Dynamic Selection Network for Image Inpainting

Ning Wang, *Student Member, IEEE*, Yipeng Zhang, *Student Member, IEEE*,
and Lefei Zhang, *Senior Member, IEEE*

***Abstract*— Image inpainting is a challenging computer vision task that aims to fill in missing regions of corrupted images with realistic contents. With the development of convolutional neural networks, many deep learning models have been proposed to solve image inpainting issues by learning information from a large amount of data. In particular, existing algorithms usually follow an encoding and decoding network architecture in which some operations with standard schemes are employed, such as static convolution, which only considers pixels with fixed grids, and the monotonous normalization style (e.g., batch normalization). However, these techniques are not well-suited for the image inpainting task because the random corrupted regions in the input images tend to mislead the inpainting process and generate unreasonable content. In this paper, we propose a novel dynamic selection network (DSNet) to solve this problem in image inpainting tasks. The principal idea of the proposed DSNet is to distinguish the corrupted region from the valid ones throughout the entire network architecture, which may help make full use of the information in the known area. Specifically, the proposed DSNet has two novel dynamic selection modules, namely, the validness migratable convolution (VMC) and regional composite normalization (RCN) modules, which share a dynamic selection mechanism that helps utilize valid pixels better. By replacing vanilla convolution with the VMC module, spatial sampling locations are dynamically selected in the convolution phase, resulting in a more flexible feature extraction process. Besides, the RCN module not only combines several normalization methods but also normalizes the feature regions selectively. Therefore, the proposed DSNet can illustrate realistic and fine-detailed images by adaptively selecting features and normalization styles. Experimental results on three public datasets show that our proposed method outperforms state-of-the-art methods both quantitatively and qualitatively.**

***Index Terms*— Image inpainting, deep learning, dynamic selection.**

I. Introduction

IMAGE inpainting, one of the challenging computer vision tasks, can restore realistic contents of corrupted images given the location of damaged areas (also called missing regions or holes). Image inpainting is widely used in image processing applications, such as face editing [1], privacy protection [2], and object removal [3], [4]. The key point of the inpainting task is to generate visually-pleasing contents in missing regions.

In recent years, many excellent methods have been proposed for image inpainting tasks. These methods always fill corrupted regions with plausible contents based on the general assumption that the contents of unknown regions (foreground) can be learned from known regions (background). Typically, these methods can be divided into two classes, namely, traditional and trainable methods. Traditional methods always reconstruct damaged regions based on diffusion [5]–[8] or patch [9]–[11]. Trainable methods refer to learning-based methods that employ convolutional neural networks (CNNs) [12]–[23]. Most of the current image inpainting methods are based on CNNs, which are proven to generate good results for images with large holes and complex backgrounds.

The development of deep learning has stimulated the emergence of trainable inpainting methods. Pathak *et al.* proposed the context encoders method [12], which is the pioneer learning-based method. Some methods [13]–[16] attempt to generate more realistic results by introducing a generative adversarial network [24]. Furthermore, some methods [16], [17] combine different networks to learn more information from the background. Other methods [18], [19] are designed by exerting the constraint of edge information. Trainable methods can hallucinate reasonable results by learning semantic information from deep layers. However, these methods still result in failure cases, which may include artifacts, distorted structures, and blurry textures, especially when the background is complex or the missing regions are immense.

One core reason for the failures of the convolutional based method is the instability of the convolutional features in both learning and inference stages of the network. In specific, due to the unpredictable corrupted regions in the inputs, it becomes difficult for the convolutional kernels to consistently capture information from the inputs, i.e. the same feature values in the convolutional feature map of an inpainting model might represent completely different things. These unstable features then mislead the inference process of the network and yield undesirable outcomes. To relieve this kind of instability, one straightforward approach is to perform boundary feature re-normalization, which re-scales the values of features based on the number of valid pixels used to generate them [21]. An alternative approach is to use the dynamic gated selection to select features, which filters the invalid/low-quality pixels

Manuscript received June 27, 2020; revised November 25, 2020 and December 28, 2020; accepted December 28, 2020. Date of publication January 8, 2021; date of current version January 14, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62076188, in part by the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170. The numerical calculations in this paper had been supported by the supercomputing system in the Supercomputing Center of Wuhan University. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sos S. Agaian. *(Corresponding author: Lefei Zhang.)*

Ning Wang and Yipeng Zhang are with the School of Computer Science, Institute of Artificial Intelligence, Wuhan University, Wuhan 430079, China (e-mail: wang_ning@whu.edu.cn; zyp91@whu.edu.cn).

Lefei Zhang is with the School of Computer Science, Institute of Artificial Intelligence, Wuhan University, Wuhan 430079, China, and also with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: zhanglefei@whu.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.3048629

using the calculated gate [22]. Meanwhile, Yu *et al.* [23] proposed the region normalization (RN) method by replacing full-spatial instance normalization (IN) with spatial region-wise normalization.

While these methods have achieved remarkable results in improving the stability of the convolutional feature maps, they ignore the essential differences between the corrupted feature maps and the intact ones, i.e. the sparsity of useful information in the hole boundary. We argue that, rather than simply adjusting the feature values, it is more important to recover the richness of information in the corrupted regions and pass the information precisely and stably during the forward pass of the network. To end this, we devise a unified model that guarantees the robustness to the interference from the invalid information in learning-based methods for high-quality inpainting.

In this paper, we propose a novel dynamic selection network (DSNet) model that adopts the dynamic selection mechanism based on the U-Net [25]. The proposed DSNet contains two dynamic selection modules. The first module is the validness migratable convolution (VMC) module. In the VMC module, the concatenated feature maps of a corrupted image and its mask image (the corresponding location of corrupted regions) are sent into modified deformable convolution [26] to select useful information dynamically. The migrated feature maps are then sent to a standard convolution layer. With the VMC module, we can extract information from the background flexibly and discard useless information in the missing regions. The second module is the regional composite normalization (RCN) module, designed to select valid information in the normalization phase. RCN is performed region-wise to satisfy the demand of the inpainting task. Furthermore, RCN takes advantage of several normalization methods simultaneously [27] to determine the most suitable combination for inpainting tasks dynamically. The proposed DSNet solves the problems mentioned above by dynamically selecting valid information. Experiments show that the proposed algorithm can reproduce missing regions with plausible semantics and achieve realistic results.

Specifically, the difference between the fixed location sampling convolution and the proposed VMC is displayed in Figure 1. When the sampling region contains valid information and invalid information, fixed grids convolution manners, including the standard convolution, the partial convolution [21], and the gated convolution [22], tend to generate unreasonable information. The standard convolution always misuses the invalid information. Although the partial convolution avoids the invalid regions, the scaling factor usually results in inaccurate pixels. The gated convolution can dynamically produce gating values to decrease the invalid information, while the issue still exists when the sampling regions mix in invalid information. The proposed VMC can borrow valid information from surrounding to replace the invalid regions in sampling. Compared with fixed grids convolution, VMC utilizes more valid information when generating new pixels, and the gradient descent is more correct in VMC due to the good updating of weights.

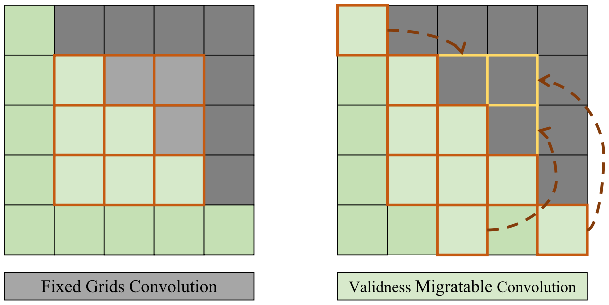


Fig. 1. Illustration of fixed grids convolution (left) and validness migratable convolution (right). The green grids represent the valid regions, and the gray grids mean the invalid regions. The grids with orange border denote the sampling regions. When the sampling region contains valid information and invalid information, fixed grids convolution manner tends to generate unreasonable information, and VMC tends to borrow valid information from surrounding to replace the invalid regions. The VMC utilizes more valid information when generating new pixels than the fixed grids convolution.

The main contributions of this paper are summarized as follows:

- We establish a novel DSNet for image inpainting. With the dynamic selection mechanism, we can avoid the interference of the invalid information from holes and utilize valid information adaptively. The DSNet contains two important modules, namely, the VMC and RCN modules.
- We devise the VMC module to solve the convolution problem, combining the deformable convolution with the regional mechanism. In addition to ignoring invalid information from missing regions, VMC can flexibly learn information from diverse locations by addressing the limitation of fixed sampling grids.
- We design the RCN module to replace general normalization, which dynamically composites three region-wise normalization manners based on circumstances. The RCN module helps obtain more valuable information suitable for inpainting tasks to avoid the expectation and variance shifts caused by single full-spatial normalization.
- Experiments on three public datasets demonstrate the remarkable performance of the proposed method qualitatively and quantitatively.

The rest of this paper is organized as follows. Section II summarizes the related works on image inpainting and normalization methods. Section III introduces the proposed DSNet model and the learning formulation in detail. Section IV discusses the experimental settings, including datasets, comparison methods, and training settings. Section V demonstrates the superior performance of the proposed algorithms on several standard datasets. The effects of the RCN and VMC modules, the important weights in the RCN module, and the training process are also analyzed in this section. Section VI concludes this paper with proposals for future work.

II. Related Works

A. Image Inpainting

Existing image inpainting methods are usually divided into two categories, namely traditional inpainting methods and trainable inpainting methods.

1) Traditional Inpainting: In the image inpainting task, many areas need to be repaired, which should acquire plausible content from the background. With the development of deep learning, the useful semantic information for mask regions can be learned by training a large amount of data. But before these trainable models, the image inpainting tasks are completed by statistic diffusing or patch matching, i.e., diffusion-based methods [5]–[8] and patch-based methods [9]–[11].

