Zoom-to-Inpaint: Image Inpainting with High-Frequency Details

Soo Ye Kim[1,2†] Kfir Aberman[2] Nori Kanazawa[2] Rahul Garg[2] Neal Wadhwa[2]
Huiwen Chang[2] Nikhil Karnad[2] Munchurl Kim[1] Orly Liba[2]

[1]KAIST
Daejeon, Republic of Korea

[2]Google Research
Mountain View CA, USA

Abstract

Although deep learning has enabled a huge leap forward in image inpainting, current methods are often unable to synthesize realistic high-frequency details. In this paper, we propose applying super-resolution to coarsely reconstructed outputs, refining them at high resolution, and then downscaling the output to the original resolution. By introducing high-resolution images to the refinement network, our framework is able to reconstruct finer details that are usually smoothed out due to spectral bias – the tendency of neural networks to reconstruct low frequencies better than high frequencies. To assist training the refinement network on large upscaled holes, we propose a progressive learning technique in which the size of the missing regions increases as training progresses. Our zoom-in, refine and zoom-out strategy, combined with high-resolution supervision and progressive learning, constitutes a framework-agnostic approach for enhancing high-frequency details that can be applied to any CNN-based inpainting method. We provide qualitative and quantitative evaluations along with an ablation analysis to show the effectiveness of our approach. This seemingly simple, yet powerful approach, outperforms existing inpainting methods.

1. Introduction

Image inpainting is a long-standing problem in computer vision and has many graphics applications. The goal of the problem is to fill in missing regions in a masked image, such that the output is a natural completion of the captured scene with (i) plausible semantics, and (ii) realistic details and textures. The latter can be achieved with traditional inpainting methods that copy patches of valid pixels, e.g., PatchMatch [3], thus preserving the textural statistics of the surrounding regions. Nevertheless, the inpainted results often lack semantic context and do not blend well with the rest of the image. With the advent of deep learning, inpainting neural networks are commonly trained in a self-supervised fashion, by generating random masks and applying them to the full image to produce masked images that are used as the network's input. These networks are able to produce semantically plausible results thanks to abundant training data. However, the results often do not have realistic details and textures, presumably due to the finding of a *spectral bias* [36] in neural networks. That is, high-frequency details are difficult to learn as neural networks are biased

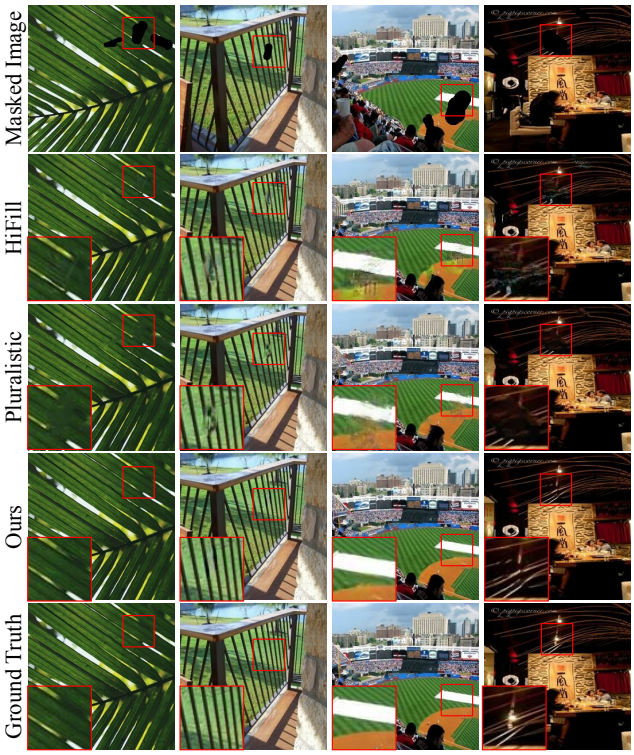


Figure 1. Qualitative comparison to existing inpainting methods, HiFill [49] and Pluralistic [56]. Our method correctly reconstructs high-frequency details, e.g., fine textures and narrow structures, and preserves the continuity and the orientation of edges.

[†]This work was done during an internship at Google Research.

towards learning low-frequency components. This is especially problematic when training neural networks for image restoration tasks such as image inpainting, because high-frequency details must be generated for realistic results.

Recent neural network architectures for image inpainting consist of (i) a coarse network that first generates a coarsely filled-in result, and (ii) a refinement network that corrects and refines the coarse output for better quality [49–51]. In this paper, with the goal of generating more realistic high-frequency details, we propose to refine after *zooming in*, therefore refining the image at a resolution higher than the target resolution. This allows the refinement network to correct local irregularities at a finer level and to learn from high-resolution (HR) labels, thus effectively reducing the spectral bias at the desired resolution and injecting more high-frequency details into the resulting image. We show that adding a simple bicubic upsampling component between the coarse and refinement networks improves inpainting results, and using a super-resolution (SR) network to upscale the intermediate result improves the results even further.

Furthermore, as an HR refinement network can be more difficult to train than a low-resolution (LR) refinement network due to a larger mask and more missing pixels in the HR input, we propose a novel *progressive learning strategy* for inpainting, where the size of masks is increased as training progresses and the framework is trained on larger masks at a later training stage. Moreover, to further enhance the high-frequency details, we propose to use an additional gradient loss [11] that minimizes the gradients of the difference between the prediction and the ground truth. We believe that these three fundamental strategies can benefit any existing inpainting network.

