Image Inpainting with Learnable Bidirectional Attention Maps

Chaohao Xie[1†], Shaohui Liu[1,3], Chao Li[2], Ming-Ming Cheng[4], Wangmeng Zuo[1,3*], Xiao Liu[2], Shilei Wen[2], Errui Ding[2]
[1]Harbin Institute of Technology, [2]Department of Computer Vision Technology (VIS), Baidu Inc.
[3]Peng Cheng Laboratory, Shenzhen, [4]Nankai University
viousxie@outlook.com, {shliu, wmzuo}@hit.edu.cn, cmm@nankai.edu.cn
{lichao40, liuxiao12, wenshilei, dingerrui}@baidu.com

Abstract

Most convolutional network (CNN)-based inpainting methods adopt standard convolution to indistinguishably treat valid pixels and holes, making them limited in handling irregular holes and more likely to generate inpainting results with color discrepancy and blurriness. Partial convolution has been suggested to address this issue, but it adopts handcrafted feature re-normalization, and only considers forward mask-updating. In this paper, we present a learnable attention map module for learning feature re-normalization and mask-updating in an end-to-end manner, which is effective in adapting to irregular holes and propagation of convolution layers. Furthermore, learnable reverse attention maps are introduced to allow the decoder of U-Net to concentrate on filling in irregular holes instead of reconstructing both holes and known regions, resulting in our learnable bidirectional attention maps. Qualitative and quantitative experiments show that our method performs favorably against state-of-the-arts in generating sharper, more coherent and visually plausible inpainting results. The source code and pre-trained models will be available at: `https://github.com/Vious/LBAM_inpainting/`.

1. Introduction

Image inpainting [3], aiming at filling in holes of an image, is a representative low level vision task with many real-world applications such as distracting object removal, occluded region completion, etc. However, there may exist multiple potential solutions for the given holes in an image, *i.e.*, the holes can be filled with any plausible hypotheses coherent with the surrounding known regions. And the holes can be of complex and irregular patterns, further increasing the difficulty of image inpainting. Traditional exemplar-based methods [2, 18, 32], *e.g.*, PatchMatch [2], gradually fill in holes by searching and copying similar patches from known regions. Albeit exemplar-based methods are effective in hallucinating detailed textures, they are still limited in capturing high-level semantics, and may fail to generate complex and non-repetitive structures (see Fig. 1(c)).

Recently, considerable progress has been made in applying deep convolutional networks (CNNs) to image inpainting [10, 20]. Benefited from the powerful representation ability and large scale training, CNN-based methods are effective in hallucinating semantically plausible result. And adversarial loss [8] has also been deployed to improve the perceptual quality and naturalness of the result. Nonetheless, most existing CNN-based methods usually adopt standard convolution which indistinguishably treats valid pixels and holes. Thus, they are limited in handling irregular holes and more likely to generate inpainting results with color discrepancy and blurriness. As a remedy, several post-processing techniques [10, 34] have been introduced but are still inadequate in resolving the artifacts (see Fig. 1(d)).

CNN-based methods have also been combined with exemplar-based one to explicitly incorporate the mask of holes for better structure recovery and detail enhancement [26, 33, 36]. In these methods, the mask is utilized to guide the propagation of the encoder features from known regions to the holes. However, the copying and enhancing operation heavily increases the computational cost and is only deployed at one encoding and decoding layers. As a result, they are better at filling in rectangular holes, and perform poorly on handling irregular holes (see Fig. 1(e)).

For better handling irregular holes and suppressing color discrepancy and blurriness, partial convolution (PConv) [17] has been suggested. In each PConv layer, mask convolution is used to make the output conditioned only on the unmasked input, and feature re-normalization is introduced for scaling the convolution output. A mask-updating rule is further presented to update a mask for the next layer, making PConv very effective in handling irregular holes. Nonetheless, PConv adopts hard 0-1 mask and handcrafted feature re-normalization by absolutely trusting all filling-in intermediate features. Moreover, PConv con-

[†]This work was done when Chaohao Xie was a research intern at Baidu
[*]Corresponding author

(a) Original (b) Input (c) PM [2] (d) GL [10] (e) CA [36] (f) PConv [17] (g) Ours

Figure 1. Qualitative comparison of inpainting results by PatchMatch (PM) [2], Global&Local (GL) [10], Context Attention (CA) [36], and Partial Convolution (PConv) [17], and Ours.

siders only forward mask-updating and simply employs all-one mask for decoder features.

In this paper, we take a step forward and present the modules of learnable bidirectional attention maps for the re-normalization of features on both encoder and decoder of the U-Net [22] architecture. To begin with, we revisit PConv without bias, and show that the mask convolution can be safely avoided and the feature re-normalization can be interpreted as a re-normalization guided by hard 0-1 mask. To overcome the limitations of hard 0-1 mask and handcrafted mask-updating, we present a learnable attention map module for learning feature re-normalization and mask-updating. Benefited from the end-to-end training, the learnable attention map is effective in adapting to irregular holes and propagation of convolution layers.

Furthermore, PConv simply uses all-one mask on the decoder features, making the decoder should hallucinate both holes and known regions. Note that the encoder features of known region will be concatenated, it is natural that the decoder is only required to focus on the inpainting of holes. Therefore, we further introduce learnable reverse attention maps to allow the decoder of U-Net concentrate only on filling in holes, resulting in our learnable bidirectional attention maps. In contrast to PConv, the deployment of learnable bidirectional attention maps empirically is beneficial to network training, making it feasible to include adversarial loss for improving visual quality of the result.