Diffusion-based methods can generate locally smooth results for tiny damaged regions by propagating neighboring information from boundary to holes. For example, Levin *et al.* [5] tried to inpaint images with the most probable contexts based on histograms of local features. However, diffusion-based methods cannot recover large regions with textures. Patch-based methods solve this problem by searching suitable patches from the background and copying them to missing regions, requiring expensive computing costs. PatchMatch [9] searches approximate nearest-neighbor matches via random sampling to reduce computing costs. But it is nevertheless more time-consuming than diffusion-based methods. What is more, patch-based methods tend to find unsuitable patches when the background is complex. Some methods [28], [29] combine diffusion-based methods with patch-based methods to solve inpainting tasks. However, these combined methods are not significantly improved compared to patch-based methods. Due to the low ability to obtain high-level information, the performance of traditional inpainting methods is still limited.

2) Trainable Inpainting: Trainable inpainting methods have seen a surge in the development of deep networks in recent years. With the proposal of Context Encoders method [12], which is the first to adopt an encoder-decoder framework to solve inpainting tasks, increasing works have attempted to handle the inpainting problem with trainable networks. Yang *et al.* proposed MNPS [17] based on context encoders to maintain structures and details for missing regions with the joint optimization of the content and texture networks. Moreover, Iizuka *et al.* proposed GL [14] to generate locally and globally consistent results by introducing global and local context discriminators.

Besides, some methods introduce special modules to take advantage of feature representations flexibly. Shift-Net [30] adds a shift-connection layer in U-Net to retain low-level information for fine details. SSDCGN [31] introduces a dense skip connection in a battery of symmetric encoder-decoder groups to maximize the semantic extraction. StructureFlow [32] employs the appearance flows learned from recovered structures to yield high-frequency details. Some methods involve different attention modules to adaptively utilize features from backgrounds as references, such as the contextual attention module [15], the multi-scale attention module [33], the attention transfer network [34], learnable bidirectional attention maps [35], and the coherent semantic attention layer [36].

Furthermore, structure and gradient information are also used as constraints. EdgeConnect [18] solves the inpainting task by separating the problem into two parts: structure prediction and image completion. Xiong *et al.* [16] proposed a foreground-aware image inpainting method that inferences the contour of saliency objects in a coarse-to-fine fashion. Li *et al.* [19] progressively inpainted structures with the special visual structure reconstruction layer. GAIN [37] is proposed by fusing features with natural gradient information to facilitate inpainting.

Some methods attempt to perform the image inpainting task by shrinking mask regions progressively. Zhang *et al.* [38] proposed the PGN method, which divides the inpainting process into several phases, and each phase aims to recover the current boundary of missing regions. Guo *et al.* [39] proposed the FRRN model with full-resolution residual blocks for progressively filling irregular holes. Li *et al.* [40] proposed the RFR-Net that identifies the mask regions and reconstructs boundaries in the feature representation space.

Trainable methods demonstrate a powerful ability to restore semantic contents and high-quality details with the support of massive data. However, the pixels from missing regions are invalid and should not be used in feature extraction. Therefore, standard convolution is unsuitable for the image inpainting task. Fortunately, some state-of-the-art methods detect this problem. For example, PConv [21] and GatedConv [22] effectively improve the performance of image inpainting models with new convolution methods.

B. Normalization

Independent and identically distributed data can simplify the training of conventional machine learning models and improve the predictive ability of machine learning models. However, the parameter update in each layer changes the data distribution. The stack layers in deep neural networks drastically change the data distribution, and internal covariate shift (ICS) occurs. Therefore, deep neural networks must use normalization to maintain data independent and identically distributed.

Batch normalization (BN) [41] is a milestone in deep learning that has been widely used in networks to accelerate the gradient descent and provide a robust training process. BN is a minibatch-wise normalization. However, BN is not very suitable for some image generation tasks (e.g., style transfer), because the generated results depend on an image instance. IN [42] is proposed to adapt image generation tasks in a channel-wise manner and achieves better performance than BN in certain image generation scenarios. Layer normalization (LN) [43], which is performed layer-wise, is proposed for recurrent neural networks (RNNs). LN is effective for RNNs, but it is not as good as BN for CNNs. LN demonstrates its effectiveness at stabilizing the hidden state dynamics in recurrent networks. Besides, BN is sensitive to the batch size. If the batch size is small, the effect will be not very good. Therefore, group normalization (GN) [44] is proposed as a simple alternative to BN by separating channels into groups and computing within each group. Weight normalization [45] is useful in RNNs and reinforcement learning. Moreover, some researchers proposed new normalization methods by combining several basic normalization methods, such as batch-instance normalization [46] and switchable normalization [27].

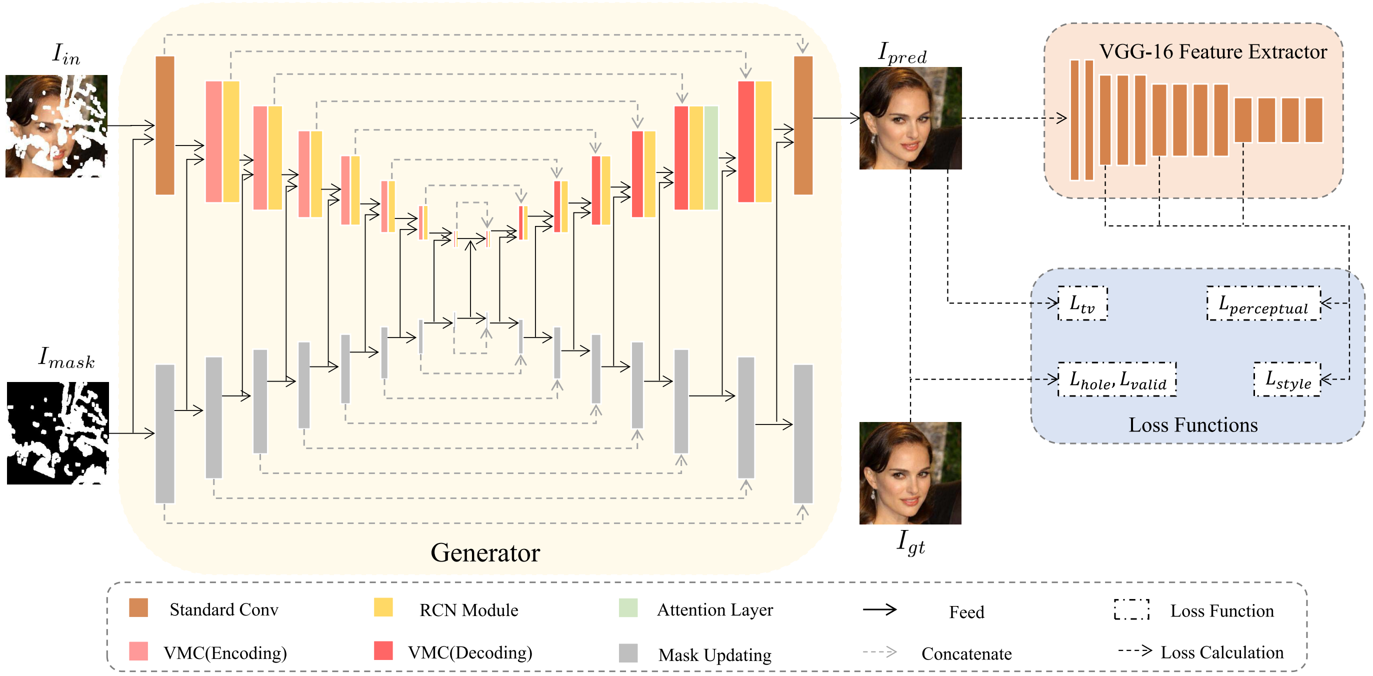


Fig. 2. The framework of proposed **Dynamic Selection Network** (DSNet). Images of damaged image I_{in} and mask I_{mask} are sent into a generator. The generator consists of standard convolutional layers at both ends, cascaded VMC and RCN modules, and a contextual attention layer. $L_{perceptual}$ and L_{style} are calculated through a VGG-16 feature extractor, L_{tv} is generated based on the predicted image I_{pred}, and L_{hole} and L_{valid} are obtained based on the ground truth I_{gt} and the predicted image I_{pred}.

In image inpainting tasks, most methods adopt a single normalization approach across the networks' layers, and the two widely used normalization approaches are BN and IN. In particular, RN [23] replaces IN with two region-wise normalization modules, namely, basic RN (RN-B) and learnable RN (RN-L), to adapt to the image inpainting task.

III. The Proposed Approach

In this section, we introduce a novel image inpainting algorithm named **Dynamic Selection Network (DSNet)**. Our DSNet model adopts the U-Net [25] as the basic network. First, the VMC module is designed to flexibly utilize valid information by learning additional offsets during convolution. Then, the RCN module combines BN [41], IN [42], and LN [43] to dynamically learn the weights corresponding to different normalizers. In particular, we consider irregular mask locations. Therefore, both the VMC and RCN modules are performed region-wise to avoid the interference of the invalid information from the mask regions. A pixel-wise attention layer is employed to provide a more reasonable detail. Finally, several losses are joint as an objective function to help generate realistic results.

The DSNet framework is displayed in **Figure 2**, and **Table I** shows the detailed architecture of the generator. Input represents the source of the input of each layer. Operator represents the operation of each layer. Normalization refers to the normalization method adopted after convolution. In_Size and Out_Size refer to the sizes of the feature maps before and after processing, respectively. Ori indicates the size of the original input image. K, S, and P represent the kernel size, stride, and padding of the operators, respectively. Besides, Num_C denotes the channel number of the output feature maps. Act_Func represents the non-linear function after the layer.

A. Validness Migratable Convolution

In learning-based methods, CNNs are widely used to extract feature representations. Standard convolution usually misuses the invalid information of missing regions, creating some artifacts, distorted structures, and blurred textures that are inconsistent with the background. Furthermore, the convolution unit always samples input feature maps at fixed locations. The fixed geometric structures of the CNN module limit the ability to model complex situations for computer vision, especially in image inpainting tasks that have arbitrary locations of mask regions and complex backgrounds. Therefore, we need a particular module for migrating valid information and making full use of them.

