Rethinking Image Inpainting via a Mutual Encoder-Decoder with Feature Equalizations

Hongyu Liu[1], Bin Jiang[1(✉)], Yibing Song[2(✉)], Wei Huang[1], and Chao Yang[1]

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
{kumapower,jiangbin,hwei,yangchaoedu}@hnu.edu.cn
[2] Tencent AI Lab, Shenzhen, China
yibingsong.cv@gmail.com

Abstract. Deep encoder-decoder based CNNs have advanced image inpainting methods for hole filling. While existing methods recover structures and textures step-by-step in the hole regions, they typically use two encoder-decoders for separate recovery. The CNN features of each encoder are learned to capture either missing structures or textures without considering them as a whole. The insufficient utilization of these encoder features hampers the performance of recovering both structures and textures. In this paper, we propose a mutual encoder-decoder CNN for joint recovery of both. We use CNN features from the deep and shallow layers of the encoder to represent structures and textures of an input image, respectively. The deep layer features are sent to a structure branch, while the shallow layer features are sent to a texture branch. In each branch, we fill holes in multiple scales of the CNN features. The filled CNN features from both branches are concatenated and then equalized. During feature equalization, we reweigh channel attentions first and propose a bilateral propagation activation function to enable spatial equalization. To this end, the filled CNN features of structure and texture mutually benefit each other to represent image content at all feature levels. We then use the equalized feature to supplement decoder features for output image generation through skip connections. Experiments on benchmark datasets show that the proposed method is effective to recover structures and textures and performs favorably against state-of-the-art approaches.

Keywords: Deep image inpainting · Feature equalizations

This work is done partially when H. Liu is an intern at Tencent AI Lab. The results and code are available at https://github.com/KumapowerLIU/Rethinking-Inpainting-MEDFE.

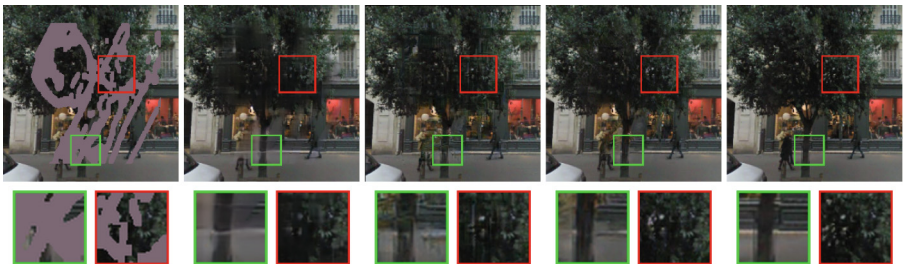
The original version of this chapter was revised: City and country of the second affiliation was corrected from "Bellevue, USA" to "Shenzhen, China". The correction to this chapter is available at https://doi.org/10.1007/978-3-030-58536-5_47

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58536-5_43) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12347, pp. 725–741, 2020.
https://doi.org/10.1007/978-3-030-58536-5_43

1 Introduction

There is a need to recover missing contents in corrupted images for visual aesthetics improvement. Deep neural networks have advanced image inpainting by introducing semantic guidance to fill hole regions. Different from the traditional methods [2,3,7,8] that propagate uncorrupted image contents to the hole regions via patch-based image matching, deep inpainting methods [13,25] utilize CNN features in different levels (i.e., from low-level features to high-level semantics) to produce more meaningful and globally consistent results.



(a) Input (b) GC [41] (c) CSA [21] (d) Ours (e) GT

Fig. 1. Visual comparison on the Paris StreetView dataset [6]. GT is the ground truth image. The proposed inpainting method is effective to reduce blur and artifacts within and around the hole regions, which are brought by inconsistent structure and texture features.

The encoder-decoder architecture is prevalent in existing deep inpainting methods [13,19,25,38]. However, a direct utilization of the end-to-end training and prediction processes generates limited results. This is due to the challenging factor that the hole region is completely empty. Without sufficient image guidance, an encoder-decoder is not able to reconstruct the whole missing content. An alternative is to use two encoder-decoders to separately learn missing structures and textures in a step-by-step manner. These two-stage methods [21,24,26,27,29,40,41] typically generate an intermediate image with recovered structures in the first stage (i.e., encoder-decoder), and send this image to the second stage for texture generation. Although structures and textures are produced on the output image, their appearances are not consistent. Figure 1 shows an example. The inconsistent structures and textures within hole regions produce blur and artifacts as shown in (b) and (c). Meanwhile, the recovered contents are not coherent to the uncorrupted contents around the hole boundaries (e.g., the leaves). This limitation is because of the independent learning of CNN features representing structures and textures. In practice, the structures and textures correlate with each other to formulate the image contents. Without considering their coherence, existing methods are not able to produce visually pleasing results.