In summary, our contributions are as follows:

- We propose a novel inpainting framework that includes an SR network to *zoom in*, allowing refinement at HR and training with HR labels, to enhance the generation of high-frequency details in the final inpainted output.
- We propose a progressive learning strategy for inpainting to aid convergence with larger masks.
- We use a gradient loss for inpainting to further improve textural details.

2. Related work

2.1. Image inpainting

Traditional image inpainting methods can be largely classified into three types: (i) propagation-based approaches that gradually fill in the missing regions from known pixel values at hole boundaries [5, 43], (ii) Markov Random Field (MRF) approaches optimizing discrete MRFs [19, 34], or (iii) patch-based approaches that search for plausible patches outside of the hole to be pasted into the missing region [3, 4] similar to texture synthesis algorithms [9, 10]. These types of approaches exploit information already present in the input image.

Deep-learning-based inpainting methods leverage information external to any one specific image by learning global semantics from an abundant corpus of training data. An early convolutional neural network (CNN) based method for inpainting was the Context Encoder [33], where the authors proposed using an L2 loss with a global generative adversarial network (GAN) loss for improved perceptual quality. GANs [12] are especially suitable for image inpainting because they are able to synthesize realistic images [6,16,20,24,26,35,44,54,55]. To consider local details as well as global semantics, Demir *et al.* [8] proposed using a PatchGAN [15] along with the global GAN. Our inpainting framework also employs a PatchGAN discriminator for enhanced local details.

When generating each missing pixel, CNNs with stacked convolution layers are limited by the local receptive field of the convolution operation, whereas previous patch-based methods are able to copy from any part in the surrounding known regions. Thus, Yu *et al.* [50] devised a contextual attention (CA) module that copies patches from the surrounding regions into the missing region, weighted by the computed similarity. We add the CA module in the bottleneck of our refinement network to get the best of both worlds – ability of GANs in synthesizing novel structures and details, as well as the ability to copy patches anywhere from the image without restrictions in the receptive field, like in [50, 51]. There also exist inpainting methods [27, 52] that use other types of attention modules.

Typically, input images for inpainting consist of some input pixels that are valid, or known, while others are invalid, or unknown/missing. Liu *et al.* [25] addressed this dichotomy using partial convolution, where only the valid pixels are taken into consideration during convolution by using a predefined mask. Yu *et al.* [51] proposed gated convolutions, where the masks are also learned, while Xie *et al.* [47] proposed using learnable bidirectional attention maps. We employ gated convolutions [51] in our framework to handle the valid and invalid pixels.

Many recent CNN-based inpainting methods use a two-stage approach, where the first network generates a coarse output and the second network refines this output [49–51, 53]. Some methods [23, 32, 48] divide the stages as (i) edge completion and (ii) image completion, or in StructureFlow [37], as (i) structure generation and (ii) texture generation. Among the coarse-to-fine two-stage methods, some methods [49, 53] handle inpainting for HR images by first downscaling the HR input to a fixed resolution before

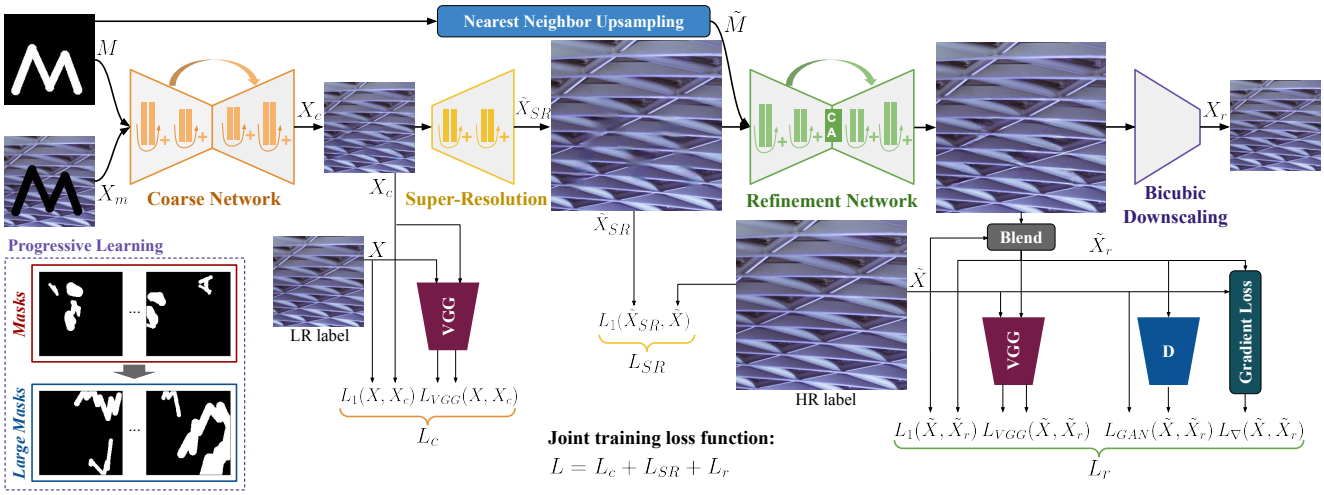


Figure 2. Our proposed inpainting framework containing three main components: a coarse network, an SR network, and an HR refinement network. The framework progressively learns to inpaint larger missing regions, and the refinement network is trained with a gradient loss to enhance the generation of high-frequency details. Images shown in this figure are actual intermediate results produced by our framework, and more examples are given in the Supplementary Material.

the inpainting stages, and increasing the resolution after inpainting using residual aggregation [49] or guided upsampling [53]. In our *zoom-to-inpaint* model, we increase the resolution of the image *between* the coarse inpainting and the refinement stages such that the coarse output is refined at HR, and then come back to the original or the desired resolution after refinement.

