especially enhances the general dilated convolution and has fewer parameters than large kernels.

Different from previous generative inpainting networks [30, 26] that apply WGAN-GP [7] for adversarial training, we propose to use RaGAN [11] to pursue more photo-realistic generated images [25]. This discriminator also considers the consistency of global and local images.

3.2. Loss functions

3.2.1 Self-guided regression loss

Here, we address the semantic structure preservation issue. We scheme to take self-guided regression constraint to correct the image semantic level estimation. Briefly, we compute the discrepancy map between generated contents and corresponding ground truth to navigate the similarity measure of the feature map hierarchy from the pre-trained VGG19 [23] network. At first, we investigate the

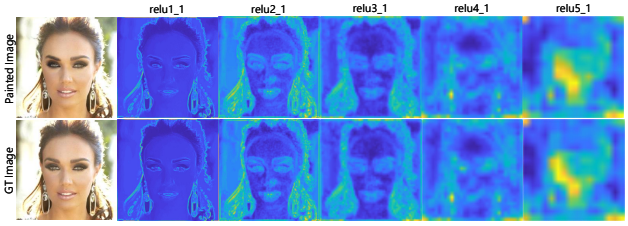


Figure 4. Visualization of average VGG feature maps.

characteristic of VGG feature maps. Given an input image $\mathbf{I}_A$, it is first fed forward through the VGG19 to yield a five-level feature map pyramid, where their spatial resolution reduces low progressively. Specifically, the l-th $(l = 1, 2, 3, 4, 5)$ level is set to the feature tensor produced by relu$l$_1 layer of VGG19. These feature tensors are denoted by F_A^l. We give an illustration of average feature maps $F_{A\_avg}^l = \frac{1}{M} \sum_{m=1}^{M} F_{A\_m}^l$ in Figure 4, which suggests that the deeper layers of a pre-trained network represent higher-level semantic information, while lower-level features more focus on textural or structural details, such as edges, corners, and other simple conjunctions. In this paper, we would intend to improve the detail fidelity of the completed image, especially for building and face images.

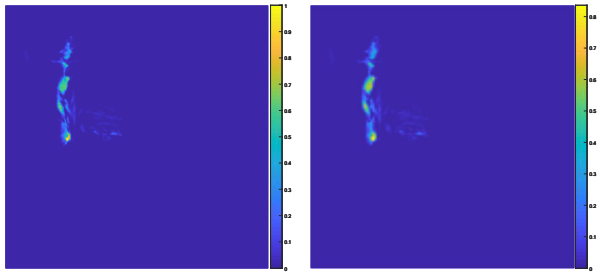


Figure 5. Visualization of guidance maps. (Left) Guidance map $\mathbf{M}_{guidance}^1$ for "relu1_1" layer. (Right) Guidance map $\mathbf{M}_{guidance}^2$ for "relu2_1" layer. These are corresponding to Figure 4.

To this end, through the error map between the output image produced by the generator and ground truth, we get the guidance map to distinguish between areas of challenging and manageable. Therefore, we propose to use the following equation to gain the average error map:

$$\mathbf{M}_{error} = \frac{1}{3}\sum_{c\in\mathcal{C}}\left(\mathbf{I}_{out,c} - \mathbf{I}_{gt,c}\right)^2, \tag{2}$$

where $\mathcal{C}$ are the three color channels, $\mathbf{I}_{out,c}$ denotes c-th channel of the output image. Then, the normalized guidance mask can be calculated by

$$\mathbf{M}_{guidance,p} = \frac{\mathbf{M}_{error,p} - \min\left(\mathbf{M}_{error}\right)}{\max\left(\mathbf{M}_{error}\right) - \min\left(\mathbf{M}_{error}\right)}, \tag{3}$$

where $\mathbf{M}_{error,p}$ is the error map value at position p. Note that our guidance mask with continuous values between 0 and 1, which is soft instead of binary. $\mathbf{M}^l_{guidence}$ corresponds l-th level feature maps and it can be expressed by

$$\mathbf{M}^{l+1}_{guidance} = AP\left(\mathbf{M}^l_{guidance}\right), \tag{4}$$