In this work, we propose a mutual encoder-decoder to jointly learn CNN features representing structures and textures. The features from the deep layers of the encoder contain structure semantics while the features from the shallow layers contain texture details. The hole regions of these two features are filled via two separate branches. In the CNN feature space, we use a multi-scale filling block within each branch for hole filling. Each block consists of 3 partial convolution streams with progressively increased kernel sizes. After hole filling in these two features, we propose a feature equalization method to ensure the structure and texture features consistent with each other. Meanwhile, the equalized features are coherent with the features of uncorrupted image content around the hole boundaries. The proposed feature equalization consists of channel reweighing and bilateral propagation. We concatenate two features first and perform channel reweighing via attention exploration [12]. The attentions across two features are set to be consistent after channel equalization. Then, we propose a bilateral propagation activation function to equalize the feature consistency in the whole feature maps. This activation function uses elements on the global feature maps to propagate channel consistency (i.e., feature coherence across the hole boundaries), while using elements within local neighboring regions to maintain channel similarities (i.e., feature consistency within the hole). To this end, we fuse the texture and structure features together to reduce inconsistency in the CNN feature maps. The equalized features then supplement the decoder features in all the feature levels via encoder-decoder skip connections. The feature consistency is then reflected in the reconstructed output image, where the blur and artifacts are effectively removed around the hole regions as shown in Fig. 1(d). Experiments on the benchmark datasets show that the proposed method performs favorably against state-of-the-art approaches.

We summarize the contributions of this work as follows:

- We propose a mutual encoder-decoder network for image inpainting. The CNN features from the shallow layer are learned to represent textures and the features from deep layers represent structures.
- We propose a feature equalization method to make structure and texture features consistent with each other. We first reweigh channels after feature concatenation and propose a bilateral propagation activation function to make the whole feature consistent.
- Extensive experiments on the benchmark datasets show the effectiveness of the proposed inpainting method in removing blur and artifacts caused by inconsistent structure and texture features. The proposed method performs favorably against state-of-the-art inpainting approaches.

2 Related Works

Empirical Image Inpainting. The empirical image inpainting methods [1,3, 18] based on diffusion techniques propagate the neighborhood appearances to the missing regions. However, they only consider surrounding pixels of missing regions, which can only deal with small holes in background inpainting tasks and

may fail to generate meaningful structures. In contrast, methods [2,4,5,28,36] based on patch match fill missing regions by transferring similar and relevant patches from the remaining image region to the hole region. Although empirical methods perform well to handle small holes on the background inpainting task, they are not able to generate semantically meaningful content. When the hole region is large, these methods suffer from a lack of semantic guidance.

Deep Image Inpainting. Image inpainting based on deep learning typically involves the generative adversarial network [9] to supplement visual perceptual guidance for hole filling. Pathak et al. [25] first bring adversarial training [9] to inpainting and demonstrate semantic hole-filling. Iizuka et al. [13] propose local and global discriminators, assisted by dilated convolution [39] to improve the inpainting quality. Nazeri et al. [24] propose EdgeConnect that predicts salient edges for inpainting guidance. Song et al. [29] utilize a segmentation prediction network to generate segmentation guidance for detail refinement around the hole region. Xiong et al. [34] present foreground-aware inpainting, which involves three stages, i.e., contour detection, contour completion and image completion, for the disentanglement of structure inference and content hallucination. Ren et al. [26] introduce a structure-aware network, which splits the inpainting task into two parts: structure reconstruction and texture generation. It uses appearance flow to sample features from contextual regions. Yan et al. [37] speculate the relationship between the contextual regions in the encoder layer and the associated hole region in the decoder layer for better predictions. Yu et al. [40] and Song et al. [27] search for a collection of background patches with the highest similarity to the generated contents in the first stage prediction. Liu et al. [20] address this inpainting task via exploiting the partial convolutional layer and mask-update operation. Following the [20], Yu et al. [41] present gate convolution that learns a dynamic mask-updating mechanism and combines with the SN-PatchGAN discriminator to achieve better predictions. Liu et al. [21] propose coherent semantic attention, which considers the feature coherency of hole regions to guarantee the pixel continuity in image level. Wang et al. [32] propose a generative multi-column convolutional neural network (GMCNN) that uses varying receptive fields in branches. Different from existing deep inpainting methods, our method produces CNN features to consistently represent structures and textures to reduce blur and artifacts around the hole region.

3 Proposed Algorithm

Figure 2 shows the pipeline of the proposed method. We use one mutual encoder-decoder to jointly learn structure and texture features and equalize them for consistent representation. The details are presented in the following:

3.1 Mutual Encoder-Decoder

We use an encoder-decoder for end-to-end image generation to fill holes. The structure of this encoder-decoder is a simplified generative network [14], where