Qualitative and quantitative experiments are conducted on the Paris SteetView [6] and Places [40] datasets to evaluate our proposed method. The results show that our proposed method performs favorably against state-of-the-arts in generating sharper, more coherent and visually plausible inpainting results. From Fig. 1(f)(g), our method is more effective in hallucinating clean semantic structure and realistic textures in comparison to PConv. To sum up, the main contribution of this work is three-fold,

- A learnable attention map module is presented for image inpainting. In contrast to PConv, the learnable attention maps are more effective in adapting to arbitrary irregular holes and propagation of convolution layers.
- Forward and reverse attention maps are incorporated to constitute our learnable bidirectional attention maps, further benefiting the visual quality of the result.
- Experiments on two datasets and real-world object removal show that our method performs favorably against state-of-the-arts in hallucinating shaper, more coherent and visually plausible results.

2. Related Work

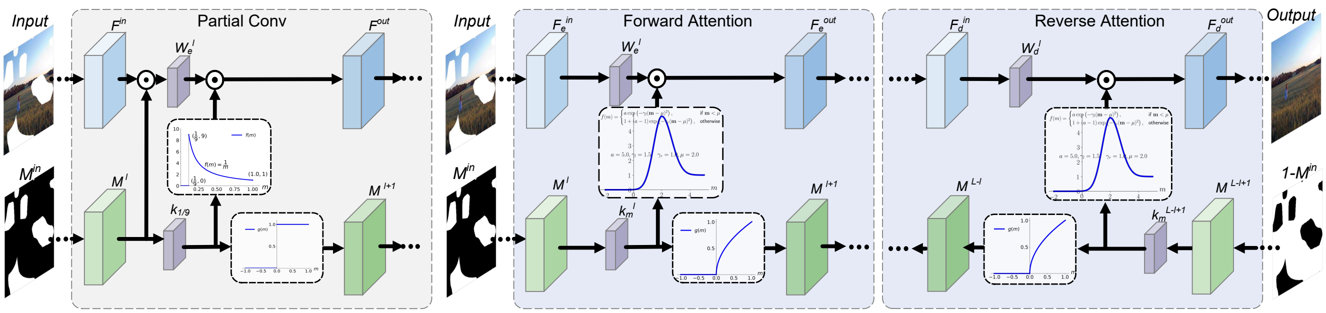
In this section, we present a brief survey on the relevant work, especially the propagation process adopted in exemplar-based methods as well as the network architectures of CNN-based inpainting methods.

2.1. Exemplar-based Inpainting

Most exemplar-based inpainting methods search and paste from the known regions to gradually fill in the holes from the exterior to the interior [2, 4, 18, 32], and their results highly depend on the propagation process. In general, better inpainting result can be attained by first filling in structures and then other missing regions. To guide the patch processing order, patch priority [15, 29] measure has been introduced as the product of confidence term and data term. While the confidence term is generally defined as the ratio of known pixels in the input patch, several forms of data terms have been proposed. In particular, Criminisi *et al*. [4] suggested a gradient-based data term for filling in linear structure with higher priority. Xu and Sun [32] assumed that structural patches are sparsely distributed in an image, and presented a sparsity-based data term. Le Meur *et al*. [18] adopted the eigenvalue discrepancy of structure tensor [5] as an indicator of structural patch.

2.2. Deep CNN-based Inpainting

Early CNN-based methods [14, 21, 30] are suggested for handling images with small and thin holes. In the past few years, deep CNNs have received upsurging interest and exhibited promising performance for filling in large holes.



(a) PConv (b) Learnable forward attention map (c) Learnable reverse attention map

Figure 2. Interplay models between mask and intermediate feature for PConv and our learnable bidirectional attention maps. Here, the white holes in $\mathbf{M}^{in}$ denotes missing region with value 0, and the black area denotes the known region with value 1.

Phatak *et al*. [20] adopted an encoder-decoder network (*i.e*., context-encoder), and incorporated reconstruction and adversarial losses for better recovering semantic structures. Iizuka *et al*. [10] combined both global and local discriminators for reproducing both semantically plausible structures and locally realistic details. Wang *et al*. [28] suggested a generative multi-column CNN incorporating with confidence-driven reconstruction loss and implicit diversified MRF (ID-MRF) term.

Multi-stage methods have also been investigated to ease the difficulty of training deep inpainting networks. Zhang *et al*. [37] presented a progressive generative networks (PGN) for filling in holes with multiple phases, while LSTM is deployed to exploit the dependencies across phases. Nazeri *et al*. [19] proposed a two-stage model EdgeConnect first predicting salient edges and then generating inpainting result guided by edges. Instead, Xiong *et al*. [31] presented 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.

In order to combine exemplar-based and CNN-based methods, Yang *et al*. [34] suggested multi-scale neural patch synthesis (MNPS) to refine the result of context-encoder via joint optimization with the holistic content and local texture constraints. Other two-stage feed-forward models, *e.g*., contextual attention [26] and patch-swap [36], are further developed to overcome the high computational cost of MNPS while explicitly exploiting image features of known regions. Concurrently, Yan *et al*. [33] modified the U-Net to form an one-stage network, *i.e*., Shift-Net, to utilize the shift of encoder feature from known regions for better reproducing plausible semantics and detailed contents. Most recently, Zheng *et al*. [39] introduced an enhanced short+long term attention layer, and presented a probabilistic framework with two parallel paths for pluralistic inpainting.

Most existing CNN-based inpainting methods are usually not well suited for handling irregular holes. To address this issue, Liu *et al*. [17] proposed a partial convolution (PConv) layer involving three steps, *i.e*., mask convolution, feature re-normalization, and mask-updating. Yu *et al*. [35] provided gated convolution which learns channel-wise soft mask by considering both corrupted images, masks and user sketches. However, PConv adopts handcrafted feature re-normalization and only considers forward mask-updating, making it still limited in handling color discrepancy and blurriness (see Fig. 1(d)).

