Painting from Part

Dongsheng Guo[1] Haoru Zhao[1] Yunhao Cheng[1] Haiyong Zheng[1,*] Zhaorui Gu[1] Bing Zheng[1,2]
[1]Underwater Vision Lab (http://ouc.ai), Ocean University of China
[2]Sanya Oceanographic Institution, Ocean University of China

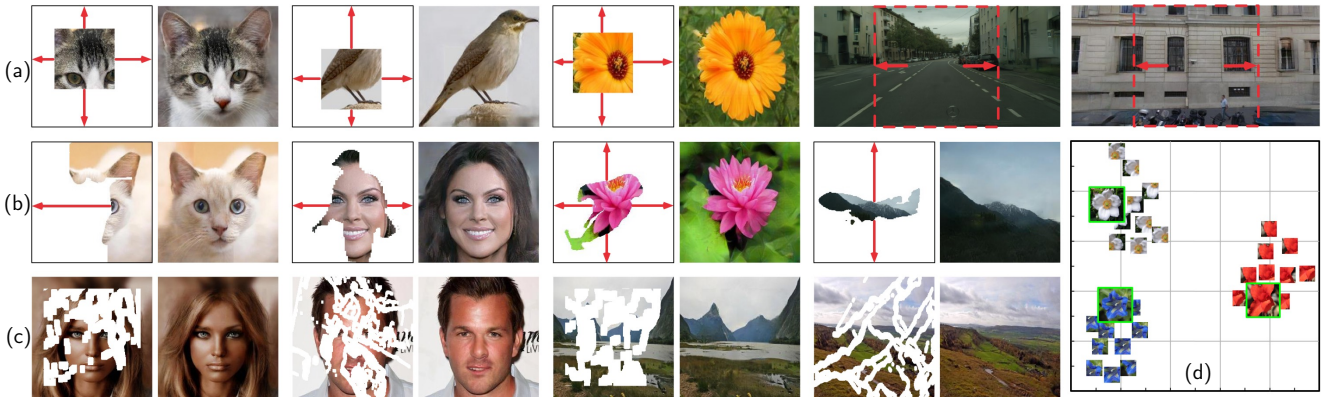

Figure 1. Our results of painting from part on various datasets, considering both object and scene, involving both outpainting and inpainting (left: input, right: output): (a) regular image outpainting, (b) irregular image outpainting, and (c) image inpainting. (d) The t-SNE [30] visualizing embeddings of three flower images and their parts, indicating strong correlations between parts and the whole image.

Abstract

This paper studies the problem of painting the whole image from part of it, namely painting from part *or* part-painting *for short, involving both inpainting and outpainting. To address the challenge of taking full advantage of both information from local domain (part) and knowledge from global domain (dataset), we propose a novel part-painting method according to the observations of relationship between part and whole, which consists of three stages: part-noise restarting, part-feature repainting, and part-patch refining, to paint the whole image by leveraging both feature-level and patch-level part as well as powerful representation ability of generative adversarial network. Extensive ablation studies show efficacy of each stage, and our method achieves state-of-the-art performance on both inpainting and outpainting benchmarks with free-form parts, including our new mask dataset for irregular outpainting. Our code and dataset are available at https://github.com/zhenglab/partpainting.*

*Corresponding author: Haiyong Zheng (zhenghaiyong@ouc.edu.cn).

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61771440 and 41776113, and the Fundamental Research Funds for the Central Universities under Grant No. 202061002.

1. Introduction

Given a part of an image, human have the natural ability to paint the unseen region (*e.g.*, outside or inside the part) [37]. Painting from part, or *part-painting* for short, is the task to paint a whole reasonable and realistic image according to a part of the image, which can be widely used in many computer vision applications such as view expansion [58, 47, 40], texture synthesis [25, 49, 42], image editing [3, 60, 9], and object removal [28, 55]. Due to intrinsic complexity of the part in an image (*e.g.*, diverse shapes and different positions), part-painting is full of challenges.

Specifically, in order to paint a reasonable and realistic whole image from a part, it is indispensable to not only require information from the given part (local domain), but also learn knowledge from other similar images (global domain). However, the multiple properties of the parts lead to fiendish complexity and huge uncertainty of the balance between local domain and global domain for part-painting. Thus, how to make full use of information from local domain (part) and knowledge from global domain (dataset) while keeping a proper balance between them, is essential and crucial for painting from part.

Recent advances in part-painting, *i.e.*, image inpainting [51, 39, 55, 32, 38, 29] and image outpainting [48,

43, 17], mainly feed the part as input into convolutional neural networks (CNNs) and learn from dataset to complete the whole painting. For inside part-painting (image inpainting), the unknown region is usually small and located inside the part, it is able to fill a small number of missing pixels via convolution with surrounding pixels that are coherent with missing ones, thus yielding promising results [35, 53, 54, 28, 50, 27]. But for outside part-painting (image outpainting), the unknown region locates outside the part and is usually large, making this task tends to be a generative problem with more challenges, and recent studies tackle it by first extending the part via feature expansion or reconstruction and then generating the result via adversarial learning [48, 17]. Therefore, current inpainting and outpainting methods are not easy to be applied to each other: the former is hard to paint large reasonable content outside [48, 43], and the latter can not handle free-form cases due to the design of requiring square part input [48, 17]. Moreover, both of previous painting methods mainly "look" the part once in the beginning, which can not take enough advantage of the information from part (*e.g.*, pixels, patches, features) during painting. In this work, we take both inpainting and outpainting with free-form parts into account as a unified part-painting framework.

We tackle part-painting basically relying on the following two observations: (1) both low-level and high-level features of the part have a strong statistical correlation with the whole image features [41, 44], and (2) small patches from the part have a high probability of abundantly recurring in the whole image [15, 61]. Figure 1(d) shows the t-SNE [30] visualizing embeddings of three flower exemplars with the parts and corresponding whole image for each, which indicates that (1) different whole images have relatively independent distributions while the parts are strongly correlated to corresponding whole image, and (2) the parts of every exemplar have very similar visual characteristics to the corresponding whole image.