Unlike conventional CNN architectures for image and video restoration that tend to reduce the size of the image (as in multi-scale architectures [17,31]), or feature map (eg. U-Net [38]), to increase the receptive field of the network, our model goes *beyond* the input and target resolutions for better refinement of high-frequency details. Upsampling can be achieved by simple bicubic interpolation, and we show that using an SR network, trained end-to-end with the coarse and refinement inpainting networks, produces even better results. The modular construction of this seemingly simple idea is in fact non-trivial, and we explain the framework in detail in Section 3. We further elaborate on how HR refinement aids the generation of high-frequency details in Section 3.1.3 and present a frequency-based analysis in Section 4.4.

2.2. Progressive learning

Learning to solve a difficult task from scratch can be challenging for neural networks. Hence, fine-tuning [14] or transfer learning is a commonly applied technique, where prior knowledge is transferred from pretrained networks to the subsequent training stages. Progressive neural networks [39] expanded on this idea and added lateral connections to previously learned features. Karras *et al.* [16] proposed to progressively train a GAN to synthesize LR images first, then to generate HR images by incrementally adding on layers to stabilize and speed up the training process. For image inpainting, filling in missing regions can be increasingly difficult as they become larger. Hence, some methods [13, 21] proposed a recurrent scheme that iteratively fills in holes from the boundary for each image as in traditional propagation-based schemes, which must also be applied during inference. In contrast, we propose to increase the size of the masks as training progresses so that our inpainting network *learns* progressively.

2.3. Gradient loss

Utilizing image gradients as a prior [42] or in the loss function [11, 22, 29] has been widely explored for image SR [29, 42] and depth estimation [11, 22] to increase sharpness in the reconstructed images. In image inpainting, Telea *et al.* [43] proposed a fast marching method that use the gradients of neighboring pixels to estimate the missing values in the inpainted region. Liu *et al.* [28] proposed a loss function to enforce the continuity of gradients in the reconstructed region and its neighboring regions. Inspired by Eigen *et al.*'s method [11] for depth estimation, we propose to minimize the image gradients of the difference between the prediction and the ground truth to further enhance high-frequency details in the inpainted result.

3. Proposed method

We propose a novel inpainting framework that is able to reconstruct high-frequency details in the final output by (i) upsampling the result of a coarse inpainting network using

an SR network and refining at HR, and (ii) employing a gradient loss. For better convergence, the framework is trained *progressively*, by increasing the size of masks.

3.1. Framework overview

Our inpainting framework consists of three trainable networks connected sequentially: a coarse inpainting network, an SR network, and an HR refinement network. The SR network and the HR refinement network are trained with HR (original) labels, $\tilde{X}$, whereas the coarse network is trained with the bicubic-downscaled versions, X. Each network is first pretrained separately, and then all networks are trained jointly. Please refer to the Supplementary Material for the detailed architectures of each network.

3.1.1 Coarse network

The coarse network, f_c, aims to characterize the LR variations in the image across its entire field-of-view, coarsely filling in the missing regions in the input masked image, X_m, given by,

$$X_m = (1 - M) \odot X, \tag{1}$$

where $M \in \mathbb{R}^{H \times W \times 3}$ is a binary mask where invalid pixels, i.e., pixels to be inpainted, are 1 and valid pixels are 0, with repeated values across the channel dimension, $X \in \mathbb{R}^{H \times W \times 3}$ is the full image, and $\odot$ is element-wise multiplication. We employ an encoder-decoder-based CNN architecture with gated convolutions similar to [51], additionally with residual blocks and batch normalization. The network output is masked with the input, yielding X_c as,

$$X_c = M \odot f_c(X_m, M, \Theta_c) + X_m, \tag{2}$$

so that the network does not attempt to reconstruct already valid regions. We train it by minimizing a loss function L_c, consisting of an L1 loss to enforce pixel-wise similarity, and a VGG loss to enforce similarity in the feature domain, given as,

$$L_c = \|X_c - X\|_1 + \lambda_c^{\phi} \cdot \|\phi_{1,4}(X_c) - \phi_{1,4}(X)\|_1, \tag{3}$$

where $\phi_{i,j}$ is the i-th convolution layer at the j-th block in VGG19 [41], and λ_c^{ϕ} is a constant.