3. Proposed Method

In this section, we first revisit PConv, and then present our learnable bidirectional attention maps. Subsequently, the network architecture and learning objective of our method are also provided.

3.1. Revisiting Partial Convolution

A PConv [17] layer generally involves three steps, *i.e*., (i) mask convolution, (ii) feature re-normalization, and (iii) mask-updating. Denote by $\mathbf{F}^{in}$ the input feature map and $\mathbf{M}$ the corresponding hard 0-1 mask. We further let $\mathbf{W}$ be the convolution filter and b be its bias. To begin with, we introduce the convolved mask $\mathbf{M}^c = \mathbf{M} \otimes \mathbf{k}_{\frac{1}{9}}$, where $\otimes$ denotes the convolution operator, $\mathbf{k}_{\frac{1}{9}}$ denotes a 3×3 convolution filter with each element $\frac{1}{9}$. The process of PConv can be formulated as,

$$\text{(i) } \mathbf{F}^{conv} = \mathbf{W}^T(\mathbf{F}^{in} \odot \mathbf{M}), \tag{1}$$

$$\text{(ii) } \mathbf{F}^{out} = \begin{cases} \mathbf{F}^{conv} \odot f_A(\mathbf{M}^c) + b, & \text{if } \mathbf{M}^c > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\text{(iii) } \mathbf{M}' = f_M(\mathbf{M}^c) \tag{3}$$

where $\mathbf{A} = f_A(\mathbf{M}^c)$ denotes the attention map, and $\mathbf{M}' = f_M(\mathbf{M}^c)$ denotes the updated mask. We further define the activation functions for attention map and updated mask as,

$$f_A(\mathbf{M}^c) = \begin{cases} \frac{1}{\mathbf{M}^c}, & \text{if } \mathbf{M}^c > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$f_M(\mathbf{M}^c) = \begin{cases} 1, & \text{if } \mathbf{M}^c > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

From Eqns. (1)$\sim$(5) and Fig. 2(a), PConv can also be explained as a special interplay model between mask and

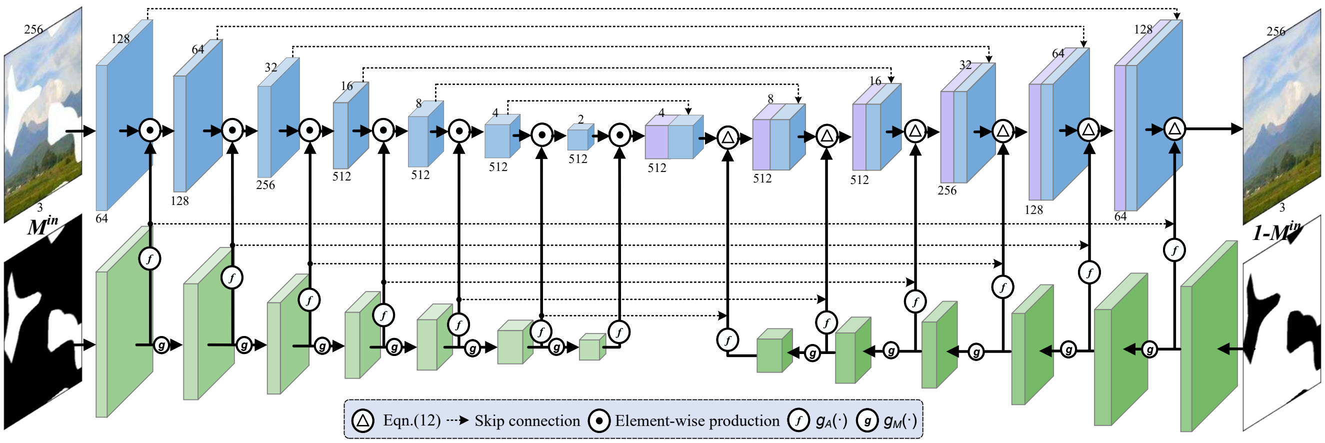


Figure 3. The network architecture of our model. The circle with triangle inside denotes operation form of Eqn.(12), g_A and g_M represent activation functions of Eqn.(9) and mask updating function of Eqn.(8).

convolution feature map. However, PConv adopts the handcrafted convolution filter $\mathbf{k}_{\frac{1}{9}}$ as well as handcrafted activation functions $f_A(\mathbf{M}^c)$ and $f_M(\mathbf{M}^c)$, thereby giving some leeway for further improvements. Moreover, the non-differential property of $f_M(\mathbf{M}^c)$ also increases the difficulty of end-to-end learning. To our best knowledge, it remains a difficult issue to incorporate adversarial loss to train a U-Net with PConv. Furthermore, PConv only considers the mask and its updating for encoder features. As for decoder features, it simply adopts all-one mask, making PConv limited in filling holes.

3.2. Learnable Attention Maps

The convolution layer without bias has been widely adopted in U-Net for image-to-image translation [11] and image inpainting [33]. When the bias is removed, it can be readily seen from Eqn. (2) that the convolution features in updated holes are zeros. Thus, the mask convolution in Eqn. (1) is equivalently rewritten as standard convolution,

$$\text{(i) } \mathbf{F}^{conv} = \mathbf{W}^T \mathbf{F}^{in}. \tag{6}$$

Then, the feature re-normalization in Eqn. (2) can be interpreted as the element-wise product of convolution feature and attention map,

$$\text{(ii) } \mathbf{F}^{out} = \mathbf{F}^{conv} \odot f_A(\mathbf{M}^c). \tag{7}$$

Even though, the handcrafted convolution filter $\mathbf{k}_{\frac{1}{9}}$ is fixed and not adapted to the mask. The activation function for updated mask absolutely trusts the inpainting result in the region $\mathbf{M}^c > 0$, but it is more sensible to assign higher confidence to the region with higher $\mathbf{M}^c$.