Therefore, for painting from part, in order to make better use of information from part (local domain), we leverage both feature-level and patch-level information of part (*part-feature* and *part-patch*) during painting; while, to balance the painting guidance between part information (local domain) and dataset knowledge (global domain), we devise a learnable adaptive strategy for both feature-level reconstruction and patch-level fusion; furthermore, we start painting from the noise sampled from local part distribution (*part-noise*), to ensure more reasonable and realistic synthesis via powerful representation of generative adversarial network (GAN). Specifically, we build a novel GAN-based network architecture for part-painting, including three stages in correlation with part-noise, part-feature and part-patch, where, part-noise is sampled from the distribution of part encoding, part-feature is extracted from part in multiple levels and injected into both high-level and low-level synthesis for further repainting, and part-patch is obtained from part mask of repainted whole image then utilized to find and replace the most strongly correlated patch from the unknown region for final refining.

Our contributions include: (1) we propose a new part-painting task, involving both image inpainting and image outpainting from free-form parts, as well as a novel architecture to solve it; (2) we devise three stages, *i.e.*, part-noise restarting, part-feature repainting, and part-patch refining, for guiding and optimizing the part-painting; (3) our method achieves state-of-the-art performance on both inpainting and outpainting benchmarks with free-form parts, including our new built irregular image outpainting dataset.

2. Related Work

2.1. Inpainting and Outpainting

Existing researches for image inpainting and outpainting can be mainly divided into non-learning methods and learning methods. Non-learning image inpainting methods usually filled unknown region by propagating contiguous information or searching and copying similar pixels from known region [2, 6, 4, 3, 10], which might work well for texture synthesis but are difficult to produce semantically meaningful content only relying on the known region. Non-learning image outpainting methods generally obtained the solutions from a pre-constructed dataset through matching and stitching [13, 36, 1, 58, 47, 40], which is not suitable for dealing with complex scenes due to the lack of semantic understanding of images and limited by the used dataset. So non-learning methods of inpainting and outpainting actually try to seek similar information from the part (local domain) and the dataset (global domain) respectively for painting.

Recently, deep CNNs have been developed to learn powerful models to tackle image inpainting and outpainting problems. Learning methods of image inpainting can be categorized into direct and progressive manners. Specifically, some methods attempted to paint a whole image from the visible region in a direct way [19, 54, 52, 28, 55, 27], for instance, Yu *et al*. [54] proposed a contextual attention layer to fill defects with more realistic textures from the visible region, Liu *et al*. [28] and Yu *et al*. [55] designed special convolution to construct CNNs that could fill in irregular/free-form regions, Li *et al*. [27] devised a recurrent feature reasoning network that recurrently infers the hole boundaries of feature maps. Progressive inpainting methods painted the unknown region in multiple stages [51, 38, 32, 26, 18, 12], which usually utilized additional prior information, for example, Xiong *et al*. [51] and Nazeri *et al*. [32] painted unknown based on pre-learning contour/edge knowledge, Ren *et al*. [38] dealt with the problem depending on prior structural information, Dong

et al. [12] painted and edited fashion images with parsing images synthesized in advance. Learning methods of image outpainting extrapolated a regular sub-image to a whole image using GANs to learn global domain knowledge [48, 43, 17], especially, Wang *et al*. [48] first proposed a cGAN-based method to address the issue of feature expansion and context prediction, Teterwak *et al*. [43] introduced semantic conditioning to the discriminator for one-side image extension, Guo *et al*. [17] performed image extrapolation in a spiral growing fashion.

2.2. Generative Adversarial Networks

Since the breakthrough made by Goodfellow *et al*. [16], GAN has become a promising approach for high-quality image synthesis. Recently, more and more works have shown GAN's strong representation ability for learning complex data distributions, especially, StyleGAN [22] proposed an alternative generator architecture that highly improved GAN's generative ability on high quality (high resolution) by combining both style information mapped from a noise of normal distribution and stochastic attributes extracted from a noise of Gaussian distribution, GauGAN [34] synthesized a high-quality photorealistic image from a random noise encoded from a style image guided by the semantic information for spatially-adaptive normalization, BigGAN [5] presented a regularization scheme with a relevant noise sampling technique to boost performance of large-scale GANs. These cutting-edge techniques indicate that GAN has powerful ability to learn the target distribution from a simple distribution (noise), and GAN's ability could be boosted by making the noise being more suitable to the target distribution. In our work, we also leverage GAN's powerful representation ability for painting and encode the part to a distribution of local domain for GAN.

3. Painting from Part

We design a part-painting network architecture, to paint from a free-form part to a whole image. Given a part $\mathbf{P} \in \mathbb{R}^{H\times W\times 3}$, mask $\mathbf{M} \in \mathbb{R}^{H\times W\times 1}$, and random noise $\mathbf{Z} \in \mathbb{R}^{d}$ sampled from a standard normal distribution, the goal of part-painting $F(\cdot)$ is to paint a reasonable and realistic whole image $\mathbf{W}_o \in \mathbb{R}^{H\times W\times 3}$: $\mathbf{W}_o = F(\mathbf{P}, \mathbf{M}, \mathbf{Z})$.

Figure 2 shows the architecture of our method, consisting of a part encoder to extract part-distribution and part-features, a painting generator to paint the whole image from noise, a whole discriminator and a painting discriminator (both are unshown in Figure 2) to distinguish whole result and painting region from corresponding reals respectively. Our painting process is divided into three stages: part-noise restarting, part-feature repainting, and part-patch refining, correspondingly, we first restart painting process from revised normally distributed part-noise, then we repaint the generated features via transferring part-feature statistics in multiple layers, finally we refine the repainted whole result by fusing with most similar part-patches.