Thus, we devise a novel VMC module to replace vanilla convolution. The VMC module can dynamically select useful information, ignoring the invalid information of missing regions and addressing the limitations of the sampling shapes. The VMC module consists of two parts: 1) validness migration, and 2) convolution with mask updating.

1) Validness Migration: The validness migration operation remains the same across different channels. For notation clarity, we introduce the feature migration in 2D.

In a vanilla convolution unit, input feature map x is sampled over regular grid τ. Then, the sampling results are weighted

TABLE I
ARCHITECTURE OF THE GENERATOR

Generator Architecture										
Layer	Input	Operator	Normalization	In_Size	Out_Size	K	S	P	Num_C	Act_Func
0	I_in, I_mask	Standard Conv-0	None	Ori	Ori/2	7	2	3	64	ReLU
1	F_Enc0	VMC-encoding-1	RCN	Ori/2	Ori/4	5	2	2	128	ReLU
2	F_Enc1	VMC-encoding-2	RCN	Ori/4	Ori/8	5	2	2	256	ReLU
3	F_Enc2	VMC-encoding-3	RCN	Ori/8	Ori/16	3	2	1	512	ReLU
4	F_Enc3	VMC-encoding-4	RCN	Ori/16	Ori/32	3	2	1	512	ReLU
5	F_Enc4	VMC-encoding-5	RCN	Ori/32	Ori/64	3	2	1	512	ReLU
6	F_Enc5	VMC-encoding-6	RCN	Ori/64	Ori/128	3	2	1	512	ReLU
7	F_Enc6	VMC-encoding-7	RCN	Ori/128	Ori/128	3	2	1	512	ReLU
8	cat(F_Enc7,F_Enc6)	VMC-decoding-8	RCN	Ori/128	Ori/64	3	1	1	512	Leaky_ReLU
9	cat(F_Dec8,F_Enc5)	VMC-decoding-9	RCN	Ori/64	Ori/32	3	1	1	512	Leaky_ReLU
10	cat(F_Dec9,F_Enc4)	VMC-decoding-10	RCN	Ori/32	Ori/16	3	1	1	512	Leaky_ReLU
11	cat(F_Dec10,F_Enc3)	VMC-decoding-11	RCN	Ori/16	Ori/8	3	1	1	512	Leaky_ReLU
12	cat(F_Dec11,F_Enc2)	VMC-decoding-12	RCN	Ori/8	Ori/8	3	1	1	256	Leaky_ReLU
13	cat(F_Dec12,F_Enc1)	VMC-decoding-13	RCN	Ori/4	Ori/4	3	1	1	128	Leaky_ReLU
	F_Dec13	Attention		Ori/4	Ori/2				128	
14	cat(F_Att,F_Enc0)	VMC-decoding-14	RCN	Ori/2	Ori	3	1	1	64	Leaky_ReLU
15	cat(F_Dec14,F_Input)	Standard Conv-15	None	Ori	Ori	3	1	1	3	None

by w. Finally, the sum of the weighted results is obtained. Location s in the output feature map y can be described as follows:

$$y(s) = \sum_{s_n \in \tau} w(s_n) \cdot x(s + s_n) \tag{1}$$

Grid τ defines the receptive field and dilation size of a convolution unit. For instance, the grid τ of 3×3 kernel with dilation 1 can be represented as follows:

$$\tau = \{(-1, -1), (-1, 0), \dots, (0, 1), (1, 1)\} \tag{2}$$

In the VMC module, we augment grid τ with feature migration Δs_n, where $n = 1, \dots, N$ and $N = |\tau|$. Then, location s in the output feature map is described as follows:

$$y(s) = \sum_{s_n \in \tau} w(s_n) \cdot x(s + s_n + \Delta s_n) \tag{3}$$

The Eqn 3 is actually the 2D mode of the validness migration process. Unlike regular sampling location s_n, the sampling location translates to the variable, that is, $s_n + \Delta s_n$ in Equation 3. Δs_n is adaptively learned from the input feature map. However, Δs_n is usually fractional, resulting in the fractional sampling location $(s_n + \Delta s_n)$ of the input feature map. The fractional sampling location fails to meet the requirement of differentiability in back-propagation. Therefore, we adopt bilinear interpolation, like STN [47], to guarantee back-propagation through this sampling mechanism.

We employ $\tilde{s}$ to represent $s + s_n + \Delta s_n$ and use t to enumerate all integer spatial locations in the feature map x. Then, the bilinear interpolation is expressed as follows:

$$x(\tilde{s}) = \sum_{t} \max(0, 1 - |t_x - \tilde{s}_x|) \cdot \max(0, 1 - |t_y - \tilde{s}_y|) \cdot x(t) \tag{4}$$

2) Convolution and Mask Updating: The proposed DSNet is built on U-Net. We must take different convolution operations in the encoding and decoding stages due to the skip connection in U-Net.

In **encoding layers**, damaged image feature map $\phi_{in}^{(l)}$ and mask feature map $\phi_m^{(l)}$ are concatenated as input $\phi^{(l)}$.

$$\phi^{(l)} = f_{concat}(\phi_{in}^{(l)}, \phi_m^{(l)}) \tag{5}$$

l represents the layer in the encoding stage. f_{concat} is the operation for concatenating feature maps.

We use f_{vm} to denote the valid migration process, which is the 3D mode of Equation 3. Then, valid information is migrated by the learning feature offsets of input feature map $\phi^{(l)}$,

$$\phi_{vm}^{(l)} = f_{vm}(\phi^{(l)}) \tag{6}$$

$\phi_{vm}^{(l)}$ is the deformed feature map of $\phi^{(l)}$.

After regional combination, the output feature map in the encoding stage of the VMC module is calculated as follows:

$$\widehat{\phi}_{vm}^{(l)} = \phi_{vm}^{(l)} * (1 - \phi_m^{(l)}) + \phi_{in}^{(l)} * \phi_m^{(l)} \tag{7}$$

$$\phi_{out}^{(l)} = f_{conv}(\widehat{\phi}_{vm}^{(l)}) \tag{8}$$

In the encoding stage, the mask is updated as follows:

$$\phi_{mu}^{(l)} = f_{max}(\phi_m^{(l)}) \tag{9}$$

f_{max} represents max-pooling, which has the same kernel size and stride as f_{conv}.

In the decoding layers, feature map $\widehat{\phi}_{vm}^{(L)}$ is obtained similarly as the encoding layers.

$$\phi_{de}^{(L)} = f_{concat}(\phi_{in}^{(L)}, \phi_m^{(L)}) \tag{10}$$

$$\phi_{vm}^{(L)} = f_{vm}(\phi_{de}^{(L)}) \tag{11}$$

$$\widehat{\phi}_{vm}^{(L)} = \phi_{vm}^{(L)} * (1 - \phi_m^{(L)}) + \phi_{in}^{(L)} * \phi_m^{(L)} \tag{12}$$

L is the layer in the decoding stage.

However, the feature maps from encoding stage $(\phi_{in}^{(l)}, \phi_m^{(l)})$ are also concatenated.

$$\phi_{en}^{(L)} = f_{concat}(\phi_{in}^{(l)}, \phi_m^{(l)}) \tag{13}$$

Then, the output feature map in the decoding stage of the VMC module is calculated as follows:

$$\phi_{out}^{(L)} = f_{conv}(f_{concat}(\phi_{en}^{(L)}, \widehat{\phi}_{vm}^{(L)})) \tag{14}$$

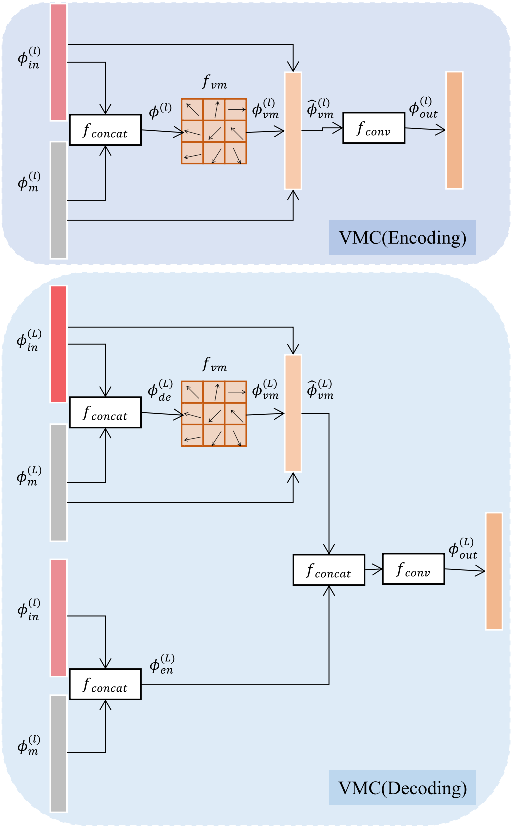


Fig. 3. The framework of **Validness Migratable Convolution** (VMC) module. VMC module has different behaviors in the encoding and decoding stages due to the skip connections in U-Net. The encoding part of VMC is shown at the top, and the decoding part is shown at the bottom.

In the decoding stage, the mask is updated by upsampling.

$$\phi_{mu}^{(L)} = f_{up}(\phi_m^{(L)}) \tag{15}$$

Upsampling function f_{up} is implemented by nearest interpolation in the DSNet.

The processes of the encoding and decoding stages of the VMC module are shown at the top and bottom of Figure 3, respectively.