where AP denotes *average pooling* with kernel size of 2 and stride of 2. Here, $\mathbf{M}^1_{guidance} = \mathbf{M}_{guidance}$ (Equation 3). In this way, the value range of $\mathbf{M}^l_{guidance}$ is still between 0 and 1. In view of the fact that lower-level feature map contains more detailed information, we choose feature tensors from "relu1_1" and "relu2_1" layers to describe image semantic structures. Thus, our self-guided regression loss is defined as

$$\mathcal{L}_{self-guided} = \sum_{l=1}^{2} w^l \frac{\left\|\mathbf{M}^l_{guidance}\odot\left(\Psi^l_{\mathbf{I}_{gt}} - \Psi^l_{\mathbf{I}_{output}}\right)\right\|_1}{N_{\Psi^l_{\mathbf{I}_{gt}}}}, \tag{5}$$

where $\Psi^l_{\mathbf{I}_*}$ is the activation map of the relu$l$_1 layer given original input $\mathbf{I}_*$, $N_{\Psi^l_{\mathbf{I}_{gt}}}$ is the number of elements in $\Psi^l_{\mathbf{I}_{gt}}$, $\odot$ is the element-wise product operator, and $w^l = \frac{1e3}{\left(C_{\Psi^l_{\mathbf{I}_{gt}}}\right)^2}$ followed by [35]. Here, C is the channel size of feature map $\Psi^l_{\mathbf{I}_{gt}}$.

An obvious benefit for this regularization is to suppress regions with higher uncertainty (as shown in Figure 5). $\mathbf{M}_{guidance}$ can be viewed as a spatial attention map, which preferably optimizes areas that are difficult to handle. Our self-guided regression loss is performed lower-level semantic space instead of pixel space. The merit of this way would appear in the perceptual image synthesis with pleasant structural information.

3.2.2 Geometrical alignment constraint

In the typical solutions, the metric evaluation in higher-level feature space is only achieved using pixel-based loss, *e.g.*, L1 or L2. It doesn't take the alignment of each high-level feature map semantic hub into account. To better measure the distance between high-level features belong to prediction and ground-truth, we impose the geometrical alignment constraint on the response maps of "relu4_1" layer. This term can help the generator create a plausible image that aligned with the target image in position. Specifically, this term encourages the output feature center to be spatially close to the target feature center. The geometrical center for the k-th feature map along axis u is calculated as

$$c^k_u = \sum_{u,v} u\cdot\mathbf{R}\left(k,u,v\right) \Big/ \sum_{u,v}\mathbf{R}\left(k,u,v\right), \tag{6}$$

where response maps $\mathbf{R} = \mathrm{VGG}\left(\mathbf{I};\theta_{\mathrm{vgg}}\right) \in \mathbb{R}^{K\times H\times W}$. $\mathbf{R}\left(k,u,v\right) \Big/ \sum_{u,v}\mathbf{R}\left(k,u,v\right)$ represents a spatial probability distribution function. c^k_u denotes coordinate expectation along axis u. Then, we pass both the completed image $\mathbf{I}_{output}$ and ground-truth image $\mathbf{I}_{gt}$ through the VGG network and obtain the corresponding response maps $\mathbf{R}'$ and $\mathbf{R}$. Given these response maps, we compute the centers $\left\langle c^{k'}_u, c^{k'}_v\right\rangle$ and $\left\langle c^k_u, c^k_v\right\rangle$ using Equation 6. Then, we formulate the geometrical alignment constraint as

$$\mathcal{L}_{align} = \sum_k\sum_{u,v}\left\|\left\langle c^{k'}_u, c^{k'}_v\right\rangle - \left\langle c^k_u, c^k_v\right\rangle\right\|^2_2. \tag{7}$$