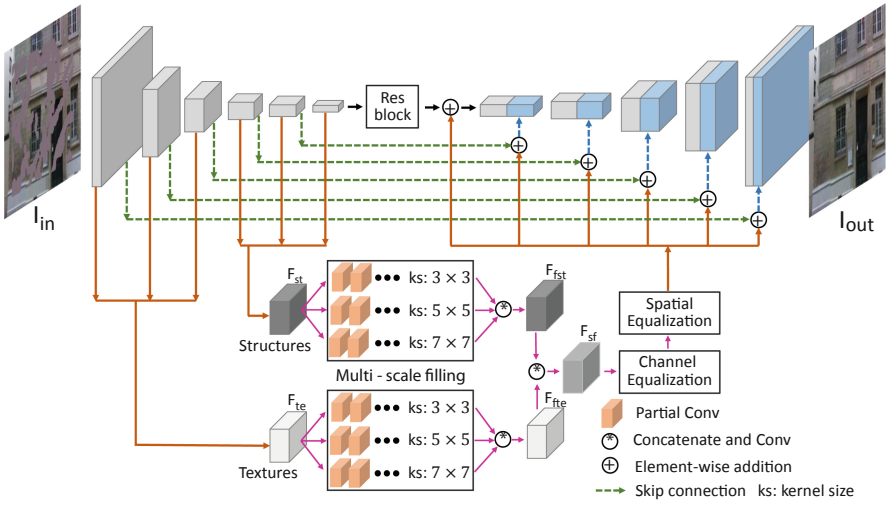


Fig. 2. The overview of the proposed pipeline. We use a mutual encoder-decoder to jointly recover structures and textures during hole filling. The deep layer features of the encoder are reorganized as structure features, while the shallow layer features are reorganized as texture features. We fill holes in multi-scales within the CNN feature space and equalize output features in both channel and spatial domains. The equalized features contain consistent structure and texture features at different CNN feature levels, and supplement the decoder via skip connections for output image generation.

there are 6 convolutional layers in the encoder and 5 convolutional layers in the decoder, respectively. Meanwhile, 4 residual blocks [10] with dilated convolutions are set between the encoders and decoders. The dilated convolutions [13,24] increase the size of the receptive field to perceive encoder features.

In the encoder, we reorganize the CNN features from deep layers as structure features where the semantics reside. Meanwhile, we reorganize the CNN features from shallow layers as texture features to represent image details. We denote the structure features as F_{st} and the texture features as F_{te} as shown in Fig. 2. The reorganization process is to resize and transform the CNN feature maps from different convolutional layers to the same size, and concatenate them accordingly.

After CNN feature reorganization, we design two branches (i.e., the structure branch and the texture branch) to separately perform hole filling on F_{te} and F_{st}. The architectures of these two branches are the same. In each branch, there are 3 parallel streams to fill holes in multiple scales. Each stream consists of 5 partial convolutions [20] with the same kernel size while the kernel size differs among different streams. By using different kernel sizes, we perform multi-scale filling in each branch for the input CNN features. The filled features from 3 streams (i.e., 3 scales) are concatenated and mapped to the same size of the input feature map via a 1×1 convolution. We denote the output of the structure branch as F_{fst}, and the output of the texture branch as F_{fte}. To ensure the hole filling to focus

on the textures and structures, we incorporate supervisions on F_{fst} and F_{fte}. We use a 1×1 convolution to separately map F_{fst} and F_{fte} to a color image I_{ost} and a color image I_{ote}, respectively. The pixel-wise L_1 loss can be written as follows:

$$\begin{aligned} L_{rst} &= \|I_{ost} - I_{st}\|_1 \\ L_{rte} &= \|I_{ote} - I_{gt}\|_1 \end{aligned} \tag{1}$$

where I_{gt} is the ground truth image and I_{st} is the structure image of I_{gt}. We use an edge-preserving smoothing method RTV [35] to generate I_{st} following [26].

The hole regions in F_{te} and F_{st} are filled via structure and texture branches, individually. The feature representations in F_{fte} and F_{fst} are not consistent to reflect the recovered structures and textures. This inconsistency leads to blur and artifacts within and around the hole regions as shown in Fig. 1. To mitigate these effects, we concatenate F_{fte} and F_{fst} first, and make a simple fusion to generate F_{sf} via a 1×1 convolutional layer. The texture and structure representations in F_{sf} are corrected via feature equalization at different CNN feature levels (i.e., across shallow to deep CNN layers).

3.2 Feature Equalizations

We equalize the fused CNN features F_{sf} in both channel and spatial domains. The channel equalization follows the squeeze and excitation operation [12] to ensure that the attentions within each channel of F_{sf} are the same. As the reweighed channels are influenced by both structure and texture representations in F_{sf}, the consistent attentions indicate that these representations are set to be consistent as well. We propagate channel equalization to the spatial domain via the proposed bilateral propagation activation function (BPA).

Formulation. BPA is inspired by the edge-preserving image smoothing [30] to generate response values based on spatial and range distances. It can be written as follows:

$$y_i^s = \frac{1}{C(x)} \sum_{j \in s} g_{\alpha_s}(\|j - i\|) x_j \tag{2}$$

$$y_i^r = \frac{1}{C(x)} \sum_{j \in v} f(x_i, x_j) x_j \tag{3}$$

$$y_i = q(y_i^s, y_i^r) \tag{4}$$

where x_i is the feature channel at position i of input feature x, x_j is a neighboring feature channel around i at position j, y_i^s and y_i^r are the feature channels after spatial and range similarity measurements. We set the normalization factor as $C(x) = N$, where N is the number of positions in x. We use q to denote the concatenation and channel reduction of y_i^s and y_i^r via a 1×1 convolutional layer.