B. Regional Composite Normalization

1) General Normalization: The most used normalization methods include BN [41], LN [43], IN [42], and GN [44]. The general normalization formulas are as follows:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}] + \epsilon}} \tag{16}$$

$$y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)} \tag{17}$$

Taking BN as an example, we normalize each dimension for a layer with d-dimensional input (i.e., $x = (x^{(1)}, \ldots, x^{(d)})$) with Equation 16. In this equation, expectation $\mathrm{E}[x^{(k)}]$ and variance $\mathrm{Var}[x^{(k)}]$ can be computed over the training dataset, where k represents the k^{th} dimension. ϵ is a small number to avoid the systematic errors caused by the denominator being equal to zero. However, when the operation in Equation 16 is adopted, the transformation inserted in the network cannot represent the identity transform. Therefore, additional scale parameter $\gamma^{(k)}$ and shift parameter $\beta^{(k)}$, which are learned along with the original model parameters, are utilized for normalized value $\widehat{x}^{(k)}$. With the help of Equation 17, the network restores the power of representation.

The difference between different normalization methods is that the set of activation values is different when calculating the expectation and the variance. The activation value set of BN comprises the activation values of different instances from the same neuron. The activation value set of LN includes the activation value of all neurons in the same layer of a single instance. For IN, the activation value set includes all values in a single feature map for a single instance. Furthermore, the activation value set of GN includes all values in the multiple feature maps of a single instance, whose number is between those in IN and LN.

2) Normalization in Image Inpainting Task: In the image inpainting task, we must always recover the masked regions. The values of these masked regions are invalid while training the datasets in CNNs, which may lead to artifacts and blurry textures in the inpainted results. Some methods attempt to solve the interference of these invalid information by replacing vanilla convolution with a new convolution, like PConv [21] and GatedConv [22]. However, they still utilize general normalization methods, which do not consider the interfere of mask distribution on normalization.

Suppose we have an image (I) with holes, we can compute its expectation and variance in two manners. In **manner I**, all pixels in image I are normalized together. Meanwhile, I is a full-spatial method. For **manner II**, the pixels from the masked and unmasked regions are normalized separately. Method II is a region-wise approach. In the image inpainting task, the basic idea is for the unknown region to have a similar distribution as the known region, indicating that the expectation and variance of the inpainted result should preferably approach the expectation and variance of the unmasked (known) area. When we normalize all regions without distinction (manner I), the expectation will gradually shift toward 255 (the expectation of the masked region), and the variance also increases. It is different from the distribution of the known area. Although we can obtain the same distribution using method II, the same problem occurs with feature maps while normalizing in method I. Therefore, method II is necessary for reducing the ICS and generating more accurate contents.

3) Formulation of Regional Composite Normalization: In our method, we devise the RCN module by integrating three normalization methods with "manner II," which normalizes the feature regionally. The RCN module is displayed in **Figure 4**.

We use F_{in} to represent the corresponding feature map of the input without removing the values of the masked region (i.e., ϕ_{out} from VMC module). Moreover, we use F_m to represent the corresponding feature map of the mask. The value of the masked region is 0, and the value of the unmasked region is 1 in F_m. Therefore, the selective feature map removing values of the masked region $\hat{F}$ is obtained by

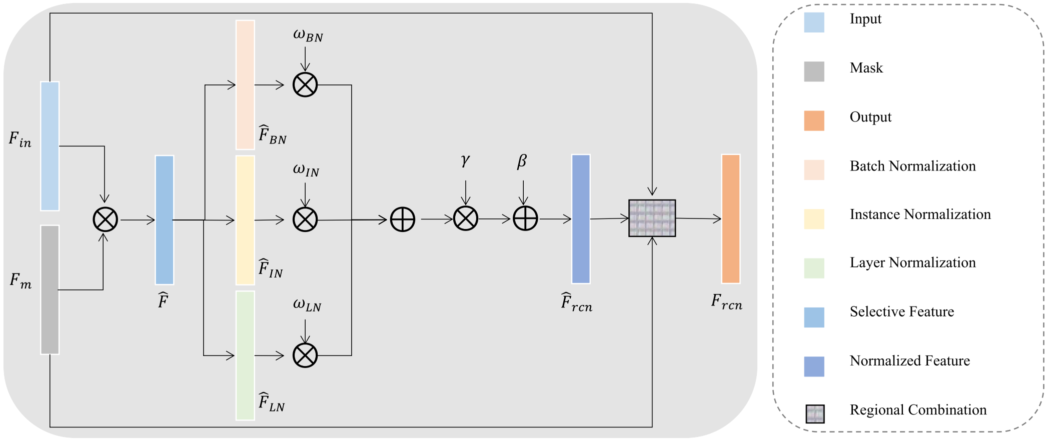


Fig. 4. The framework of **Regional Composite Normalization** (RCN) module. In this module, feature maps are selected based on the location of the mask. Then, the particular features are normalized regionally in three manners, namely batch normalization, instance normalization, and layer normalization. They are composited and normalized dynamically. After the regional combination operation, we obtain the output feature map F_{rcn}.

calculating the Hadamard product of F_{in} and F_m:

$$\hat{F} = F_{in} * F_m \tag{18}$$

We use $x_{n,c,i,j}$, $x'_{n,c,i,j}$, and $x^*_{n,c,i,j}$ to indicate the pixel values at location (n, c, i, j) in feature maps $\hat{F}$, F_m, and F_{in}, respectively. The sizes of $\hat{F}$, F_m, and F_{in} are $N \times C \times H \times W$. Then, RCN is expressed as follows:

$$\hat{x}_{n,c,i,j} = \gamma \frac{x_{n,c,i,j} - \sum_{k\in\rho} w_k^{(e)} \mu_k}{\sqrt{\sum_{k\in\rho} w_k^{(v)} \sigma_k^2 + \epsilon}} + \beta \tag{19}$$

$\hat{x}_{n,c,i,j}$ represents the pixel value of feature map $\hat{F}_{rcn}$. In Equation 19, ϵ is a small number to avoid systematic errors. Moreover, γ is a scale parameter and β is a shift parameter learned along with the original model parameters. Meanwhile, ρ is a set containing different normalization methods, $\rho = \{bn, in, ln\}$. μ_k and σ_k are defined in Equations 22–27.

In feature map $\hat{F}_{rcn}$, the pixels are normalized based on the region. Then, we must restore the removing region of the feature map as follows:

$$x_{rcn} = \hat{x}_{n,c,i,j} * x'_{n,c,i,j} + x^*_{n,c,i,j} * (1 - x'_{n,c,i,j}) \tag{20}$$

x_{rcn} represents the pixel value of the generated feature map (F_{rcn}), which is the well-normalized feature map of the RCN module.

Now, we introduce the detailed operation of RCN. Given that the mask is repeated in number and channel, the ratio of the unmasked region (r_u) can be calculated as follows:

$$r_u = \frac{1}{H \times W} \sum_{i,j}^{H,W} x'_{n,c,i,j} \tag{21}$$

Then, the expectations of $\hat{F}$ with three normalization methods are calculated as follows:

$$\mu_{bn} = \frac{1}{r_u}\left(\frac{1}{N \times H \times W} \sum_{n,i,j}^{N,H,W} x_{n,c,i,j}\right) \tag{22}$$

$$\mu_{in} = \frac{1}{r_u}\left(\frac{1}{H \times W} \sum_{i,j}^{H,W} x_{n,c,i,j}\right) \tag{23}$$

$$\mu_{ln} = \frac{1}{r_u}\left(\frac{1}{C \times H \times W} \sum_{c,i,j}^{C,H,W} x_{n,c,i,j}\right) \tag{24}$$

Besides, the variances of $\hat{F}$ with different normalization methods are calculated as follows:

$$\sigma_{bn}^2 = \frac{1}{r_u}\left[\frac{1}{N \times H \times W} \sum_{n,i,j}^{N,H,W} [(x_{n,c,i,j} - \mu_{bn}) * x'_{n,c,i,j}]^2\right] \tag{25}$$

$$\sigma_{in}^2 = \frac{1}{r_u}\left[\frac{1}{H \times W} \sum_{i,j}^{H,W} [(x_{n,c,i,j} - \mu_{in}) * x'_{n,c,i,j}]^2\right] \tag{26}$$

$$\sigma_{ln}^2 = \frac{1}{r_u}\left[\frac{1}{C \times H \times W} \sum_{c,i,j}^{C,H,W} [(x_{n,c,i,j} - \mu_{ln}) * x'_{n,c,i,j}]^2\right] \tag{27}$$

Meanwhile, $w_k^{(e)}$ and $w_k^{(v)}$ are used to obtain the weighted average of the expectations and variances, respectively. Each $w_k^{(e)}$ or $w_k^{(v)}$ is a scalar variable and learned by back-propagation. Moreover, each $w_k^{(e)}$ or $w_k^{(v)}$ is computed using a softmax function, which is shared across all channels. The relationships are as follows:

$$\sum_{k\in\rho} w_k^{(e)} = 1, \quad \forall w_k^{(e)} \in [0, 1] \tag{28}$$

$$\sum_{k\in\rho} w_k^{(v)} = 1, \quad \forall w_k^{(v)} \in [0, 1] \tag{29}$$

C. Pixel-Wise Attention

The attention mechanism is important in image inpainting [15], [33], [48] because it can help learn related features from the background. In the proposed DSNet, we employ a contextual attention layer to explore the region similarity.

For the irregular region inpainting task, we must adopt a pixel-wise attention module [48] to avoid the misuse of invalid information. We add the pixel-wise contextual attention layer to the third-to-last layer in the decoding stage. The pixel-wise attention layer can borrow useful information on the entire image by measuring the similarity of patches in the background and the foreground. With the help of the pixel-wise attention layer, we can generate realistic results with global semantics and subtle details.

D. Loss Function

The loss function we use is composed of the perceptual, style [49], total variation, hole, and valid losses. In many image inpainting methods, these losses are applied as the objective function [18], [19], [21], [23], [33], [39].

1) Perceptual Loss and Style Loss: Perceptual loss ($L_{perceptual}$) and style loss (L_{style}) are generated by a 16-layer VGG network [50] pretrained on ImageNet [51]. ϕ_{pool_i} is the feature from the i^{th} pooling layer of VGG-16.