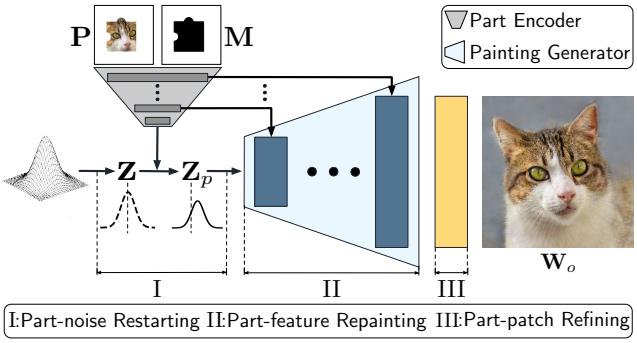

Figure 2. Overview of our network architecture with three stages for painting outside or inside from free-form part.

3.1. Part-noise Restarting

To leverage GAN's powerful representation ability, we propose to start painting from a noise not directly from the part. Not only that, we restart the painting from the part-noise $\mathbf{Z}_p \in \mathbb{R}^d$ following the distribution of part domain. To do this, we revise the noise $\mathbf{Z}$ with parameters $\beta_p \in \mathbb{R}^d$ and $\gamma_p \in \mathbb{R}^d$ learnt from part encoder $F_{enc}(\cdot)$, as follows:

$$\beta_p, \gamma_p = F_{enc}(\mathbf{P}, \mathbf{M}), \quad \mathbf{Z}_p = \gamma_p \odot \mathbf{Z} + \beta_p, \qquad (1)$$

where $\odot$ represents Hadamard product. This is similar to reparameterization trick in VAE [23] building exclusive distribution for each image, while we attempt to boost the painting from a closer distribution.

3.2. Part-feature Repainting

During GAN's painting, we consider making better use of part-features as repainting, mimicking human's "painting by watching". Thus, we design a new Repainting Residual Block (**RRB**), with a core repainting layer, to replace the normal residual block for painting generator, as shown in Figure 3. RRB receives the whole-feature generated from last block of painting generator and part-feature extracted from corresponding layer of part encoder as inputs, and produces repainted whole-feature as output.

Repainting layer plays a key role in repainting, which can be regarded as a special adaptive normalization layer, using one feature to normalize another feature while learning to keep a balance between them. To do this, we design the repainting via part-feature transfer then whole-feature reconstruction, for respectively transferring part-feature to whole-feature then learning to balance them adaptively.

Formally, we denote input feature maps of repainting layer as f_w, which joint generated whole-feature and extracted part-feature, then we use downscaled mask $\mathbf{M}\downarrow$ to separate f_w into part-location feature f_p and exclusion-of-part-location feature f_g. Noting that, part-location feature

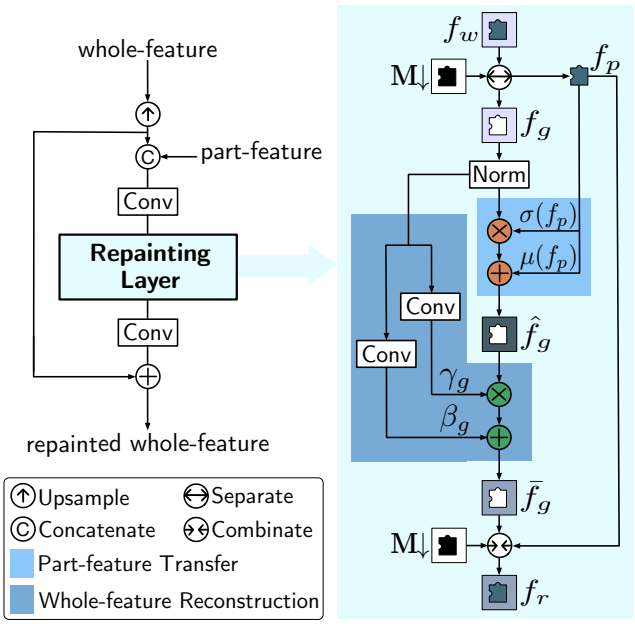

Figure 3. The details of our Repainting Residual Block constituting painting generator, with a core Repainting Layer transfers part-feature statistics to whole feature then reconstructs whole feature.

is extracted from corresponding part encoder layer, so we utilize it to repaint the exclusion-of-part-location feature.

Part-feature Transfer. We calculate mean $\mu(f)$ and standard deviation $\sigma(f)$ of f_p independently for each channel, representing part-feature statistics, then transfer them to normalized f_g channel-wisely by:

$$\hat{f}_g^i = \sigma(f_p^i)\left(\frac{f_g^i - \mu(f_g^i)}{\sigma(f_g^i)}\right) + \mu(f_p^i), \tag{2}$$

where i represents i-th channel and $\hat{f}_g$ is the transferred feature. By doing so, $\hat{f}_g$ will have the same statistics as f_p. Naturally, part and whole should be strongly correlated, but shouldn't be the same statistically (refer to Figure 1(d)). Thus, we adaptively learn to balance pre-transferred feature f_g and post-transferred feature $\hat{f}_g$ via reconstruction.

Whole-feature Reconstruction. We adopt 1×1 convolution to produce affine parameters γ_g and β_g from f_g. Then we reconstruct the transferred feature $\hat{f}_g$ element-wisely for each channel as well:

$$\bar{f}_g^i = \gamma_g^i \odot \hat{f}_g^i + \beta_g^i, \tag{3}$$

where i represents i-th channel and $\bar{f}_g$ is the reconstructed feature. In this way, it can retain valuable whole-feature statistics avoiding being washed away by part-feature transfer. Finally, we combinate $\bar{f}_g$ with part-location feature f_p to obtain the final output f_r of repainting layer.

3.3. Part-patch Refining

According to the internal statistics of a single natural image having recurred small patches abundantly, we further reuse part-feature for refining the painting, via part-patch transmit then whole-patch fusion, to respectively transmit part-patch to whole-patch then learn to fuse transmitted patches with whole-patches adaptively, constituting our Patch Refining Module (*PRM*) shown in Figure 4.