3.2.3 Feature matching losses

The VGG feature matching loss $\mathcal{L}_{fm\_vgg}$ compares the activation maps in the intermediate layers of well-trained VGG19 [23] model, which can be written as

$$\mathcal{L}_{fm\_vgg} = \sum_{l=1}^{5} w^l \frac{\left\|\Psi^l_{\mathbf{I}_{gt}} - \Psi^l_{\mathbf{I}_{output}}\right\|_1}{N_{\Psi^l_{\mathbf{I}_{gt}}}}, \tag{8}$$

where N^l is the number of elements in $\Psi^{\mathrm{relu}l\_1}_{\mathbf{I}_{gt}}$. We also introduce local branch in discriminator feature matching loss $\mathcal{L}_{fm\_dis}$, which is reasonable to assume that the output image are consistent with the ground-truth images under any measurements (*i.e.*, any high-dimensional spaces). This feature matching loss is defined as

$$\mathcal{L}_{fm\_dis} = \sum_{l=1}^{5} w^l \frac{\left\|D^l_{local}\left(\mathbf{I}_{gt}\right) - D^l_{local}\left(\mathbf{I}_{output}\right)\right\|_1}{N_{D^l_{local}(\mathbf{I}_{gt})}}, \tag{9}$$

where $D^l_{local}\left(\mathbf{I}_*\right)$ is the activation in the l-th selected layer of the discriminator given input $\mathbf{I}_*$ (see in Figure 3). Note that the hidden layers of the discriminator are trainable, which is slightly different from the well-trained VGG19 network trained on the ImageNet dataset. It can adaptively

update based on specific training data. This complementary feature matching can dynamically extract features that may be not mined in VGG model.

3.2.4 Adversarial loss

For improving the visual quality of inpainted results, we use relativistic average discriminator [11] as in ESRGAN [25], which is the recent state-of-the-art perceptual image super-resolution algorithm. For the generator, the adversarial loss is defined as

$$\begin{aligned}\mathcal{L}_{adv} = &-\mathbb{E}_{x_r}\left[\log\left(1 - D_{Ra}\left(x_r, x_f\right)\right)\right] \\ &- \mathbb{E}_{x_f}\left[\log\left(D_{Ra}\left(x_f, x_r\right)\right)\right],\end{aligned} \tag{10}$$

where $D_{Ra}\left(x_r, x_f\right) = sigmoid\left(C\left(x_r\right) - \mathbb{E}_{x_f}\left[C\left(x_f\right)\right]\right)$ and $C\left(\cdot\right)$ indicates the discriminator network without the last *sigmoid* function. Here, real/fake data pairs $\left(x_r, x_f\right)$ are sampled from ground-truth and output images.

3.2.5 Final objective

With self-guided regression loss, geometrical alignment constraint, VGG feature matching loss, discriminator feature matching loss, adversarial loss, and mean absolute error (MAE) loss, our overall loss function is defined as

$$\begin{aligned}\mathcal{L}_{total} = &\mathcal{L}_{mae} + \lambda\left(\mathcal{L}_{self-guided} + \mathcal{L}_{fm\_vgg}\right) \\ &+ \eta\mathcal{L}_{fm\_dis} + \mu\mathcal{L}_{adv} + \gamma\mathcal{L}_{align},\end{aligned} \tag{11}$$

where λ, η, μ, and γ are used to balance the effects between the losses mentioned above.

4. Experiments

We evaluate the proposed inpainting model on Paris Street View [20], Places2 [34], CelebA-HQ [12], and a new challenging facial dataset FFHQ [13].

4.1. Experimental settings

For our experiments, we set $\lambda = 25$, $\eta = 5$, $\mu = 0.03$ and $\gamma = 1$ in Equation 11. The training procedure is optimized using Adam optimizer [14] with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. We set the learning rate to $2e-4$. The batch size is 16. We apply PyTorch framework to implement our model and train them using NVIDIA TITAN Xp GPU (12GB memory).