By comparing features $\phi^{gt}_{pool_i}$ from ground truth I_{gt} with features $\phi^{pred}_{pool_i}$ from predicted images I_{pred}, perceptual loss is calculated as follows:

$$L_{perceptual} = \mathbb{E}[\sum_{i=1}^{N} \|\phi^{gt}_{pool_i} - \phi^{pred}_{pool_i}\|_1]. \tag{30}$$

The style loss is calculated as follows:

$$L_{style} = \mathbb{E}[\sum_{i=1}^{N} \|G(\phi^{gt}_{pool_i}) - G(\phi^{pred}_{pool_i})\|_1], \tag{31}$$

where G means the Gram matrix on the feature maps, $G(\phi) = \phi^T \phi$.

Perceptual and style losses utilize the features from VGG-16 to learn high-level structure and overall style information.

2) Total Variation Loss: The total variation loss (L_{tv}) is applied to eliminate the checkerboard artifacts problem.

$$L_{tv} = \mathbb{E}[\sum_{x,y}(\|m_{x,y} - m_{x+1,y}\|_1 + \|m_{x,y} - m_{x,y+1}\|_1)], \tag{32}$$

where $m_{x,y}$ means pixel value at location (x, y) of damaged regions.

3) Hole Loss and Valid Loss: Hole loss (L_{hole}) and valid loss (L_{valid}) are calculated as follows:

$$L_{hole} = \mathbb{E}[\sum_{i,j,k} \|p^{gt\_hole}_{i,j,k} - p^{pred\_hole}_{i,j,k}\|_1], \tag{33}$$

$$L_{valid} = \mathbb{E}[\sum_{i,j,k} \|p^{gt\_valid}_{i,j,k} - p^{pred\_valid}_{i,j,k}\|_1], \tag{34}$$

where i, j, k represent the height, weight, and channel of the generated image/ground truth, respectively. For L_{hole}, we calculate the L1 distance of the mask regions in I_{gt} and I_{pred}. For L_{valid}, we calculate the L1 distance of the background regions in I_{gt} and I_{pred}.

4) Overall Loss: In summary, the overall loss function of the proposed algorithm is defined as follows:

$$\begin{aligned} L_{overall} = {} & \lambda_{perceptual} L_{perceptual} + \lambda_{style} L_{style} \\ & + \lambda_{tv} L_{tv} + \lambda_{hole} L_{hole} + \lambda_{valid} L_{valid} \end{aligned} \tag{35}$$

IV. Experiments

A. Datasets

In image inpainting algorithms, three public datasets are commonly used in experiments, namely, the Paris StreetView [52], CelebA [53], and Places2 [54] datasets. Besides, an external mask dataset from [21] is augmented to simulate missing regions. These datasets are used with the proposed and comparison methods.

1) Paris StreetView: The Paris StreetView dataset collects information from Google StreetView. The Paris StreetView dataset contains sufficient structure information, such as windows, doors, and some Paris-style buildings. The Paris StreetView dataset is composed of 15,000 images. We use 14,900 images for training and 100 images for testing. We reuse test images ten times for testing (1000 different mask images) to produce more objective and accurate results.

2) CelebA: CelebA is the CelebFaces attributes dataset, which contains more than 200,000 celebrity facial images. CelebA includes 162,770 training images, 19,867 validation images, and 19,962 testing images. We combine the training and validation images for the training process and use 2000 testing images for testing.

3) Places: Places refers to the Places365-Standard dataset, which contains 1,803,460 training images from 365 scene categories. We use the first 2000 images in the validation dataset for testing.

4) Masks: In the image inpainting task, we must identify the location of missing regions. Therefore, an external mask dataset is often employed to imitate the location of holes. In our experiments, we augment the irregular masks dataset from PConv [21] as the training mask dataset by introducing four rotations (i.e., 0°, 90°, 180°, 270°). The testing mask dataset is the original testing mask dataset from PConv, divided into six groups according to the ratio of the mask regions.

B. Comparison Methods

We compare the proposed method to four state-of-the-art image inpainting models.

1) PConv [21]: The PConv method proposes a particular partial convolution layer to solve the convolution problem in the inpainting task. PConv is one of the earliest trainable models that can generate reasonable results for irregular regions.

2) PRVS [19]: The progressive reconstruction of visual structure (PRVS) method devises a PRVS network, which employs edge information as an extra constraint. In particular, the visual structures are recovered progressively to ensure accuracy.

3) EC [18]: The EC method refers to the EdgeConnect model, which has two stages: the edge generator and the image completion network. EC also employs image structure information to restore holes.

4) RN [23]: The RN method devises two kinds of region normalization methods, namely, RN-B and RN-L, to avoid the mean and variance shifts in image inpainting tasks.

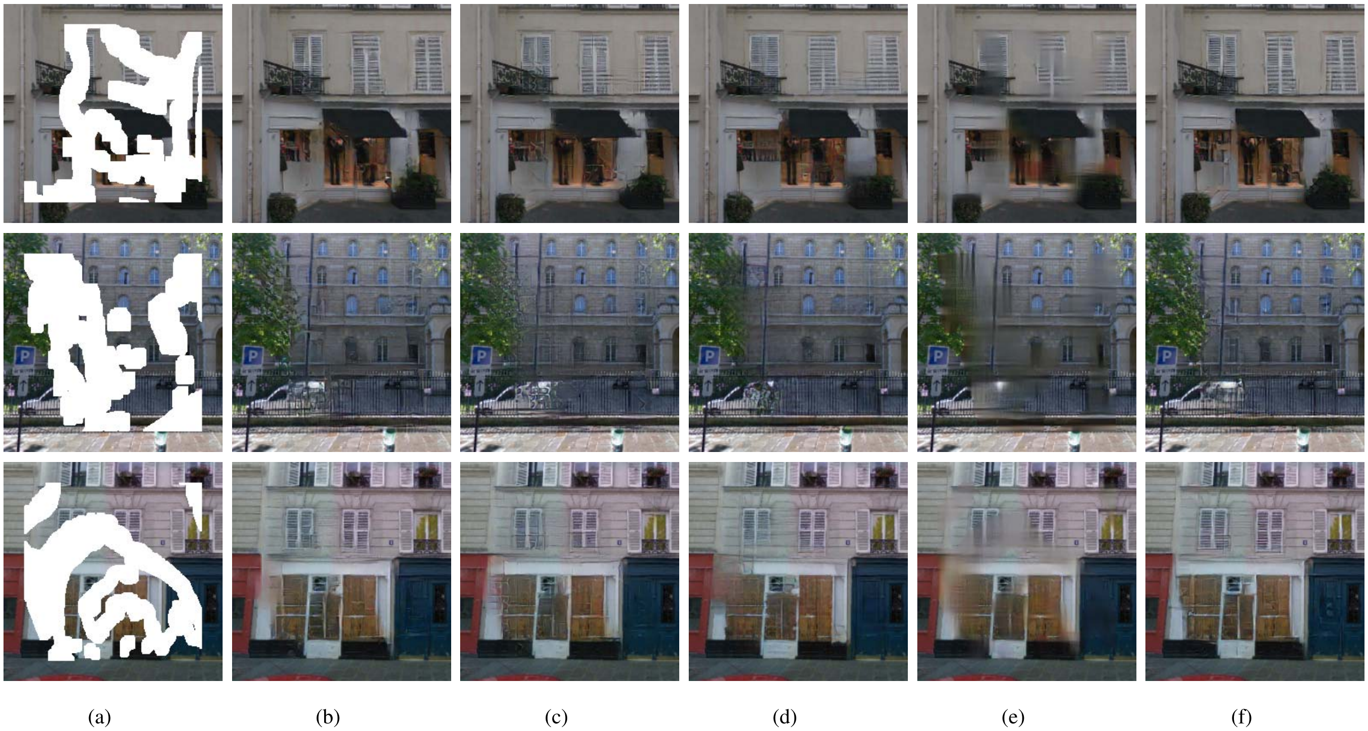


Fig. 5. Qualitative comparisons on Paris StreetView dataset. From left to right are: (a) Masked image, (b) PConv, (c) PRVS, (d) EC, (e) RN, (f) Ours.

C. Training Settings

All models are trained with Python on Ubuntu 17.10 system, i7-6800K 3.40 GHz CPU, and 12G NVIDIA Titan Xp GPU. The batch size in our experiments is six. The size of the training image is 256 × 256. For the CelebA dataset, the raw images have a size of 178 × 218. To preserve the faces region and satisfy the size requirement, we resize the minimum width or length to 256 and crop a 256 × 256 sub-image at the center as the input image. The images from the Paris StreetView and Places2 datasets are square, thus resizing is the only operation before training. For the hyper-parameters, we adopt 0.05 for $\lambda_{perceptual}$, 120 for λ_{style}, 0.1 for λ_{tv}, 6 for λ_{hole}, and 1 for λ_{valid}. These values are empirically based on the experimental observations in our model.

V. Results

The proposed model is qualitatively and quantitatively compared with comparison methods in three datasets. Ablation studies are conducted on the Paris StreetView dataset, and the weights of different normalization methods in the RCN module are analyzed.

A. Qualitative Analysis

Figures 5, 6, and 7 present the qualitative comparison results on Paris StreetView, CelebA, and Places2, respectively. As shown in Figure 5, the PConv, PRVS, EC, RN, and the proposed DSNet can generate approximate contents for masked regions. However, there exist many artifacts and distorted structures in the comparison methods PConv, PRVS, EC, and RN. For example, in the second case of Figure 5(b), the PConv can restore the brick wall's texture, but the windows at the boundary of masked regions are not recovered well. The results of PConv usually contain basic textures but distorted structures. The PRVS (Figure 5(c)) and EC (Figure 5(d)) methods utilize the structure information in inpainting process, so these two methods can generate more reasonable structures than PConv. However, EC is a little weak in deducing plausible context from surrounding information. For instance, the EC cannot generate windows for large missing regions in the second case. The PRVS tends to generate checkboard-like textures while in large corrupted areas, like the tree and car in the second row. As for the RN method (Figure 5(e)), it can recover the overall style of the image but with blurry details. More specifically, the RN method can infer the plausible color and structure of the damaged regions, while it cannot generate detailed textures. The proposed DSNet (figure 5(f)) can generate reasonable structures and exquisite details for missing regions. Compared with other methods, the proposed DSNet significantly reduces the appearance of checkerboard artifacts and distorted structures.