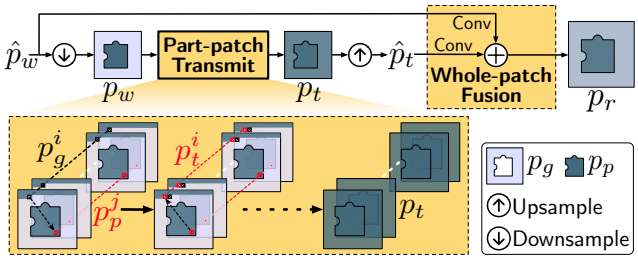

Figure 4. The details of our Patch Refining Module following painting generator, which replaces similar whole patches by matching part patches then fuses them with previous patches.

We first downscale the repainted whole feature maps $\hat{p}_w$ generated by painting generator to p_w, to reduce the computational complexity and to represent patches as pixels. Then we use the part-location pixels p_p (part region in p_w) to replace exclusion-of-part-location pixels p_g (repainted region in p_w), *i.e.*, part-patch transmit.

Part-patch Transmit. We transmit p_p to p_g by finding the most similar pixel sequence p_p^j for p_g^i, where i and j represent spatial locations, and each pixel sequence is composed of pixels at the same location across all channels (see p_g^i and p_p^j in Figure 4). Then we replace p_g^i with p_p^j, via:

$$p_t^i = \arg\max_{p_p^j} \cos(p_p^j, p_g^i), \tag{4}$$

where p_t^i represents transmitted pixel sequence at location i. After replacing all locations in p_g, we get output p_t and upscale p_t to $\hat{p}_t$ which represents transmitted part-patches. By this way, we actually utilize part-patches to refine the repainted result for further making better use of the part.

Whole-patch Fusion. However, the patch refining carried out by finding and replacing is some kind of rough, especially for the cases that unknown region and part region have large differences. Therefore, we finally seek to learn a fusion of transmitted part-patches and whole-patches repainted by painting generator, so that the final painting result will adaptively keep effective information from local domain (the part) and global domain (the dataset). And the learnable fusion can be expressed as:

$$p_r = \omega_1 \star \hat{p}_w + \omega_2 \star \hat{p}_t, \tag{5}$$

where $\star$ indicates convolution, ω_1 and ω_2 are learnable weights, $\hat{p}_w$ and $\hat{p}_t$ represent input whole-patches and transmitted part-patches respectively, and p_r denotes final refined whole patches. At last, we adopt a convolution layer after PRM to output whole part-painting result $\mathbf{W}_o$.

3.4. Loss Design

Our total loss includes KL-Divergence loss, adversarial loss, reconstruction loss, perceptual loss, and style loss.

KL-Divergence Loss. Referring to [23], we adopt a KL-Divergence loss term to maintain similar part-noise distribution: $\mathcal{L}_{kl} = D_{KL}(q(\mathbf{Z}_p \mid \mathbf{P}) \parallel p(\mathbf{Z}))$ where $\mathbf{Z}_p \sim F_{enc}(\mathbf{P}) = q(\mathbf{Z}_p \mid \mathbf{P})$, $\mathbf{Z} \sim \mathbb{N}(0,1) = p(\mathbf{Z})$ and D_{KL} means the Kullback-Leibler divergence.

Adversarial Loss. We devise a whole discriminator D_W and a painting discriminator D_R to distinguish whole result $\mathbf{W}_o$ and painting region $\mathbf{R}_o$ from corresponding real ones $\mathbf{W}_{gt}$ and $\mathbf{R}_{gt}$ respectively, where $\mathbf{R}_o = \mathbf{W}_o \odot \mathbf{M}$, so the adversarial losses are:

$$\begin{aligned}\mathcal{L}^W_{adv}(F, D_W) =& \mathbb{E}_{\mathbf{W}_{gt}}[\log(D_W(\mathbf{W}_{gt}))] \\ &+ \mathbb{E}_{\mathbf{W}_o}[\log(1 - D_W(\mathbf{W}_o))], \quad (6)\\ \mathcal{L}^R_{adv}(F, D_R) =& \mathbb{E}_{\mathbf{R}_{gt}}[\log(D_R(\mathbf{R}_{gt}))] \\ &+ \mathbb{E}_{\mathbf{R}_o}[\log(1 - D_R(\mathbf{R}_o))], \quad (7)\end{aligned}$$

where F is the painting function, which is trained to minimize this objective against D_W and D_R that try to maximize it. Our total adversarial loss is:

$$\mathcal{L}_{adv} = \left(\mathcal{L}^W_{adv} + \mathcal{L}^R_{adv}\right)/2. \quad (8)$$

Reconstruction Loss. Inspired by SpiralNet [17], we combine Hue-Color loss with L1 loss to reconstruct $\mathbf{W}_o$ by $\mathbf{W}_{gt}$ in pixel-wise color and intensity:

$$\begin{aligned}\mathcal{L}_{rec} = 1 + \frac{1}{h \times w} \sum_v \Big[&\frac{\lambda}{3} \|\mathbf{W}^v_o - \mathbf{W}^v_{gt}\|_1 - \\ &\min[\cos(\mathbf{W}^v_o, \mathbf{W}^v_{gt}), \cos(\mathbb{1} - \mathbf{W}^v_o, \mathbb{1} - \mathbf{W}^v_{gt})]\Big], \quad (9)\end{aligned}$$

Total Loss. The total loss of our network is:

$$\begin{aligned}\mathcal{L} =& \lambda_{kl}\mathcal{L}_{kl} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{perc}\mathcal{L}_{perc} \\ &+ \lambda_{style}\mathcal{L}_{style}, \quad (10)\end{aligned}$$

where $\mathcal{L}_{perc}$ and $\mathcal{L}_{style}$ denote perceptual loss [20] and style loss [14] respectively, λs are weights to balance different losses. We empirically set $\lambda_{kl} = 0.001$, $\lambda_{adv} = 0.1$, $\lambda_{rec} = 10$, $\lambda_{perc} = 10$ and $\lambda_{style} = 250$ in experiments.

4. Experiments

We conduct experiments to compare our method with state-of-the-art inpainting and outpainting methods respectively. Particulary, for outpainting, we build a new mask dataset for painting from an irregular part, *i.e.*, irregular outpainting. We further conduct ablation studies to validate the efficacy of three stages on CelebA-HQ [21] in regular outpainting. Please refer to *supplementary file* for implementation details, dataset splitting and more compared results.

