Image Inpainting with Local and Global Refinement

Weize Quan, Ruisong Zhang, Yong Zhang, Zhifeng Li, Jue Wang, and Dong-Ming Yan

***Abstract*—Image inpainting has made remarkable progress with recent advances in deep learning. Popular networks mainly follow an encoder-decoder architecture (sometimes with skip connections) and possess sufficiently large receptive field, *i.e.*, larger than the image resolution. The receptive field refers to the set of input pixels that are path-connected to a neuron. For image inpainting task, however, the size of surrounding areas needed to repair different kinds of missing regions are different, and the very large receptive field is not always optimal, especially for the local structures and textures. In addition, a large receptive field tends to involve more undesired completion results, which will disturb the inpainting process. Based on these insights, we rethink the process of image inpainting from a different perspective of receptive field, and propose a novel three-stage inpainting framework with local and global refinement. Specifically, we first utilize an encoder-decoder network with skip connection to achieve coarse initial results. Then, we introduce a shallow deep model with small receptive field to conduct the local refinement, which can also weaken the influence of distant undesired completion results. Finally, we propose an attention-based encoder-decoder network with large receptive field to conduct the global refinement. Experimental results demonstrate that our method outperforms the state of the arts on three popular publicly available datasets for image inpainting. Our local and global refinement network can be directly inserted into the end of any existing networks to further improve their inpainting performance. Code is available at https://github.com/weizequan/LGNet.git.**

***Index Terms*—Image inpainting, Neural networks, Receptive field.**

I. Introduction

IMAGE inpainting refers to the completion of missing regions in digital images. The goal of image inpainting is to fill the missing regions with semantically reasonable and visually realistic content, which is also consistent with the remaining parts of the image (see Fig. 1 for examples). It can be used as an image editing tool, *e.g.*, to remove unwanted objects from an image or to restore detective regions in damaged paintings.

Early works can be classified into two categories: diffusion-based approaches [1]–[3] and patch-based approaches [4]–[8]. The former diffuse the information from the surrounding regions to the interior of the missing regions based on partial differential equations and variational methods, while the latter propagate the image contents from known regions to unknown regions via appearance copying and pasting. These methods have achieved great visual effects when handling small missing regions. However, for large missing regions they cannot create large structures and objects that are not present anywhere else in the image.

To solve this problem, recent work resorts to deep learning, *e.g.*, using convolutional neural networks (CNNs) [9], [10] and generative adversarial networks (GANs) [11]. From the perspective of network design, these inpainting methods can be roughly categorized into one-stage networks (one generator) [12]–[18], two-stage networks (two generators) [19]–[24], and progressive networks (one or multiple generators applied in an iterative manner) [25]–[29]. In literature, several aspects have gained more attentions, *e.g.*, simultaneous or step-wise structure and texture inpainting [22], different attention strategies for context information collection [19], [30], progressive filling from border to center [28], and so on.

In this paper, we design an inpainting network from a different perspective, *i.e.*, *receptive field*, which refers to the set of input pixels that are path-connected to a neuron [31]. Our work is inspired from three points: (1) The image inpainting problem is relevant to the receptive field, and the scope of neighboring areas needed to repair different kinds of missing regions are different. (2) Very large receptive field, as most previous methods pursued, may not be optimal, especially for repairing the local structures and details. Meanwhile, a large receptive field tends to contain more undesired completion results, which potentially have the negative effect on the inpainting process, *e.g.*, the inpainting of local patterns as shown in the first two rows of Fig. 2. (3) The receptive field is an important aspect of deep neural networks, which has gained more attention in image classification and semantic segmentation [32], [33]. However, it is less focused in image inpainting with deep learning. Therefore, we propose a three-stage inpainting framework with respect to the receptive field. We first apply a coarse inpainting network with large receptive field (covering the whole image) to fill the holes, which can complete the primary structure and partial texture details. Then, we propose the local and global refinement networks with different sizes of receptive fields to improve the inpainting results. These two sub-networks separately pay attention to the "local inpainting" and "global inpainitng", and are combined to obtain the "whole inpainting".

Our work provides the following contributions:

- We propose a local refinement network with small receptive field to improve the inpainted image. This shallow deep network can repair some missing regions, *e.g.*, the local structures and texture details, according to the surrounding local regions and prevent from the interference

W. Quan, R. Zhang, and D.-M. Yan are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with School of Artificial Intelligence, the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yandongming@gmail.com) (Corresponding author: Dong-Ming Yan.)

Y. Zhang and J. Wang are with the Tencent AI Lab, ShenZhen, P.R.China.

Z. Li is with the Tencent Data Platform, ShenZhen, P.R.China.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes a Supplementary Material, which provides the network details of our method. Contact qweizework@gmail.com for further questions about this work.

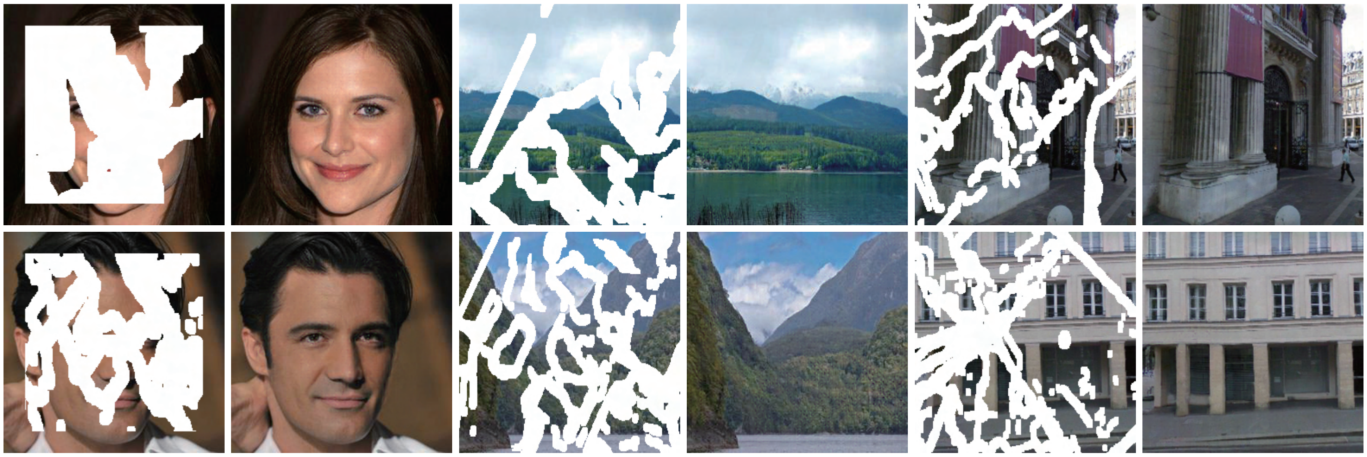


Fig. 1. Selected image inpainting results of our proposed method on CelebA-HQ (left), Places2 (middle), and Paris StreetView (right) datasets, respectively.

of long-distance failed completion results after the coarse inpainting stage.

- We propose an attention-based global refinement network with large receptive field to further enhance the completion result. This network can further improve the visual quality using the global information, especially for the large structures and long-distance texture patterns. In addition, the attention computation is more robust and stable due to the relatively good quality of the output of the local refinement network.
- Our proposed inpainting framework achieves the state-of-the-art performance on three popular public inpainting datasets: CelebA-HQ [34], Places2 [35], and Paris StreetView [36].

The rest of this paper is organized as follows. Section II reviews related work and highlights the differences between our method and previous works. Section III presents the motivation, network architecture, and loss functions of the proposed method. Section IV evaluates the performance of our method and compares with state-of-the-art methods. Section V draws the conclusions and discusses the future works.

II. Prior Art

Existing image inpainting methods are mainly divided into two categories: traditional inpainting methods and deep-learning-based inpainting methods.

A. Traditional Inpainting Methods

This kind of works mainly include diffusion-based methods [1]–[3] and patch-based methods [4]–[8]. We refer readers to the surveys [37], [38] for more details of the traditional approaches.

Diffusion-based methods. The term diffusion describes the process of propagating local information with smoothness constraints. The use of diffusion for image inpainting was pioneered by Bertalmio *et al.* [1], specifically, the anisotropic diffusion was iteratively applied without losing sharpness in the reconstruction process. Based on the joint interpolation of the image gray levels and the gradient directions, Ballester *et al.* [2] formulated the image inpainting as a variational problem. Tschumperlé and Deriche [3] proposed a single generic anisotropic diffusion equation and obtained the better inpainting results. However, these approaches cannot handle relatively large missing regions due to the limited extended prediction from the boundary.

Patch-based methods. The core idea of patch-based methods is to propagate the appearance information from the background regions to the missing regions based on the patch-level similarity. To reduce the time costs for patch matching, Barnes *et al.* [4] proposed a randomized nearest-neighbor patch matching algorithm, namely, PatchMatch, which is widely used in the editing tools. Through using mid-level structural cues, Huang *et al.* [6] proposed an automatic completion algorithm based on augmented patch-based matching. Ding *et al.* [8] proposed a non-local texture similarity measure to select multiple candidate patches, and then applied the α-trimmed mean filter to obtain the inpainted results. Moreover, patch-based method is also often used in exemplar-based inpainting [5], [7]. These methods mainly depend on the low-level information, and thus cannot generate semantically correct results for large missing regions.

B. Deep-learning-based Inpainting Methods

From the perspective of network design, existing deep-learning-based image inpainting methods can be roughly classified into three types: one-stage, two-stage, and progressive methods.

One-stage methods. In early works, Pathak *et al.* [12] designed an encoder-decoder architecture which is trained via a combination of pixel-wise reconstruction loss and adversarial loss [11]. To improve the consistency of image completion, Iizuka *et al.* [13] introduced the global and local context discriminators to train the fully-convolutional completion network. They mainly focus on the discriminator design while adopting a simple encoder-decoder network as the generator. However, our core idea is to design an efficient inpainting generator according to the receptive field, which is important for image inpainting. In addition, a more efficient patch-based discriminator was proposed by [20] and adopted by many following works. Liu *et al.* [14] designed a partial convolution operation followed by an automatic mask-update step to better fill irregular holes. Inspired by several existing methods [19], [39], [40] that use attention mechanism to fill in the missing regions, Zeng *et al.* [15] proposed a pyramid-context encoder network for image completion via attention transfer. Xie *et al.* [16] generalized the partial convolution [14] by introducing learnable bidirectional attention maps. To jointly recover the structure and texture, Liu *et al.* [17] fused the texture features (from the shallow layers of the encoder) and structure features (from the deep layers of the encoder) together via feature equalization. Liao *et al.* [41] utilized the semantic segmentation map to guide the inpainting process of mixed scenes, which needs additional semantic segmentation annotations during the training stage. More recently, Zhang *et al.* [18] proposed a pixel-wise dense detector to localize the artifacts of completed results, and this position information is inserted into the reconstruction loss to better train the completion network. With the same purpose, Hui *et al.* [42] devised the feature center alignment constraint, and designed a self-guided regression loss to enhance the semantic details. Wang *et al.* [43] proposed the validness migratable convolution and regional composite normalization modules to better utilize the valid pixels during the inpainting process. Similarly, Zhu *et al.* [44] proposed a mask-aware convolution and point-wise normalization for image inpainting, standing from the dynamic concept as well. These methods sometimes suffer from the noticeable artifacts, *e.g.*, smooth textures and incorrect semantics, due to a lack of adequate constraints.