The same performance can be observed on the CelebA and Places2 datasets. For the CelebA dataset (Figure 6), the inpainted faces of the DSNet have more regular facial features and better hair than those of the other models. For the Places dataset (Figure 7), the reconstructed scenes are closer to reality. According to the qualitative results, we can find that the proposed DSNet model outperforms state-of-the-art models with a consistent style and excellent details.

B. Quantitative Analysis

We also compare our model with comparison methods quantitatively. In this study, the following two kinds of metrics are employed for comparison: (1) the pixel-level metrics

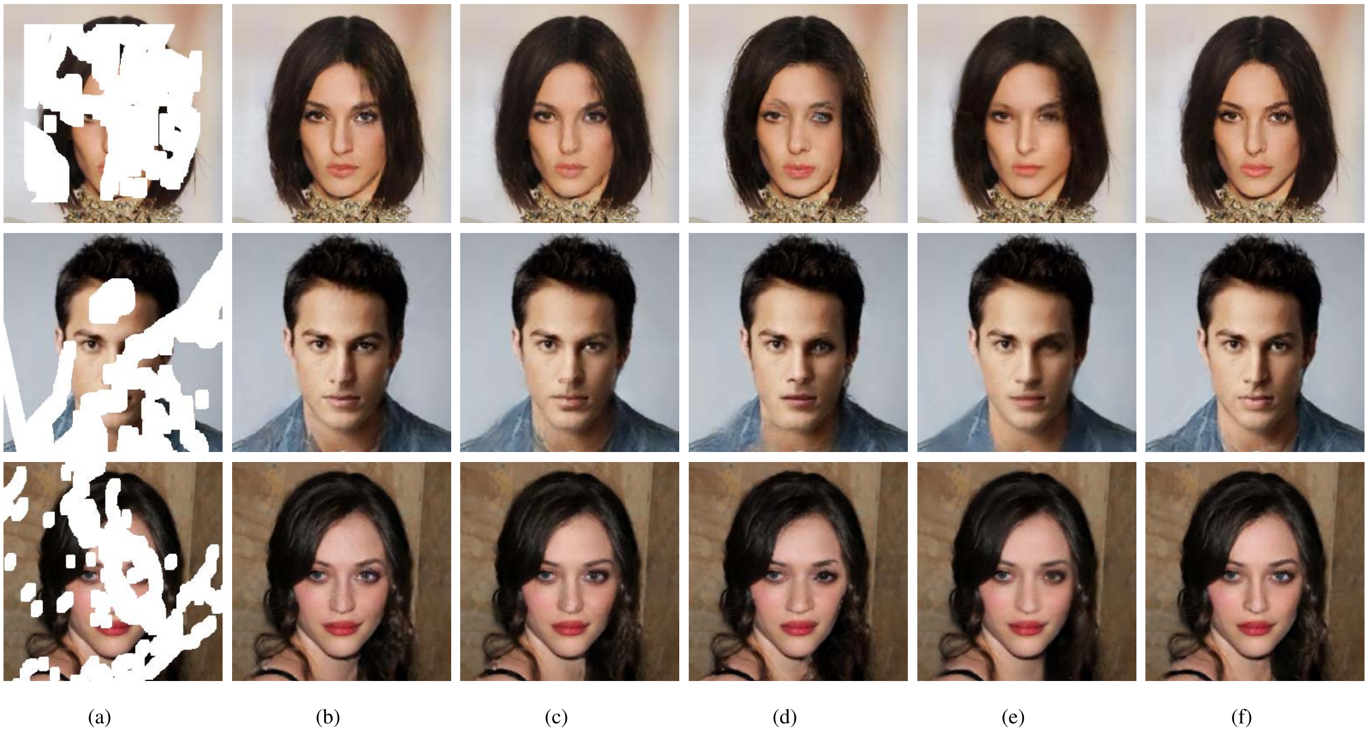


Fig. 6. Qualitative comparisons on CelebA dataset. From left to right are: (a) Masked Image, (b) PConv, (c) PRVS, (d) EC, (e) RN, (f) Ours.

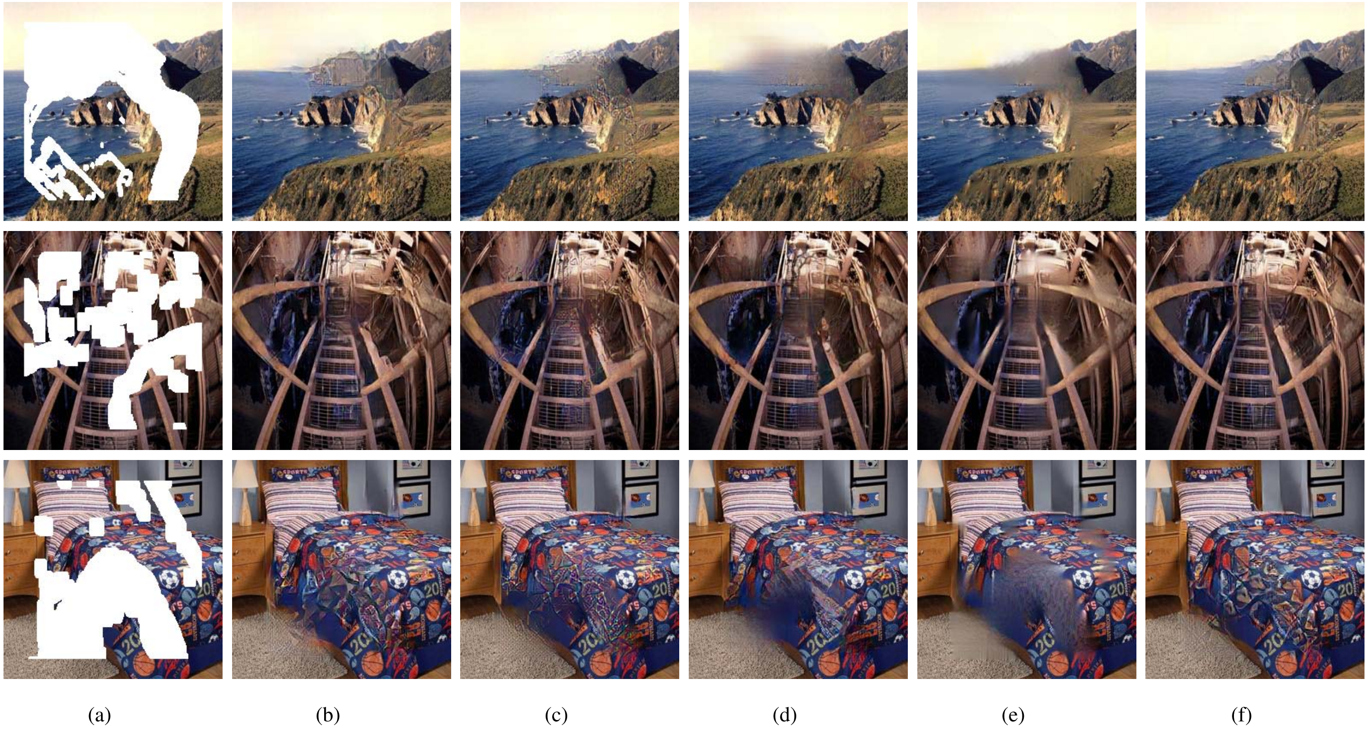


Fig. 7. Qualitative comparisons on Places2 dataset. From left to right are: (a) Masked Image, (b) PConv, (c) PRVS, (d) EC, (e) RN, (f) Ours.

(including multi-scale structural similarity (MS-SSIM) and mean absolute error (MAE)), and (2) the human perception-level metrics (including Fréchet inception distance (FID) [55] and learned perceptual image patch similarity (LPIPS) [56]). Pixel-level metrics (like the PSNR, SSIM, MAE, and MSE) are widely accepted for quantitative comparison in some classical literature. However, in the image inpainting task, the output images with good performance on pixel-level metrics may reportedly lead to blurry (bad visual performance) patches [18]. A similar phenomenon has also been

TABLE II

NUMERICAL COMPARISON ON PARIS STREETVIEW DATASET. BEST RESULTS IN EACH GROUP ARE HIGHLIGHTED IN **BOLD**

Paris StreetView						
Metric	Mask	PConv	PRVS	EC	RN	Ours
FID	(0.0,0.1]	0.046	0.059	0.067	0.248	**0.035**
	(0.1,0.2]	0.293	0.382	0.398	1.345	**0.203**
	(0.2,0.3]	0.845	1.141	1.416	4.306	**0.547**
	(0.3,0.4]	2.056	2.999	3.855	10.389	**1.325**
	(0.4,0.5]	3.838	5.513	7.565	19.284	**2.627**
	(0.5,0.6]	9.281	13.244	17.763	38.712	**6.292**
LPIPS	(0.0,0.1]	0.0171	0.0183	0.0176	0.0321	**0.0159**
	(0.1,0.2]	0.0457	0.0493	0.0469	0.0791	**0.0425**
	(0.2,0.3]	0.0804	0.0872	0.0845	0.1326	**0.0752**
	(0.3,0.4]	0.1189	0.1295	0.1290	0.1897	**0.1117**
	(0.4,0.5]	0.1644	0.1789	0.1813	0.2511	**0.1535**
	(0.5,0.6]	0.2369	0.2576	0.2694	0.3473	**0.2200**
MS-SSIM	(0.0,0.1]	0.9843	0.9842	0.9822	0.9774	**0.9852**
	(0.1,0.2]	0.9518	0.9524	0.9473	0.9401	**0.9548**
	(0.2,0.3]	0.9035	0.9053	0.8943	0.8885	**0.9092**
	(0.3,0.4]	0.8393	0.8445	0.8253	0.8233	**0.8492**
	(0.4,0.5]	0.7605	0.7691	0.7424	0.7420	**0.7747**
	(0.5,0.6]	0.5947	0.6156	0.5660	0.5690	**0.6219**
MAE (%)	(0.0,0.1]	0.314	0.311	0.323	0.373	**0.305**
	(0.1,0.2]	0.839	0.820	0.847	0.926	**0.814**
	(0.2,0.3]	1.492	**1.445**	1.506	1.581	1.451
	(0.3,0.4]	2.246	**2.149**	2.267	2.318	2.179
	(0.4,0.5]	3.109	**2.957**	3.131	3.181	3.015
	(0.5,0.6]	4.597	**4.324**	4.645	4.682	4.425