The bilateral propagation utilizes the distances of feature channels from both spatial and range domains. We explore j within a neighboring region s, which is

set as the same spatial size as the input feature for global propagation. The spatial contributions from neighboring feature channels are adjusted via a Gaussian function g_{α_s}. When computing y_i^T, we measure the similarities between feature channels x_i and x_j via $f(.)$ within a neighboring region v around i. The size of v is 3×3. To this end, the bilateral propagation considers both global continuity via y_s^i and local consistency via y_r^i.

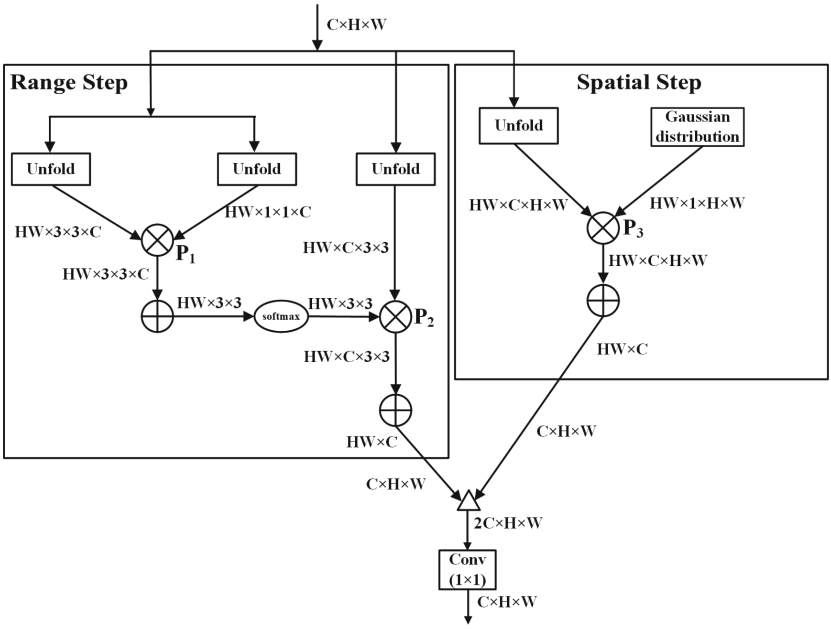


Fig. 3. The pipeline of the bilateral propagation activation function. We denote the broadcast dot product operation as $\otimes$, element-wise addition in the selected channel as $\oplus$, and the concatenation as $\triangle$. For two matrices with different dimensions, broadcast operations first broadcast features in each dimension to match the dimensions of the two matrices.

During the range similarity computation step, we define the pairwise function $f(.)$ as a dot product operation, which can be written as follows:

$$f(x_i, x_j) = (x_i)^T(x_j). \tag{5}$$

The proposed bilateral propagation shares similarity to the non-local block [31] that for each i, $\frac{1}{C(x)} f(x_i, x_j)$ becomes the softmax computation along dimension j. The difference resides on the region design of propagation. The non-local block uses feature channels from all the positions to generate y_i and the similarity is only measured between x_i and x_j. In contrast, BPA considers both feature channel similarity and spatial distance between x_i and x_j during bilateral weight computation. In addition, we use a global region s to compute

spatial distance while using a local region v to compute range distance. The advantage of global and local region selections is that we ensure both long-term continuity in the whole spatial region and local consistency around the current feature channel. The boundaries of hole regions are unified with the neighboring image content and the contents within the hole regions are set to be consistent.

Implementations. Figure 3 shows how bilateral propagation operates in the network. The range step corresponds to the computation of y_i^r in Eq. 3 and the spatial step corresponds to y_i^s in Eq. 2. During range computation, the operations until the element-wise multiplication P_1 represent Eq. 5 at all spatial locations. We use the unfold function in PyTorch to reshape the features to vectors (i.e., $HW \times 3 \times 3 \times C$) for obtaining all the neighboring x_j for each x_i, so that we can make efficient element-wise matrix multiplications. Similarly, the operations until P_2 represent the term $\sum_j f(x_i, x_j) \cdot x_j$ in Eq. 3. During spatial computation, the operations until P_3 represent the term $\sum_j g_{\alpha_s}(\|j - i\|)x_j$. As a result, the bilateral propagation operation can be efficiently executed via the element-wise matrix multiplications and additions shown in Fig. 3.

3.3 Loss Functions

We introduce several loss functions to measure structure and texture differences including pixel reconstruction loss, perceptual loss, style loss, and relativistic average LS adversarial loss [16] during training. We also employ a discriminator with local and global operations to ensure local-global contents consistency. And the spectral normalization [23] is applied in both local and global discriminators to achieve stable training.