Two-stage methods. Yu *et al.* [19] proposed an improved generative inpainting framework consisting of a coarse network and a refinement network. In the refinement network, they introduced contextual attention to model the long-term correlation. Inspired by [14], Yu *et al.* [20] subsequently upgraded the previous work [19] by introducing gated convolution and a patch-based GAN discriminator. Nazeri *et al.* [21] proposed an edge-guided two-stage image inpainting method. They first recovered the edge map of the missing region, and then combined this edge map with the incomplete image as the input of the second stage to perform the inpainting task. Due to the limited structural guidance of edge images, Ren *et al.* [22] employed images processed with edge-preserving smoothing as the representation of structure, and then modeled the inpainting task as the combination of structure reconstruction and texture generation. Wu *et al.* [45] combined a local binary pattern learning network and a generative inpainting network. For these approaches, the network architectures of two stages usually are very similar, and thus their receptive fields are close (greater than or approximately equal to the input image resolution). And they rarely focus on the impact of the size of receptive field in the refinement network.

Our network also designed in a coarse-to-fine manner, however, there exists two apparent differences compared to previous two-stage frameworks: (1) We deeply analyze the impact of the refinement networks with different receptive fields on image inpainting (see Sec. III-A and IV-C), which is unfortunately omitted by previous approaches. The existing coarse-to-fine methods only focus on the encoder-decoder generator with large receptive field by introducing the dilated convolution or contextual attention. However, we highlight the importance of network with small receptive filed for image inpainting. And this point provides interesting observations and insights for future studies. (2) The combination of local refinement network (with small receptive field) and global refinement network (with large receptive field) can handle different inpainting scenarios, including local structures, local texture details, large structures, and long-distance texture patterns. This is a new design idea. In addition, the attention computation in global refinement network (our third stage) is more robust attributing to the better representation provided by the local refinement network, compared to the previous two-stage methods that compute the contextual attention in the second stage with coarse results.

Progressive methods. Zhang *et al.* [25] divided the image inpainting process into four different phases, and used an LSTM (long short-term memory) architecture [46] to control the information flow of the progressive process. However, they cannot handle irregular holes commonly appeared in real-world applications. To address this limitation, Guo *et al.* [26] proposed full-resolution residual networks with several dilation modules. Li *et al.* [27] progressively reconstructed the visual structure to entangle the visual feature reconstruction for image inpainting. Different from existing progressive methods, Li *et al.* [28] followed a progressive framework in feature space and devised a recurrent feature reasoning network with consistent attention. Zeng *et al.* [29] proposed a confidence-based iterative inpainting method. Their network is similar to [19], whereas the output of the second generator is the inpainted image along with a confidence map, and this map is used to guide the next iteration of the completion process. Due to the iterative nature, these methods inevitably suffer from the high computational costs.

III. Proposed Method

A. Observation and Motivation

For image inpainting, we observe that many existing works often follow the common design concept, where their networks have very large receptive field, *e.g.*, a U-Net like architecture or using multiple dilation convolution layers. In this work, however, we highlight that a network with small receptive field is also important. To clearly illustrate and analyze the impact of networks with different receptive fields and motivate our work, we train three different networks following a coarse-to-fine framework: (1) a U-Net network with large receptive field (greater than the input image resolution), denoted as "C"; (2) a U-Net network adding a shallow network with small receptive field (approximately a quarter of the input resolution), denoted as "C+F_S"; (3) a U-Net network adding another U-Net network with large receptive field (greater than the input image resolution), denoted as "C+F_L". The corresponding inpainting results on three different datasets are shown in Fig. 2.

Comparing "C+F_S" and "C+F_L" of the first and second rows, we observe that "F_S" can better repair the local

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2022.3152624, IEEE Transactions on Image Processing

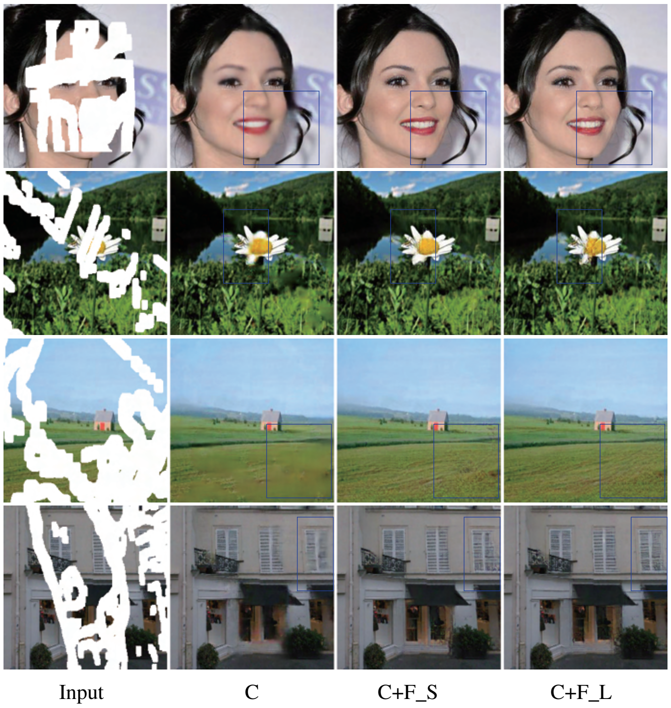


Fig. 2. The inpainting performance of networks with different receptive fields.

structure (the nose and curls) and the local texture details (the flower heart), where the missing regions are mainly related to the surrounding local regions. On the other hand, when comparing "C+F_S" and "C+F_L" of third and fourth rows, we can find that the long-distance texture and large structure are better inpainted by "F_L". The grass in the lower right corner is almost missing, and the network needs to access the remaining information on the left to infer; the inpainting of right window needs to perceive the global information, *i.e.*, the layout and texture of all windows. In summary, network with small receptive field is more efficient for repairing the local structures and textures, while network with large receptive field is more useful for inpainting the long-distance details and large structures. Inspired by the above observation and analysis, in this paper, we propose a three-stage network for image inpainting to combine the networks with different receptive fields, considering the complexity of missing regions. Our network has the local and global refinement sub-networks, therefore, it is called LGNet, as shown in Fig. 3. We describe our network architecture and the corresponding loss functions in the following.

B. Coarse Inpainting Network

Our coarse inpainting network (Net_C) adopts an encoder-decoder architecture with the skip connection. This network consists of eight downsampling and upsampling operations. The long skip connections are applied to propagate the information from the encoder to the decoder to recover the information lost during downsampling. In the end of encoder, the receptive filed is already 766×766, which is much larger than the input image resolution (256×256). Large receptive field is beneficial to the completion of the whole structure. As input, the network accepts an input image $\mathbf{I}_{in}$ and a binary mask $\mathbf{M}$ describing the missing regions (where 0 means the valid pixel and 1 means the missing pixel). The output of our coarse inpainting network Net_C is an inpainted image $\mathbf{I}^C_{out}$. To reduce the blur effect and enhance the realism of inpainted results, we also apply a patch-based discriminator with spectral normalization [48]. This discriminator takes the ground-truth image and inpainted image as input, and outputs a 2D feature map of shape $\mathbb{R}^{32\times32}$. Each element in this feature map is discriminated as real or fake.

The training objective of Net_C consists of a pixel-wise reconstruction loss and an adversarial loss. In our work, we use the weighted L_1 loss for the pixel-wise reconstruction,

$$\begin{aligned}\mathcal{L}^C_{valid} &= \frac{1}{\mathrm{sum}(\mathbb{1}-\mathbf{M})}||(\mathbf{I}^C_{out}-\mathbf{I}_{gt})\odot(\mathbb{1}-\mathbf{M})||_1,\\ \mathcal{L}^C_{hole} &= \frac{1}{\mathrm{sum}(\mathbf{M})}||(\mathbf{I}^C_{out}-\mathbf{I}_{gt})\odot\mathbf{M}||_1,\end{aligned} \tag{1}$$

where $\mathbf{I}_{gt}$ is the ground-truth image, $\odot$ is the element-wise product operation, and $\mathrm{sum}(\mathbf{M})$ is the number of non-zero elements in $\mathbf{M}$. Then the pixel-wise reconstruction loss is formulated as:

$$\mathcal{L}^C_r = \mathcal{L}^C_{valid} + \lambda_h \cdot \mathcal{L}^C_{hole}, \tag{2}$$

where λ_h is a balancing factor.

For GAN loss, we use the least square loss [49], the corresponding loss functions for the coarse inpainting network and the discriminator are defined as:

$$\mathbf{I}^C_{mer} = \mathbf{I}_{in}\odot(\mathbb{1}-\mathbf{M}) + \mathbf{I}^C_{out}\odot\mathbf{M}, \tag{3}$$

$$\mathcal{L}^C_G = \mathbb{E}_{\mathbf{I_{mer}}\sim p_{\mathbf{I}_{mer}}(\mathbf{I_{mer}})}\Big[(D(\mathbf{I}^C_{mer})-1)^2\Big], \tag{4}$$

$$\begin{aligned}\mathcal{L}_D = &\frac{1}{2}\mathbb{E}_{\mathbf{I}\sim p_{data}(\mathbf{I})}\Big[(D(\mathbf{I}_{gt})-1)^2\Big]\\ &+\frac{1}{2}\mathbb{E}_{\mathbf{I_{mer}}\sim p_{\mathbf{I}_{mer}}(\mathbf{I_{mer}})}\Big[(D(\mathbf{I}^C_{mer}))^2\Big],\end{aligned} \tag{5}$$

where $\mathbf{I}^C_{mer}$ is the merged image.

To this end, the total loss for Net_C is $\mathcal{L}_C = \mathcal{L}^C_{valid} + \lambda_h \cdot \mathcal{L}^C_{hole} + \lambda_g \cdot \mathcal{L}^C_G$, and we set $\lambda_h = 6$ and $\lambda_g = 0.1$ in all experiments.

C. Local Refinement Network

For the local refinement, we design a shallow deep network. The local refinement network (Net_L) includes two downsampling operations, four residual blocks, and two upsampling operations (see middle row of Fig. 3). Due to the shallow nature, this network has small receptive field (*i.e.*, 109×109 for each output neuron), and then processes the local region of the above coarse inpainted result in the sliding window manner. Based on this design, some missing regions, *e.g.*, the local structures and textures, can be appropriately repaired using the surrounding local information, and this repair process has no influence from distant and failed filling contents. We have also tested to use more residual blocks, which can gradually

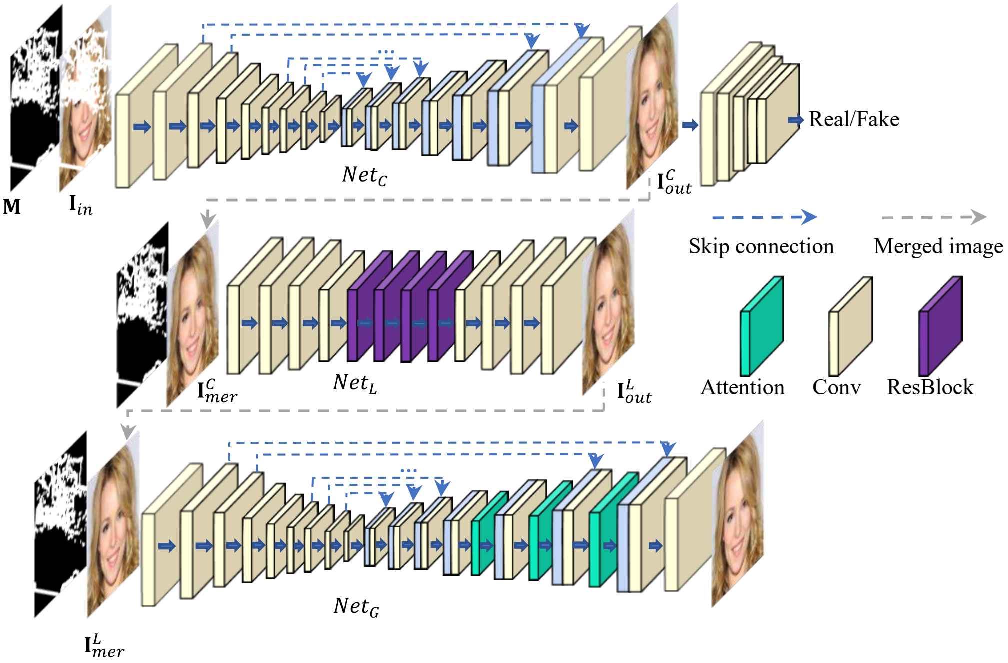


Fig. 3. The network architecture of our proposed LGNet. The purple block in local refinement network (Net_L) represents a two-layer residual block [47]. Three green blocks in global refinement network (Net_G) represent the attention modules, where the resolution is 16×16, 32×32, and 64×64, respectively. In our framework, the output of each stage (*i.e.*, $\mathbf{I}^C_{out}$ or $\mathbf{I}^L_{out}$) is first merged with original incomplete image ($\mathbf{I}_{in}$) and then concatenated with binary mask ($\mathbf{M}$) as the input of the next stage. Gray dotted line represents the merged image (*i.e.*, $\mathbf{I}^C_{mer}$ or $\mathbf{I}^L_{mer}$), which is obtained by using the valid (undamaged) regions in original incomplete image ($\mathbf{I}_{in}$) replaces the corresponding regions of sub-network's output. Please refer to Eqn.(3) for the detailed formula.

increase the receptive field, however, the inpainted results only have negligible improvements.