TABLE III

NUMERICAL COMPARISON ON CELEBA DATASET. BEST RESULTS IN EACH GROUP ARE HIGHLIGHTED IN **BOLD**

CelebA						
Metric	Mask	PConv	PRVS	EC	RN	Ours
FID	(0.0,0.1]	0.012	0.015	0.024	0.039	**0.009**
	(0.1,0.2]	0.061	0.085	0.129	0.157	**0.049**
	(0.2,0.3]	0.192	0.267	0.445	0.463	**0.150**
	(0.3,0.4]	0.416	0.644	1.125	1.043	**0.362**
	(0.4,0.5]	0.774	1.286	2.333	1.867	**0.660**
	(0.5,0.6]	1.595	2.931	6.613	3.973	**1.328**
LPIPS	(0.0,0.1]	0.0095	0.0096	0.0118	0.0207	**0.0086**
	(0.1,0.2]	0.0252	0.0259	0.0303	0.0436	**0.0232**
	(0.2,0.3]	0.0451	0.0466	0.0546	0.0705	**0.0415**
	(0.3,0.4]	0.0672	0.0700	0.0842	0.0994	**0.0622**
	(0.4,0.5]	0.0928	0.0975	0.1204	0.1331	**0.0863**
	(0.5,0.6]	0.1361	0.1440	0.1970	0.1897	**0.1266**
MS-SSIM	(0.0,0.1]	0.9914	0.9919	0.9887	0.9875	**0.9920**
	(0.1,0.2]	0.9739	0.9755	0.9674	0.9702	**0.9757**
	(0.2,0.3]	0.9482	0.9513	0.9344	0.9457	**0.9519**
	(0.3,0.4]	0.9149	0.9198	0.8900	0.9125	**0.9209**
	(0.4,0.5]	0.8736	0.8805	0.8316	0.8691	**0.8821**
	(0.5,0.6]	0.7871	0.7956	0.6852	0.7674	**0.8012**
MAE (%)	(0.0,0.1]	0.236	**0.223**	0.277	0.297	0.226
	(0.1,0.2]	0.639	**0.604**	0.732	0.709	0.614
	(0.2,0.3]	1.157	**1.093**	1.329	1.217	1.113
	(0.3,0.4]	1.768	**1.675**	2.049	1.832	1.699
	(0.4,0.5]	2.490	**2.366**	2.923	2.609	2.397
	(0.5,0.6]	3.855	3.704	4.827	4.115	**3.690**

TABLE IV

NUMERICAL COMPARISON ON PLACES DATASET. BEST RESULTS IN EACH GROUP ARE HIGHLIGHTED IN **BOLD**

Places						
Metric	Mask	PConv	PRVS	EC	RN	Ours
FID	(0.0,0.1]	0.038	0.037	0.051	0.069	**0.027**
	(0.1,0.2]	0.291	0.264	0.340	0.438	**0.181**
	(0.2,0.3]	0.935	0.965	1.190	1.449	**0.650**
	(0.3,0.4]	2.100	2.181	2.662	3.256	**1.531**
	(0.4,0.5]	4.038	4.365	5.411	6.639	**2.869**
	(0.5,0.6]	8.435	9.071	10.956	13.594	**6.096**
LPIPS	(0.0,0.1]	0.0207	0.0209	0.0218	0.0296	**0.0186**
	(0.1,0.2]	0.0549	0.0577	0.0577	0.0733	**0.0499**
	(0.2,0.3]	0.0962	0.1440	0.1026	0.1244	**0.0883**
	(0.3,0.4]	0.1386	0.1532	0.1504	0.1758	**0.1284**
	(0.4,0.5]	0.1883	0.2117	0.2073	0.2333	**0.1754**
	(0.5,0.6]	0.2629	0.3000	0.2983	0.3200	**0.2462**
MS-SSIM	(0.0,0.1]	0.9805	0.9808	0.9769	0.9811	**0.9817**
	(0.1,0.2]	0.9407	0.9413	0.9311	**0.9457**	0.9434
	(0.2,0.3]	0.8843	0.8856	0.8678	**0.8967**	0.8890
	(0.3,0.4]	0.8186	0.8210	0.7961	**0.8382**	0.8248
	(0.4,0.5]	0.7358	0.7398	0.7104	**0.7612**	0.7438
	(0.5,0.6]	0.5831	0.5926	0.5605	**0.6121**	0.5940
MAE (%)	(0.0,0.1]	0.470	0.453	0.498	**0.431**	0.454
	(0.1,0.2]	1.233	1.186	1.293	**1.093**	1.198
	(0.2,0.3]	2.188	2.104	2.282	**1.916**	2.132
	(0.3,0.4]	3.196	3.061	3.315	**2.796**	3.122
	(0.4,0.5]	4.377	4.189	4.502	**3.865**	4.288
	(0.5,0.6]	6.206	5.919	6.339	**5.621**	6.093

observed in other image restoration tasks, including super-resolution [57]. Furthermore, pixel-level metrics are not suitable for all image inpainting tasks, such as the objects removal task, which should generate a reasonable background for removed regions rather than the original objects. For a comprehensive evaluation, FID and LPIPS demonstrate that deep visual representations are useful in imitating human perceptual judgments. These two metrics both employ information from the high-level features of generated and ground-truth images to calculate the distance and have been introduced in state-of-the-art image inpainting works to measure the perceptual quality [18], [58], [59].

Evaluations involve testing with six mask groups, which are classified based on mask ratios (0.0, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5], and (0.5, 0.6]. The quantitative results on Paris StreetView are shown in Table II. Our method performs best in the perception-level metrics FID and LPIPS, consistent with the qualitative results. As for pixel-level metrics, the evaluation results on MS-SSIM are also the best among all methods and the performance of MAE is near the best MAE performance of the PRVS method on every mask group, which has the best evaluation results on the MAE metric. In the Paris StreetView dataset, the PConv method ranks second on the perception-level metrics and third on the pixel-level metrics. The PRVS ranks second in terms of the pixel-level metrics and third in terms of the perception-level metrics. The EC has a comparable performance of PRVS in terms of LPIPS metric in the Paris StreetView dataset.

Table III lists the quantitative results of the CelebA dataset. The proposed DSNet also obtains the best evaluation results of the FID, LPIPS, and MS-SSIM metrics in the CelebA dataset. The MAE of DSNet still ranks second in CelebA. The performance of other methods has a similar distribution to the Paris StreetView dataset. Table IV presents the quantitative results of the Places2 dataset. At the perception level, the proposed DSNet has the first rank as usual. The RN has the best performance at the pixel level (MS-SSIM and MAE), but its high performance is acquired by generating blurring contents (according to the images in Fig. 7).

Tables II, III, and IV indicate that the proposed DSNet is superior to other competitors on all four models from the

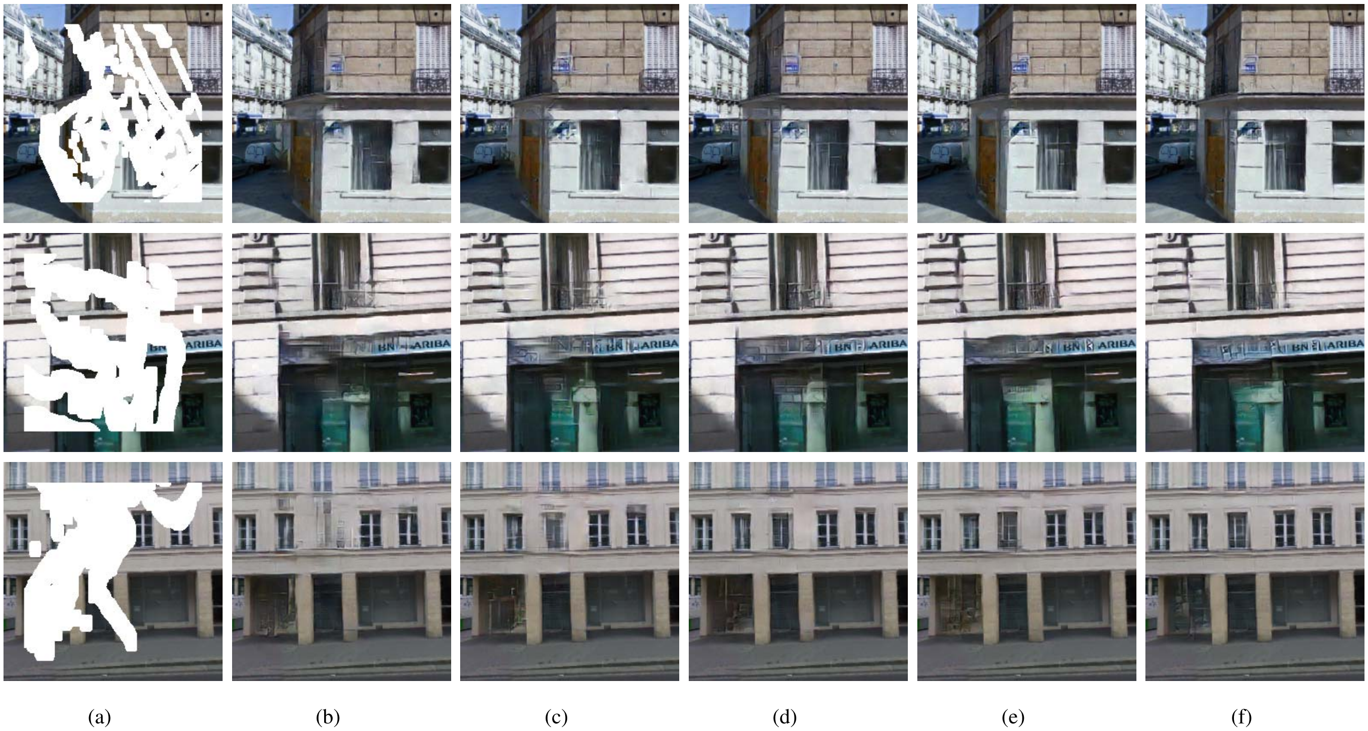


Fig. 8. Qualitative results of different experimental settings. From left to right are: (a) Masked Image, (b) Experiment I, (c) Experiment II, (d) Experiment III, (e) Experiment IV, (f) DSNet (Ours).