To overcome the above limitations, we suggest learnable attention map which generalizes PConv without bias from three aspects. First, to make the mask adaptive to irregular holes and propagation along with layers, we substitute $\mathbf{k}_{\frac{1}{9}}$ with layer-wise and learnable convolution filters $\mathbf{k}_{\mathbf{M}}$. Second, instead of hard 0-1 mask-updating, we modify the activation function for updated mask as,

$$g_M(\mathbf{M}^c) = (ReLU(\mathbf{M}^c))^{\alpha}, \tag{8}$$

where $\alpha \geq 0$ is a hyperparameter and we set $\alpha = 0.8$. One can see that $g_M(\mathbf{M}^c)$ degenerates into $f_M(\mathbf{M}^c)$ when $\alpha = 0$. Third, we introduce an asymmetric Gaussian-shaped form as the activation function for attention map,

$$g_A(\mathbf{M}^c) = \begin{cases} a \exp\left(-\gamma_l (\mathbf{M}^c - \mu)^2\right), & \text{if } \mathbf{M}^c < \mu \\ 1 + (a-1) \exp\left(-\gamma_r (\mathbf{M}^c - \mu)^2\right), & \text{else} \end{cases} \tag{9}$$

where a, μ, γ_l, and γ_r are the learnable parameters, we initialize them as $a = 1.1, \mu = 2.0, \gamma_l = 1.0, \gamma_r = 1.0$ and learn them in an end-to-end manner.

To sum up, the learnable attention map adopt Eqn. (6) in Step (i), and the next two steps are formulated as,

$$\text{(ii) } \mathbf{F}^{out} = \mathbf{F}^{conv} \odot g_A(\mathbf{M}^c), \tag{10}$$

$$\text{(iii) } \mathbf{M}' = g_M(\mathbf{M}^c). \tag{11}$$

Fig. 2(b) illustrates the interplay model of learnable attention map. In contrast to PConv, our learnable attention map is more flexible and can be end-to-end trained, making it effective in adapting to irregular holes and propagation of convolution layers.

3.3. Learnable Bidirectional Attention Maps

When incorporating PConv with U-Net for inpainting, the method [17] only updates the masks along with the convolution layers for encoder features. However, all-one mask is generally adopted for decoder features. As a result, the $(L-l)$-th layer of decoder feature in both known regions and holes should be hallucinated using both $(l+1)$-th layer of encoder feature and $(L-l-1)$-th layer of decoder feature. Actually, the l-th layer of encoder feature will be concatenated with the $(L-l)$-th layer of decoder feature, and we can only focus on the generation of the $(L-l)$-th layer of decoder feature in the holes.

Figure 4. Qualitative comparison on Paris StreetView dataset. Comparison with PatchMatch (PM) [2], Global&Local(GL) [10], Context Attention(CA) [36], PConv [17] and Ours.

We further introduce learnable reverse attention maps to the decoder features. Denote by $\mathbf{M}_e^c$ the convolved mask for encoder feature $\mathbf{F}_e^{in}$. Let $\mathbf{M}_d^c = \mathbf{M}_d \otimes \mathbf{k}_{\mathbf{M}_d}$ be the convolved mask for decoder feature $\mathbf{F}_d^{in}$. The first two steps of learnable reverse attention map can be formulated as,

$$\text{(i\&ii) } \mathbf{F}_d^{out} = (\mathbf{W}_e^T \mathbf{F}_e^{in}) \odot g_A(\mathbf{M}_e^c) + (\mathbf{W}_d^T \mathbf{F}_d^{in}) \odot g_A(\mathbf{M}_d^c). \quad (12)$$

where $\mathbf{W}_e$ and $\mathbf{W}_d$ are the convolution filters. And we define $g_A(\mathbf{M}_d^c)$ as the reverse attention map. Then, the mask $\mathbf{M}_d^c$ is updated and deployed to the former decoder layer,

$$\text{(iii) } \mathbf{M}_d' = g_M(\mathbf{M}_d^c). \quad (13)$$

Fig. 2(c) illustrates the interplay model of reverse attention map. In contrast to forward attention maps, both encoder feature (mask) and decoder feature (mask) are considered. Moreover, the updated mask in reverse attention map is applied to the former decoder layer, while that in forward attention map is applied to the next encoder layer.

By incorporating forward and reverse attention maps with U-Net, Fig. 3 shows the full learnable bidirectional attention maps. Given an input image I^{in} with irregular holes, we use $\mathbf{M}^{in}$ to denote the binary mask, where ones indicate the valid pixels and zeros indicate the pixels in holes. From Fig. 3, the forward attention maps take $\mathbf{M}^{in}$ as the input mask for the re-normalization of the first layer of encoder feature, and gradually update and apply the mask to next encoder layer. In contrast, the reverse attention maps take $1 - \mathbf{M}^{in}$ as the input for the re-normalization of the last (*i.e.*, L-th) layer of decoder feature, and gradually update and apply the mask to former decoder layer. Benefited from the end-to-end learning, our learnable bidirectional attention maps (LBAM) are more effective in handling irregular holes. The introduction of reverse attention maps allows the decoder concentrate only on filling in irregular holes, which is also helpful to inpainting performance. Our LBAM is also beneficial to network training, making it feasible to exploit adversarial loss for improving visual quality.

