Deep Generative Model for Image Inpainting with Local Binary Pattern Learning and Spatial Attention

Haiwei Wu, *Student Member, IEEE*, Jiantao Zhou, *Senior Member, IEEE*, and Yuanman Li, *Student Member, IEEE*

***Abstract*—Deep learning (DL) has demonstrated its powerful capabilities in the field of image inpainting. The DL-based image inpainting approaches can produce visually plausible results, but often generate various unpleasant artifacts, especially in the boundary and highly textured regions. To tackle this challenge, in this work, we propose a new end-to-end, two-stage (coarse-to-fine) generative model through combining a local binary pattern (LBP) learning network with an actual inpainting network. Specifically, the first LBP learning network using U-Net architecture is designed to accurately predict the structural information of the missing region, which subsequently guides the second image inpainting network for better filling the missing pixels. Furthermore, an improved spatial attention mechanism is integrated in the image inpainting network, by considering the consistency not only between the known region with the generated one, but also within the generated region itself. Extensive experiments on public datasets including `CelebA-HQ`, `Places` and `Paris StreetView` demonstrate that our model generates better inpainting results than the state-of-the-art competing algorithms, both quantitatively and qualitatively. The source code and trained models will be made available at https://github.com/HighwayWu/ImageInpainting.**

***Index Terms*—Image inpainting, LBP, spatial attention, deep learning.**

I. Introduction

IMAGE inpainting is to fill the missing region of an image with plausible contents. It has a wide range of applications in the field of computer vision, e.g., repairing damaged photos or removing unwanted objects. The major challenge faced by image inpainting is the generation of visually realistic and semantically plausible contents for the missing region that is consistent with the known part.

Several traditional approaches [1]–[14] attempted to solve the inpainting problem via the image-level texture synthesis. Sun *et al.* [7] proposed to adopt user-specified curves to complete missing structures, and then fill the missing region via patch-based texture synthesis. Hays and Efros [8] built a huge database of photographs, from which similar patches can be fetched for image inpainting. Similarly, Simakov *et al.* [9] suggested an approach based on the bidirectional patch similarity to better summarize visual data for re-targeting, object removal and image inpainting. Barnes *et al.* [11] designed the Patch-Match algorithm, significantly speeding up the searching of similar patches, which could be used for image inpainting. Recently, Huang *et al.* [14] presented a novel structure-guided inpainting method by maintaining the neighborhood consistence and structure coherence of inpainted regions. As these inpainting methods essentially assumed that the missing region shares the same structural features with the known one, they cannot create novel contents for the challenging cases where the missing region involves complex structures (e.g., faces) and high-level semantics [15].

Fig. 1. Inpainting results generated by our proposed model on images of face, natural scene, texture and street view. In each pair, the left is the input image with centering or irregular mask, and the right is the inpainting result.

With the rapid development of deep convolutional neural networks (CNNs) and generative adversarial networks (GANs) [16] in recent years, many deep generative models [15], [17]–[38] have been proposed for image inpainting, achieving promising results. Generally, these models employ a convolutional encoder-decoder network, where the encoder extracts high-level information from image-level pixels, from which the decoder then generates semantically coherent contents. During this process, adversarial networks are often jointly trained to promote consistency between the generated and existing pixels. These deep generative models were reported to be able to produce plausible new contents, e.g., faces, objects and scenes [15]. Nevertheless, they also tend to generate unpleasant boundary artifacts, distorted structures, and blurry textures that are incoherent with the available parts. As pointed out in [15], such inferior performance may be due to the ineffectiveness of CNNs in modeling long-term correlations between the distant contextual information and the missing region.

In this paper, inspired by the image inpainting procedure conducted by human artists, we propose a new end-to-end, coarse-to-fine deep generative image inpainting model that consists of two networks. The first network, called Local Binary Pattern (LBP) learning network, aims to recover the LBP feature of the missing region, based on the known region. One of the reasons why we select LBP feature under this

The authors are with the State Key Laboratory of Internet of Things for Smart City, and also with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China (*Corresponding Author: Jiantao Zhou, email: jtzhou@umac.mo*).

circumstance is that it contains a great amount of structural information, capable of well guiding the subsequent image inpainting task. As verified in [39], an image visually close to the original one could be reconstructed solely from its LBP feature. Also, from the perspective of practical implementation, LBP is easy to be computed and very few parameters are involved. The second network, called image inpainting network, performs the actual operations for filling the missing pixels, by using the learned LBP feature as the guidance. To further promote the semantic relevance of the filled pixels, we propose to integrate a novel spatial attention layer into the image inpainting network. The proposed attention layer has a unique mechanism that models the correlations not only between the known region with the generated one, but also within the generated region itself. The latter correlation was largely ignored by *all* the existing methods [15], [22], [27], [28], [30], [35], [37]. As a result, our proposed generative model leads to better global and local consistency in the inpainting results. Meanwhile, in order to make the training process more stable, we design a multi-level loss such that multi-level features could be optimized. Experiments on three publicly available datasets `CelebA-HQ` [40], `Places` [41] and `Paris StreetView` [42] demonstrate that our proposed model can generate better results than the state-of-the-art competitors, both quantitatively and qualitatively. Fig. 1 shows some example results. We also would like to emphasize that, among all the two-stage networks for the image inpainting [15], [22], [24], [26], [28], [34], the key differences lie in how to design these two networks, and naturally, different designs could lead to dramatically different inpainting results.

Our major contributions can be summarized as follows:

- We propose a new end-to-end, coarse-to-fine deep generative model that incorporates LBP learning to provide structural information for the inpainting task. A multi-level loss is also designed to ensure more stable training process.
- We introduce a novel spatial attention layer that models the correlations not only between the known region and the filled one, but also within the filled region itself. This leads to better global and local consistency of the inpainting results.
- Our model achieves better inpainting performance in comparison with several state-of-the-art methods [15], [24], [35], [43] over a variety of challenging datasets including `CelebA-HQ`, `Places` and `Paris StreetView`.

The rest of this paper is organized as follows. Section II reviews the related works on LBP and deep generative models for image inpainting. Section III presents our proposed model. Experimental results are given in Section IV and Section V concludes.

II. Related Works

A. Local Binary Pattern (LBP)

LBP is a simple yet very effective texture descriptor originally proposed by Ojala *et al.* [44]. The LBP feature extraction process is to label each pixel of an image by thresholding its spatial neighborhood. Specifically, to extract the LBP feature associated with the pixel I, we first obtain its 3×3 neighborhood denoted by $I_1, I_2, \cdots, I_8$. Then the LBP feature associated with I is an 8-bit long binary string $\mathbf{b} = b_1, b_2, \cdots, b_8$, where

$$b_i = \begin{cases} 0 & \text{if } I_i \leq I \\ 1 & \text{otherwise} \end{cases}, \text{for } i = 1, 2, \cdots, 8. \tag{1}$$

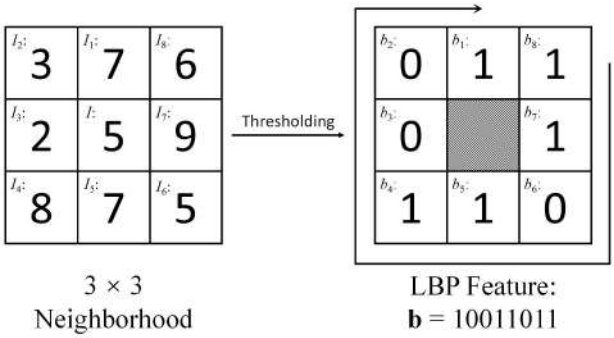


Fig. 2. An example of the LBP extraction. Left is the original 3×3 neighborhood. Right is the thresholded neighborhood, and the LBP feature of the centering pixel I is $\mathbf{b} = 10011011$.