4.1. Image Inpainting

We conduct experiments on CelebA-HQ and Places2 with the commonly used mask dataset [28] for image inpainting. We compare our method with three state-of-the-art inpainting methods: PC [28], GC [55], and MEDFE [29] with output resolution of 256×256. Following [55, 29], we adopt PSNR, SSIM, FID, mean ℓ_1 error and mean ℓ_2 error for quantitative evaluation. Table 1 and Figure 5 show the quantitative and qualitative comparison results respectively, demonstrating the superiority of our method.

Metric	CelebA-HQ				Places2			
	PC	GC	MEDFE	Ours	PC	GC	MEDFE	Ours
PSNR↑	27.19	27.44	26.82	**27.97**	26.62	27.36	27.17	**28.24**
SSIM↑	0.9283	0.9347	0.9265	**0.9364**	0.8635	0.8813	0.8755	**0.8957**
FID↓	6.23	5.83	5.48	**5.29**	41.57	30.49	35.57	**30.16**
ℓ_1 err.↓	0.0716	0.0735	0.0786	**0.0687**	0.0865	0.0778	0.0802	**0.0706**
ℓ_2 err.↓	0.0111	0.0106	0.0118	**0.0100**	0.0149	0.0142	0.0140	**0.0110**

Table 1. Quantitative comparison of image inpainting. The ↑ indicates the higher the better, and ↓ indicates the lower the better.

4.2. Image Outpainting

Regular Image Outpainting. Following previous outpainting studies [48, 17], we evaluate our method on both object datasets (CelebA-HQ [21], CUB [46], AFHQ Cat [7], Flowers [33]) and scene datasets (Paris StreetView [11], Cityscapes [8], Places2 Desert Road [59]), using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Frechet Inception Distance (FID) as metrics. Besides, we append Learned Perceptual Image Patch Similarity (LPIPS) [57] to measure the perceptual similarity between real images and painting images for evaluating reasonability. Similar to [48, 17], we also consider three different cases: (1) four-side outpainting of $128 \times 128 \rightarrow 256 \times 256$ on CelebA-HQ, CUB, AFHQ Cat and Flowers; (2) two-side outpainting of $256 \times 256 \rightarrow 512 \times 256$ on Cityscapes and Paris StreetView; and (3) one-side outpainting of $256 \times 256 \rightarrow 512 \times 256$ on Places2 Desert Road. We compare our method with state-of-the-art outpainting methods: Boundless [43] in one-side case, SRN [48] and SpiralNet [17], as well as an inpainting method MEDFE [29] retrained for outpainting in all three cases.

Irregular Image Outpainting. To supplement painting whole image from an irregular part, we build a new mask dataset and compare our method with state-of-the-art inpainting method MEDFE [29] on object dataset CelebA-HQ and scene dataset Places2, using the same evaluation metrics as regular outpainting. Our irregular outpainting mask dataset consists of 15672 masks for training and 2600 masks for testing, with resolution of 256×256, covering mask ratio of $50\% - 90\%$. We produce masks mainly considering (1) random overlap of diverse shapes (*e.g.*, circle,

Figure 5. Qualitative comparison results of image inpainting on CelebA-HQ (top) and Places2 (bottom).

Method	CelebA-HQ (Four-side)				Cub (Four-side)				AFHQ Cat (Four-side)				Flowers (Four-side)			
	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓
MEDFE	15.00	0.6424	18.24	0.2977	15.11	0.4971	58.38	0.3969	14.13	0.4971	23.63	0.4306	14.57	0.4818	41.13	0.3877
SRN	15.17	0.6752	32.25	0.2839	15.31	0.5112	80.13	0.3858	14.43	0.5380	25.48	0.3578	13.49	0.4660	66.01	0.4245
SpiralNet	**16.05**	0.6815	21.17	0.2910	**16.22**	0.5313	56.50	0.3807	15.49	0.5488	21.62	0.3594	15.67	0.5078	52.14	0.3894
Ours	15.76	**0.6820**	**18.20**	**0.2652**	16.16	**0.5326**	**39.83**	**0.3737**	**15.50**	**0.5709**	**19.62**	**0.3286**	**15.80**	**0.5193**	**39.40**	**0.3698**

Method	Paris StreetView (Two-side)				Cityscapes (Two-side)				Places2 Desert Road (One-side)			
	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓
MEDFE	17.08	0.6361	24.44	0.3193	20.26	0.6967	23.89	0.2111	19.40	0.6798	85.79	0.2485
Boundless	–	–	–	–	–	–	–	–	19.04	0.6825	86.10	0.2423
SRN	17.08	0.6457	21.53	0.2993	20.33	0.6980	28.90	0.2171	19.45	0.6877	85.59	0.2357
SpiralNet	17.20	0.6480	27.56	0.2789	20.43	0.7125	22.34	0.2141	20.22	0.7026	80.66	0.2448
Ours	**17.63**	**0.6677**	**20.88**	**0.2711**	**20.59**	**0.7141**	**19.89**	**0.1963**	**20.56**	**0.7080**	**78.03**	**0.2264**

Table 2. Quantitative comparison of regular image outpainting.

Method	CelebA-HQ				Places2			
	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓
MEDFE	16.27	0.6911	14.23	0.2493	17.87	0.6360	60.54	0.3235
Ours	**16.81**	**0.7093**	**13.98**	**0.2294**	**18.77**	**0.6679**	**58.79**	**0.3099**

Table 3. Quantitative comparison of irregular image outpainting.

ellipse, rectangular, and triangle) and (2) real object shapes (*e.g.*, person, dog, leaf, and plane).