3.1.2 Super-resolution network

We use an SR network that *zooms in* on the coarse output X_c by scale factor $s > 1$, yielding $\tilde{X}_{SR} \in \mathbb{R}^{sH \times sW \times 3}$. Our SR network architecture is designed as a cascade of four residual blocks with a pixel shuffle layer [40] at the end. Contrary to the coarse network output, we do not mask $\tilde{X}_{SR}$ since the refinement network in the following stage can propagate the HR patches from valid regions to the inpainted region using contextual attention (CA) [50]. Therefore, we train the SR network by directly minimizing $L_{SR} = \|\tilde{X}_{SR} - \tilde{X}\|_1$, where $\tilde{X} \in \mathbb{R}^{sH \times sW \times 3}$ is the full HR image.

3.1.3 High-resolution refinement network

Unlike common refinement schemes of previous inpainting frameworks, our proposed refinement is achieved by *zooming in*, refining, then *zooming out* back to the input resolution, in order to benefit from the supervision of HR labels during refinement and aid the learning of high-frequency components. Specifically, given $\tilde{X}_{SR}$ and $\tilde{M}$ as input, where $\tilde{M} \in \mathbb{R}^{sH \times sW \times 3}$ is M upscaled by nearest neighbor upsampling, the HR refinement network, f_r, generates the refined image $f_r(\tilde{X}_{SR}, \tilde{M}, \Theta_r) \in \mathbb{R}^{sH \times sW \times 3}$. Then, $\tilde{X}_r$, which is used for optimizing the training losses in the refinement network, is obtained by blending the network output with the label, $\tilde{X}$:

$$\tilde{X}_r = \tilde{M} \odot f_r(\tilde{X}_{SR}, \tilde{M}, \Theta_r) + (1 - \tilde{M}) \odot \tilde{X}. \tag{4}$$

By masking with the original label and not the input, $\tilde{X}_{SR}$, no loss occurs outside the inpainted regions. As the loss is zero in the valid regions, the network does not spend its capacity on reconstructing these regions, which will be replaced by the input image, X_m, after downscaling, in the final output (Equation 7). The architecture of the refinement network is similar to the coarse network and is encoder-decoder-based, except that we add a CA module [50] to its bottleneck.

For training, we use a gradient loss, L_{∇}, between $\tilde{X}_r$ and $\tilde{X}$ to further encourage the generation of high-frequency details. Inspired by [11], L_{∇} is given as,

$$L_{\nabla} = \frac{1}{2}(\|(\tilde{X}_r - \tilde{X})_{\nabla_x}\|_2^2 + \|(\tilde{X}_r - \tilde{X})_{\nabla_y}\|_2^2), \tag{5}$$

where ∇_x and ∇_y are horizontal and vertical image gradients, respectively, obtained by 1-tap filters, $[-1, 1]$ and $[-1, 1]^{\mathrm{T}}$. Then, f_r is trained with L_r that consists of an L1 loss, VGG loss, hinge GAN loss – L_h, and L_{∇}, given by,

$$\begin{aligned} L_r = \|\tilde{X}_r - \tilde{X}\|_1 + \lambda_r^{\phi} \cdot \|\phi_{1,4}(\tilde{X}_r) - \phi_{1,4}(\tilde{X})\|_1 \\ + \lambda_r^{GAN} \cdot L_h(\tilde{X}_r) + \lambda_r^{\nabla} \cdot L_{\nabla}. \end{aligned} \tag{6}$$

A PatchGAN [15] approach is adopted for good perceptual results, with spectral normalization [30] similar to [51] for stable training of GANs. By training the refinement network with HR labels, we drive the CNN to explicitly learn high-frequency details, additionally to the low-frequencies that are inherently preferred according to the empirical evidence of a spectral bias in neural networks [36].

After pretraining each of the three components separately, the entire framework is trained end-to-end with a total loss $L = L_c + L_{SR} + L_r$.

Figure 3. Example masks used during progressive learning.

Downscaling. As a last step, the refined HR output, $f_r(\tilde{X}_{SR}, \tilde{M}, \Theta_r)$, is downscaled back (*zoomed out*) by scale factor s to the original resolution by bicubic downsampling, and blended with the input, X_m. The final output $X_r \in \mathbb{R}^{H \times W \times 3}$ is then given as,

$$X_r = M \odot f_r(\tilde{X}_{SR}, \tilde{M}, \Theta_r) \downarrow_s + (1 - M) \odot X_m. \quad (7)$$

Note that the refinement network would not be able to learn from the HR labels if losses are only imposed on X_r.

3.2. Progressive learning for image inpainting

In image inpainting, the training time until convergence tends to increase proportionally as invalid regions in masks become larger and increasingly more difficult to fill in [25]. This is problematic if we want to refine at HR with the image enlarged by scale factor s, since the number of missing pixels would increase by s^2. Thus, we propose a progressive learning strategy for inpainting, where we train the network in N steps by increasing the size of masks at each n-th step, where $n = 1, 2, ..., N$. We set $N = 2$ in our experiments, where masks at $n = 2$ are the same as random masks used in [50] and masks at $n = 1$ are generated by modifying the random generation parameters of masks in [50] to produce smaller and more confined masks. Example masks are shown in Figure 3 and a detailed configuration of the parameters is provided in the Supplementary Material.