An example of the LBP feature extraction is illustrated in Fig. 2.

LBP feature essentially records the relative ordering within a block of pixels, capturing the information of edges, spots and other local structures [45]. LBP shows very good performance in many vision tasks, e.g., unsupervised texture segmentation [46], face recognition [47] and image reconstruction [39].

B. Image Inpainting by Deep Generative Models

Many DL- and GAN-based inpainting methods have been proposed in recent years, achieving promising results. Pathak *et al.* [25] pioneered the research in this direction by training deep generative adversarial networks for inpainting large holes in images. However, the proposed networks cannot satisfactorily maintain global consistency and tends to produce severe visual artifacts. Iizuka *et al.* [18] designed a generative network with two context discriminators to encourage global and local consistency, where the global discriminator evaluates whether the image is coherent as a whole, and the local discriminator ensures local consistency of the generated patches. Instead of merely using the features of the encoder layer, Yan *et al.* [35] proposed an attention mechanism, which jointly uses the encoder layer and the corresponding decoder layer to estimate the missing features. To further improve the attention mechanism, Wang *et al.* [30] suggested a multi-stage image contextual attention learning strategy to deal with the rich background information flexibly while avoiding abuse them. Meanwhile, several works [21], [43] adopted partial or gated convolutions to reduce the color discrepancy and blurriness, where the convolutions are masked, re-normalized, and operated only on the known region.

Attempting to further improve the inpainting performance, there is a recent trend of using two-stage networks, where

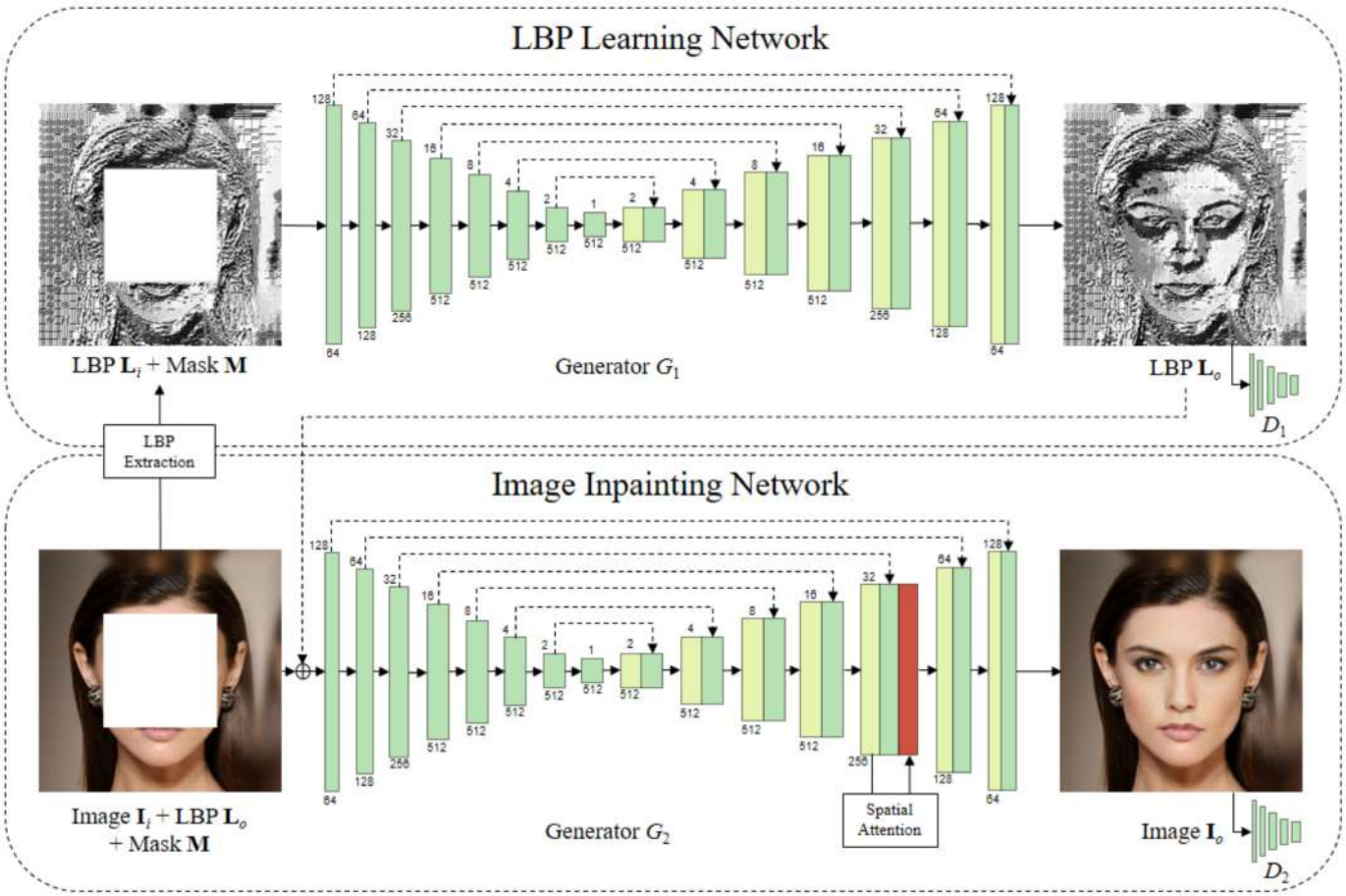


Fig. 3. Overview of our proposed generative inpainting network. Note that a spatial attention layer is concatenated in the decoder of the image inpainting network. The number above each layer represents the size of the resolution, while the number below means the dimension.

the first stage estimates the missing structures and the second stage aims to generate the final results assisted by the estimated structural information. Along this line, Yu *et al.* [15] proposed to use a simple dilated convolutional network in the first stage to rough out the missing contents, and then integrated the contextual attention in the second stage. Song *et al.* [28] introduced a patch-swap layer to propagate the high-frequency texture details from the boundary to the hole, where a VGG network is used as the feature extractor. By incorporating the prior knowledge on local patches continuity, Liu *et al.* [22] suggested a coherent semantic attention layer to model the semantic relevance between the holes, and iteratively optimize them to achieve better spatial consistency. Ren *et al.* [26] designed the StructureFlow, which employs edge-preserved smooth images to train a structure reconstructor in the first stage, and then uses a texture generator with appearance flow in the second stage to yield the image details. Xiong *et al.* [34] built a foreground-aware image inpainting model that detects and completes the foreground contour first, and then fills the missing region using the predicted contour as a guidance. Furthermore, Nazeri *et al.* [24] proposed an edge-connect model that comprises of an edge generator followed by an image completion network. These two-stage methods demonstrate promising visual results by using the learned information to assist the ultimate inpainting task.