Tables 2 and 3 show the quantitative comparison results of regular and irregular image outpainting respectively, indicating that our method performs the best across all datasets of painting from part. Noting that, although SpiralNet obtains two better PSNR results, it has been suggested that reconstruction-based metrics (*e.g.*, PSNR) are not true reflections of photo-realism due to multi-modal image completion possibility [54, 28, 24]. We show qualitative comparison of different methods across various datasets of regular and irregular outpainting in Figures 6 and 7, respectively. The results demonstrate that our method achieves to paint more reasonable and realistic whole images.

4.3. Efficacy of Part-noise Restarting

To validate the efficacy of part-noise restarting, we conduct ablation study: (1) remove the noise and FCs to result in a normal cGAN [31] (w/o noise), (2) start to paint from a standard normal noise instead, and (3) ours (part-noise).

Method	PSNR↑	SSIM↑	FID↓	LPIPS↓
w/o noise	15.18	0.6580	22.78	0.2839
standard normal noise	15.49	0.6649	21.44	0.2816
part-noise (ours)	**15.76**	**0.6820**	**18.20**	**0.2652**

Table 4. Quantitative comparison about efficacy of part-noise restarting on CelebA-HQ. Refer to Section 4.3 for details.

Table 4 indicates that painting from both standard normal noise and part-noise outperforms painting without noise, and painting from part-noise performs best. Figure 8 shows that painting without noise has poor representation ability (see the ghosting glasses in Figure 8a), while painting from standard normal noise generates incomplete result (see glasses temples in Figure 8b). Obviously, part-noise restarting attempt to leverage GAN's representation ability on the part-distribution, thus painting more reasonable and accu-

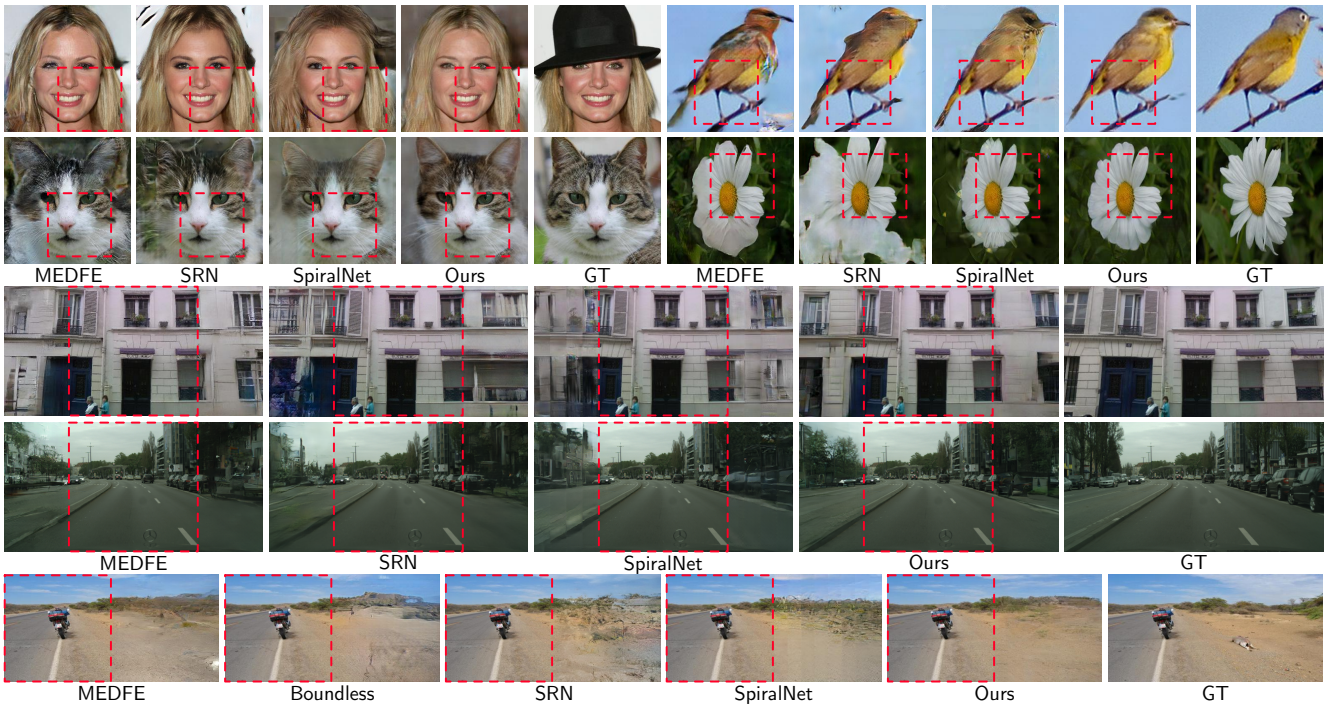

Figure 6. Qualitative comparison of regular image outpainting in four-side, two-side and one-side cases (red boxes mark parts).

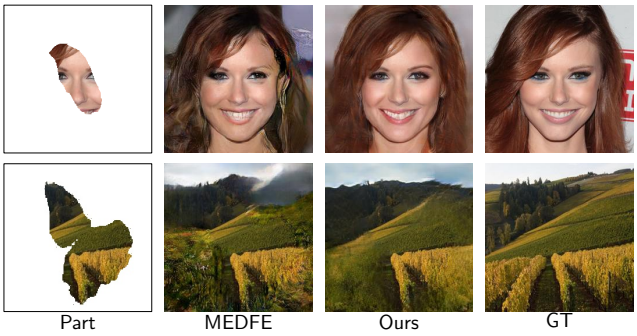

Figure 7. Qualitative comparison results of irregular image outpainting on CelebA-HQ and Places2 with our designed masks.

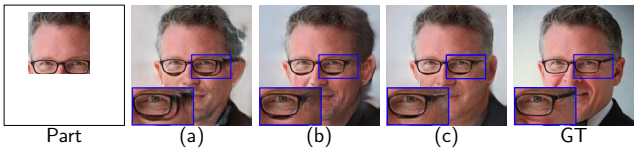

Figure 8. Qualitative comparison about efficacy of part-noise restarting: (a) w/o noise, (b) standard normal noise, (c) part-noise (ours). Refer to Section 4.3 for details.

rate results. Besides, corresponding loss curves in Figure 9 demonstrate the part-noise also benefits our model for faster convergence speed. Notably, different part-noises of same part introduce very slight diversity to the output due to the same part distribution and strong effect of following stages.