Empirically, if our framework is trained directly on masks at $n = 2$ without progressive learning, it takes 2M iterations to converge. With progressive learning, it takes 80K iterations to converge on masks at $n = 1$ then only 1.2M iterations on masks at $n = 2$. A mere addition of 80K iterations on $n = 1$ drops the total number of iterations by $\sim 40\%$. In the following sections, *masks* denote masks at $n = 1$ and *large masks* denote masks at $n = 2$.

4. Experiment results and evaluation

4.1. Implementation details

Training configuration. For training our *zoom-to-inpaint* model, we first pretrain the coarse and refinement networks on Places2 [57] at 256×256 resolution for 180K iterations using *masks* as defined in Section 3.2. For both networks, we only use the L1 loss and the VGG loss, with $\lambda^{\phi} = 0.01$. For the SR network, we use an upsampling scale factor of $s = 2$ and pretrain it on DIV2K [2] for 400K iterations with randomly cropped 64×64 patches. Then, we jointly train the entire framework on DIV2K using the proposed progressive learning strategy, i.e., 80K iterations with *masks*, and then another 1.2M iterations with *large masks*. We randomly crop 512×512 patches from DIV2K and use them as HR labels ($\tilde{X}$), and bicubic-downsample them to 256×256 to generate LR labels (X). We use the following loss coefficients: $\lambda_c^{\phi} = 0.01$, $\lambda_r^{\phi} = 10^{-5}$, $\lambda_r^{GAN} = 0.5$ and $\lambda_r^{\nabla} = 1$. Our implementation is in TensorFlow [1] and trained on 8 NVIDIA V100 GPUs using Adam optimizer [18] with a mini-batch size of 16 and a learning rate of 10^{-5}. Please refer to the Supplementary Material for details of our model architecture and a complexity analysis.

Test dataset. For the test dataset, we apply both masks on 200 images from the validation and test sets of Places2 (100 images each), and on 100 images in the validation set of DIV2K. For DIV2K, 256×256 patches were center-cropped from the full images and used as X.

Inpainting methods. We compare our method with the following inpainting approaches: DeepFillv2 [51], EdgeConnect [32], Pluralistic [56], and HiFill [49]. Similar to our method, [32, 51, 56] were trained on 256×256 images, and therefore, our test set can be used as is. However, since HiFill [49] was originally trained on 512×512 images, we bicubic-upscale the test images to 512×512 for input and then downscale the output back to 256×256 before computing the metrics. We used the publicly released weights trained on Places2 for all methods.

4.2. Quantitative evaluation

For quantitative evaluation, we report the results of four metrics – PSNR, SSIM [45], MS-SSIM (multi-scale SSIM) [46] and L1 error – that are frequently used in inpainting literature, on both mask sizes in Table 1. Two additional perceptual metrics – FID and LPIPS – that are less frequently used, are reported in the Supplementary Material. As shown in Table 1, our *zoom-to-inpaint* model outperforms all compared methods on both mask sizes on all metrics, with at most 0.97 dB PSNR and 0.0055 MS-SSIM gain over the next best method. It shows that our framework is able to generate results that are more consistent with the ground truth compared to the compared inpainting methods.

In Table 1, the improvement over the next best method on PSNR and L1 error, which are pixel-wise error metrics, is larger on *masks* compared to *large masks*, showing that our framework adds on better pixel-level details that are closer to the ground truth when the missing region is rel-

Method	Places2 (256 × 256)				DIV2K (256 × 256, center-cropped)			
	PSNR ↑	SSIM ↑	MS-SSIM ↑	L1 Error ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	L1 Error ↓
Masks								
HiFill [49]	31.12	0.9586	0.9742	0.00744	30.91	0.9633	0.9777	0.00700
Pluralistic [56]	33.23	0.9670	0.9807	0.00558	32.73	0.9703	0.9820	0.00543
DeepFill-v2 [51]	34.03	0.9719	0.9834	0.00485	33.11	0.9741	0.9852	0.00467
EdgeConnect [32]	33.98	0.9718	0.9841	0.00388	33.11	0.9734	0.9845	0.00388
Ours	**34.78**	**0.9755**	**0.9863**	**0.00357**	**34.08**	**0.9787**	**0.9886**	**0.00329**
Large Masks								
HiFill [49]	24.94	0.8891	0.9134	0.02034	24.23	0.8739	0.8993	0.02302
Pluralistic [56]	26.17	0.9022	0.9191	0.01784	25.62	0.8890	0.9071	0.01949
DeepFill-v2 [51]	26.77	0.9158	0.9326	0.01536	26.07	0.9018	0.9229	0.01735
EdgeConnect [32]	27.61	0.9166	0.9382	0.01328	26.87	0.9036	0.9291	0.01494
Ours	**27.71**	**0.9202**	**0.9415**	**0.01314**	**27.07**	**0.9094**	**0.9346**	**0.01462**

Table 1. Quantitative evaluation. Values in **bold** denote the best performance.

Compared method	[49]	[56]	[51]	[32]
Preference of ours over compared	75.49%	89.13%	69.23%	64.21%

Table 2. User study results on *masks*, indicating the percentage (%) of users who selected ours over the compared method. Results on *large masks* are provided in the Supplementary Material.

atively smaller. The gain on MS-SSIM, which measures the structural similarity at multiple scales, is greater for *large masks*, showing that our model is able to recover better global structures for large missing regions.