3.4. Model Architecture

We modify the U-Net architecture [11] of 14 layers by removing the bottleneck layer and incorporating with bidirectional attention maps (see Fig. 3). In particular, forward attention layers are applied to the first six layers of encoder, while reverse attention layers are adopted to the last six layers of decoder. For all the U-Net layers and the forward and reverse attention layers, we use convolution filters with the kernel size of 4×4, stride 2 and padding 1, and no bias parameters are used. In the U-Net backbone, batch normalization and leaky ReLU nonlinearity are used to the features after re-normalization, and tanh nonlinearity is deployed right after convolution for the last layer. Fig. 3 also provides the size of feature map for each layer, and more details of the network architecture are given in the suppl.

3.5. Loss Functions

For better recovery of texture details and semantics, we incorporate pixel reconstruction loss, perceptual loss [12], style loss [7] and adversarial loss [8] to train our LBAM.

Pixel Reconstruction Loss. Denote by I^{in} the input image with holes, $\mathbf{M}^{in}$ the binary mask region, and I^{gt} the ground-truth image. The output of our LBAM can be defined as $I^{out} = \Phi(I^{in}, \mathbf{M}^{in}; \Theta)$, where Θ denotes the model parameters to be learned. We adopt the ℓ_1-norm error of the output image as the pixel reconstruction loss,

$$\mathcal{L}_{\ell_1} = \parallel I^{out} - I^{gt} \parallel_1. \quad (14)$$

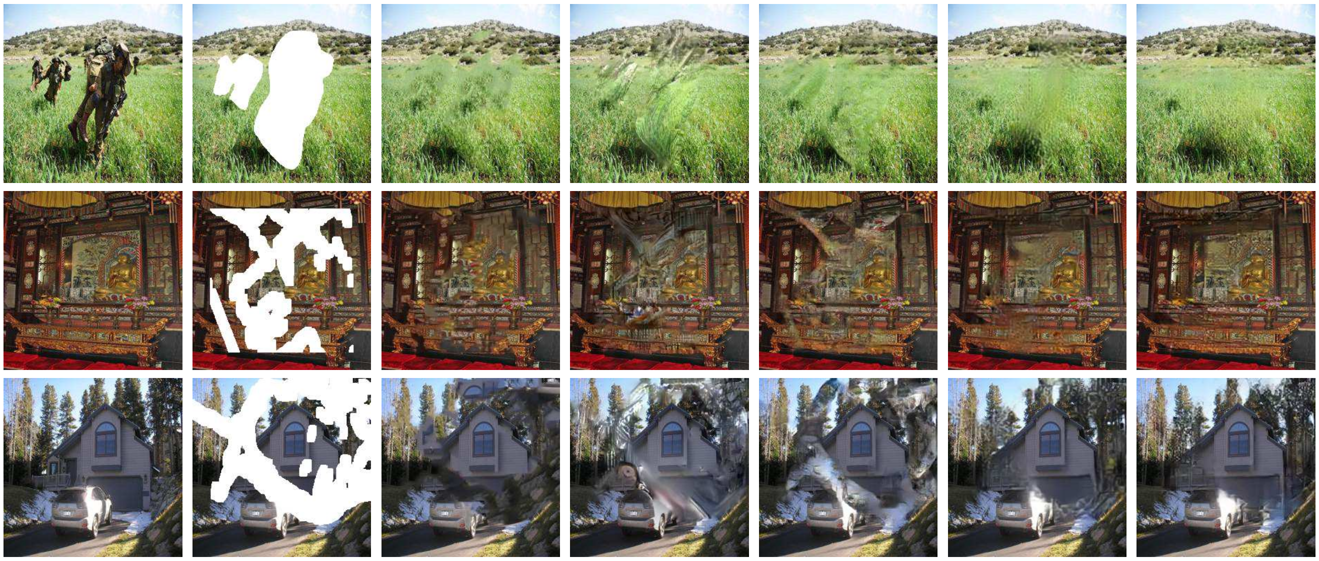


Figure 5. Qualitative comparison on Places dataset. Comparison with PatchMatch (PM) [2], Global&Local(GL) [10], Context Attention(CA) [36], PConv [17] and Ours.

Perceptual Loss. The ℓ_1-norm loss is limited in capturing high-level semantics and is not consistent with the human perception of image quality. To alleviate this issue, we introduce the perceptual loss $\mathcal{L}_{perc}$ defined on the VGG-16 network [25] pre-trained on ImageNet [23],

$$\mathcal{L}_{perc} = \frac{1}{N}\sum\nolimits_{i=1}^{N} \parallel \mathcal{P}^i(I^{gt}) - \mathcal{P}^i(I^{out}) \parallel^2 \tag{15}$$

where $\mathcal{P}^i(\cdot)$ is the feature maps of the i-th pooling layer. In our implementation, we use *pool*-1, *pool*-2, and *pool*-3 layers of the pre-trained VGG-16.

Style Loss. For better recovery of detailed textures, we further adopt the style loss defined on the feature maps from the pooling layers of VGG-16. Analogous to [17], we construct a Gram matrix from each layer of feature map. Suppose that the size of feature map $\mathcal{P}^i(I)$ is $H_i \times W_i \times C_i$. The style loss can then be defined as,

$$\begin{aligned}\mathcal{L}_{style} = & \frac{1}{N}\sum\nolimits_{i=1}^{N} \frac{1}{C_i \times C_i} \times \\ & \parallel \mathcal{P}^i(I^{gt})(\mathcal{P}^i(I^{gt}))^T - \mathcal{P}^i(I^{out})(\mathcal{P}^i(I^{out}))^T \parallel^2\end{aligned} \tag{16}$$