The first item of training objective of Net_L is the weighted reconstruction loss $\mathcal{L}^L_r$, which is the same as Eqn.(2) except for replacing $\mathbf{I}^C_{out}$ with $\mathbf{I}^L_{out}$ in Eqn.(1). Following [14], the total variation (TV) loss is used as the smoothing penalty. Its formulation is:

$$\begin{aligned}\mathcal{L}^L_{tv} = &||\mathbf{I}^L_{mer}(i, j+1) - \mathbf{I}^L_{mer}(i, j)||_1 \\ &+||\mathbf{I}^L_{mer}(i+1, j) - \mathbf{I}^L_{mer}(i, j)||_1.\end{aligned} \tag{6}$$

Here, the computation process of $\mathbf{I}^L_{mer}$ is the same as that of $\mathbf{I}^C_{mer}$, *i.e.*, Eqn.(3).

Similar to many previous works [14], [16], [17], [28], the perceptual loss [50] and style loss [51] defined on the VGG-16 [52] (pre-trained on ImageNet [53]) are also applied to better recover the structural and textual information. Different from the above pixel-wise reconstruction loss and TV loss, which are conducted in the pixel space, these two losses are computed in the feature space. The perceptual loss can be formulated as:

$$\mathcal{L}^L_{per} = \sum_i ||\mathcal{F}_i(\mathbf{I}^L_{out}) - \mathcal{F}_i(\mathbf{I}_{gt})||_1 + ||\mathcal{F}_i(\mathbf{I}^L_{mer}) - \mathcal{F}_i(\mathbf{I}_{gt})||_1, \tag{7}$$

where $\mathcal{F}_i$ means the feature map of i-th layer in pre-trained VGG-16 network ($i \in \{5, 10, 17\}$).

Similarly, the style loss is defined as:

$$\mathcal{L}^L_{sty} = \sum_i ||\mathcal{G}_i(\mathbf{I}^L_{out}) - \mathcal{G}_i(\mathbf{I}_{gt})||_1 + ||\mathcal{G}_i(\mathbf{I}^L_{mer}) - \mathcal{G}_i(\mathbf{I}_{gt})||_1, \tag{8}$$

where $\mathcal{G}_i(\cdot) = \mathcal{F}_i(\cdot)\mathcal{F}_i(\cdot)^T$ is the Gram matrix [51].

To summarize, the objective for local refinement network is:

$$\mathcal{L}_L = \mathcal{L}^L_{valid} + \lambda_h \cdot \mathcal{L}^L_{hole} + \lambda_{tv} \cdot \mathcal{L}^L_{tv} + \lambda_{per} \cdot \mathcal{L}^L_{per} + \lambda_{sty} \cdot \mathcal{L}^L_{sty}. \tag{9}$$

For image inpainting, [14] is a pioneer work combining the weighted reconstruction loss, perceptual loss, style loss, and TV loss to train the inpainting network. In [14], the corresponding loss weights are chosen by performing a hyper-parameter search on validation images, and the similar weight settings are also adopted in the following works [17], [28], [44]. In our experiments, we also found that losses with these weights in [14] have relatively balanced order of magnitude, therefore, we simply adopt the weight setting [14]. Specifically, $\lambda_h = 6$, $\lambda_{tv} = 0.1$, $\lambda_{per} = 0.05$, and $\lambda_{sty} = 120$.

D. Attention-based Global Refinement Network

After the local refinement process, some visual artifacts are appropriately eliminated with the guidance of surrounding local regions. However, some missing regions (*e.g.*, the large structures or long-distance detail patterns) could be better refined when catching information from relatively large surrounding region. For this purpose, we introduce an attention-based global refinement network to enlarge the extent of captured information for a neuron through two ways, *i.e.*, large receptive field and attention scheme. Because our coarse inpainting network already has enough receptive field covering the whole image, we simply utilize the network architecture of Net_C and add three attention modules in the front of decoder to obtain our attention-based global refinement network Net_G (see three green blocks in the bottom row of Fig. 3).

Local refinement network can provide relatively correct completion result. Therefore, the attention computation in the global refinement network tends to be more stable and robust. Attention scheme has been extensively used in existing works [19], [20], [28] to model the correlations between contextual information and the missing regions, *e.g.*, the symmetry and repeated patterns. In this work, we use the simple self-attention method [19], [28], and more advanced attention method can also be used in our framework. This is not the main focus of our paper. Given a feature map $F \in \mathbb{R}^{C\times HW}$, the affinity $s_{i,j} \in \mathbb{R}^{HW\times HW}$ of F_i and F_j is calculated by:

$$s_{i,j} = \frac{exp(\hat{s}_{i,j})}{\sum_k exp(\hat{s}_{i,k})}, \hat{s}_{i,j} =< \frac{F_i}{||F_i||}, \frac{F_j}{||F_j||} > . \qquad (10)$$

Then, the weighted average version of F is $\tilde{F} = F * S \in \mathbb{R}^{C\times HW}$ via the matrix multiplication. Finally, we concatenate F and $\tilde{F}$, and apply a 1×1 convolutional layer to preserve the original channel number of F.

The training objective $\mathcal{L}_G$ of Net_G is similar with $\mathcal{L}_L$ of Net_L, only replacing $\mathbf{I}^L_{out}$ with $\mathbf{I}^G_{out}$ in the corresponding locations of $\mathcal{L}_L$.

To this end, our proposed inpainting network LGNet is trained in an "end-to-end" manner, and the final training loss is the summation of losses of three sub-networks and a discriminator, *i.e.*, $\mathcal{L}_C + \mathcal{L}_L + \mathcal{L}_G + \mathcal{L}_D$.

IV. Experiments

In this section, we first explain our experimental settings, the datasets, competing methods, and implementation details. Then, we evaluate and analyze our methods via comparison experiments and ablation studies. We finally show several real-world applications.

A. Experimental Settings

Datasets. We conduct experiments on three public datasets, which are commonly used to evaluate image inpainting tasks.

- CelebA-HQ dataset [34]: The high-quality version of the CelebA [54] consists of 30,000 face images. We randomly select 27,000 for training and the remaining 3,000 for testing.
- Places2 dataset [35]: A large-scale scene recognition dataset. We select 20 categories to construct the inpainting dataset. Specifically, 2,000 images are randomly selected for each category from the training set of Places2, in total, the training set includes 40,000 images. All images in the test set of these 20 categories (in total, 2,000 images) are used as our test set.
- Paris StreetView dataset [36]: This dataset consists of street-level imagery. Following the original setting, we use 14,900 images as the training set and 100 images as the test set.

TABLE I
QUANTITATIVE COMPARISONS OF OUR METHOD WITH FIVE ADVANCED INPAINTING METHODS ON CELEBA-HQ DATASET. ‡ HIGHER IS BETTER. † LOWER IS BETTER. THE BEST TWO SCORES ARE INDICATED BY RED AND BLUE FONTS, RESPECTIVELY.

	Masks	1-10%	10-20%	20-30%	30-40%	40-50%	50-60%
$\ell_1(\%)$ †	PEN	0.80	2.15	3.88	5.83	8.02	11.77
	GConv	0.65	1.81	3.41	5.33	7.53	12.05
	MEDFE	1.02	2.15	3.68	5.51	7.65	11.67
	RFR	1.59	2.47	3.58	4.90	6.44	9.47
	MADF	0.47	1.30	2.40	3.72	5.26	8.43
	Ours	0.46	1.28	2.38	3.72	5.27	8.38
PSNR ‡	PEN	35.34	29.76	26.79	24.70	23.06	20.85
	GConv	37.14	31.02	27.57	25.03	23.10	20.22
	MEDFE	36.13	30.97	27.75	25.36	23.47	20.85
	RFR	36.39	31.87	29.07	26.87	25.09	22.51
	MADF	39.68	33.77	30.42	27.95	25.99	23.07
	Ours	40.04	33.99	30.54	27.99	26.01	23.12
SSIM ‡	PEN	0.988	0.965	0.933	0.894	0.849	0.764
	GConv	0.991	0.971	0.941	0.902	0.856	0.750
	MEDFE	0.990	0.971	0.943	0.908	0.865	0.775
	RFR	0.991	0.976	0.957	0.932	0.902	0.834
	MADF	0.995	0.984	0.967	0.945	0.917	0.848
	Ours	0.995	0.985	0.968	0.945	0.917	0.849
FID †	PEN	1.41	4.19	8.38	12.68	18.73	23.38
	GConv	0.78	2.05	3.93	5.86	8.64	12.75
	MEDFE	0.84	2.06	3.71	5.22	7.12	10.07
	RFR	0.86	1.68	2.67	3.77	5.21	7.60
	MADF	0.52	1.55	3.28	5.43	8.35	13.54
	Ours	0.39	1.06	2.08	3.16	4.61	7.07
LPIPS †	PEN	0.020	0.053	0.092	0.134	0.180	0.240
	GConv	0.012	0.034	0.061	0.091	0.125	0.181
	MEDFE	0.014	0.032	0.055	0.080	0.101	0.156
	RFR	0.015	0.028	0.042	0.060	0.081	0.118
	MADF	0.009	0.025	0.048	0.077	0.109	0.168
	Ours	0.006	0.017	0.031	0.048	0.069	0.108

To train the networks, we construct the irregular masks on the basis of QD-IMD (quick draw irregular mask dataset) [55] with several simple operations like [18]. Following the existing methods, we use the irregular mask data shared by Liu *et al.* [14] as the testing masks to evaluate the trained models. The irregular mask data contains 6 categories with different hole ratios, *i.e.*, $(0.01, 0.1]$, $(0.1, 0.2]$, $(0.2, 0.3]$, $\cdots$, $(0.5, 0.6]$. Each category has 2,000 masks. In [14], these masks are generated via random dilation, rotation, and cropping from the raw masks obtained by occlusion/dis-occlusion mask estimation method [56] between two consecutive frames of videos.