Pixel Reconstruction Loss. We measure the pixel-wise difference from two aspects. The first one is the loss terms illustrated in Eq. 1 where we add supervisions on the texture and structure branches. The second one measures the similarity between the network output and the ground truth, which can be written as follows:

$$L_{re} = \|I_{out} - I_{gt}\|_1 \tag{6}$$

where I_{out} is the finally predicted image by the network.

Perceptual Loss. To capture the high-level semantics and simulate human perception of images quality, we utilize the perceptual loss [15] L_{perc} defined on the ImageNet-pretrained VGG-16 feature backbone:

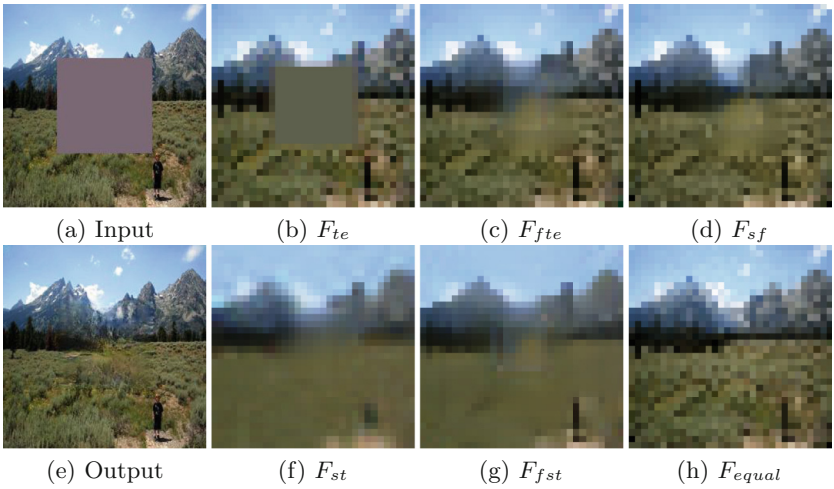
$$L_{prec} = \mathbb{E}\Big[\sum_i \frac{1}{N_i}\|\Phi_i(I_{out}) - \Phi_i(I_{gt})\|_1\Big] \tag{7}$$

where Φ_i is the activation map of the i-th layer of the VGG-16 backbone. In our work, Φ_i corresponds to the activation maps from layers ReLu1_1, ReLu2_1, ReLu3_1, ReLu4_1, and ReLu5_1.

Style Loss. The transposed convolutional layers from the decoder will bring artifacts that resemble checkerboard. To mitigate this effect, we introduce the style loss. Given feature maps of size $C_j \times H_j \times W_j$, we compute the style loss as follows:

$$L_{style} = \mathbb{E}_j \left[\| G_j^{\Phi}(I_{out}) - G_j^{\Phi}(I_{gt}) \|_1 \right] \tag{8}$$

where G_j^{Φ} is a $C_j \times C_j$ Gram matrix constructed from the selected activation maps. These activation maps are the same as those used in the perceptual loss.



(a) Input (b) F_{te} (c) F_{fte} (d) F_{sf}

(e) Output (f) F_{st} (g) F_{fst} (h) F_{equal}

Fig. 4. Visualization of the feature map response. The input and output images are shown in (a) and (e), respectively. We use a 1×1 convolutional layer to map high dimensional feature maps to the color images as shown in (b)–(d) and (f)–(h).

Relativistic Average LS Adversarial Loss. We follow [40] to utilize global and local discriminators for perception enhancement. The relativistic average LS adversarial loss is adopted for our discriminators. For the generator, the adversarial loss is defined as:

$$L_{adv} = -\mathbb{E}_{x_r}[\log(1 - D_{ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{ra}(x_f, x_r))] \tag{9}$$

where $D_{ra}(x_r, x_f) = \text{sigmoid}(C(x_r) - \mathbb{E}_{x_f}[C(x_f)])$ and $C(.)$ indicates the local or global discriminator without the last sigmoid function. To this end, real and fake data pairs (x_r, x_f) are sampled from the ground-truth and output images.

Total Losses. The whole objective function of the proposed network can be written as:

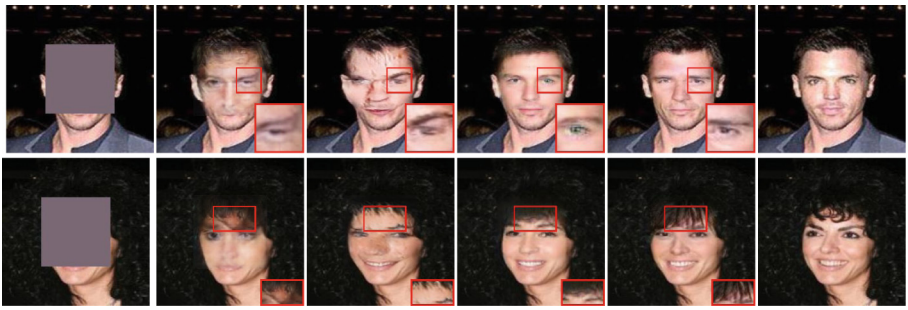
$$L_{total} = \lambda_r L_{re} + \lambda_p L_{prec} + \lambda_s L_{style} + \lambda_{adv} L_{adv} + \lambda_{st} L_{rst} + \lambda_{te} L_{rte} \tag{10}$$