Adversarial Loss. Adversarial loss [8] has been widely adopted in image generation [24, 27, 38] and low level vision [16] for improving the visual quality of generated images. In order to improve the training stability of GAN, Arjovsky *et al*. [1] exploit the Wasserstein distance for measuring the distribution discrepancy between generated and real images, and Gulrajani *et al*. [9] further introduce gradient penalty for enforcing the Lipschitz constraint in discriminator. Following [9], we formulate the adversarial loss as,

$$\begin{aligned}\mathcal{L}_{adv} = & \min_{\Theta}\max_{D} E_{I^{gt}\sim p_{data}(I^{gt})} D(I^{gt}) \\ & - E_{I^{out}\sim p_{data}(I^{out})} D(I^{out}) \\ & + \lambda E_{\hat{I}\sim p_{\hat{I}}}((\parallel \nabla_{\hat{I}} D(\hat{I}) \parallel)^2 - 1)^2\end{aligned} \tag{17}$$

where $D(\cdot)$ represents the discriminator. $\hat{I}$ is sampled from I^{gt} and I^{out} by linear interpolation with a randomly selected factor, λ is set to 10 in our experiments. We empirically find that it is difficult to train the PConv model when including adversarial loss. Fortunately, the incorporation of learnable attention maps is helpful to ease the training, making it feasible to learn LBAM with adversarial loss. Please refer to the suppl. for the network architecture of the 7-layer discriminator used in our implementation.

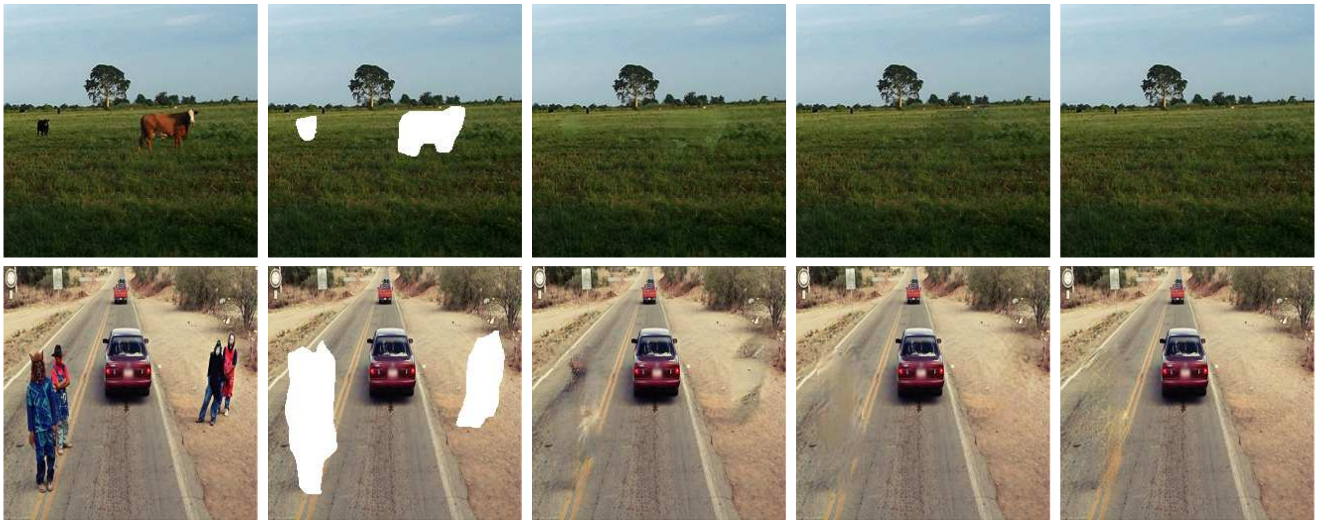
Model Objective Taking the above loss functions into account, the model objective of our LBAM can be formed as,

$$\mathcal{L} = \lambda_1\mathcal{L}_{\ell_1} + \lambda_2\mathcal{L}_{adv} + \lambda_3\mathcal{L}_{perc} + \lambda_4\mathcal{L}_{style} \tag{18}$$

where λ_1, λ_2, λ_3, and λ_4 are the tradeoff parameters. In our implementation, we empirically set $\lambda_1 = 1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.05$ and $\lambda_4 = 120$.

4. Experiments

Experiments are conducted for evaluating our LBAM on two datasets, *i.e*., Paris StreetView [6] and Places (Places365-standard) [40], which have been extensively adopted in image inpainting literature [20, 33, 34, 36]. For Paris StreetView, we use its original splits, $14,900$ images for training, and 100 images for testing. In our experiments, 100 images are randomly selected and removed from the training set to form our validation set. As for Places, we randomly select 10 categories from the 365 categories, and use all the $5,000$ images per category from the original training set to form our training set of $50,000$ images. Moreover, we divide the original validation set from each category of $1,000$ images into two equal non-overlapped sets of 500 images respectively for validation and testing. Our LBAM takes ~ 70 ms for processing a 256×256 image, $5\times$ faster



Original Input CA [36] PCony [17] Ours

Figure 6. Results on real-world images. From left to right are: original image, input with objects masked (white area), Context Attention (CA) [36], PConv [17], and Ours.

than Context Attention [36] ($\sim 400ms$) and $\sim 3\times$ faster than Global&Local(GL) [10] ($\sim 200ms$).

In our experiments, all the images are resized where the minimal height or width is 350, and then randomly cropped to the size of 256×256. Data augmentation such as flipping is adopted during training. We generate $18,000$ masks with random shape, and $12,000$ masks from [17] for training and testing. Our model is optimized using the ADAM algorithm [13] with initial learning rate of $1e-4$ and $\beta = 0.5$. The training procedure ends after 500 epochs, and the mini-batch size is 48. All the experiments are conducted on a PC equipped with 4 parallel NVIDIA GTX 1080Ti GPUs.

4.1. Comparison with State-of-the-arts

Our LBAM is compared with four state-of-the-art methods, *i.e.*, Global&Local [10], PatchMatch [2], Context Attention [36], and PConv [17].