Comparison methods. In this work, we compare our method with five state-of-the-art inpainting methods, which are summarized as follows:

- PEN [15]: A pyramid-context encoder network to fill the holes by progressively learning region affinity with attention.
- GConv [20]: A coarse-to-fine generative network, which is an enhanced version of their previous work [19] by introducing the gated convolution.
- MEDFE [17]: A mutual encoder-decoder CNN with feature equalizations for joint recovery of structure and texture.
- RFR [28]: A progressive inpainting method in the feature space with recurrent feature reasoning and knowledge consistent attention.
- MADF [44]: A U-shaped framework with mask-aware dynamic filtering for image inpainting with a point-wise normalization.

Implementation details. Our LGNet is implemented with PyTorch 1.3.1. As GPU we use a TITAN RTX from NVIDIA®.

TABLE II

QUANTITATIVE COMPARISONS OF OUR METHOD WITH FIVE ADVANCED INPAINTING METHODS ON PLACES2 DATASET. ‡ HIGHER IS BETTER. † LOWER IS BETTER. THE BEST TWO SCORES ARE INDICATED BY RED AND BLUE FONTS, RESPECTIVELY.

	Masks	1-10%	10-20%	20-30%	30-40%	40-50%	50-60%
ℓ_1(%) †	PEN	1.10	2.94	5.18	7.54	10.16	13.76
	GConv	1.16	3.03	5.30	7.66	10.28	14.24
	MEDFE	1.22	2.77	4.84	7.12	9.76	13.93
	RFR	0.83	2.20	3.93	5.83	7.96	11.37
	MADF	0.80	2.18	3.96	5.91	8.10	11.68
	Ours	0.68	1.89	3.51	5.33	7.41	10.86
PSNR ‡	PEN	33.42	27.90	25.09	23.21	21.74	20.07
	GConv	32.86	27.42	24.65	22.81	21.34	19.53
	MEDFE	34.08	29.05	25.92	23.78	22.07	19.93
	RFR	35.74	30.24	27.24	25.13	23.48	21.33
	MADF	36.17	30.37	27.17	25.00	23.31	21.10
	Ours	37.62	31.61	28.18	25.84	24.05	21.69
SSIM ‡	PEN	0.975	0.927	0.867	0.801	0.727	0.619
	GConv	0.968	0.917	0.856	0.792	0.722	0.610
	MEDFE	0.978	0.941	0.888	0.825	0.752	0.630
	RFR	0.983	0.952	0.911	0.862	0.805	0.699
	MADF	0.984	0.953	0.910	0.859	0.800	0.690
	Ours	0.988	0.963	0.925	0.878	0.823	0.714
FID †	PEN	4.60	11.65	20.78	31.12	45.72	60.43
	GConv	5.17	11.70	18.53	25.76	34.60	42.29
	MEDFE	3.59	8.76	15.12	22.15	30.43	40.72
	RFR	2.62	5.99	9.47	12.90	16.62	22.13
	MADF	2.15	5.58	9.20	13.08	17.36	24.42
	Ours	1.97	5.25	8.90	13.02	17.60	25.99
LPIPS †	PEN	0.035	0.093	0.160	0.226	0.295	0.365
	GConv	0.037	0.086	0.134	0.180	0.229	0.298
	MEDFE	0.028	0.063	0.105	0.150	0.201	0.268
	RFR	0.021	0.047	0.074	0.106	0.142	0.201
	MADF	0.014	0.038	0.068	0.102	0.141	0.209
	Ours	0.014	0.035	0.064	0.096	0.132	0.198

TABLE III

QUANTITATIVE COMPARISONS OF OUR METHOD WITH FIVE ADVANCED INPAINTING METHODS ON PARIS STREETVIEW DATASET. ‡ HIGHER IS BETTER. † LOWER IS BETTER. THE BEST TWO SCORES ARE INDICATED BY RED AND BLUE FONTS, RESPECTIVELY.

	Masks	1-10%	10-20%	20-30%	30-40%	40-50%	50-60%
ℓ_1(%) †	PEN	0.97	2.58	4.65	6.84	9.35	13.00
	GConv	0.93	2.55	4.67	6.99	9.58	14.19
	MEDFE	1.15	2.46	4.24	6.25	8.63	12.73
	RFR	0.71	1.88	3.38	5.04	6.95	10.28
	MADF	0.64	1.73	3.19	4.86	6.79	10.33
	Ours	0.58	1.59	2.97	4.57	6.44	9.88
PSNR ‡	PEN	34.25	28.97	26.03	24.12	22.56	20.72
	GConv	34.72	28.95	25.73	23.62	21.95	19.59
	MEDFE	35.12	30.25	27.08	24.91	23.12	20.76
	RFR	36.81	31.43	28.39	26.30	24.60	22.27
	MADF	37.64	31.99	28.71	26.44	24.65	22.14
	Ours	38.52	32.77	29.38	27.01	25.15	22.56
SSIM ‡	PEN	0.979	0.939	0.884	0.821	0.745	0.624
	GConv	0.980	0.940	0.885	0.825	0.757	0.629
	MEDFE	0.984	0.954	0.909	0.854	0.787	0.660
	RFR	0.987	0.962	0.928	0.886	0.836	0.733
	MADF	0.989	0.966	0.933	0.892	0.841	0.732
	Ours	0.991	0.971	0.940	0.900	0.851	0.742
FID †	PEN	9.63	25.71	46.52	67.88	91.65	117.94
	GConv	7.84	20.27	34.50	46.92	59.73	75.11
	MEDFE	6.58	16.20	28.95	41.80	56.02	76.17
	RFR	5.15	12.83	21.73	29.98	38.73	51.41
	MADF	4.00	10.98	19.99	29.53	40.06	58.04
	Ours	3.78	10.53	19.39	28.81	39.58	58.74
LPIPS †	PEN	0.025	0.067	0.119	0.174	0.237	0.324
	GConv	0.024	0.063	0.108	0.152	0.199	0.271
	MEDFE	0.019	0.046	0.083	0.122	0.169	0.242
	RFR	0.016	0.040	0.067	0.096	0.130	0.188
	MADF	0.011	0.032	0.059	0.090	0.126	0.195
	Ours	0.011	0.030	0.055	0.085	0.120	0.187

The Adam optimizer [57] with a minibatch size of 4 is used to train our network, where $\beta_1 = 0.5$ and $\beta_2 = 0.999$. For the learning rate schedule, we set its initial value as 0.0002 for the first 100 epochs and linearly decay it to zero in the next 100 epochs. In our experiments, all the images and masks are of size of 256×256.

B. Comparisons with State-of-the-art Methods

We quantitatively and qualitatively compare our method with five representative state-of-the-art image inpainting methods: PEN [15], GConv [20], MEDFE [17], RFR [28], and MADF [44]. We also analyze and compare the computational complexity of these methods.

Quantitative comparisons. For the evaluation metrics, we adopt several common metrics in the image inpainting task: ℓ_1 error, PSNR (peak signal-to-noise ratio), SSIM (the structural similarity index) [58], FID (Fréchet inception distance) [59], and LPIPS (learned perceptual image patch similarity) [60]. The first three metrics are based on the low-level pixel values, while the last two metrics are related to the high-level visual perception. For LPIPS, we use the latest version (*i.e.*, V0.1) [61]. From the Table I-III, it is obvious that our proposed method has the best performance among all the inpainting methods. Only for large mask on Places2 and Paris StreetView dataset (hole ratio is larger than 30%), the FID of our method is not the best, but competitive. A possible reason is that the progressive strategy and corresponding multi-level attention fusion in RFR are slightly better for large missing regions of complex natural scene.

Qualitative comparisons. Fig. 4 illustrates the visual results of the six inpainting methods. Three groups (each group includes three or four rows) separately correspond to CelebA-HQ [34], Places2 [35], and Paris StreetView [36]. As shown in Fig. 4, we observe that PEN and GConv have relatively poor visual quality, which is also consistent with the quantitative results (see Table I-III). Comparing face completion results, our method can better repair the facial features, such as eyes, nose, and mouth. Compared with GConv and RFR, our attention computation in the global refinement network is more robust and stable based on the relatively good completion result of the local refinement network. In addition, compared with existing inpainting methods, our method can better restore the structures and details. For instance, the structure of the iron grating (in the eighth row) and the windows (in the last row) are successfully restored. Moreover, our method achieves a clear and sharp boundary (the last image of the fifth row), and recovers the flower texture (the last image of the seventh row) as well.

Except for the irregular masks, we also compare the inpainting performance of our method with other competitors on the regular masks, *e.g.*, square and circles. The corresponding results are shown in Fig. 5. The hole size of square mask is 128×128. The radii of the four circles are 30, 30, 40, and 40, respectively. It can be seen that our method still has the

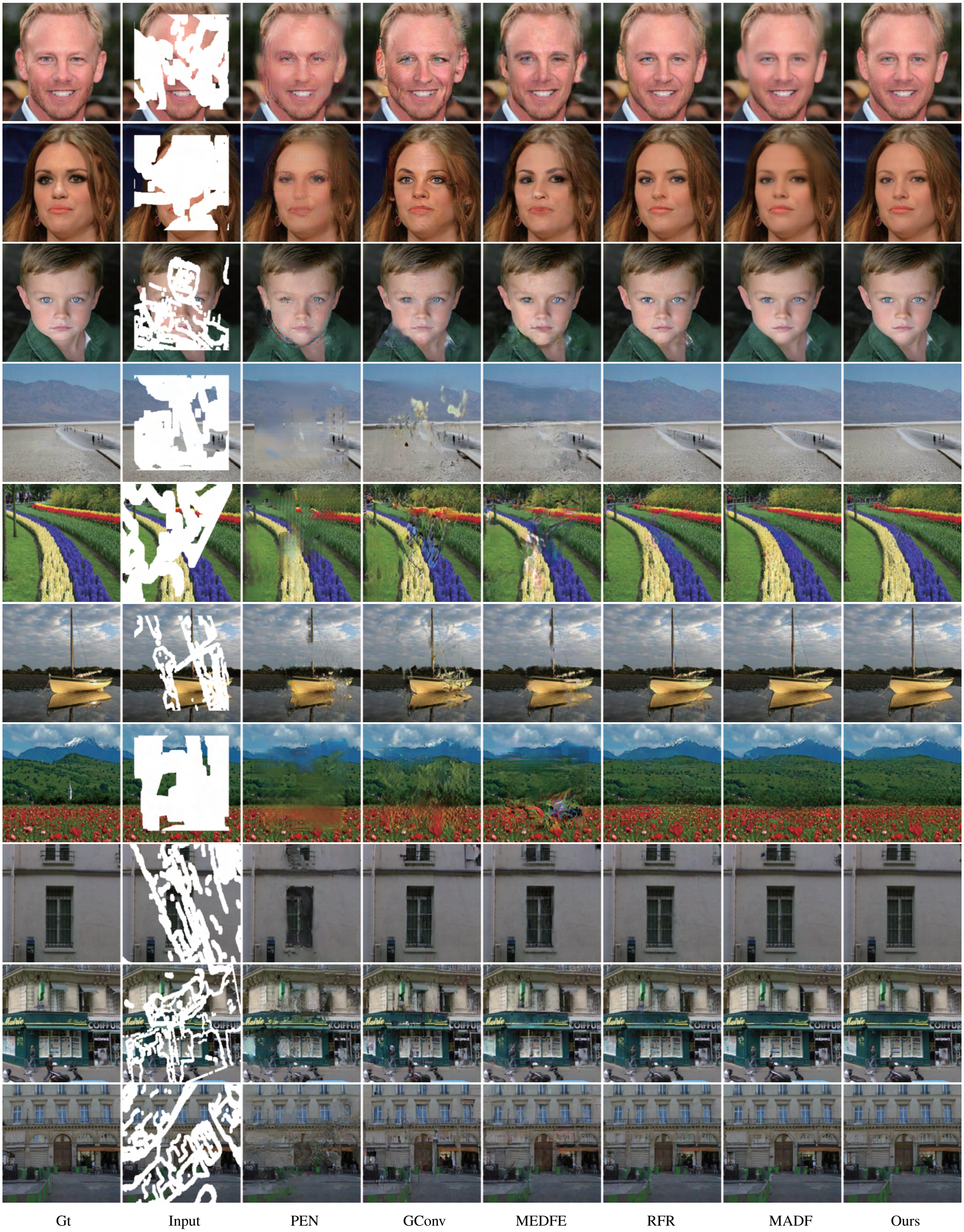


Fig. 4. Qualitative comparisons of our method with PEN, GConv, MEDFE, RFR, and MADF on three datasets with irregular masks. From top to bottom: CelebA-HQ, Places2, and Paris StreetView, respectively. These irregular masks are shared by [14], and we illustrate the inpainted results with different masks just for diverse comparisons like other existing inpainting methods [20], [28], [44]. [Best view with zoom-in.]