For training, given a raw image $\mathbf{I}_{gt}$, a binary image mask $\mathbf{M}$ (value 0 for known pixels and 1 denotes unknown ones) at a random position. In this way, the input image $\mathbf{I}_{in}$ is obtained from the raw image as $\mathbf{I}_{in} = \mathbf{I}_{gt} \odot (\mathbf{1} - \mathbf{M})$. Our inpainting generator takes $[\mathbf{I}_{in}, \mathbf{M}]$ as input, and produces prediction $\mathbf{I}_{pred}$. The final output image is $\mathbf{I}_{out} = \mathbf{I}_{in} + \mathbf{I}_{pred} \odot \mathbf{M}$. All input and output are linearly scaled to $[-1, 1]$. We train our network on the training set and evaluate it on the validation set (Places2, CelebA-HQ, and FFHQ) or testing set (Paris street view and CelebA). For training, we use images of resolution 256×256 with the largest hole size 128×128 as in [30, 26]. For Paris street view (936×537), we randomly crop patches with resolution 537×537 and then scale down them to 256×256 for training. Similarly, for Places2 ($512 \times *$), 512×512 sub-images are cropped at a random location. These images are scaled down to 256×256 for our model. For CelebA-HQ and FFHQ face datasets (1024×1024), images are directly scaled to 256. We use the irregular mask dataset provided by [18]. All results generated by our model are not post-processed.

4.2. Qualitative comparisons

As shown in Figures 6, 1, and 7, compared with other state-of-the-art methods, our model gives a noticeable visual improvement on textures and structures. For instance, our network generates plausible image structures in Figure 6, which mainly stems from the dense multi-scale fusion architecture and well-designed losses. The realistic textures are hallucinated via feature matching and adversarial training. For Figure 1, we show that our results with more realistic details and fewer artifacts than the compared approaches. Besides, we give partial results of our method and PICNet [33] on Places2 dataset in Figure 7. The proposed DMFN creates more reasonable, natural, and photo-realistic images. Additionally, we also show some example results (masks at random position) of our model trained on FFHQ in Figure 8. In Figure 9, our method performs more stable and fine for large-area irregular masks than compared algorithms. More compelling results can be found in the *supplementary material*.

4.3. Quantitative comparisons

Following [30, 26], we measure the quality of our results using peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Learned perceptual image patch similarity (LPIPS) [32] is a new metric that can better evaluate the perceptual similarity between two images. Because the purpose of image inpainting is to pursue visual effects, we adopt LPIPS as the main qualitative assessment. The lower the values of LPIPS, the better. In Places2, 100 validation images from "canyon" scene category are chosen for evaluation. As shown in Table 1, our method produces acceptable results compared with CA [30], GMCNN [26], and PICNet [33] in terms of all evaluation measurements.

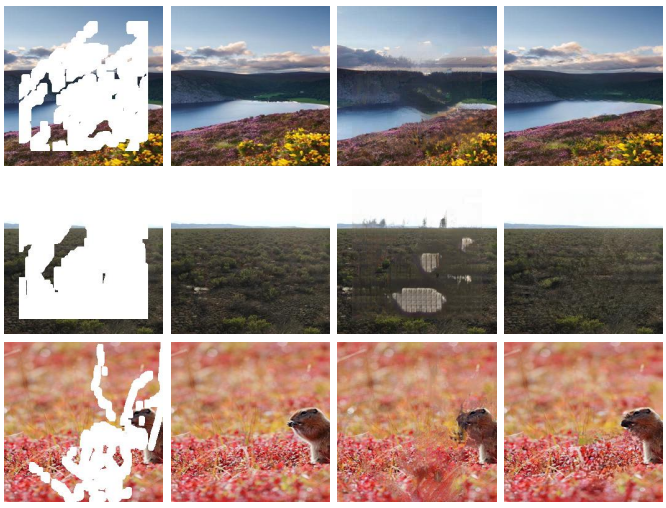
We also conducted user studies as illustrated in Figure 10. The scheme is based on blind randomized A/B/C tests deployed on Google Forms platform as in [26]. Each survey includes 40 single-choice questions. Each question involves three options (completed images that are generated from the same corrupted input by three different methods).

Input Image CE [20] Shift-Net [28] GMCNN [26] PICNet [33] DMFN (Ours)

Figure 6. Visual comparisons on Paris street view.

Table 1. Quantitative results (center regular mask) on four testing datasets.