III. Proposed Deep Generative Model with LBP Learning and Spatial Attention

The proposed deep generative model for image inpainting falls into the category of two-stage scheme, which can be shown in Fig. 3. As mentioned previously, different designs of these two stages could lead to dramatically different inpainting performance. Our model consists of two networks: LBP learning network and image inpainting network. The first network predicts the LBP information of the missing region, serving as a guidance for the actual image inpainting task in the second network. The major reason why we choose the LBP in the first network is because LBP contains richer amount of structural information, compared with other alternatives, e.g., edges. An inspiring example is given in Fig. 4, where we compare the quality of the reconstructed images from the edges and the LBP. As can be seen, the reconstructed result from LBP is much better than the one obtained from the edges, especially in the fine textured regions, e.g., the hairs. In addition, LBP feature extraction is of low complexity, and involves very few parameters. In contrast, some other structural information (e.g., edges and contours) extractions typically involve many parameters, e.g., the pre-filtering strength and the threshold for the edge response, whose optimal setting should vary for different images. These facts would suggest that LBP is a more

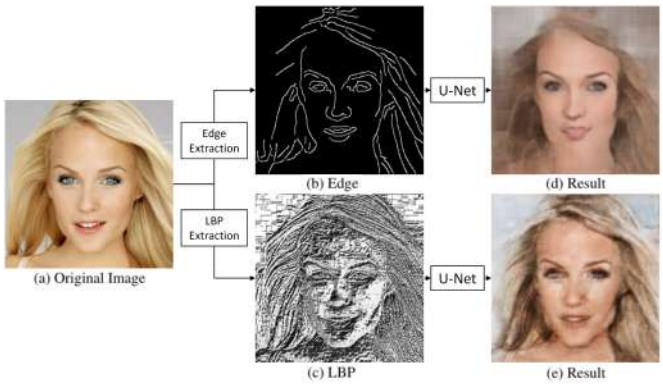


Fig. 4. Image reconstruction results from edges and LBP feature. Here, Canny edge detection [48] is used with $\sigma = 2$ (the recommended setting in [24]).

appropriate candidate for providing structural information to be incorporated into the image inpainting network.

Both networks follow an adversarial model [16]. More specifically, each network contains a generator based on U-Net architecture [49], and a discriminator based on the PatchGAN [19]. Let $\mathbf{I}_i$ be an input image with white pixels filled in the missing region, and $\mathbf{M}$ be the corresponding binary mask, where 1's are assigned to the known region and 0's elsewhere. $\mathbf{M}$ is randomly sampled from the mask dataset and can be either centering or irregular. Denote $\mathbf{L}_i$ as the LBP extracted from $\mathbf{I}_i$ in the grayscale channel. At the training stage, the generator of the LBP learning network G_1 takes the pair $(\mathbf{L}_i, \mathbf{M})$ as input, and outputs the learned LBP $\mathbf{L}_o$, where the LBP feature for the missing region has been restored. During this process, the discriminator D_1 works together with the G_1 to produce the result $\mathbf{L}_o$. Upon having a well-estimated LBP, we then use it to guide the inpainting process in the inpainting network. Specifically, the generator G_2 takes ($\mathbf{I}_i$, $\mathbf{L}_o$, $\mathbf{M}$) as input, and outputs the final inpainting result $\mathbf{I}_o$, with the assistance of the discriminator D_2. At the testing stage, the procedure is similar, but without the need of using the two discriminators D_1 and D_2.

In the following, we present the details regarding the LBP learning network and the image inpainting network.

A. LBP Learning Network

The LBP learning network is formed by two components: the generator G_1 and the discriminator D_1. For G_1, we adopt a pruned U-Net architecture [49] composed of an encoder and a decoder. In the encoder, each layer has a 4×4 convolution, an Instance Norm [50] and a LeakyReLU [51] with $\alpha = 0.2$. The decoder has a symmetric structure, except that the convolution and LeakyReLU are replaced with the deconvolution and ReLU [52], respectively. Additionally, skip connections are used to concatenate the features from each layer of the encoder with the corresponding layer of the decoder. Experimentally, we find that the dilated convolutions in the original U-Net architecture [49] bring negligible improvements to the final inpainting results. We hence prune the U-Net architecture by removing the dilated convolutions, so as to reduce the number of model parameters, which could speed up the training process. For the D_1, we adopt the PatchGAN architecture [19].

In addition to the network architecture, another important ingredient for achieving a desirable LBP learning network is the loss function. To better deal with the training instability, we design a multi-level loss to penalize the feature-domain deviation of the model. Specifically, let $\mathbf{L}_g$ be the ground-truth LBP, and its corresponding high-level features be $\Phi_h(\mathbf{L}_g)$, where h is the layer index within G_1. As the training direction, $\Phi_h(\mathbf{L}_g)$ can optimize high-level features of the G_1 globally. The multi-level loss function is then defined as:

$$\mathcal{L}_m = \sum_{h \in \mathcal{H}} ||\Phi_h(\mathbf{L}_o) - \Phi_h(\mathbf{L}_g)||_2, \tag{2}$$

where $\mathcal{H}$ accommodates the indexes of *all* the convolution and deconvolution layers in G_1.

Besides the multi-level loss, the reconstruction loss and the adversarial loss need to be included as well. In this work, the reconstruction loss is naturally defined as:

$$\mathcal{L}_r = ||\mathbf{L}_o - \mathbf{L}_g||_2. \tag{3}$$

Also, the adversarial loss [35] can be calculated as follows:

$$\begin{aligned}\mathcal{L}_a = \min_{G_1} \max_{D_1} &\mathbb{E}_{\mathbf{L}_g}[\log D_1(\mathbf{L}_g)] + \\ &\mathbb{E}_{\mathbf{L}_i}[\log(1 - D_1(G_1(\mathbf{L}_i, \mathbf{M})))].\end{aligned} \tag{4}$$

Finally, the loss function for the LBP learning network is defined by integrating the above three types of loss.

$$\mathcal{L}_{LBP} = \lambda_m \mathcal{L}_m + \lambda_r \mathcal{L}_r + \lambda_a \mathcal{L}_a, \tag{5}$$

where λ_m, λ_r and λ_a are the parameters trading off different types of loss, whose settings will be clarified in the next Section.

B. Image Inpainting Network

The architecture of the image inpainting network is similar to our LBP learning network, except that $\mathbf{I}_i$, $\mathbf{L}_o$ and $\mathbf{M}$ are used as inputs together, and a newly designed spatial attention layer is embedded in the fifth layer of the decoder. The feature map of the spatial attention layer is of size 32×32, aiming at more effectively modeling the correlations not only between the known region with the filled one, but also within the filled region itself. In the following, let us first explain the loss function for the image inpainting network, and then present the details of our proposed spatial attention layer.