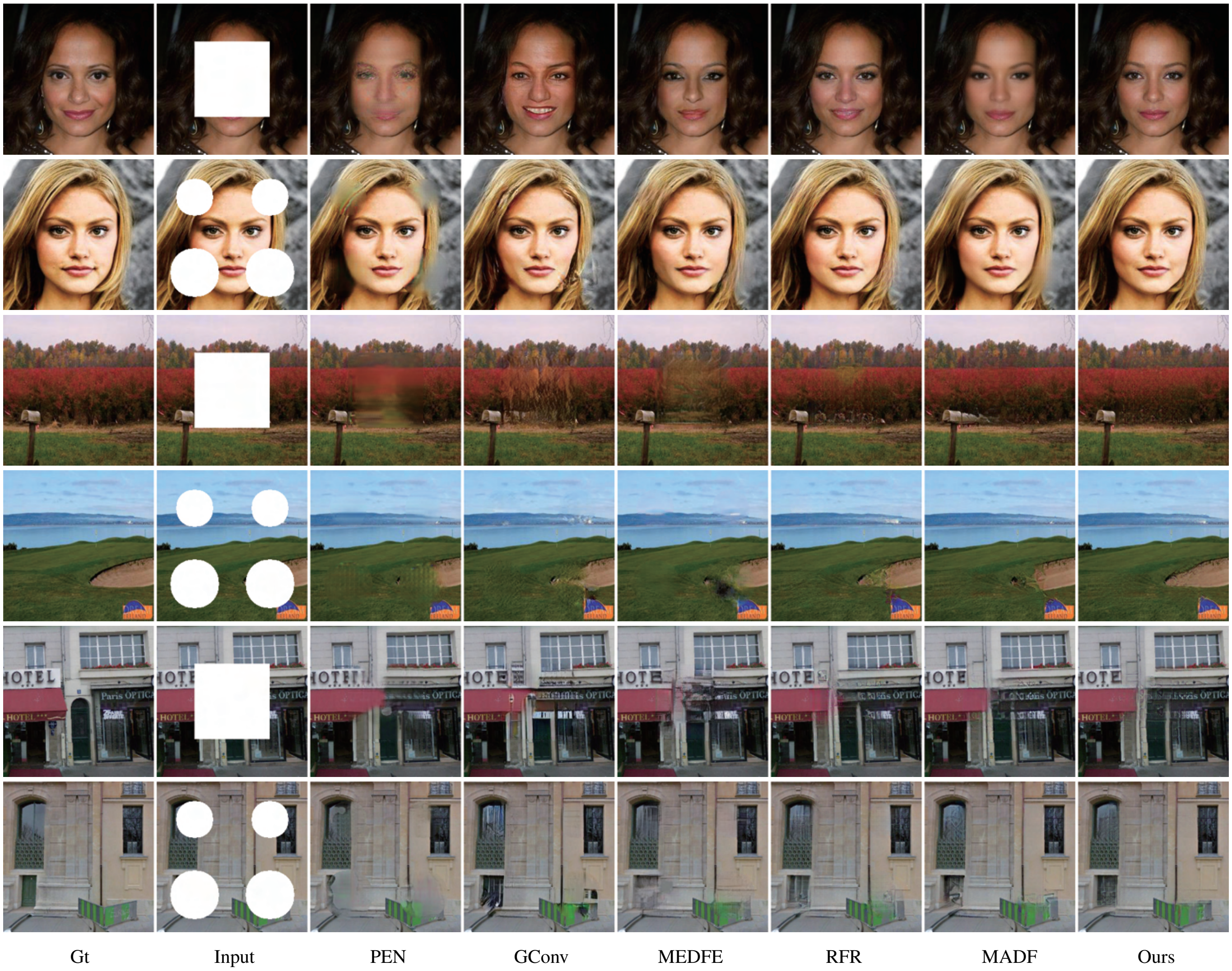


Fig. 5. Qualitative comparisons of our method with PEN, GConv, MEDFE, RFR, and MADF on three datasets with regular masks. From top to bottom: CelebA-HQ, Places2, and Paris StreetView, respectively. [Best view with zoom-in.]

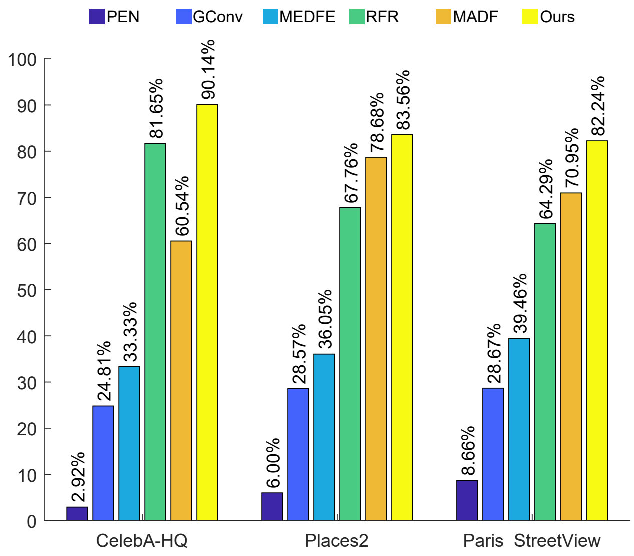


Fig. 6. The statistical results of user study. The value at the top of the bar indicates the percentage of being selected as the more natural one.

superior performance. In the first row, our method recovers the more realistic face; In the fourth row, our network restores more natural structure and details of the scene.

User study. The evaluation metrics are not strictly consistent with the human perception. To further compare the visual quality of our method with other five advanced image inpainting methods, we also conduct a user study. We randomly select 16 inpainted images from each of three datasets (in total 48 images). Then, we invite 27 volunteers for choosing the more natural one from two images generated by different methods without showing the mask and ground truth image. In the end, we collect 1,296 votes. The corresponding statistical results are shown in Fig. 6. We can find that our method has the highest probability of being selected.

Computational complexity analysis. In this part, we evaluate the computational complexity of LGNet and select FLOPs, the number of parameters, and inference time (only on CPU and GPU) as statistics. Inference time (Infer. time) is the time of a forward pass of networks. We count the inference time only on the CPU (Intel® Xeon® Processor E5-

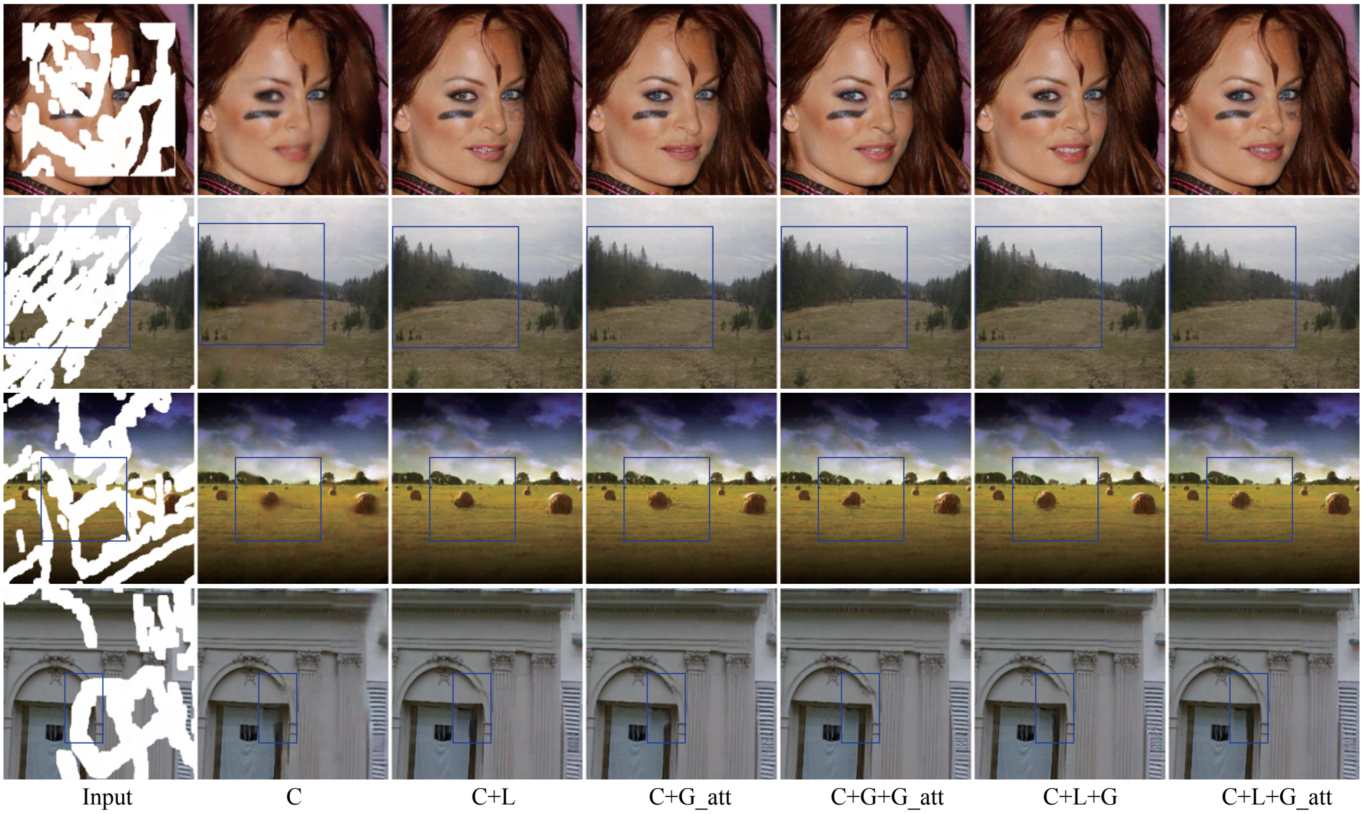


Fig. 7. Qualitative comparisons of our network LGNet with its different variants. The meaning of symbols is the same as in Table VI. [Best view with zoom-in.]

TABLE IV
MODEL COMPUTATIONAL COMPLEXITY STATISTICS.