where λ_r, λ_p, λ_s, λ_{adv}, λ_{st} and λ_{te} are the tradeoff parameters. In our implementation, we empirically set $\lambda_r = 1$, $\lambda_p = 0.1$, $\lambda_s = 250$, $\lambda_{adv} = 0.2$, $\lambda_{st} = 1$, $\lambda_{te} = 1$.

3.4 Visualizations

We use a structure branch and a texture branch to separately fill holes in CNN feature space. Then, we perform feature equalization to enable consistent feature representations in different feature levels for output image reconstruction. In this section, we visualize the feature maps during different steps to show whether they correspond to our objectives. We use a 1×1 convolutional layer to map CNN feature maps to color images for a clear display.

Figure 4 shows the visualization results. The input image is shown in (a) with a mask in the center. The visualized F_{te} and F_{st} are shown in (b) and (f), respectively. We observe that textures are preserved in (b) while the structures are in (f). By multi-scale hole filling, the hole regions in F_{fte} and F_{fst} are effectively reduced as shown in (c) and (g). After equalization, the hole regions in (h) are effectively filled and the equalized features contribute to the decoders to generate the output image as shown in (e).



(a) Input (b) CE [25] (c) CA [40] (d) SH [37] (e) Ours (f) GT

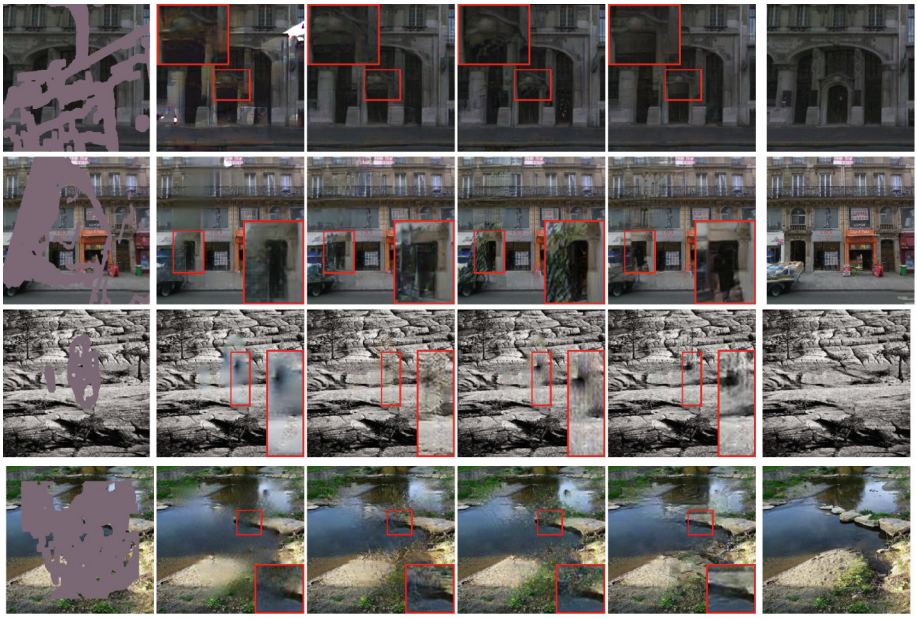
Fig. 5. Visual evaluations for filling center holes. Our method performs favorably against existing approaches to retain both structures and textures.

4 Experiments

We evaluate our method on three datasets: Paris StreetView [6], Place2 [43] and CelebA [22]. We follow the training, testing, and validation splits of these three datasets. Data augmentation such as flipping is also adopted during training. Our model is optimized by the Adam optimizer [17] with a learning rate of 2×10^{-4} on a single NVIDIA 2080TI GPU. The training process of the CelebA model, Paris StreetView model and Place2 model are stopped after 6 epochs, 30 epochs and 60 epochs, respectively. All the masks and images for training and testing are with the size of 256×256.

We compare our method with six state-of-the-art method: CE [25], CA [40], SH [37], CSA [21], SF [26] and GC [41]. For a fair evaluation on model generalization abilities, we conduct experiments on filling center holes and irregular holes

on the input images. The center hole is brought by a mask that covers the image center with a size of 128×128. We obtain irregular masks from PConv [20]. These masks are in different categories according to the ratios of the hole regions versus the entire image size (i.e., below 10%, from 10% to 20%, etc.). For holes in the image center, we compare with CA [40], SH [37] and CE [25] on the CelebA [22] validation set. We choose these three methods because they are more effective to fill holes in the image center than fill irregular holes. When handling irregular holes on the input images, we compare with CSA [21], SF [26] and GC [41] using Paris StreetView [6] and Place2 [43] validation datasets.