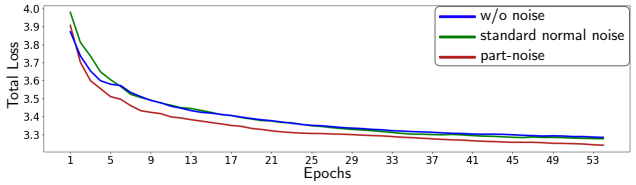

Figure 9. Curves of total loss during training on CelebA-HQ.

4.4. Efficacy of Part-feature Repainting

We first visualize features and illustrate distributions in part-feature repainting stage. Figure 10a indicates that generated feature f_g is pulled closer to transferred/part feature $\hat{f}_g$ with $\bar{f}_g$ as reconstructed result, further Figure 10b demonstrates that synthesized result approaches ground truth rather part, thanks to part-feature repainting.

Besides, we conduct ablation study to validate the efficacy of part-feature repainting: (1) without repainting layer (w/o RL), (2) without part-feature transfer (w/o FT), (3) without whole-feature reconstruction (w/o FR), replace RL with IN to (4) normalize part region and painted region separately (SN) and (5) normalize part region and painted region together (TN), then (6) ours (w/ RL). IN means instance normalization [45] being widely used in GANs.

Table 5 demonstrates the efficacy about our design of RL (w/ RL performs best while w/o RL performs worst) as well as its components (each of FT and FR improves performance). Figure 11 draws the same conclusion, specifically, background color in the part cannot be transferred to the

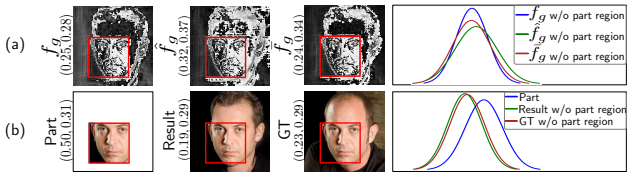

Figure 10. Feature visualization and statistical distribution (Gaussian with $(mean, variance)$) for part-feature repainting: (a) generated feature f_g, transferred/part feature $\hat{f}_g$, and reconstructed feature $\bar{f}_g$, (b) the part, painting result, and ground truth (GT).

whole painting if without FT (w/o RL, w/o FT, SN and TN), while background color in the part will be unduly transferred to the face resulting in unreasonable content if without FR (w/o FR), SN and TN produce unrealistic results, especially, SN leads to inharmony between whole and part due to their independent feature statistics, and TN messes up image color due to direct mixup of part and whole feature statistics, yet our method (w/ RL) can transfer the feature from part to whole (both face and background) as well as keep their independent characteristics, yielding more reasonable and realistic results.

Metric	w/o RL	w/o FT	w/o FR	SN	TN	w/ RL
PSNR↑	15.19	15.42	15.40	15.39	15.29	**15.76**
SSIM↑	0.6530	0.6639	0.6623	0.6616	0.6538	**0.6820**
FID↓	23.39	22.18	21.99	21.58	22.83	**18.20**
LPIPS↓	0.2967	0.2904	0.2855	0.2930	0.2917	**0.2652**

Table 5. Quantitative comparison about efficacy of part-feature repainting on CelebA-HQ. Refer to Section 4.4 for details.

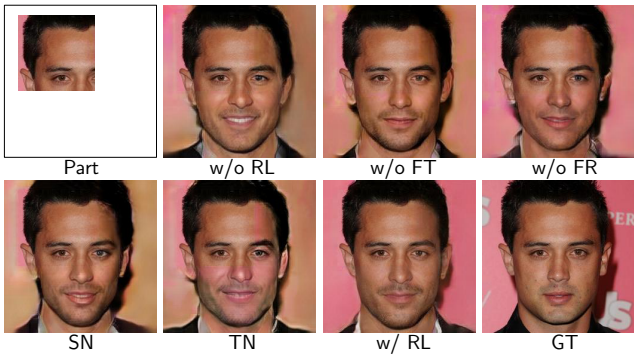

Figure 11. Qualitative comparison about efficacy of part-feature repainting. Refer to Section 4.4 for details.

4.5. Efficacy of Part-patch Refining

We first visualize features in Figure 12, where part-patch information in generated feature p_g is transmitted to surrounding regions for producing p_t that is further refined as p_r, demonstrating the refining efficacy.

We also conduct ablation study to validate the efficacy of part-patch refining: (1) without patch refining module (w/o PRM), insert PRM into painting generator (2) after the first layer (PRM$_{fst}$) and (3) after the middle layer (PRM$_{mid}$),

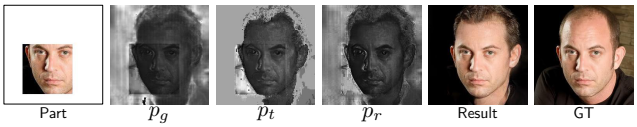

Figure 12. Feature visualization for part-patch refining: generated feature p_g, transmitted feature p_t, and refined feature p_r.

(4) replace cosine distance with L1 distance (PRM$_{L1}$), (5) replace cosine distance with L2 distance (PRM$_{L2}$), (6) replace PRM with self-attention [56], then (7) ours (w/ PRM).

Table 6 and Figure 13 demonstrate the efficacy of our part-patch refining. Particularly, the painting looks coarse if without PRM (w/o PRM) or with attention instead (Atten), and the synthesis seems unrealistic if inserting PRM into the painting generator (PRM$_{fst}$ and PRM$_{mid}$), especially, PRM$_{mid}$ contributes a little for refining since it may disorder high-level structural information of the feature, and PRM$_{fst}$ does not work for refining at the beginning of GAN's generation due to noise property of the feature, while, L1 distance and L2 distance achieve similar acceptable performance, also our method (w/ PRM) helps to refine the painting better (*e.g.*, clear hairs).