Model	FLOPs	#Parameter	Infer. time (CPU)	Infer. time (GPU)
PEN [15]	48.07 G	10.23 M	418.16 ms	39.78 ms
GConv [20]	55.57 G	4.05 M	361.92 ms	13.98 ms
MEDFE [17]	137.93 G	130.32 M	946.82 ms	113.91 ms
RFR [28]	206.11 G	30.59 M	910.33 ms	28.95 ms
MADF [44]	51.77 G	85.14 M	925.65 ms	15.59 ms
LGNet	69.62 G	115.00 M	589.82 ms	13.59 ms
LGNet_share	69.62 G	60.58 M	589.82 ms	13.59 ms
LGNet_light	53.73 G	67.10 M	515.28 ms	12.66 ms

TABLE V
NUMERICAL RESULTS OF LGNET, LGNET_LIGHT, AND LGNET_SHARE ON PARIS STREETVIEW DATASET. ‡ HIGHER IS BETTER. † LOWER IS BETTER.

	Masks	1-10%	10-20%	20-30%	30-40%	40-50%	50-60%
$\ell_1(\%)$ †	LGNet	0.58	1.59	2.97	4.57	6.44	9.88
	LGNet_light	0.57	1.57	2.95	4.54	6.41	9.88
	LGNet_share	0.59	1.62	3.02	4.66	6.59	10.26
PSNR ‡	LGNet	38.52	32.77	29.38	27.01	25.15	22.56
	LGNet_light	38.59	32.81	29.38	26.99	25.11	22.52
	LGNet_share	38.30	32.54	29.16	26.77	24.88	22.19
SSIM ‡	LGNet	0.991	0.971	0.940	0.900	0.851	0.742
	LGNet_light	0.991	0.971	0.940	0.900	0.850	0.741
	LGNet_share	0.990	0.970	0.938	0.896	0.843	0.725
FID †	LGNet	3.78	10.53	19.39	28.81	39.58	58.74
	LGNet_light	3.83	10.50	19.44	28.80	39.84	59.78
	LGNet_share	3.90	11.01	20.54	30.93	42.59	64.44
LPIPS †	LGNet	0.011	0.030	0.055	0.085	0.120	0.187
	LGNet_light	0.011	0.030	0.056	0.086	0.121	0.189
	LGNet_share	0.011	0.031	0.058	0.090	0.127	0.200

2690 v4) and the GPU (NVIDIA® TITAN RTX), respectively. The corresponding results are reported in Table IV. Because our LGNet adopts a coarse-to-fine strategy, the number of parameters of our LGNet are relatively larger. However, the FLOPs of LGNet is comparable to other methods and thus LGNet has competitive inference time on CPU, especially compared to RFR and MADF that have better inpainting performance among the existing methods. Importantly, LGNet has the fastest inference time on GPU. The reason is that our network only has simple convolutional operation and self-attention operation (essentially the matrix multiplication), which are GPU-friendly.

Furthermore, we evaluate two simplified versions of our LGNet: (1) LGNet_share is a weight sharing version of LGNet, where all the weights of Net_C are the same as Net_G since they have almost same architectures. (2) LGNet_light is a light version of LGNet, where we simply change the base channel of Net_C and Net_G from 64 to 48. In Net_C and Net_G, the number of channel of feature maps doubles from a base number of 64, *i.e.*, 64, 128, 256, etc. LGNet_share can significantly decrease the number of parameters of LGNet (from 115.00 M to 60.58 M). Because the inference process of LGNet_share and LGNet is same, *i.e.*, the sequence is Net_C, Net_L, and Net_G, “FLOPs” and “Infer. time (CPU, GPU)” are also same. For LGNet_light, “#Parameter” and “FLOPs” of Net_C and Net_G both decline due to the decrease

TABLE VI
THE FID COMPARISONS OF SIX DIFFERENT NETWORK ARCHITECTURES ON THREE DATASETS. "C" STANDS FOR COARSE INPAINTING NETWORK; "L" STANDS FOR LOCAL REFINEMENT NETWORK; "G" STANDS FOR GLOBAL REFINEMENT NETWORK; "G_ATT" STANDS FOR ATTENTION-BASED GLOBAL REFINEMENT NETWORK. LOWER IS BETTER.

	Masks	C	C+L	C+G_att	C+G+G_att	C+L+G	C+L+G_att
CelebA-HQ	1-10%	0.67	0.40	0.46	0.44	0.40	**0.39**
	10-20%	1.88	1.12	1.23	1.16	1.08	**1.06**
	20-30%	3.82	2.27	2.38	2.29	2.12	**2.08**
	30-40%	6.10	3.53	3.55	3.46	3.24	**3.16**
	40-50%	9.37	5.20	5.26	4.97	4.71	**4.61**
	50-60%	15.39	8.04	7.90	7.61	7.35	**7.07**
Places2	1-10%	2.96	2.03	2.29	2.23	1.99	**1.97**
	10-20%	8.58	5.51	6.02	5.78	5.35	**5.25**
	20-30%	16.07	9.39	10.19	9.79	9.08	**8.90**
	30-40%	25.46	13.79	14.86	14.03	13.40	**13.02**
	40-50%	37.32	18.69	20.37	18.93	18.10	**17.60**
	50-60%	54.67	27.21	29.76	27.33	26.68	**25.99**
Paris StreetView	1-10%	5.68	3.99	4.40	4.24	3.88	**3.78**
	10-20%	15.83	11.13	11.79	11.48	10.65	**10.53**
	20-30%	30.33	20.69	21.31	20.67	19.87	**19.39**
	30-40%	46.64	31.16	31.44	30.55	29.32	**28.81**
	40-50%	64.79	42.68	42.97	41.21	40.07	**39.58**
	50-60%	93.75	63.19	61.07	59.20	60.01	**58.74**

of channel number of all their feature maps . Therefore, compared to LGNet, "FLOPs", "#Parameter", and "Infer. time (CPU, GPU)" of LGNet_light are smaller. In addition, Table V reports the quantitative results of LGNet, LGNet_light, and LGNet_share on Paris StreetView dataset. The inpainting performance of LGNet_light and LGNet are very close. LGNet_share is slightly inferior to LGNet and the reason is that the input distributions are different for Net_C (incomplete images with missing regions) and Net_G (complete images with artifacts). In summarize, LGNet_light is a better way to decrease the computational complexity for our framework.

C. Ablation Studies

Network design. We validate and evaluate our network design by comparing different variants of LGNet. The corresponding numerical results are reported in Table VI and visual comparisons are shown in Fig. 7. Comparing the columns of "C", "C+L", "C+L+G", and "C+L+G_att" in Table VI, we observe that the inpainting performance is getting better. This indicates the effectiveness of our proposed local refinement network and attention-based global refinement network. The FID value of column "C+L+G_att" is lower than that of column "C+G+G_att", and this phenomenon is also consistent with the case of "C+L" and "C+G_att" (only except for 50-60% on CelebA-HQ and Paris StreetView). These results support our network design: on the basis of the coarse inpainted results, the shallow deep model with small receptive field (local refinement network) works well, and the cascade of local refinement network and attention-based global refinement network is superior to the "brute-force" concatenation of two deep models with large receptive field. The reason is that networks with different receptive fields can handle different kinds of visual artifacts.

Next, we analyze and compare the visual results of different networks in Fig. 7. The results of "C" have roughly complete structures and the obvious blur. The introduce of local refinement network can enrich the local details, *e.g.*, the regions of blue boxes in the second and third rows. "C+G_att"can repair some regions according to the global information, such as the left eye in the first row and lawn in the blue box of the second row, meanwhile, "C+G+G_att" can further eliminate some visual artifacts. "C+L+G_att" combines the completion capability of "L" and "G_att" to obtain more natural and realistic inpainted results (see last column).

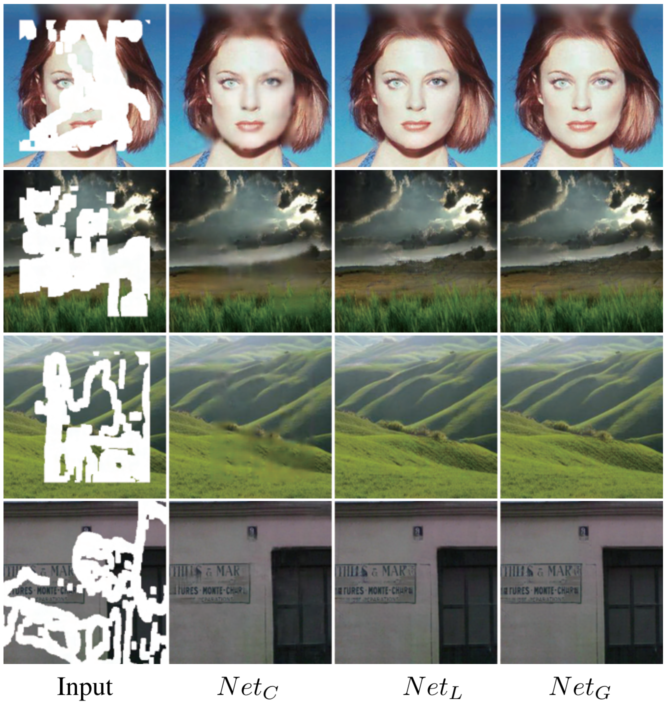


Fig. 8. The outputs of our three sub-networks: coarse inpainting network (Net_C), local refinement network (Net_L), and attention-based global refinement network (Net_G).

Moreover, we analyze our LGNet by presenting and comparing the outputs of three sub-networks as shown in Fig. 8. It is obvious that the visual quality of inpainted images are getting better (2-4 columns). Take the first row as an example, Net_C provides an initial completion result, Net_L removes some local blur in the face using the local information, and Net_G finally recovers a plausible eye using global information with attention. The remaining samples have similar process.

Loss weights. [14] is a pioneer inpainting work, which combines the weighted reconstruction loss, perceptual loss, style loss, and TV loss as the total training objective. The weights of these losses are similar for the following works [17], [28], [44]. In this work, we mainly focus on the network design according to the receptive field. Therefore, we simply adopt the same loss weights as [14], [44]. To further evaluate the inpainting performance of different weight settings, we conduct the ablation experiments on Paris StreetView dataset, and the corresponding results are reported in Table VII. All evaluation metrics go worse when setting the $\mathcal{L}_{valid}$ and $\mathcal{L}_{hole}$ with the same weight, *i.e.*, $\lambda_h = 1$. Pixels in hole regions are unknown, and thus are more difficult to recover than valid regions. Therefore, the larger weight for λ_h is needed. For

TABLE VII
QUANTITATIVE RESULTS OF OUR METHOD WITH DIFFERENT LOSS WEIGHTS ON PARIS STREETVIEW DATASET. ‡ HIGHER IS BETTER. † LOWER IS BETTER. "λ_h" REPRESENTS $\lambda_h = 1$; "λ_{per}" REPRESENTS $\lambda_{per} = 0.1$; AND "λ_{sty}" REPRESENTS $\lambda_{sty} = 180$.