Evaluation on Paris StreetView and Places. Fig. 4 and Fig. 5 show the results by our LBAM and the competing methods. Global&Local [10] is limited in handling irregular holes, producing many matchless and meaningless textures. PatchMatch [2] performs poorly for recovering complex structures, and the results are not consistent with surrounding context. For some complex and irregular holes, context attention [36] still generates blurry results and may produce unwanted artifacts. PConv [17] is effective in handling irregular holes, but over-smoothing results are still inevitable in some regions. In contrast, our LBAM performs well generating visually more plausible results with fine-detailed, and realistic textures.

Quantitative Evaluation. We also compare our LBAM quantitatively with the competing methods on Places [40] with mask ratio $(0.1, 0.2]$, $(0.2, 0.3]$, $(0.3, 0.4]$ and $(0.4, 0.5]$. From Table 1, our LBAM performs favorably in terms PSNR, SSIM, and mean ℓ_1 loss, especially when the mask ratio is higher than 0.3.

Table 1. Quantitative comparison on Places. Results of PConv* are taken from [17].

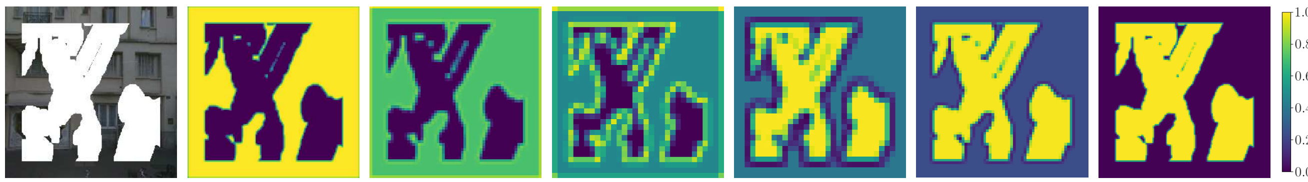
	Mask	GL [10]	PM [2]	CA [36]	PConv* [17]	Ours
PSNR	(0.1-0.2]	23.36	26.67	26.27	28.32	**28.51**
	(0.2, 0.3]	20.53	24.21	23.56	25.25	**25.59**
	(0.3, 0.4]	19.37	21.95	21.20	22.89	**23.31**
	(0.4, 0.5]	17.86	20.02	19.95	21.38	**21.66**
SSIM	(0.1-0.2]	0.828	0.876	**0.881**	0.870	0.872
	(0.2, 0.3]	0.744	0.763	0.769	0.779	**0.785**
	(0.3, 0.4]	0.643	0.657	0.667	0.689	**0.708**
	(0.4, 0.5]	0.545	0.572	0.563	0.595	**0.602**
Mean l_1(%)	(0.1-0.2]	2.45	1.43	2.05	**1.09**	1.12
	(0.2, 0.3]	4.01	2.38	3.74	**1.88**	1.93
	(0.3, 0.4]	5.86	3.59	5.65	2.84	**2.55**
	(0.4, 0.5]	7.92	5.22	7.43	3.85	**3.67**

Object Removal from Real-world Images. Using the model trained on Places, we further evaluate LBAM on the real world object removal task. Fig. 6 shows the results by our LBAM, context attention [36] and PConv [17]. We mask the object area either with contour shape or with rectangular bounding box. In contrast to the competing methods, our LBAM can produce realistic and coherent contents by both global semantics and local textures.

User Study. Besides, user study is conducted on Paris StreetView and Places for subjective visual quality evaluation. We randomly select 30 images from the test set covering with different irregular holes, and the inpainting results are generated by PatchMatch [2], Global&Local [10], Context Attention [36], PConv [17] and ours. We invited 33 volunteers to vote for the most visually plausible inpainting result, which is assessed by the criteria including coherency with the surrounding context, semantic structure and fine

(a) (b) (c) (d) (e) (f) (g)

Figure 7. Visualization of features from the first encoder layer and 13-th decoder layer. (a) Input, (b)(c) Ours(unlearned), (d)(e) Ours(forward), (f)(g) Ours(full).



(a) (b) (c) (d) (e) (f) (g)

Figure 8. Visualization of updated masks after activation function $g_A(\cdot)$ for forward and reverse attention maps. (a) Input, (b)(c)(d) forward masks from the first three (1,2,3) layers, (e)(f)(g) reverse masks from the last three (11, 12, 13) layers.

(a) Input (b) Ours(unlearned) (c) Ours(forward) (d) Ours(full)

Figure 9. Visual quality comparison of the effect on the learnable bidirectional attention maps.

details. For each test image, the 5 inpainting results are randomly arranged and presented to user along with the input image. Our LBAM has 63.2% chance to win out as the most favorable result, largely surpassing PConv [17] (15.2%), PatchMatch [2] (11.1%), Context Attention [36] (6.33%) and Global&Local [10] (4.17%).

4.2. Ablation Studies

Ablation studies are conducted to compare the performance of several LBAM variants on Paris StreetView, *i.e.*, (i) Ours(full): the full LBAM model, (ii) Ours(unlearned): the LBAM model where all the elements in mask convolution filters are set as $\frac{1}{16}$ because the filter size is 4×4, and we adopt the activation functions defined in Eqn. (4) and Eqn. (5), (iii) Ours(forward): the LBAM model without reverse attention map, (iv) Ours(w/o $\mathcal{L}_{adv}$): the LBAM model without (w/o) adversarial loss, (v) Ours(Sigmoid/LReLU/ReLU/3×3): the LBAM model using Sigmoid/LeakyReLU/ReLU as activation functions or 3×3 filter for mask updating.