Metric	w/o PRM	PRM$_{fst}$	PRM$_{mid}$	PRM$_{L1}$	PRM$_{L2}$	Atten	w/ PRM
PSNR↑	15.35	15.45	15.54	15.35	15.13	15.39	**15.76**
SSIM↑	0.6595	0.6695	0.6719	0.6645	0.6629	0.6664	**0.6820**
FID↓	21.43	22.42	20.76	20.09	20.71	20.86	**18.20**
LPIPS↓	0.2868	0.2849	0.2729	0.2875	0.2852	0.2856	**0.2652**

Table 6. Quantitative comparison about efficacy of part-patch refining on CelebA-HQ. Refer to Section 4.5 for details.

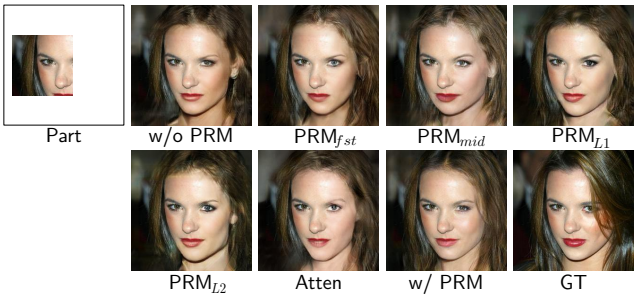

Figure 13. Qualitative comparison about efficacy of part-patch refining. Refer to Section 4.5 for details.

5. Conclusion

In this paper, we propose an unified part-painting task to paint a whole image from the free-form part, and devise a novel method that includes three stages to fully and properly take advantage of both the local domain (the part) and the global domain (the dataset). Both extensive experiments across various datasets of different part-painting tasks and ablation studies demonstrate superiority of our method. We hope that our work opens up new avenues for unifying free-form image outpainting and inpainting.

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM TOG*, 26(3):10, 2007.

[2] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE TIP*, 10(8):1200–1211, 2001.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3):24, 2009.

[4] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE TIP*, 12(8):882–889, 2003.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[6] Tony F Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *JVCIR*, 12(4):436–449, 2001.

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[9] Tali Dekel, Chuang Gan, Dilip Krishnan, Ce Liu, and William T Freeman. Sparse, smart contours to represent and edit images. In *CVPR*, pages 3511–3520, 2018.

[10] Ding Ding, Sundaresh Ram, and Jeffrey J Rodríguez. Image inpainting using nonlocal texture matching and nonlinear filtering. *IEEE TIP*, 28(4):1705–1719, 2018.

[11] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A Efros. What makes paris look like paris? *ACM TOG*, 31(4):101, 2012.

[12] Haoye Dong, Xiaodan Liang, Yixuan Zhang, Xujie Zhang, Xiaohui Shen, Zhenyu Xie, Bowen Wu, and Jian Yin. Fashion editing with adversarial parsing learning. In *CVPR*, pages 8120–8128, 2020.

[13] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *CVPR*, pages 1033–1038, 1999.

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[15] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *ICCV*, pages 349–356, 2009.

[16] 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.

[17] Dongsheng Guo, Hongzhi Liu, Haoru Zhao, Yunhao Cheng, Qingwei Song, Zhaorui Gu, Haiyong Zheng, and bing Zheng. Spiral generative network for image extrapolation. In *ECCV*, pages 701–717, 2020.

[18] Zongyu Guo, Zhibo Chen, Tao Yu, Jiale Chen, and Sen Liu. Progressive image inpainting with full-resolution residual network. In *ACM MM*, pages 2496–2504, 2019.

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM TOG*, 36(4):1–14, 2017.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

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

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

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[24] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided GAN based semantic inpainting. In *CVPR*, pages 13696–13705, 2020.

[25] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, pages 702–716, 2016.

[26] 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.

[27] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *CVPR*, pages 7760–7768, 2020.

[28] 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.

[29] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *ECCV*, 2020.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008.

[31] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[32] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. EdgeConnect: Structure guided image inpainting using edge prediction. In *ICCV Workshops*, Oct 2019.

[33] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729, 2008.

[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.

[35] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016.

[36] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM TOG*, 22(3):313–318, 2003.

[37] Luiz Pessoa, Evan Thompson, and Alva Noë. Finding out about filling-in: A guide to perceptual completion for visual

science and the philosophy of perception. *Behavioral and Brain Sciences*, 21(6):723–748, 1998.

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

[39] 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.

[40] Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M Seitz. Photo uncrop. In *ECCV*, pages 16–31, 2014.

[41] Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.

[42] Ron Slossberg, Gil Shamai, and Ron Kimmel. High quality facial surface and texture synthesis via generative adversarial networks. In *ECCV*, pages 498–513, 2018.

[43] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for image extension. In *ICCV*, pages 10521–10530, 2019.

[44] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391–412, 2003.

[45] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, pages 6924–6932, 2017.

[46] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[47] Miao Wang, Yukun Lai, Yuan Liang, Ralph Robert Martin, and Shi-Min Hu. BiggerPicture: Data-driven image extrapolation using graph matching. *ACM TOG*, 33(6):173, 2014.

[48] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *CVPR*, pages 1399–1408, 2019.

[49] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. TextureGAN: Controlling deep image synthesis with texture patches. In *CVPR*, pages 8456–8465, 2018.

[50] Chaohao Xie, Shaohui Liu, Chao Li, Ming-Ming Cheng, Wangmeng Zuo, Xiao Liu, Shilei Wen, and Errui Ding. Image inpainting with learnable bidirectional attention maps. In *CVPR*, pages 8858–8867, 2019.

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

[52] 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.

[53] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017.

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018.

[55] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, pages 4471–4480, 2019.

[56] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, pages 7354–7363, 2019.

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

[58] Yinda Zhang, Jianxiong Xiao, James Hays, and Ping Tan. FrameBreak: Dramatic image extrapolation by guided shift-maps. In *CVPR*, pages 1171–1178, 2013.

[59] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, pages 1452–1464, 2017.

[60] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, pages 597–613, 2016.

[61] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR*, pages 977–984, 2011.