1) Loss Function for the Inpainting Network: To better optimize the high-level features of the image inpainting network, we further introduce two loss terms, namely, the perceptual loss [53] and the style loss [54]. Specifically, the perceptual loss penalizes the inpainting results that are not perceptually similar to the ground-truth $\mathbf{I}_g$, and it can be defined as:

$$\mathcal{L}_p = \sum_{h \in \mathcal{A}} ||\varphi_h(\mathbf{I}_o) - \varphi_h(\mathbf{I}_g)||_2, \tag{6}$$

where φ_h is the activation map corresponding to the h-th layer of an ImageNet-pretrained VGG-16 network. The set $\mathcal{A}$ is formed by the layer indexes of conv2_1, conv3_1, conv4_1 layers. On the other hand, the style loss is used to measure

the differences between the covariances of the activation maps, which is an effective strategy to eliminate the "checkerboard" artifacts cased by deconvolution layers [55]. Typically, the style loss can be defined as:

$$\mathcal{L}_s = \sum_{h\in\mathcal{A}} ||\mathbf{G}^{\varphi_h}(\mathbf{I}_o) - \mathbf{G}^{\varphi_h}(\mathbf{I}_g)||_2, \tag{7}$$

where $\mathbf{G}^{\varphi_h}$ is a 3×3 Gram matrix constructed from the activation map φ_h.

For the total loss function of the image inpainting network, we also add the multi-level loss, the reconstruction loss and the adversarial loss, which can be similarly defined as in (2), (3) and (4), respectively. Finally, the loss function of the image inpainting network can be expressed as:

$$\mathcal{L}_{Img} = \lambda_m\mathcal{L}_m + \lambda_r\mathcal{L}_r + \lambda_a\mathcal{L}_a + \lambda_p\mathcal{L}_p + \lambda_s\mathcal{L}_s, \tag{8}$$

where λ_m, λ_r, λ_a, λ_p and λ_s are parameters used for trading off different losses.

2) Spatial Attention Layer: Another crucial element in our proposed scheme is a new spatial attention layer, further improving the semantic consistency, not only between the known region and the filled region, but also within the filled region itself. This is quite different from the existing attention models [15], [22], [27], [28], [30], [35], [37], which only paid attention to the relevant patches of the known region, while totally ignoring the correlations among the generated patches.

More specifically, let $\Phi_h(\mathbf{I}_i)$ be the feature map of the h-th layer in the generator G_2 when using $\mathbf{I}_i$ as the input. Denote Ω and $\bar{\Omega}$ as the missing region and the known region of $\Phi_h(\mathbf{I}_i)$, respectively. We extract all 1×1 patches $\{\mathbf{P}_j\}_{j=1}^K$ from $\Phi_h(\mathbf{I}_i)$ and group them into two sets $\mathcal{P}$ and $\bar{\mathcal{P}}$, where

$$\mathcal{P} = \Big\{\mathbf{P}_j | \mathbf{P}_j \in \Omega\Big\}, \tag{9}$$

$$\bar{\mathcal{P}} = \Big\{\bar{\mathbf{P}}_k | \bar{\mathbf{P}}_k \in \bar{\Omega}\Big\}. \tag{10}$$

For each patch $\mathbf{P}_j \in \mathcal{P}$, its intra- cosine similarities within $\mathcal{P}$ and inter- cosine similarities with $\bar{\mathcal{P}}$ can be respectively computed as

$$S_{j,k} = \Big\langle \frac{\mathbf{P}_j}{||\mathbf{P}_j||}, \frac{\mathbf{P}_k}{||\mathbf{P}_k||} \Big\rangle, \quad \mathbf{P}_k \in \mathcal{P}, \tag{11}$$

$$\bar{S}_{j,k} = \Big\langle \frac{\mathbf{P}_j}{||\mathbf{P}_j||}, \frac{\bar{\mathbf{P}}_k}{||\bar{\mathbf{P}}_k||} \Big\rangle, \quad \bar{\mathbf{P}}_k \in \bar{\mathcal{P}}. \tag{12}$$

Upon computing all $S_{j,k}$'s and $\bar{S}_{j,k}$'s, we can readily obtain the top-T similar patches for $\mathbf{P}_j$ from $\mathcal{P}$ and $\bar{\mathcal{P}}$, respectively. Let $\mathcal{N} = \{n_1, ..., n_T\}$ and $\bar{\mathcal{N}} = \{\bar{n}_1, ..., \bar{n}_T\}$ record the indexes of these top-T similar patches in Ω and $\bar{\Omega}$, respectively. The process of similarity search can be conducted via a convolutional layer, as explained in [15], [35]. We then propose to update each $\mathbf{P}_j \in \mathcal{P}$ via a non-local mean [56] strategy:

$$\mathbf{P}_j^* = \sum_{k\in\mathcal{N}} \frac{\exp(S_{j,k})}{Z_j}\mathbf{P}_k + \sum_{k\in\bar{\mathcal{N}}} \frac{\exp(\bar{S}_{j,k})}{Z_j}\bar{\mathbf{P}}_k, \tag{13}$$

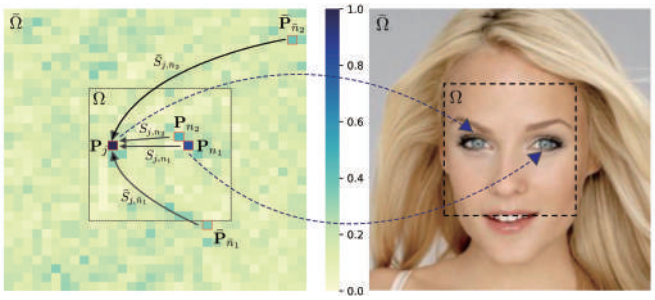


Fig. 5. An example of our spatial attention layer. The depth of the color in the left image represents the level of similarity with the feature patch $\mathbf{P}_j$. The right image helps better understand the semantic correlations in the pixel domain.

where Z_j is the normalization factor:

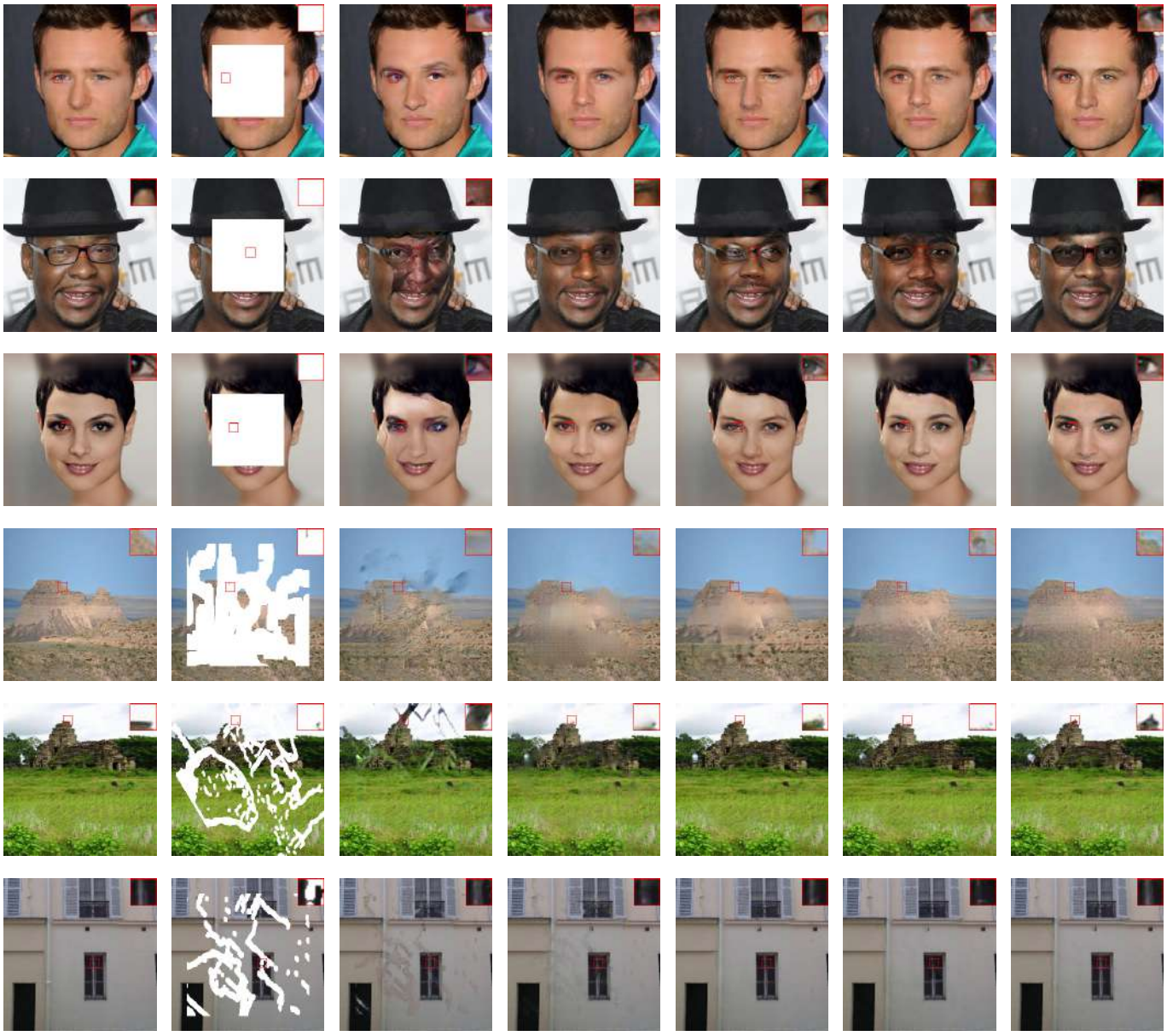
$$Z_j = \sum_{k\in\mathcal{N}} \exp(S_{j,k}) + \sum_{k\in\bar{\mathcal{N}}} \exp(\bar{S}_{j,k}). \tag{14}$$

Therefore, the updated $\mathbf{P}_j^*$ absorbs the information from not only the top-T most similar feature patches in the known region, but also from the ones in the missing region. As expected and will be verified experimentally, the new $\mathbf{P}_j^*$ could better promote semantic coherence both globally and locally.

An illustrative example is given in Fig. 5 where the layer index $h = 13$, the number of the most similar patches $T = 2$ and Ω is a centering rectangle region indicated by dotted lines. The feature patch $\mathbf{P}_j$ in Fig. 5 would correspond to the left eye in the pixel domain. $\mathbf{P}_{n_1}$ and $\mathbf{P}_{n_2}$ are the patches having the highest similarities with $\mathbf{P}_j$ in Ω, while $\bar{\mathbf{P}}_{\bar{n}_1}$ and $\bar{\mathbf{P}}_{\bar{n}_2}$ are the most similar two patches in $\bar{\Omega}$. When the missing region is included in the attention scope, we can find the most relevant patch $\mathbf{P}_{n_1}$ with $S_{j,n_1} = 0.8$. Very likely, $\mathbf{P}_{n_1}$ would correspond to the right eye in the pixel domain. However, if the search range is constrained to the known region only, then the globally most similar patch $\mathbf{P}_{n_1}$ will be missed out. Therefore, by paying extra attention to the generated patches, we may not only provide more relevant patches as references for optimization, but also enhance the network's ability to understand the semantic correlations within the missing region.

IV. Experiments

The proposed deep generative model is implemented using PyTorch framework. The training is performed on a desktop equipped with a Core-i7 and a single GTX 2080 GPU. To stabilize the training process and alleviate the gradient vanishing problem, we first train the generator G_1 and the discriminator D_1 in the LBP network. Then we concatenate the G_1 to the image inpainting network, and perform an end-to-end training over G_1, G_2 and D_2 simultaneously. Adam [57] algorithm is adopted, where the parameters in Adam are $\beta_1 = 0.5$, $\beta_2 = 0.999$ and learning rate $r = 2\times10^{-4}$. We train the model with the batch size of 1 and set the parameters of the spatial attention layer to be $h = 13$ and $T = 2$. In the loss functions, the parameters trading off different terms are set to be $\lambda_m = 0.01$, $\lambda_r = 10$, $\lambda_a = 0.2$, $\lambda_p = 1$ and $\lambda_s = 10$. Note that all the parameters are fixed when performing the subsequent experiments.



(a) GT. (b) Input (c) CA (d) SH (e) GC (f) EC (g) Ours

Fig. 6. Qualitative comparisons for image inpainting performance with centering and irregular masks on CelebA-HQ, Places and Paris StreetView. For each row, the images from left to right are ground truth, input images with centering or irregular mask, results generated by CA [15], SH [35], GC [43], EC [24] and our proposed method, respectively.

We evaluate the inpainting performance of our method over three publicly available datasets: a high-quality human face dataset CelebA-HQ [40], a natural scene dataset collected from the real world Places [41], and a street view dataset containing various objects Paris StreetView [42]. The CelebA-HQ dataset contains 28000 training images and 2000 testing images. The Places dataset includes 365 categories, each containing 5000 training images and 100 validation images, where these validation images are used for testing. The Paris StreetView dataset has 14900 images in the training set and 100 images in the testing set.

For comparison purpose, we adopt four state-of-the-art inpainting methods: Contextual Attention (CA) [15], Shift-net (SH) [35], Gated Convolutions (GC) [43], and EdgeConnect (EC) [24], with both the centering and the irregular loss patterns. Same with the settings of the aforementioned methods, the centering masks are of sizes 120×120, and irregular masks are obtained from [21], with various missing-to-known area ratios.