Fig. 7 shows the visualization of features from the first encoder layer and 13-th decoder layer by Ours(unlearned), Ours(forward), and Ours(full). For Ours(unlearned), blurriness and artifacts can be observed from Fig. 9(b). Ours(forward) is beneficial to reduce the artifacts and noise, but the decoder hallucinates both holes and known regions and produces some blurry effects (see Fig. 9(c)). In contrast, Ours(full) is effective in generating semantic structure and detailed textures (see Fig. 9(d)), and the decoder focus mainly on hallucinating holes (see Fig. 7(g)). Table 2 gives the quantitative results of the LBAM variants on Paris StreetView, and the performance gain of Ours(full) can be explained by (1) learnable attention maps, (2) reverse attention maps, and (3) proper activation functions.

Table 2. Ablation studies (PSNR/SSIM) on Paris StreetView.

Method	(0.1, 0.2]	(0.2, 0.3]	(0.3, 0.4]	(0.4, 0.5]
Ours(unlearned)	26.95/0.853	24.39/0.763	22.54/0.677	21.20/0.583
Ours(forward)	27.80/0.869	25.13/0.775	23.04/0.688	21.76/0.598
Ours(Sigmoid)	26.93/0.857	24.15/0.768	22.24/0.683	20.32/0.582
Ours(LReLU)	26.61/0.852	23.59/0.762	20.63/0.667	18.38/0.562
Ours(ReLU)	27.62/0.864	25.16/0.776	22.96/0.685	21.48/0.596
Ours(3x3)	28.74/0.886	26.10/0.793	24.03/0.703	22.43/0.617
Ours(w/o $\mathcal{L}_{adv}$)	29.19/0.903	26.55/0.817	24.46/0.729	22.70/0.626
Ours(full)	28.73/0.889	26.16/0.795	24.26/0.716	22.62/0.621

Mask Updating. Fig. 8 shows the visualization of updated masks from different layers. From the first to third layers, the masks of encoder are gradually updated to reduce the size of holes. Analogously, from the 13-th to 11-th layers, the masks of decoder are gradually updated to reduce the size of known region.

Effect of Adversarial Loss. Table 2 also gives the quantitative result w/o $\mathcal{L}_{adv}$. Albeit Ours(w/o $\mathcal{L}_{adv}$) improves PSNR and SSIM, the use of $\mathcal{L}_{adv}$ generally benefits the visual quality of the inpainting results. The qualitative results are given in the suppl.

5. Conclusion

This paper proposed a learnable bidirectional attention maps (LBAM) for image inpainting. With the introduction of learnable attention maps, our LBAM is effective in adapting to irregular holes and propagation of convolution layers. Furthermore, reverse attention maps are presented to allow the decoder of U-Net concentrate only on filling in holes. Experiments shows that our LBAM performs favorably against state-of-the-arts in generating sharper, more coherent and fine-detailed results.

Acknowledgement

This work was supported in part by the NSFC grant under No. 61671182 and 61872116, and National Key Research and Development Project 2018YFC0832105.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 214–223, 2017. 6

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, pages 24:1–24:11, 2009. 1, 2, 5, 6, 7, 8

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 417–424, 2000. 1

[4] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing (TIP)*, pages 1200–1212, 2004. 2

[5] Silvano Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, pages 116–125, 1986. 2

[6] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A Efros. What makes paris look like paris? *Communications of the ACM*, pages 103–110, 2015. 2, 6

[7] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 5

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014. 1, 5, 6

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5767–5777, 2017. 6

[10] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, pages 107:1–107:14, 2017. 1, 2, 3, 5, 6, 7, 8

[11] 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 (CVPR)*, pages 5967–5976, 2017. 4, 5

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *The European Conference on Computer Vision (ECCV)*, volume 9906, pages 694–711, 2016. 5

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[14] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *Pattern Recognition (GCPR)*, pages 523–534, 2014. 2

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

[16] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 6

[17] Guilin Liu, Fitsum A. Reda, Kevin Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, volume 11215, pages 89–105, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[18] Olivier Le Meur, Josselin Gautier, and Christine Guillemot. Examplar-based inpainting based on local geometry. In *IEEE International Conference on Image Processing (ICIP)*, pages 3401–3404, 2011. 1, 2

[19] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. 3

[20] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016. 1, 3, 6

[21] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 901–909, 2015. 2

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351, pages 234–241, 2015. 2

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2234–2242, 2017. 6

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 6

[26] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Hao Li, and Qin Huang. Contextual Based Image Inpainting: Infer, Match and Translate. In *The European Conference on Computer Vision (ECCV)*, volume 11206, pages 3–18, 2018. 1, 3

[27] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Condi-

tional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4790–4798, 2016. 6

[28] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 329–338, 2018. 3

[29] Marta Wilczkowiak, Gabriel J. Brostow, Ben Tordoff, and Roberto Cipolla. Hole filling through photomontage. In *British Machine Vision Conference (BMVC)*, pages 492–501, 2005. 2

[30] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 341–349, 2012. 2

[31] 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 (CVPR)*, 2019. 3

[32] Zongben Xu and Jian Sun. Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing (TIP)*, pages 1153–1165, 2010. 1, 2

[33] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *The European Conference on Computer Vision (ECCV)*, volume 11218, pages 3–19, 2018. 1, 3, 4, 6

[34] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4076–4084, 2017. 1, 3, 6

[35] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 3

[36] 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 (CVPR)*, pages 5505–5514, 2018. 1, 2, 3, 5, 6, 7, 8

[37] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. Semantic image inpainting with progressive generative networks. In *ACM International Conference on Multimedia (ACM MM)*, pages 1939–1947, 2018. 3

[38] 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 (ICCV)*, pages 5908–5916, 2017. 6

[39] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1438–1447, 2019. 3

[40] 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 (TPAMI)*, pages 1452–1464, 2017. 2, 6, 7