	Masks	1-10%	10-20%	20-30%	30-40%	40-50%	50-60%
$\ell_1(\%)$ †	λ_h	0.63	1.73	3.24	5.00	7.05	10.87
	λ_{per}	0.57	1.57	2.94	4.53	6.40	9.85
	λ_{sty}	0.59	1.62	3.02	4.64	6.53	10.03
	$\lambda_{per}, \lambda_{sty}$	0.58	1.59	2.98	4.60	6.49	10.02
	Ours	0.58	1.59	2.97	4.57	6.44	9.88
PSNR ‡	λ_h	37.83	32.04	28.61	26.22	24.36	21.76
	λ_{per}	38.59	32.80	29.39	27.00	25.13	22.53
	λ_{sty}	38.39	32.64	29.25	26.87	25.02	22.43
	$\lambda_{per}, \lambda_{sty}$	38.49	32.69	29.27	26.87	25.00	22.39
	Ours	38.52	32.77	29.38	27.01	25.15	22.56
SSIM ‡	λ_h	0.989	0.966	0.931	0.886	0.830	0.709
	λ_{per}	0.991	0.971	0.940	0.901	0.851	0.742
	λ_{sty}	0.991	0.970	0.939	0.898	0.847	0.737
	$\lambda_{per}, \lambda_{sty}$	0.991	0.970	0.939	0.898	0.847	0.736
	Ours	0.991	0.971	0.940	0.900	0.851	0.742
FID †	λ_h	3.97	11.10	20.28	30.45	41.81	61.53
	λ_{per}	3.78	10.46	19.26	28.66	39.30	58.80
	λ_{sty}	3.82	10.52	19.42	29.18	39.94	59.66
	$\lambda_{per}, \lambda_{sty}$	3.77	10.46	19.45	28.83	39.44	58.33
	Ours	3.78	10.53	19.39	28.81	39.58	58.74
LPIPS †	λ_h	0.011	0.030	0.056	0.088	0.124	0.193
	λ_{per}	0.011	0.030	0.055	0.085	0.120	0.188
	λ_{sty}	0.011	0.030	0.055	0.085	0.119	0.186
	$\lambda_{per}, \lambda_{sty}$	0.010	0.029	0.054	0.084	0.119	0.186
	Ours	0.011	0.030	0.055	0.085	0.120	0.187

the remaining settings ("λ_{per}"-"Ours" in Table VII), their inpainting performance are very close.

D. Generality of Local and Global Refinement

Based on the observations about the impact of networks with different receptive fields on image inpainting, we propose an inpaiting framework with local and global refinement. It is natural to directly insert our local and global refinement network (LG) in the end of any existing networks, *i.e.*, regarding the existing networks as the first stage, to further improve their inpainting performance. In this subsection, we conduct the experiments on PEN [15] and MEDFE [17]. The training strategy also adopts their original setting [15], [17]. The corresponding numerical results are reported in Table VIII. For all three datasets, we find that our local and global refinement network can stably and consistently improve the metrics of inpainting results (comparing the rows of "X" and "X_LG" in Table VIII). Furthermore, we also illustrate the improvement of visual quality with LG, as shown in Fig. 9. For example, the results of PEN have more natural texture (the second row) and structural windows (the third row); the results of MEDFE have realistic and symmetrical face (the fourth row) and complete telegraph pole (the fifth row).

E. Real-World Applications

In this subsection, we apply our method to several real-world applications, including object removal, text editing, and logo removal.

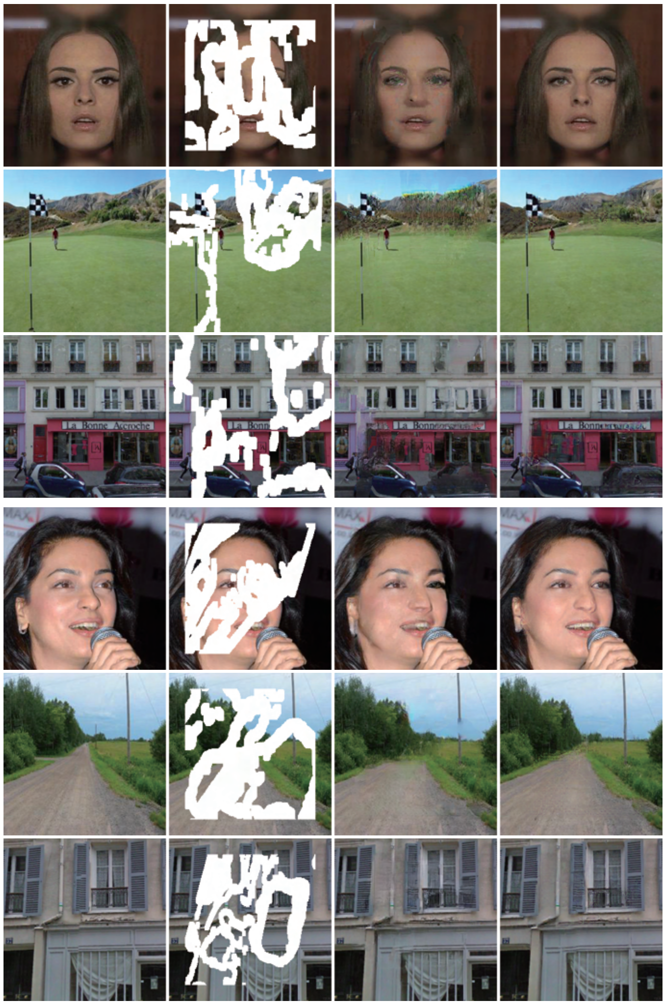


Fig. 9. The first three rows: PEN without/with LG; The remaining three rows: MEDFE without/with LG. Each group (one row) includes Gt, Input, original PEN (or MEDFE), and PEN (or MEDFE) with LG, respectively.

Object removal. Image inpainting technique is often used in image editing tools to removal unwanted objects. To evaluate the performance of our method on object removal, we apply our trained models to remove objects from selected real-world images, and the corresponding results are shown in Fig. 10. The models are separately trained on CelebA-HQ, Places2, and Paris StreetView. Our method can achieve visually realistic results, successfully removing the objects indicated by binary masks. In details, our method can recover the mouth and eyes of human, preserve the original details for the natural scene images, and preserve the consistent building structures.

Text editing. For text editing task, text erasing and text replacement are two common operations. The former can hide the important information, and the latter can be used in text translation. We conduct the experiments on the recent real-world text erasing datasets [62], including 11,040 training samples and 1,080 testing samples. Fig. 11 shows the examples of text translation (the first row) and information hiding (the second row). For text translation, the original text is erased using our inpainting method, and then the translated content is

TABLE VIII
QUANTITATIVE COMPARISONS OF EXISTING METHODS ("X") AND THESE METHODS WITH OUR LG ("X_LG") ON THREE PUBLIC DATASETS. ‡ HIGHER IS BETTER. † LOWER IS BETTER.

	Dataset	CelebA-HQ			Places2			Paris StreetView		
	Masks	10%-20%	30%-40%	50%-60%	10%-20%	30%-40%	50%-60%	10%-20%	30%-40%	50%-60%
$\ell_1(\%)$ †	PEN	2.15	5.83	11.77	2.94	7.54	13.76	2.58	6.84	13.00
	PEN_LG	1.30	3.90	9.04	1.96	5.55	11.31	1.64	4.78	10.55
	MEDFE	2.15	5.51	11.67	2.77	7.12	13.93	2.46	6.25	12.73
	MEDFE_LG	1.66	4.26	9.27	2.29	5.80	11.46	2.08	5.19	10.89
PSNR ‡	PEN	29.76	24.70	20.85	27.90	23.21	20.07	28.97	24.12	20.72
	PEN_LG	33.65	27.50	22.47	31.26	25.49	21.38	32.48	26.64	22.04
	MEDFE	30.97	25.36	20.85	29.05	23.78	19.93	30.25	24.91	20.76
	MEDFE_LG	33.18	27.23	22.40	30.75	25.39	21.36	31.69	26.26	21.84
SSIM ‡	PEN	0.965	0.894	0.764	0.927	0.801	0.619	0.939	0.821	0.624
	PEN_LG	0.983	0.941	0.833	0.961	0.872	0.703	0.969	0.894	0.724
	MEDFE	0.971	0.908	0.775	0.941	0.825	0.630	0.954	0.854	0.660
	MEDFE_LG	0.982	0.937	0.831	0.957	0.869	0.701	0.965	0.887	0.716
FID †	PEN	4.19	12.68	23.38	11.65	31.12	60.43	25.71	67.88	117.94
	PEN_LG	1.16	3.62	8.65	5.70	14.54	29.45	11.70	33.25	70.65
	MEDFE	2.06	5.22	10.07	8.76	22.15	40.72	16.20	41.80	76.17
	MEDFE_LG	1.30	3.82	8.30	6.27	15.15	29.63	13.11	34.15	67.60
LPIPS †	PEN	0.053	0.134	0.240	0.093	0.226	0.365	0.067	0.174	0.324
	PEN_LG	0.018	0.054	0.123	0.038	0.106	0.219	0.033	0.096	0.212
	MEDFE	0.032	0.080	0.156	0.063	0.150	0.268	0.046	0.122	0.242
	MEDFE_LG	0.020	0.056	0.122	0.045	0.111	0.220	0.037	0.102	0.218

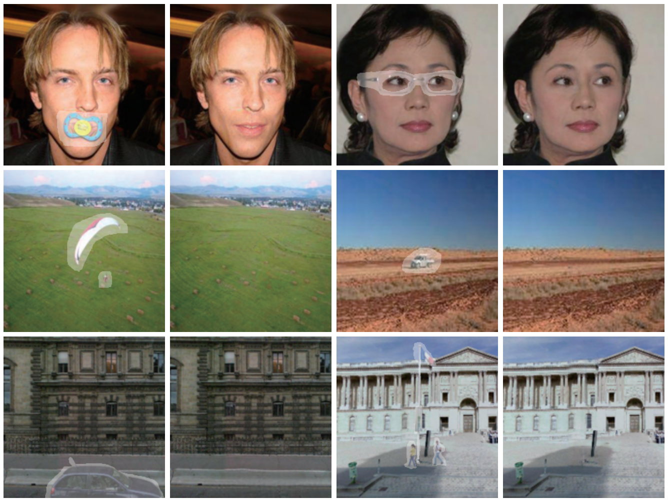


Fig. 10. Examples of object removal on different scenes.

putted on the image. Our method obtains the plausible results.

Logo removal. Automatic logo removal is prevalent in the process of commercial advertising and product packaging design. The core of this technique is to restore the reasonable content in the original logo regions. Through several representative examples (Fig. 12), we show that our method can be used for logo removal application. In this work, we use a public logo detection dataset, QMULOpenLogo [63]. We follow the original train/test splitting, where we train our model on 15,975 images, and test on 8,331 images. We resize all images so that the shorter edge of each resized image has 256 pixels. During training, we randomly crop image patch of 256×256, and we crop center patch in the testing stage. As shown in Fig. 12, our method can recover the reasonable color transition (the first row), preserve the original structure and shape (the second row), and fill the consistent content with background (the third row).

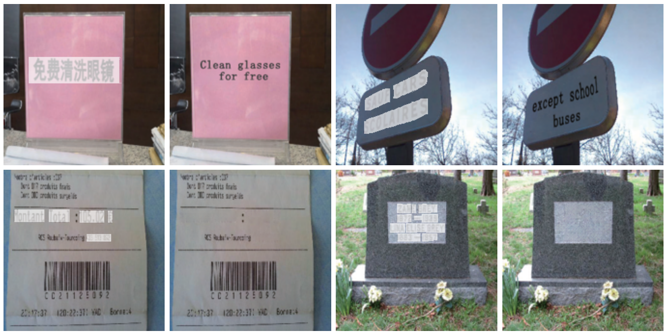


Fig. 11. Examples of text translation and information hiding.

F. Failed Examples

Our method might not restore the correct semantic objects when the specific objects are scarce in the training samples (Fig. 13(left)) and the unmasked regions of object are very small (Fig. 13(right)), especially for the very large missing regions. Fig. 13 illustrates such two unsatisfied cases of our method. Fig. 13(left) is predicted via the model trained on Places2 dataset, which mainly contains natural scene and lacks the training samples of cars. By contrast, when we train our inpainting network on the CelebA-HQ dataset (only including face images), the model works well for inpainting the face image with very large missing regions. Fig. 13(right) cannot restore humans because it is difficult to predict the correct semantic with very small remaining regions of humans. For this challenging scenario, a possible solution is to provide the semantic prior or a reference image.