4.3. Qualitative evaluation

Visual results. We show qualitative comparisons of visual results in Figures 1 and 4. In Figure 1, our method accurately reconstructs edges and fine lines in the correct orientation while other methods find it difficult to preserve the continuity of the fine lines or fail to produce any edges at all in the missing region. Figure 4 shows the qualitative results on both mask sizes. Similar to Figure 1, our method accurately reconstructs high-frequency details such as edges and fine texture, as well as global structures for *large masks*. Please refer to the Supplementary Material for additional results, including full images of the crops shown in Figure 4.

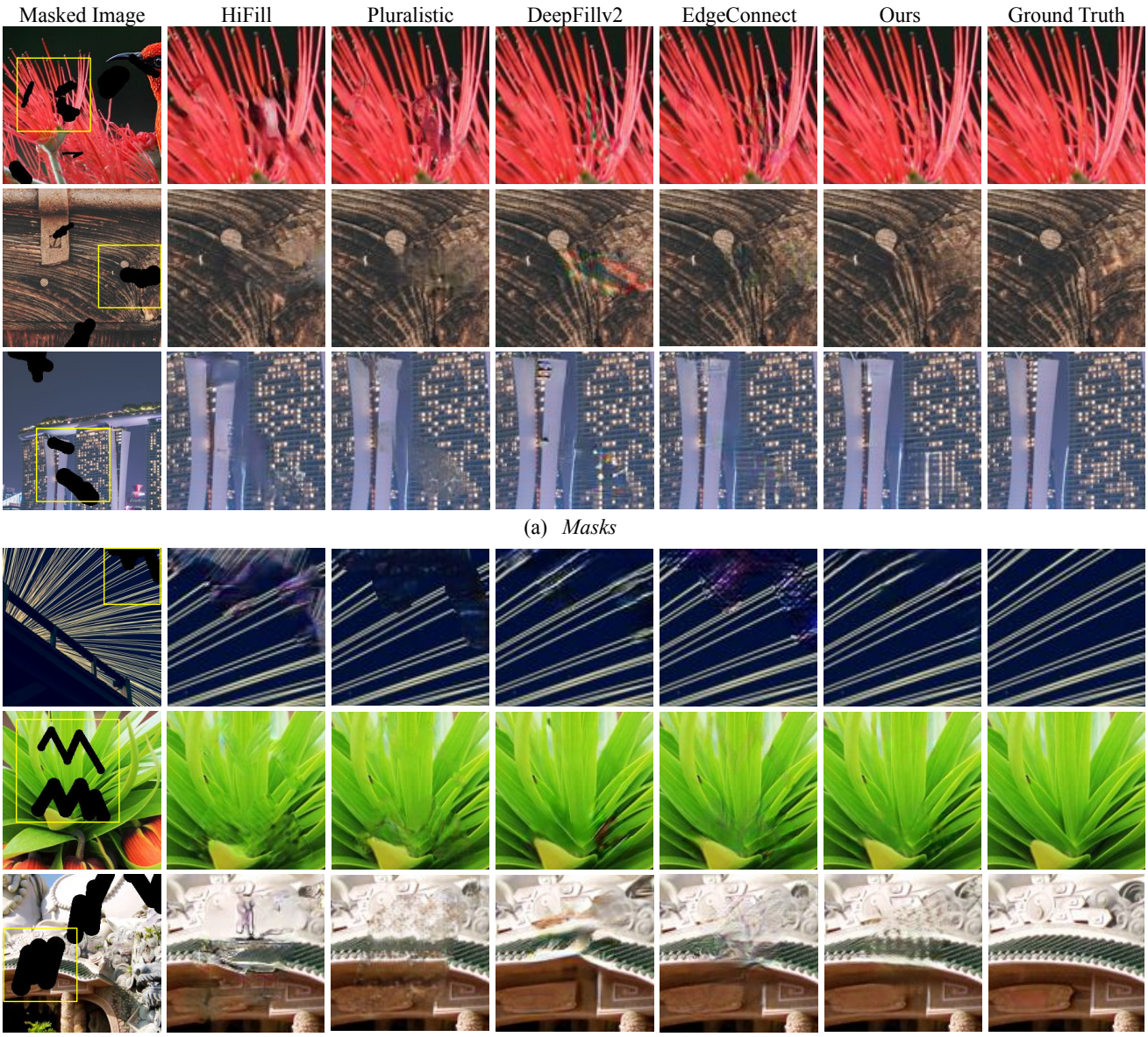
User study. We conduct a user study to evaluate the preferences of users on the results produced by our approach compared to the other methods. We asked 13 users to evaluate 300 pairs of inpainted images in a random order, where one image is generated by our method, and the other image is generated by a method among [32, 49, 51, 56]. Users are asked to select their preferred result based on the question: "Which of these images looks better?". The percentage of users that prefer our method over the others for *masks* is summarized in Table 2, where users more frequently prefer our method over all other methods, with at least 64.21% preference rate. More details of the user study including a screenshot, the raw number of counts, and results on *large masks* are provided in the Supplementary Material. We observed that *masks* are more suitable for comparing the ability to generate pleasing and comparable results rather than *large masks*, due to objectionable artifacts being generated by all methods for the latter, as shown in the Supplementary Material.