(a) Input (b) GC [41] (c) SF [26] (d) CSA [21] (e) Ours (f) GT

Fig. 6. Visual evaluations for filling irregular holes. Our method performs favorably against existing approaches to retain both structures and textures.

4.1 Visual Evaluations

The visual comparison on the results for filling center holes are in Fig. 5 and the results for filling irregular holes are in Fig. 6. We also display ground truth images in (f) to show the actual image content. In Fig. 5, the input images are shown in (a). The results produced by CE and CA contain distorted structures and blurry textures as shown in (b) and (c). Although more visually pleasing contents are generated in (d), the semantics remain unreasonable. By utilizing

Table 1. Numerical evaluations on the CelebA dataset where the inputs are with centering hole regions. ↓ indicates lower is better while ↑ indicates higher is better.

	CE	CA	SH	Ours
FID↓	52.17	37.61	29.72	**25.51**
PSNR↑	8.53	23.65	26.10	**26.32**
SSIM↑	0.137	0.870	0.902	**0.910**

Table 2. Numerical comparisons on the Place2 dataset. ↓ indicates lower is better while ↑ indicates higher is better.

	Mask	GC	SF	CSA	Ours
FID↓	10–20%	19.04	8.78	7.85	**6.91**
	20–30%	28.45	16.38	13.95	**8.06**
	30–40%	40.71	27.54	25.74	**19.36**
	40–50%	60.72	40.93	38.74	**28.79**
PSNR↑	10–20%	27.10	29.50	**31.31**	31.13
	20–30%	25.18	27.22	28.66	**28.87**
	30–40%	22.51	24.37	25.01	**25.34**
	40–50%	20.35	21.90	22.54	**22.81**
SSIM↑	10–20%	0.929	0.926	0.954	**0.957**
	20–30%	0.878	0.885	0.918	**0.923**
	30–40%	0.823	0.802	0.843	**0.854**
	40–50%	0.670	0.678	0.702	**0.719**

consistent structure and texture features, our method is effective to generate results with realistic textures.

Figure 6 shows the comparison for filling irregular holes, which is more challenging than filling centering holes. The results from GC contain noisy patterns shown in (b). The details are missing and the structures are distorted in (c) and (d). These methods are not effective to recover image contents without bringing in obvious artifacts (i.e., the second row around the door regions). In contrast, our method learns to represent structures and textures in a consistent formation. The results shown in (e) indicate the effectiveness of our method to produce visually pleasing contents. The evaluations on filling both centering holes and irregular holes indicate that our method performs favorably against existing hole filling approaches.

4.2 Numerical Evaluations

We conduct numerical evaluations on the Place2 dataset with different mask ratios. Besides, we evaluate numerically on the CelebA dataset with centering holes in the input images. There are 100 validation images from the "valley"

scene category chosen for evaluations. In CelebA, we randomly choose 500 images for evaluation. For the evaluation metrics, we follow [26] to use SSIM [33] and PSNR. Moreover, we introduce FID (Frèchet Inception Distance) metric [11] as it indicates the perceptual quality of the results. The evaluation results are shown in Tables 1 and 2. Our method outperforms existing methods to fill centering holes. Meanwhile, favorable performance is achieved by our method to fill irregular holes under various hole versus image ratios.

Human Subject Evaluation. We follow [42] to involve over 35 volunteers for evaluating the results on CelebA, Place2 and Paris StreetView datasets. The volunteers are all image experts with image processing background. There are 20 questions for each subject. In each question, the subject needs to select the most realistic result from 4 results generated by different methods without knowing the hole region in advance. We tally the votes and show the statistics in Table 3. Our method performs favorably against existing methods.

Table 3. Human Subject Evaluation results. Each subject selects the most realistic result without knowing hole regions in advance.

	CE	CA	SH	GC	SF	CSA	Ours
Paris StreetView	N/A	N/A	N/A	5.3%	21.0%	29.8%	**43.7%**
Place2	N/A	N/A	N/A	3.0%	25.0%	29.6%	**42.4%**
CelebA	1.2%	2.0%	40.4%	N/A	N/A	N/A	**56.4%**

Table 4. Ablation study on the Paris StreetView dataset. Our performance is improved by using structure and texture branches.

	Ours without textures	Ours without structures	Ours
FID↓	30.37	27.46	**25.10**
PSNR↑	22.80	22.96	**23.38**
SSIM↑	0.818	0.823	**0.833**

Table 5. Ablation study on the Place2 dataset. Non-local aggregation improves our baseline while feature equalization makes further improvement.

	Ours without equalization	Non-local aggregation	Ours
FID↓	29.11	24.07	**21.26**
PSNR↑	23.14	23.64	**24.57**
SSIM↑	0.837	0.848	**0.852**

(a) Input image (b) Ours w/o textures (c) Ours w/o structures (d) Ours (e) Ground truth

Fig. 7. Abalation studies on structure and texture branches. A joint utilization of these two branches improves the content quality.