Method	Paris street view (100)	Places2 (100)	CelebA-HQ (2,000)	FFHQ (10,000)
	LPIPS / PSNR / SSIM	LPIPS / PSNR / SSIM	LPIPS / PSNR / SSIM	LPIPS / PSNR / SSIM
CA [30]	N/A	0.1524 / 21.32 / 0.8010	0.0724 / 24.13 / 0.8661	N/A
GMCNN [26]	0.1243 / 24.38 / 0.8444	0.1829 / 19.51 / 0.7817	0.0509 / 25.88 / 0.8879	N/A
PICNet [33]	0.1263 / 23.79 / 0.8314	0.1622 / 20.70 / 0.7931	N/A	N/A
DMFN (Ours)	**0.1018 / 25.00 / 0.8563**	**0.1361 / 21.53 / 0.8079**	**0.0460 / 26.50 / 0.8932**	**0.0457 / 26.49 / 0.8985**



Input GT PICNet [33] DMFN (Ours)

Figure 7. Visual comparisons on Places2. **Best viewed with zoom-in.**

Input DMFN (Ours) GT

Figure 8. Visual results on FFHQ dataset.

There are 20 participants invited to accomplish this survey. They are asked to select to the most realistic item in each question. The option order is shuffled each time. Finally, our method outperforms compared approaches by a large margin.

4.4. Ablation study

4.4.1 Effectiveness of DMFB

To validate the representation ability of our DMFB, we replace its middle part (4 dilated convolutions and combination operation) to a 3×3 dilated convolution (256 chan-

Input GT PICNet [33] DMFN (Ours)

Figure 9. Inpainted images with irregular masks on Paris StreetView and CelebA-HQ. **Best viewed with zoom in.**

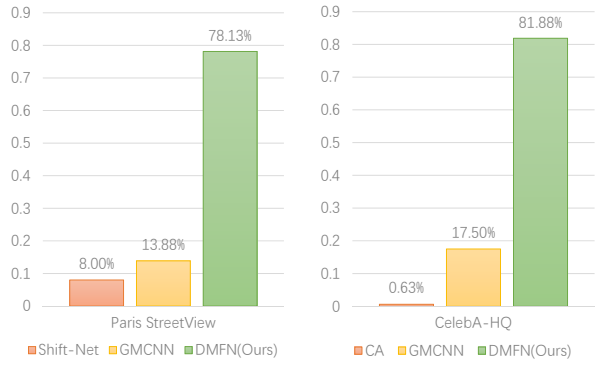


Figure 10. Results of user study.

nels) with dilation rate of 2 or 8 ("rate=2" or "rate=8", see in Table 2). Additionally, to verify the strength of $K_i(\cdot)$ in combination operation, we perform the DMFB without $K_i(\cdot)$ that denoted as "w/o $K_i(\cdot)$" in Table 2. Combined with Table 2 and Figure 11, we can clearly see that our model with DMFB (**Parms**: $471,808$) predicts reasonable and less artifact images than ordinary dilated convolutions (**Parms**: $803,392$). Meanwhile, the results of "rate=2" and "rate=8" suggest the importance of spatial support as discussed in [9]. It also demonstrates large and dense receptive field is beneficial to completing images with large holes.

Table 2. Quantitative results of different structures on Paris street view dataset.

Model	rate=2	rate=8	w/o combination	w/o $K_i(\cdot)$	DMFB
Params	803,392	803,392	361,024	361,024	471,808
LPIPS↓	0.1059	0.1067	0.1083	0.1026	**0.1018**
PSNR↑	24.93	24.91	24.24	24.93	**25.00**
SSIM↑	0.8530	0.8549	0.8505	0.8561	**0.8563**

Table 3. Investigation of self-guided regression loss and geometrical alignment constraint.

Metric	w/o self-guided	w/o align	w/o dis_fm	with all
LPIPS↓	0.0537	0.0534	0.0542	**0.0530**
PSNR↑	25.73	25.63	25.65	**25.83**
SSIM↑	0.8892	0.8884	0.8870	**0.8892**

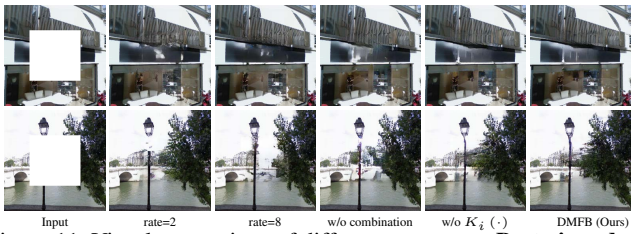


Figure 11. Visual comparison of different structures. **Best viewed with zoom-in.**