A. *Qualitative Comparisons*

Fig. 6 shows the inpainting results of different algorithms for some representative testing images. Additional inpainting results can be found in the complementary materials. For CA [15], the main semantic information of the missing area can be well restored; but the incoherent artifacts around the boundary

TABLE I
QUANTITATIVE COMPARISONS OVER CELEBA-HQ WITH CENTERING MASK AMONG CA [15], SH [35], GC [43], EC [24] AND OURS. $^-$LOWER IS BETTER. $^+$HIGHER IS BETTER

Method	CA [15]	SH [35]	GC [43]	EC [24]	Ours
ℓ_1^-(%)	4.98	3.20	4.46	3.64	**3.16**
SSIM$^+$	0.882	0.924	0.897	0.912	**0.926**
PSNR$^+$	25.05	28.13	26.43	27.20	**28.34**

TABLE II
QUANTITATIVE COMPARISONS OVER PLACES WITH IRREGULAR MASK AMONG CA [15], SH [35], GC [43], EC [24] AND OURS. $^-$LOWER IS BETTER. $^+$HIGHER IS BETTER

Method	Mask	CA [15]	SH [35]	GC [43]	EC [24]	Ours
ℓ_1^-(%)	10-20%	5.11	2.82	3.07	2.78	**2.43**
	20-30%	9.05	5.03	5.45	4.95	**4.47**
	30-40%	13.40	7.53	8.25	7.42	**6.84**
	40-50%	17.61	10.43	11.57	10.33	**9.65**
SSIM$^+$	10-20%	0.887	0.921	0.929	0.924	**0.936**
	20-30%	0.805	0.860	0.869	0.868	**0.880**
	30-40%	0.724	0.793	0.802	0.804	**0.817**
	40-50%	0.642	0.719	0.726	0.732	**0.745**
PSNR$^+$	10-20%	24.08	27.93	27.67	27.29	**28.85**
	20-30%	20.99	24.94	24.24	24.82	**25.59**
	30-40%	18.92	22.80	21.82	22.63	**23.30**
	40-50%	17.56	21.11	19.90	20.91	**21.48**

are quite obvious (e.g., the first three rows). This phenomenon is especially visible when the missing region is located in the homogenous parts (e.g., the sixth row). SH [35] can generate visually much more realistic images due to the shift layer, the guidance loss and the improved network architecture. However,the boundary inconsistency is still very severe (e.g., the fourth row), and often some fine textures are highly blurry (e.g., the fifth row). GC [43] generally can produce pretty good results; but also leads to rather inconsistent, blurry artifacts in the texture regions and also visible distortions around the mask boundary (e.g., the fourth row). Furthermore, even though EC [24] can produce visually good results by first complementing the edge contours, some broken or blurred edges can be observed (e.g., the fourth and fifth rows). Also, in some cases, the mask boundaries are still quite obvious (e.g., the sixth row). This may be due to the inadequate guidance offered by the edges. Compared with these methods, our proposed model can learn more reasonable semantic relevance and generate more realistic inpainting results (especially those fine structures and texture regions), primarily thanks to the rich amount of guiding information provided by LBP learning and the employment of the new spatial attention layer in the inpainting network.

B. Quantitative Comparisons

In addition to the qualitative comparisons, we also compare different methods quantitatively for both centering and irregular masks, as shown in Tables I-II. Here, we adopt the commonly used metrics, namely, ℓ_1 loss, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). It can be seen that our method consistently outperforms all the competing algorithms.

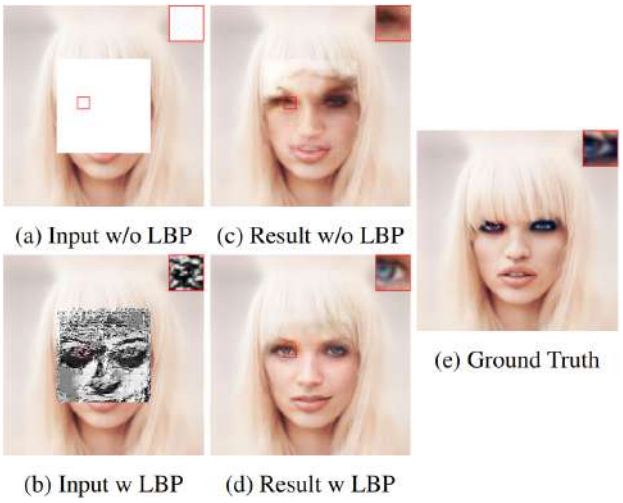


Fig. 7. Effect of the LBP learning network. (a)-(b) Inputs without or with the learned LBP. (c)-(d) Results without or with the LBP learning network. (e) Ground truth.

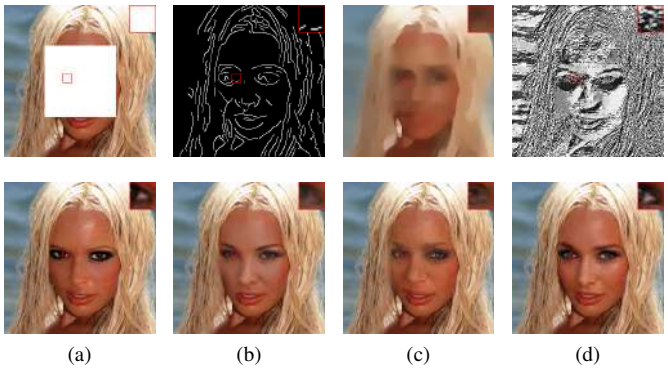


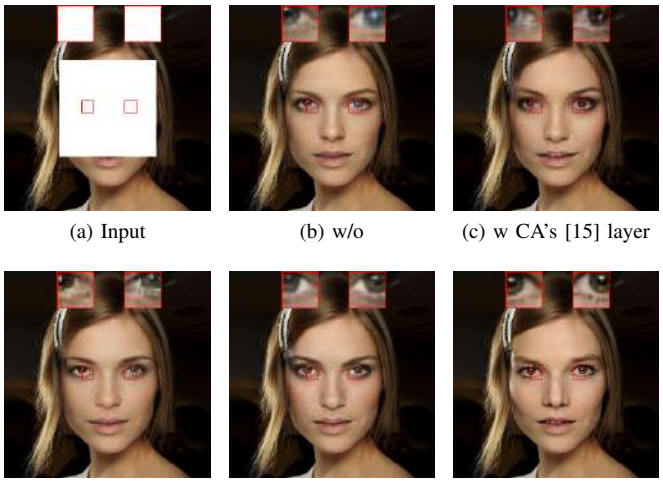
Fig. 8. Effect of the guidance provided by edges [48], RTV [58] and LBP [44]. (a) Input and ground truth. (b)-(d) The first row shows the predicted edges, RTV and LBP, and the second row presents corresponding results guided by them.

C. Ablation Studies

In this section, additional experiments are conducted to analyze how each component (e.g., LBP learning, the new spatial attention layer and the multi-level loss) of our proposed model contributes to the final inpainting results.

1) Effect of the LBP learning network: To investigate the effectiveness of the LBP learning network, we inpaint the images with or without the learned LBP in the missing region, respectively. Fig. 7 reports the comparison results, where Fig. 7(c) is the result without LBP learning, and Fig. 7(d) is the one guided by the learned LBP. It can be seen that, by learning the LBP of the missing region first, more structural information can be provided to significantly improve the inpainting performance, making the results sharper and more semantically reasonable.

Additionally, by replacing the LBP learning network with the edge generator [24] and structure reconstructor based on



(a) Input (b) w/o (c) w CA's [15] layer

(d) w SH's [35] layer (e) w our layer (f) Ground Truth

Fig. 9. Effect of different spatial attention layer. (a) Input. (b)-(e) Results without spatial layer, with CA's [15] layer, with SH's [35] layer and with our layer, respectively. (f) Ground truth.