(a) Input image (b) Ours w/o equalization (c) Non-Local aggregation (d) Ours (e) Ground truth

Fig. 8. Ablation studies on feature equalizations. More realistic and visually pleasing contents are generated via feature equalizations.

5 Ablation Study

Structure and Texture Branches. To evaluate the effects of structure and texture branches, we use each of these branches separately for network training. For fair comparisons, we expand the channel number of the texture and structure branch outputs via additional convolutions. So the single branch output contains the same size as that of F_{sf}. As shown in Fig. 7, the output of our method without a texture branch contains rich structure information (i.e., the window in the red and green boxes) while the textures are missing. In comparison, the output of our method without a structure branch does not contain meaningful structure (i.e., the window in the red and green boxes). By utilizing both branches, our method achieves favorable results on both structures and textures. Table 4 shows the similar numerical performance on the Paris StreetView dataset where these two branches improve our method significantly.

Feature Equalizations. We show the contributions of feature equalizations by removing them from the pipeline and showing the performance degradation. Moreover, we show that the bilateral propagation activation function (BPA) is more effective to fill hole regions than the Non-local attentions [31]. As shown in Fig. 8, without using equalization our method generates visually unpleasant contents and visible artifacts. In comparison, the contents generated by [31] are more natural. However, the recovered contents are still blurry and inconsistent because the Non-local block ignores the local coherency and global distance

of features. This limitation is effectively solved via our method with feature equalizations. Similar performance has been shown numerically in Table 5 where our method achieves favorable results.

6 Concluding Remarks

We propose a mutual encoder-decoder with feature equalizations to correlate filled structures with textures during image inpainting. The shallow and deep layer features are reorganized as texture and structure features, respectively. In the CNN feature space, we introduce a texture branch and a structure branch to fill holes in multi-scales and fuse the outputs together via feature equalizations. During equalization, we first ensure consistent attentions among individual channels and propagate them to the whole spatial feature map region via the proposed bilateral propagation activation function. The experiments carried out over the benchmark datasets have shown the effectiveness of the proposed method when compared to state-of-the-art approaches on filling both regular and irregular hole regions.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China under Grant No. 61702176.

References

1. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. TIP **10**, 1200–1211 (2001)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: PatchMatch: a randomized correspondence algorithm for structural image editing. In: SIGGRAPH (2009)
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH (2000)
4. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. TIP **13**, 1200–1212 (2004)
5. Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P.: Image melding: combining inconsistent images using patch-based synthesis. ACM Trans. Graph. **31**, 18 (2012)
6. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes Paris look like Paris? Commun. ACM **58**, 103–110 (2015)
7. Efros, A., Freeman, W.: Image quilting for texture synthesis and transfer. In: SIGGRAPH (2001)
8. Efros, A., Freeman, W.: Texture synthesis by nonparametric sampling. In: ICCV (2001)
9. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: NIPS (2017)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)

13. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. In: SIGGRAPH (2017)
14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
16. Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard GAN. In: ICLR (2018)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Levin, A., Zomet, A., Weiss, Y.: Learning how to inpaint from global image statistics. In: ICCV (2003)
19. Li, Y., Liu, S., Yang, J., Yang, M.H.: Generative face completion. In: CVPR (2017)
20. Liu, G., Reda, F.A., Shih, K.J., Wang, T.-C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 89–105. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_6
21. Liu, H., Jiang, B., Xiao, Y., Yang, C.: Coherent semantic attention for image inpainting. In: ICCV (2019)
22. Liu, Z., LuoPi, n., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
23. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
24. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: EdgeConnect: generative image inpainting with adversarial edge learning. In: ICCV Workshops (2019)
25. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: feature learning by inpainting. In: CVPR (2016)
26. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: StructureFlow: image inpainting via structure-aware appearance flow. In: ICCV (2019)
27. Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Kuo, C.-C.J.: Contextual-based image inpainting: infer, match, and translate. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 3–18. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_1
28. Song, Y., Bao, L., He, S., Yang, Q., Yang, M.H.: Stylizing face images via multiple exemplars. CVIU **162**, 135–145 (2017)
29. Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., Kuo, J.: SPG-Net: segmentation prediction and guidance network for image inpainting. arXiv preprint arXiv:1805.03356 (2018)
30. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: CVPR (1998)
31. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
32. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: NIPS (2018)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**, 600–612 (2004)
34. Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: CVPR (2019)

35. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. SIGGRAPH **31**, 139 (2012)
36. Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. TIP **19**, 1153–1165 (2010)
37. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-Net: image inpainting via deep feature rearrangement. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_1
38. Yeh, R., Chen, C., Lim, T., Johnson, M.H., Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arXiv preprint arXiv:1607.07539 (2016)
39. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
40. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: CVPR (2018)
41. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV (2019)
42. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: CVPR (2019)
43. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. PAMI **40**, 1452–1464 (2017)