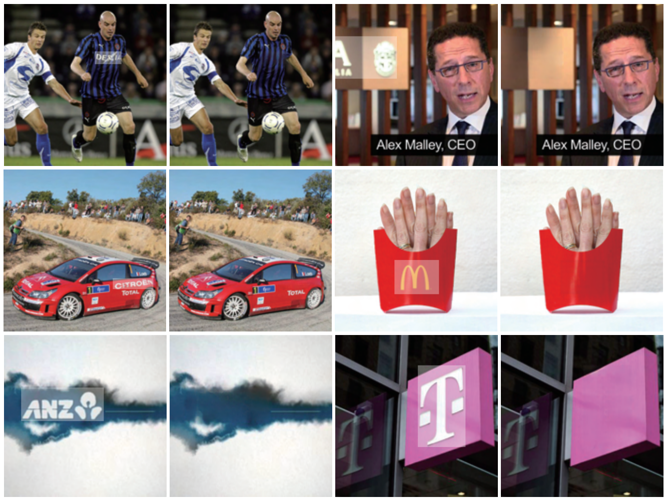


Fig. 12. Examples of logo removal.

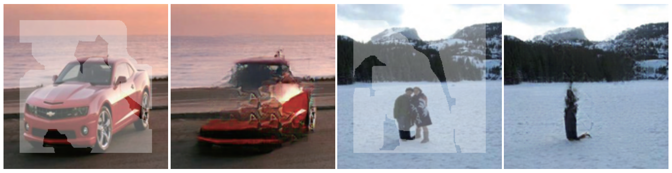


Fig. 13. Two failed examples of our method.

V. Conclusions and Future Work

From the perspective of receptive field, we proposed a three-stage generative network for image inpainting. A coarse inpainting network with large receptive field is applied to complete the whole structure and partial texture details. A local refinement network with small receptive field is designed to eliminate the visual artifacts strongly related to its local region and prevent the negative effect from far and failed filling contents. An attention-based global refinement network with large receptive field is proposed to further improve the visual quality of inpainted results using the global information and the more stable attention computation. Extensive results demonstrate the superiority of our proposed method.

In this work, our framework concatenates two sub-networks with small and large receptive fields to handle different types of missing regions and artifacts. Our study implies that it is beneficial to introduce different receptive fields for image inpainting task. In the future, we would like to improve our network architecture by designing a sub-network with diverse receptive fields or bring the local and global refinement as a whole for more efficient storage and computation. In addition, attention has been prevalent in many existing image inpainting methods, however, the attention computation is sensitive to the filled contents and has no appropriate supervision. We would also like to design more accurate and robust attention computation mechanism.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (62102418 and 62172415), the National Key R&D Program of China (2019YFB2204104), the Tencent AI Lab Rhino-Bird Focused Research Program (No: JR202127), and the Open Project Program of National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University (No.2021SCUVS002).

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proc. ACM SIGGRAPH*, 2000, pp. 417–424.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, “Filling-in by joint interpolation of vector fields and gray levels,” *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, 2001.

[3] D. Tschumperlé and R. Deriche, “Vector-valued image regularization with pdes: a common framework for different applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, 2005.

[4] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “PatchMatch: A randomized correspondence algorithm for structural image editing,” *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[5] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, “Image Melding: combining inconsistent images using patch-based synthesis,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 1–10, 2012.

[6] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Image completion using planar structure guidance,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 33, no. 4, pp. 1–10, 2014.

[7] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray, “Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions,” *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1809–1824, 2015.

[8] D. Ding, S. Ram, and J. J. Rodríguez, “Image inpainting using nonlocal texture matching and nonlinear filtering,” *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1705–1719, 2019.

[9] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.

[10] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Adv. Neural Inform. Process. Syst.*, 2014, pp. 2672–2680.

[12] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2536–2544.

[13] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 36, no. 4, pp. 1–14, 2017.

[14] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *Eur. Conf. Comput. Vis.*, 2018, pp. 85–100.

[15] Y. Zeng, J. Fu, H. Chao, and B. Guo, “Learning pyramid-context encoder network for high-quality image inpainting,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1486–1494.

[16] C. Xie, S. Liu, C. Li, M.-M. Cheng, W. Zuo, X. Liu, S. Wen, and E. Ding, “Image inpainting with learnable bidirectional attention maps,” in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8858–8867.

[17] H. Liu, B. Jiang, Y. Song, W. Huang, and C. Yang, “Rethinking image inpainting via a mutual encoder-decoder with feature equalizations,” in *Eur. Conf. Comput. Vis.*, 2020.

[18] R. Zhang, W. Quan, B. Wu, Z. Li, and D.-M. Yan, “Pixel-wise dense detector for image inpainting,” *Comput. Graph. Forum*, vol. 39, no. 7, pp. 471–482, 2020.

[19] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5505–5514.

[20] ——, “Free-form image inpainting with gated convolution,” in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4471–4480.

[21] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, “EdgeConnect: Structure guided image inpainting using edge prediction,” in *Int. Conf. Comput. Vis. Worksh.*, 2019.

[22] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, “Structureflow: Image inpainting via structure-aware appearance flow,” in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 181–190.

[23] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 7508–7517.

[24] Y. Wang, Y.-C. Chen, X. Tao, and J. Jia, "Vcnet: A robust approach to blind image inpainting," in *Eur. Conf. Comput. Vis.*, 2020.

[25] H. Zhang, Z. Hu, C. Luo, W. Zuo, and M. Wang, "Semantic image inpainting with progressive generative networks," in *ACM Int. Conf. Multimedia*, 2018, p. 1939–1947.

[26] Z. Guo, Z. Chen, T. Yu, J. Chen, and S. Liu, "Progressive image inpainting with full-resolution residual network," in *ACM Int. Conf. Multimedia*, 2019, p. 2496–2504.

[27] J. Li, F. He, L. Zhang, B. Du, and D. Tao, "Progressive reconstruction of visual structure for image inpainting," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5961–5970.

[28] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 7757–7765.

[29] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," in *Eur. Conf. Comput. Vis.*, 2020.

[30] H. Liu, B. Jiang, Y. Xiao, and C. Yang, "Coherent semantic attention for image inpainting," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4170–4179.

[31] J. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *Adv. Neural Inform. Process. Syst.*, 2014, p. 1601–1609.

[32] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.

[33] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Adv. Neural Inform. Process. Syst.*, 2016, p. 4905–4913.

[34] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Int. Conf. Learn. Represent.*, 2018.

[35] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, 2017.

[36] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 101:1–101:9, 2012.

[37] C. Guillemot and O. L. Meur, "Image inpainting : Overview and recent advances," *IEEE Sign. Process. Magazine*, vol. 31, no. 1, pp. 127–144, 2014.

[38] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, "Image Inpainting: A Review," *Neural Process. Letters*, vol. 51, no. 2, pp. 2007–2028, 2020.

[39] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C.-C. J. Kuo, "Contextual-based image inpainting: Infer, match, and translate," in *Eur. Conf. Comput. Vis.*, 2018, pp. 3–18.

[40] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, "Shift-net: Image inpainting via deep feature rearrangement," in *Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[41] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. Satoh, "Guidance and evaluation: Semantic-aware image inpainting for mixed scenes," in *Eur. Conf. Comput. Vis.*, 2020.

[42] Z. Hui, J. Li, X. Wang, and X. Gao, "Image fine-grained inpainting," *arXiv preprint arXiv:2002.02609*, 2020.

[43] N. Wang, Y. Zhang, and L. Zhang, "Dynamic selection network for image inpainting," *IEEE Trans. Image Process.*, vol. 30, pp. 1784–1798, 2021.

[44] M. Zhu, D. He, X. Li, C. Li, F. Li, X. Liu, E. Ding, and Z. Zhang, "Image inpainting by end-to-end cascaded refinement with mask awareness," *IEEE Trans. Image Process.*, vol. 30, pp. 4855–4866, 2021.

[45] H. Wu, J. Zhou, and Y. Li, "Deep generative model for image inpainting with local binary pattern learning and spatial attention," *arXiv preprint arXiv:2009.01031*, 2020.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, 1997.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[48] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Int. Conf. Learn. Represent.*, 2018.

[49] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2813–2821.

[50] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[51] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2414–2423.

[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[54] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.

[55] K. Iskakov, "Qd-imd: Quick draw irregular mask dataset," https://github.com/karfly/qd-imd. Accessed 29 July 2021.

[56] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *Eur. Conf. Comput. Vis.*, 2010, pp. 438–451.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Represent.*, 2015.

[58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[59] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Adv. Neural Inform. Process. Syst.*, 2017, pp. 6626–6637.

[60] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 586–595.

[61] R. Zhang, "Perceptualsimilarity," 2018, https://github.com/richzhang/PerceptualSimilarity. Accessed 29 July 2021.

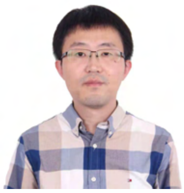
[62] X. Bian, C. Wang, W. Quan, J. Ye, X. Zhang, and D.-M. Yan, "Scene text removal via cascaded text stroke detection and erasing," *Computational Visual Media*, 2021.

[63] H. Su, X. Zhu, and S. Gong, "Open logo detection challenge," in *Brit. Mach. Vis. Conf.*, 2018.

Weize Quan received his Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (China) and Université Grenoble Alpes (France) in 2020, and his Bachelor's degree from Wuhan University of Technology in 2014. He is currently an assistant professor at the National Laboratory of Pattern Recognition of the Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics and image processing.

Ruisong Zhang is currently pursuing the Ph.D. degree in the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. He received the B.S. degree of information security from Xidian University in 2019. His research interests include image processing and image forensics.

Yong Zhang received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences in 2018. From 2015 to 2017, he was a Visiting Scholar with the Rensselaer Polytechnic Institute. He is currently with the Tencent AI Lab. His research interests include computer vision and machine learning.



Zhifeng Li (M'06-SM'11) is currently a top-tier principal research scientist with Tencent. He received the Ph. D. degree from the Chinese University of Hong Kong in 2006. After that, he was a postdoctoral fellow at the Chinese University of Hong Kong and Michigan State University for several years. Before joining Tencent, he was a full professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include deep learning, computer vision and pattern recognition, and face detection and recognition. He was one of the 2020 Most Cited Chinese Researchers (Elsevier-Scopus) in computer science and technology. He is currently serving on the Editorial Boards of Pattern Recognition, IEEE Transactions on Circuits and Systems for Video Technology, and Neurocomputing. He is a fellow of British Computer Society (FBCS).

Jue Wang is Distinguished Scientist at Tencent, leading research efforts in intelligent content generation for the gaming and entertainment industry. He is the Director of the Visual Computing Center at Tencent AI lab, a multidisciplinary research hub for Computer Vision, Computer Graphics and HCI. He received his BE and MS from Tsinghua University in Beijing, and his PhD from the University of Washington at Seattle. He was Senior Director at MEGVII Research from 2017 to 2020, and was Principal Scientist at Adobe Research before that. He has published more than 150 research articles in top-tier academic journals and conferences in the areas of Computer Vision, Computer Graphics, Machine Learning and HCI, and holds more than 80 international patents. He is an Associated Editor for IEEE Transactions on Pattern Analysis and Machine Intelligence.

Dong-Ming Yan is a professor in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CAS). He received his Ph.D. degree in computer science from Hong Kong University in 2010, and his master and bachelor degrees in computer science and technology from Tsinghua University in 2005 and 2002, respectively. His research interests include image processing, geometric processing, and visualization.