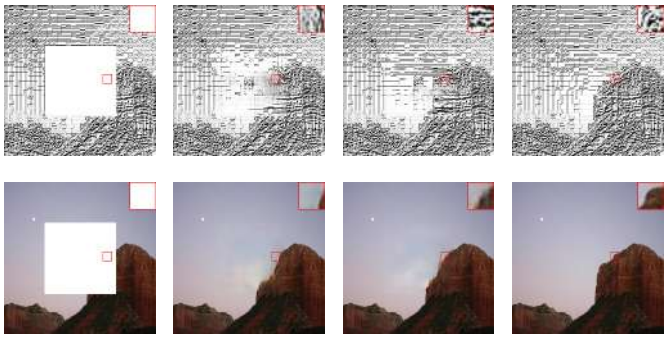
TABLE III
EFFECT OF DIFFERENT SPATIAL ATTENTIONS ON PLACES WITH CENTERING MASK. $^-$LOWER IS BETTER. $^+$HIGHER IS BETTER

Method	w/o	w CA's	w SH's	w our
$\ell_1^-(\%)$	7.68	7.64	7.65	**7.33**
SSIM$^+$	0.722	0.723	0.721	**0.729**
PSNR$^+$	22.33	22.39	22.41	**22.72**

relative total variation (RTV) [26], [58], respectively, we explore the guidance provided by different structural information (edges, RTV or LBP) for performing the inpainting tasks. In Fig. 8, we can see that the predicted edges contain discontinuities, and RTV, as an edge-preserved smooth method, inherently misses a lot of structural details. As a comparison, the sufficient structural information contained in LBP makes the result sharper (e.g., eyes and nose). This would suggest that LBP is a more appropriate candidate for providing structural information in the case of image inpainting.

2) Effect of the spatial attention layer: We evaluate the effectiveness of our spatial attention layer by removing it, replacing it with the contextual attention layer [15] or the shift layer [35], respectively. The results in Fig. 9 indicate that our spatial attention layer can better guarantee semantic coherence within the missing region (e.g., the restored eyes), while some artifacts and inconsistent contents are produced by CA's and SH's layer. The statistics in Table III also verify the superiority of our proposed spatial attention layer quantitatively (e.g., over 0.3 dB PSNR gains).

3) Hyperparameters of spatial attention layer: The considerable operations of convolutional filters performed in the spatial attention layer may cause memory overhead for GPUs [15]. One of the main factors influencing the calculation time is the size of the feature map, i.e., when the index h of the spatial attention layer is bigger, the feature map size goes larger, which requires more computational resources. However, when h is smaller, the performance could be degraded due to the insufficient number of extracted patches. To achieve better tradeoff between the efficiency and the performance, we set the resolution of the attention layer to 32×32, i.e., h = 13. Another influencing factor is the number of extracted patches used for spatial attention operations. Since negligible improvements can be brought when involving more patches to update the missing region, we heuristically determine $T = 2$ to save the cost.



(e) Input (f) w/o $\mathcal{L}_m$ (g) w $\mathcal{L}_m$ (h) GT.

Fig. 10. Effect of the multi-level loss $\mathcal{L}_m$ on Places with a centering mask. (a) Inputs. (b)-(c) LBP and inpainting results of our model without or with $\mathcal{L}_m$. (d) Ground truth.

TABLE IV
EFFECT OF THE MULTI-LEVEL LOSS $\mathcal{L}_m$ ON PLACES WITH CENTERING MASKS. $^-$LOWER IS BETTER. $^+$HIGHER IS BETTER

Method	$\ell_1^-(\%)$	SSIM$^+$	PSNR$^+$
w/o $\mathcal{L}_m$	4.80	0.844	24.20
w $\mathcal{L}_m$	**4.71**	**0.845**	**24.31**

4) Effect of the multi-level loss $\mathcal{L}_m$: We evaluate the effectiveness of the multi-level loss by adding or dropping $\mathcal{L}_m$ in the loss functions of the LBP learning and the image inpainting networks. As shown in Fig. 10, without the multi-level loss, the learned LBP could not satisfactorily recover some structural information, leading to inferior performance of the inpainting result. By incorporating the multi-level loss term, the LBP feature can be more faithfully predicted, and the inpainting result gets improved. Besides, quantitative comparisons in Table IV also verify the superiority of multi-level loss (e.g., around 0.11 dB PSNR gains).

V. CONCLUSIONS

In this paper, we have proposed a deep generative model for image inpainting with LBP learning and a new spatial attention mechanism. The proposed model has been formed with two networks: a LBP learning network, which aims to learn the LBP feature of the missing region, and an image inpainting network, which generates the inpainting results by using the learned LBP as a guidance. Within the two-stage framework, how to select these two sub-networks is very crucial and could lead to vastly different results. Compared with edges and RTV, LBP features contain much more structural information and are almost parameter-free. Furthermore, we have designed a new spatial attention layer, and have incorporated it into

the image inpainting network. The proposed spatial attention strategy not only considers the dependency between the know region and the filled region, but also the one within the filled region. Such dependency, though obviously important, has been overlooked by all the existing schemes. Experimental results have been provided to demonstrate the superiority of the proposed model.

APPENDIX A

The architectures of the generators G_1 or G_2, the discriminators D_1 or D_2 are shown in Table V. Conv(f, k, s, p) means a convolutional layer with f filters, kernel size k, stride s and padding p. DeConv denotes deconvolutional layer. IN represents InstanceNorm and LReLU is LeakyReLU with slop of 0.2. Cat(Layer b_1, Layer b_2) concatenates the outputs of Layer b_1 and Layer b_2. Tanh and Sigmoid are the activation functions. We embed our spatial attention layer *only* in Layer 13 of G_2.

TABLE V
THE ARCHITECTURES OF THE GENERATORS G_1 OR G_2, AND THE DISCRIMINATORS D_1 OR D_2.

The architecture of the generators G_1 or G_2
[Layer 1] Conv(64, 4, 2, 1);
[Layer 2] LReLU; Conv(128, 4, 2, 1); IN;
[Layer 3] LReLU; Conv(256, 4, 2, 1); IN;
[Layer 4] LReLU; Conv(512, 4, 2, 1); IN;
[Layer 5] LReLU; Conv(512, 4, 2, 1); IN;
[Layer 6] LReLU; Conv(512, 4, 2, 1); IN;
[Layer 7] LReLU; Conv(512, 4, 2, 1); IN;
[Layer 8] LReLU; Conv(512, 4, 2, 1); ReLU; DeConv(512, 4, 2, 1); IN;
[Layer 9] Cat(Layer 8, Layer 7); ReLU; DeConv(512, 4, 2, 1); IN;
[Layer 10] Cat(Layer 9, Layer 6); ReLU; DeConv(512, 4, 2, 1); IN;
[Layer 11] Cat(Layer 10, Layer 5); ReLU; DeConv(512, 4, 2, 1); IN;
[Layer 12] Cat(Layer 11, Layer 4); ReLU; DeConv(256, 4, 2, 1); IN;
[Layer 13] Cat(Layer 12, Layer 3); **(SpatialAttention)**; ReLU; DeConv(128, 4, 2, 1); IN;
[Layer 14] Cat(Layer 13, Layer 2); ReLU; DeConv(64, 4, 2, 1); IN;
[Layer 15] Cat(Layer 14, Layer 1); ReLU; DeConv(3, 4, 2, 1); Tanh;

The architectures of the discriminators D_1 or D_2
[Layer 1] Conv(64, 4, 2, 1);
[Layer 2] LReLU; Conv(128, 4, 2, 1); IN;
[Layer 3] LReLU; Conv(256, 4, 2, 1); IN;
[Layer 4] LReLU; Conv(512, 4, 2, 1); IN;
[Layer 5] LReLU; Conv(1, 4, 2, 1); Sigmoid;

REFERENCES

[1] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2. IEEE, 1999, pp. 1033–1038.