Ablations	PSNR ↑	SSIM ↑	MS-SSIM ↑	L1 Error ↓
Masks				
No zoom	32.12	0.9714	0.9812	0.00441
Bicubic zoom	32.80	0.9753	0.9832	0.00391
SR zoom	33.40	0.9770	0.9853	0.00363
SR zoom+L_∇	**34.08**	**0.9787**	**0.9886**	**0.00329**
Large Masks				
No zoom	25.89	0.8977	0.9180	0.01747
Bicubic zoom	26.09	0.9016	0.9219	0.01657
SR zoom	26.95	0.9080	0.9307	0.01497
SR zoom+L_∇	**27.07**	**0.9094**	**0.9346**	**0.01462**

Table 3. Quantitative comparison of our model (SR zoom+L_∇) with its ablations.

4.4. Ablation study

Ablation study on framework components. In order to analyze the contributions of the individual components, we compare against three ablations of our inpainting framework: (i) No zoom, (ii) Bicubic zoom, and (iii) SR zoom. For (i), we replace the SR component (described in Section 3.1.2) with an identity transform that simply copies the coarse output without an upsampling component, so that refinement is applied at the original resolution like in other conventional two-stage inpainting frameworks. For (ii), we



(a) *Masks*

(b) *Large Masks*

Figure 4. Qualitative comparison to other methods on (a) *Masks* and (b) *Large Masks*. When images with high-frequency regions are fed into inpainting networks, HiFill [49] and Pluralistic [56] tend to generate blurry texture as seen in the examples in the 1st, 2nd and 3rd rows, and DeepFillv2 [51] and EdgeConnect [32] generate color artifacts as shown in the 2nd, 3rd and 5th rows. Our method is able to accurately reconstruct high-frequency details, as well as global structures.

replace the SR component with bicubic upsampling, and for (iii), we add back the SR zoom. (i), (ii) and (iii) are trained without the gradient loss L_∇. Lastly, SR zoom+L_∇ corresponds to our full framework with L_∇. The results are shown in Table 3 and Figure 5.

As shown in Table 3, zooming in with bicubic upsampling improves all metrics compared to refining at the original resolution (*No zoom*), showing the benefit of refining at a higher resolution and training the refinement network with HR supervision. This indicates that as long as the refinement network is trained on HR labels so that local irregularities generated by the coarse network can be corrected with the magnification, the *zoom* can even be achieved by bicubic upsampling. Adding the SR network further improves the accuracy by a large margin, with $>$1 dB gain in PSNR compared to *No zoom*. Compared to Bicubic zoom, SR zoom is able to generate sharper results for the surrounding regions, that can then be propagated by the CA module into the in-

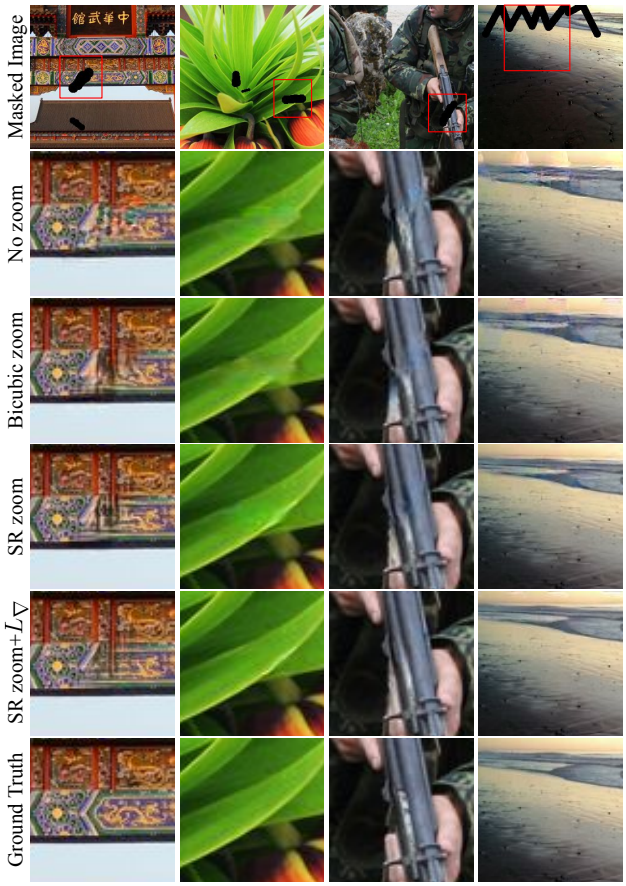


Figure 5. Visual comparison of results produced by our model and its ablations. HR refinement and the gradient loss improves the generation of high-frequency details.

painted region. Adding the gradient loss further improves the quantitative metrics for both mask sizes. Its benefit is especially prominent for *masks*, where the network is more likely to reconstruct the image more accurately, and thus, fine details contribute more to the evaluation metrics. Figure 5 shows that each component improves the reconstruction quality, analogously to the quantitative results. Please refer to the Supplementary Material for additional results.

Frequency-domain analysis. We provide insights into the benefits of zooming in and refining at HR even though the final output is of lower resolution. Using a frequency-domain analysis, we demonstrate that our strategy introduces desirable frequencies into the inpainted result that survive downsampling. Specifically, instead of directly computing the metrics corresponding to a ground truth image and a prediction, we first construct a 2-level Laplacian pyramid [7] for each of them using a traditional 5-tap Gaussian kernel, and report *per-level* metric values. This allows us to measure the accuracy in different frequency bands.