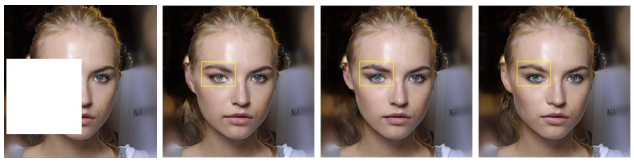


Figure 12. Visual comparison of results using different losses. **Best viewed with zoom-in.**

4.4.2 Self-guided regression and geometrical alignment constraint

To investigate the effect of proposed self-guided regression loss and geometrical alignment constraint, we train a complete DMFN on CelebA-HQ dataset without the corresponding loss. As shown in Figure 12, "w/o self-guided" model cannot restore some structural details and "w/o alignment" item shows some misalignment in the yellow box, while "with all" model (DMFN trained all losses) can mitigate these problems. And we give the quantitative results in Table 3, which validates the effectiveness of various proposed losses. More discussions about loss functions are provided in the *supplementary material*.

5. Conclusion

In this paper, we proposed a dense multi-scale fusion network with self-guided regression loss and geometrical alignment constraint for image fine-grained inpainting, which highly improves the quality of produced images. Specifically, dense multi-scale fusion block is developed to extracted better features. With the assistance of self-guided regression loss, the restoration of semantic structures becomes easier. Additionally, geometrical alignment constraint is inductive to the coordinate registration between generated image and ground-truth, which promotes the reasonableness of painted results.

References

[1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001. 2

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24:1–24:11, 2009. 1, 2

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424, 2000. 2

[4] Min cheol Sagong, Yong goo Shin, Seung wook Kim, Seung Park, and Sung jea Ko. Pepsi: Fast image inpainting with parallel decoding network. In *CVPR*, pages 11360–11368, 2019. 3

[5] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346, 2001. 2

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 1, 2

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, pages 5767–5777, 2017. 2, 4

[8] Zheng Hui, Jie Li, Xinbo Gao, and Xiumei Wang. Progressive perception-oriented network for single image super-resolution. *arXiv:1907.10399v1*, 2019. 4

[9] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107:1–107:14, 2017. 1, 2, 3, 8

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 3

[11] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. In *ICLR*, 2019. 2, 4, 6

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 6

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 6

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[15] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (TOG)*, 24(3):795–802, 2005. 2

[16] Jingyuan Li, Fengxiang He, Lefei Zhang, Bo Du, and Dacheng Tao. Progressive reconstruction of visual structure for image inpainting. In *ICCV*, pages 5962–5971, 2019. 3

[17] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *CVPR*, pages 3911–3919, 2017. 1, 2

[18] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018. 2, 6

[19] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCVW*, 2019. 3

[20] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPP*, pages 2536–2544, 2016. 1, 2, 6, 7

[21] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *ICCV*, pages 181–190, 2019. 3

[22] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *CVPR*, pages 1–8, 2008. 2

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 4, 5

[24] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C.-C. Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *ECCV*, pages 3–19, 2018. 2, 3

[25] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, pages 63–79, 2018. 4, 6

[26] Yi Wang, Xin Tao, Xiaojun Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative mulit-column convolutional neural networks. In *NeurIPS*, pages 331–340, 2018. 1, 2, 3, 4, 6, 7

[27] Wei Xiong, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *CVPR*, pages 5840–5848, 2019. 3

[28] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *ECCV*, pages 1–17, 2018. 2, 3, 7

[29] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, pages 6721–6729, 2017. 1, 2, 3

[30] Jiahui Yu, Zhe Lin, Jimie Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018. 1, 2, 3, 4, 6, 7

[31] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *CVPR*, pages 1486–1494, 2019. 2

[32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[33] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, pages 1438–1447, 2019. 2, 6, 7, 8

[34] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464. 6

[35] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics (TOG)*, 37(4):49:1–49:13, 2018. 5