[2] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. Conf. Comput. Graph. Interactive Tech.* ACM, 2001, pp. 341–346.

[3] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 453–465, 2011.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, 2003.

[5] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.

[6] C. Ling, C. Lin, C. Su, Y. Chen, and H. M. Liao, "Virtual contour guided video object inpainting using posture mapping and retrieval," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 292–302, 2011.

[7] J. Sun, L. Yuan, J. Y. Jia, and H. Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, 2005.

[8] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. Graph.*, vol. 26, no. 3, p. 4, 2007.

[9] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.* IEEE, 2008, pp. 1–8.

[10] W. Lie, C. Hsieh, and G. Lin, "Key-frame-based background sprite generation for hole filling in depth image-based rendering," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1075–1087, 2018.

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

[12] C. H. Cheung, K. N. Ngan, and L. Sheng, "Spatio-temporal disocclusion filling using novel sprite cells," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1376–1391, 2018.

[13] T. Nguyen, B. Kim, and M. Hong, "New hole-filling method using extrapolated spatio-temporal background information for a synthesized free-view," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1345–1358, 2019.

[14] J. Liu, S. Yang, Y. Fang, and Z. Guo, "Structure-guided image inpainting using homography transformation," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3252–3265, 2018.

[15] J. H. Yu, Z. Lin, J. M. Yang, X. H. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2018, pp. 5505–5514.

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

[17] B. Dolhansky and C. C. Ferrer, "Eye in-painting with exemplar generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2018, pp. 7902–7911.

[18] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, p. 107, 2017.

[19] P. Isola, J. Y. Zhu, T. H. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2017, pp. 1125–1134.

[20] A. Li, J. Z. Qi, R. Zhang, X. J. Ma, and K. Ramamohanarao, "Generative image inpainting with submanifold alignment," in *Proc. Int. Jt. Conf. AI*, 2019, pp. 811–817.

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

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

[23] Y. Q. Ma, X. L. Liu, S. H. Bai, L. Wang, D. L. He, and A. Liu, "Coarse-to-fine image inpainting via region-wise convolutions and non-local correlation," in *Proc. Int. Jt. Conf. AI*, 2019.

[24] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edgeconnect: generative image inpainting with adversarial edge learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[25] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2016, pp. 2536–2544.

[26] Y. R. Ren, X. M. Yu, R. N. Zhang, T. H. Li, S. Liu, and G. Li, "Structureflow: image inpainting via structure-aware appearance flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 181–190.

[27] M. Sagong, Y. Shin, S. Kim, S. Park, and S. Ko, "Pepsi: fast image inpainting with parallel decoding network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2019, pp. 11 360–11 368.

[28] Y. H. Song, C. Yang, Z. Lin, X. F. Liu, Q. Huang, H. Li, and C. J. Kuo, "Contextual-based image inpainting: infer, match, and translate," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[29] Y. H. Song, C. Yang, Y. J. Shen, P. Wang, Q. Huang, and C. J. Kuo, "Spg-net: segmentation prediction and guidance network for image inpainting," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–12.

[30] N. Wang, J. Y. Li, L. F. Zhang, and B. Du, "Musical: multi-scale image contextual attention learning for inpainting," in *Proc. Int. Jt. Conf. AI*, 2019.

[31] Y. Wang, X. Tao, X. J. Qi, X. Y. Shen, and J. Y. Jia, "Image inpainting via generative multi-column convolutional neural networks," in *Proc. Neural Info. Process. Syst.*, 2018, pp. 331–340.
[32] J. Xiao, L. Liao, Q. G. Liu, and R. M. Hu, "Cisi-net: explicit latent content inference and imitated style rendering for image inpainting," in *Proc. AAAI Conf. Arti. Intell.*, vol. 33, 2019, pp. 354–362.
[33] C. H. Xie, S. H. Liu, C. Li, M. M. Chen, W. M. Zou, X. Liu, S. L. Wen, and E. Ding, "Image inpainting with learnable bidirectional attention maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8858–8867.
[34] W. Xiong, J. H. Yu, Z. Lin, J. M. Yang, X. Lu, C. Barnes, and J. B. Luo, "Foreground-aware image inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2019, pp. 5840–5848.
[35] Z. Y. Yan, X. M. Li, M. Li, W. M. Zuo, and S. G. Shan, "Shift-net: image inpainting via deep feature rearrangement," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–17.
[36] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2017, pp. 5485–5493.
[37] Y. H. Zeng, J. L. Fu, H. Y. Chao, and B. N. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2019, pp. 1486–1494.
[38] D. Zhao, B. L. Guo, and Y. Y. Yan, "Parallel image completion with edge and color map," *Appl. Sci.*, vol. 9, no. 18, p. 3856, 2019.
[39] B. Waller, M. S. Nixon, and J. N. Carter, "Image reconstruction from local binary patterns," in *Proc. Int. Conf. Signal-Image Tech. Internet-Based Syst.* IEEE, 2013, pp. 118–123.
[40] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.
[41] B. L. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: a 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, 2017.
[42] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, "What makes paris look like paris?" *ACM Trans. Graph.*, vol. 31, no. 4, 2012.
[43] J. H. Yu, Z. Lin, J. M. Yang, X. H. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4471–4480.
[44] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recogn.*, vol. 29, no. 1, pp. 51–59, 1996.
[45] B. C. Zhang, Y. S. Gao, S. Q. Zhao, and J. Z. Liu, "Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, 2010.
[46] T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," *Pattern Recogn.*, vol. 32, no. 3, pp. 477–486, 1999.
[47] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, 2006.
[48] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.
[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Computer-Assisted Int.* Springer, 2015, pp. 234–241.
[50] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: the missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
[51] B. Xu, N. Y. Wang, T. Q. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.
[52] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[53] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 694–711.
[54] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2016, pp. 2414–2423.
[55] M. S. Sajjadi, B. Scholkopf, and M. Hirsch, "Enhancenet: single image super-resolution through automated texture synthesis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4491–4500.
[56] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, vol. 2. IEEE, 2005, pp. 60–65.
[57] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[58] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, 2012.