In Figure 6, we show the per-level improvement of *Bicubic zoom*, *SR zoom* and *SR zoom+L_∇* over the baseline *No zoom*. We use the SSIM metric that is known to be sensitive to local structural changes, and use *masks* to avoid the effect of artifacts generated with very large masks. While we see improvements in all frequency bands, we observe that the improvements are skewed towards higher frequency bands for all models. This indicates that all components of our framework, i.e., refining at HR with HR labels, SR zoom and gradient loss, improve the overall reconstruction, more so at higher frequencies.

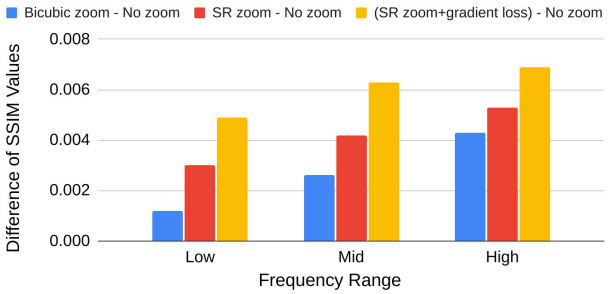


Figure 6. Frequency domain comparison of the ablation models using 2-level Laplacian pyramids [7], evaluated on *masks*. Each component of our framework adds to improving the reconstruction of the inpainted regions, with high frequencies benefitting more than lower frequencies.

5. Conclusion

We propose a novel inpainting framework with HR refinement, by inserting an SR network between coarse and refinement networks. By training the refinement network with HR labels, our model is able to learn from high-frequency components present in the HR labels, reducing the spectral bias [36] at the desired resolution. Furthermore, we propose a progressive learning strategy for inpainting that increases the area of the missing regions as training progresses, and a gradient loss for inpainting to generate even more accurate texture and details. The HR refinement, progressive learning and gradient loss can each or together be applied to any inpainting framework. These simple but non-trivial modular constructions greatly improve the final inpainted result quantitatively and qualitatively.

Limitations. In cases of challenging inputs with very large holes where all methods tend to generate severe artifacts, we find that our method produces artifacts containing a high-frequency repetitive pattern that is displeasing and sometimes more objectionable than artifacts produced by other methods. We analyze these artifacts with the user study on *large masks* in the Supplementary Material.

Acknowledgement. We thank Jonathan T. Barron for insightful discussions and Yael Pritch and David Salesin for useful comments.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 5

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017. 5

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 2009. 1, 2

[4] Connelly Barnes, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, 2010. 2

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *SIGGRAPH*, 2000. 2

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 2

[7] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. 8

[8] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. In *arXiv:1803.07422*, 2018. 2

[9] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001. 2

[10] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, 1999. 2

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, 2015. 2, 3, 4

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *International Conference on Neural Information Processing Systems*, 2014. 2

[13] Zongyu Guo, Zhibo Chen, Tao Yu, Jiale Chen, and Sen Liu. Progressive image inpainting with full-resolution residual network. In *ACM Multimedia*, 2019. 3

[14] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 3

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 4

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2, 3

[17] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *AAAI Conference on Artificial Intelligence*, 2020. 3

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[19] Nikos Komodakis and Georgios Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007. 2

[20] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided gan based semantic inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[21] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[22] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[23] Liang Liao, Ruimin Hu, Jing Xiao, and Zhongyuan Wang. Edge-aware context encoder for image inpainting. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018. 2

[24] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *European Conference on Computer Vision*, 2020. 2

[25] 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 *European Conference on Computer Vision*, 2018. 2, 5

[26] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *European Conference on Computer Vision*, 2020. 2

[27] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *IEEE International Conference on Computer Vision*, 2019. 2

[28] Huaming Liu, Guanming Lu, Xuehui Bi, Jingjie Yan, and Weilan Wang. Image inpainting based on generative adversarial networks. In *IEEE International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, 2018. 3

[29] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[30] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 4

[31] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[32] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *IEEE International Conference on Computer Vision Workshops*, 2019. 2, 5, 6, 7

[33] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[34] Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg. Shift-map image editing. In *IEEE International Conference on Computer Vision*, 2009. 2

[35] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 2

[36] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, 2019. 1, 4, 8

[37] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *IEEE International Conference on Computer Vision*, 2019. 2

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015. 3

[39] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. In *arXiv:1606.04671*, 2016. 3

[40] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 4

[42] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 3

[43] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 2004. 2, 3

[44] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. In *European Conference on Computer Vision*, 2020. 2

[45] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[46] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. *Asilomar Conference on Signals, Systems & Computers*, 2:1398–1402, 2003. 5

[47] 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 *IEEE International Conference on Computer Vision*, 2019. 2

[48] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[49] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 5, 6, 7

[50] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4, 5

[51] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *IEEE International Conference on Computer Vision*, 2019. 2, 4, 5, 6, 7

[52] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[53] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *European Conference on Computer Vision*, 2020. 2, 3

[54] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision*, 2017. 2

[55] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[56] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 5, 6, 7

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database

for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5