TABLE V

THE ANALYSIS OF RCN MODULE AND VMC MODULE ON PARIS STREETVIEW DATASET. EXPERIMENT I IS VMC + BN, EXPERIMENT II IS VMC + RBN, EXPERIMENT III IS PCONV + RCN, AND EXPERIMENT IV IS VMC + RCN. THE DETAILED SETTINGS OF EXPERIMENTS I - IV ARE INTRODUCED IN SECTION V-C

Metric	Mask	I	II	III	IV	DSNet
FID	(0.0,0.1]	0.167	0.060	0.054	0.039	**0.035**
	(0.1,0.2]	0.824	0.383	0.326	0.232	**0.203**
	(0.2,0.3]	2.574	1.226	0.982	0.658	**0.547**
	(0.3,0.4]	4.954	2.754	2.233	1.418	**1.325**
	(0.4,0.5]	9.024	5.736	4.569	2.932	**2.627**
	(0.5,0.6]	18.968	13.999	10.183	6.816	**6.292**
LPIPS	(0.0,0.1]	0.0278	0.0176	0.0173	0.0160	**0.0159**
	(0.1,0.2]	0.0643	0.0472	0.0461	0.0432	**0.0425**
	(0.2,0.3]	0.1072	0.0840	0.0815	0.0766	**0.0752**
	(0.3,0.4]	0.1491	0.1253	0.1211	0.1144	**0.1117**
	(0.4,0.5]	0.1997	0.1737	0.1667	0.1587	**0.1535**
	(0.5,0.6]	0.2713	0.2512	0.2378	0.2288	**0.2200**
MAE (%)	(0.0,0.1]	0.383	0.320	0.315	0.306	**0.305**
	(0.1,0.2]	0.981	0.854	0.840	0.820	**0.814**
	(0.2,0.3]	1.715	1.529	1.495	1.462	**1.451**
	(0.3,0.4]	2.497	2.301	2.251	2.198	**2.179**
	(0.4,0.5]	3.419	3.190	3.114	3.047	**3.015**
	(0.5,0.6]	4.964	4.723	4.584	4.486	**4.425**

perception level and obtains a comparable performance at the pixel level. They demonstrate the superiority of the proposed model.

C. Ablation Analysis

Here, we conduct some experiments to explore the effectiveness of the RCN and VMC modules in the DSNet (VMC + RCN + Attention). The attention module is removed in Experiments I–IV. The settings of the four experiments are as follows:

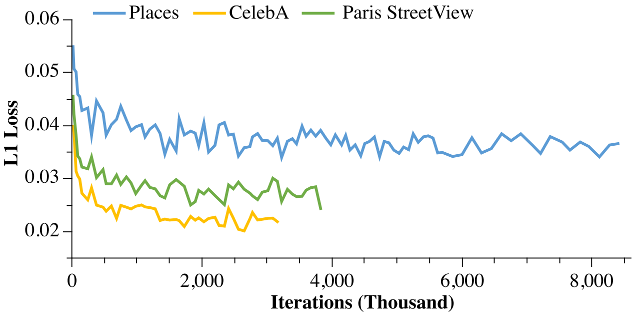


Fig. 9. Training loss (L1 Loss) on three datasets.

Experiment I (VMC + BN): We replace the RCN module with the general BN method.

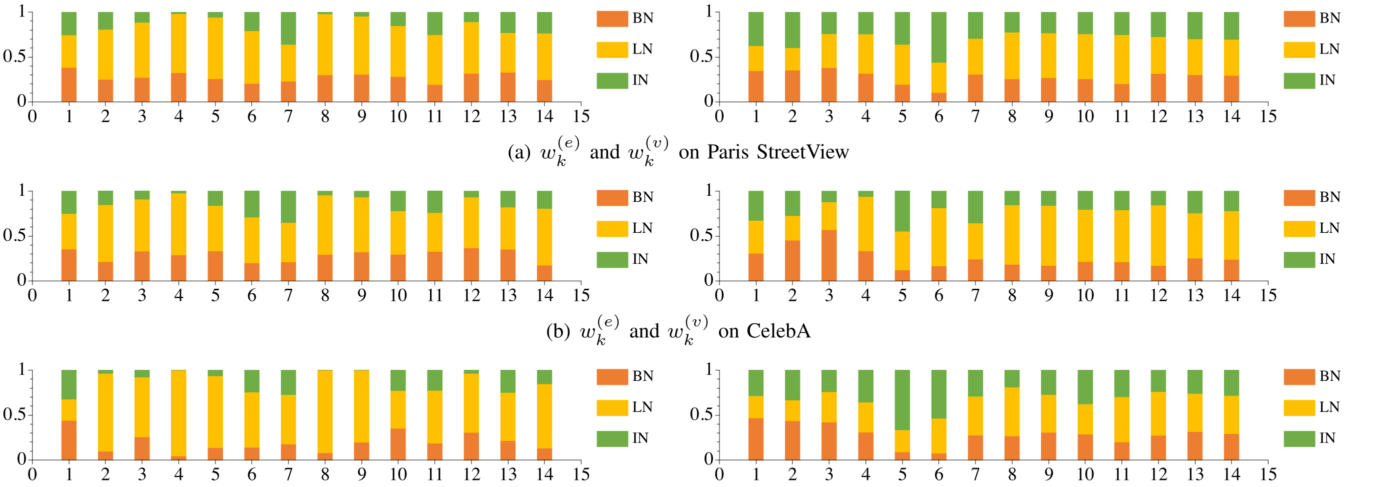
Experiment II (VMC + RBN): We replace the RCN module with the regional BN (RBN) method.

Experiment III (PConv + RCN): We replace the VMC module with the PConv module [21].

Experiment IV (VMC + RCN): We remove the attention module in the DSNet.

The quantitative results of Experiments I–IV are displayed in Table V, and the corresponding qualitative results are shown in Figure 8.

1) Effect of RCN: The proposed RCN module described in Section III-B employs three normalization methods regionally. To verify the effectiveness of RCN, we replace the RCN module (Experiment IV) in the attention-deleted DSNet with BN module (Experiment I) and RBN module (Experiment II), respectively. According to the quantitative and qualitative results shown in Table V and Figure 8, Experiment II generates better results than Experiment I, demonstrating the influence of



(a) $w_k^{(e)}$ and $w_k^{(v)}$ on Paris StreetView

(b) $w_k^{(e)}$ and $w_k^{(v)}$ on CelebA

(c) $w_k^{(e)}$ and $w_k^{(v)}$ on Places

Fig. 10. Important weights of μ_k and σ_k on three datasets. From top to bottom are $w_k^{(e)}$ and $w_k^{(v)}$ on Paris StreetView, CelebA, and Places datasets respectively. The left part is $w_k^{(e)}$, and the right part is $w_k^{(v)}$.

the region-wise operation in the normalization stage. It verifies that the problem in the normalization phase must be solved for image inpainting tasks. Moreover, Experiment IV has the best performance among Experiment I to IV, indicating that the proposed RCN module is very suitable and useful for solving the normalization problem.

2) Effect of VMC: The proposed VMC module is described in Section III-A and adopts modified deformable convolution to solve the convolution problem. According to the results of Experiment III and Experiment IV in Table V and Figure 8, we can find that Experiment IV has better performance than Experiment III. This phonomenon shows that the proposed VMC module is more suitable than the PConv module for image inpainting tasks.

3) Training Loss: Figure 9 displays the changes in training loss (L1 Loss) during the iteration on three datasets. Among these three datasets, CelebA has the fastest descent speed and can generate good results at approximately 3.12 million iterations. The Paris StreetView stops at approximately 3.84 million iterations. Places is a large dataset and requires approximately 8.4 million iterations.

D. Important Weights in RCN Module

Figure 10 presents the weights in the RCN module. $w_k^{(e)}$ is the weight of expectation μ_k, and $w_k^{(v)}$ is the weight of variance σ_k, where $k \in \rho, \rho = \{bn, ln, in\ \}$. The y-axis represents the importance weights that add up to 1, while the x-axis shows different layers of the DSNet. In the proposed DSNet, the encoding stage contains seven RCN modules, as does the decoding stage. Meanwhile, 1∼7 corresponds to the RCN modules in the encoding stage, 8∼14 corresponds to the decoding stage. The weights are distinct in different datasets, indicating that different datasets should choose different normalization methods even if the models and tasks are the same. Furthermore, the diverse distributions of weights in different layers demonstrate that different normalization methods should be combined dynamically in deep networks to adapt to different scenarios. The various distributions of these weights verify the necessity of different normalization methods in image inpainting tasks. Therefore, the proposed RCN module is also proved to be necessary and useful.

TABLE VI

COMPARISON OF RUNNING TIME

Time (ms)	PConv	PRVS	EC	RN	Ours
Training	32.75	116.7	56.5+84	55.6	139
Testing	12	62.5	27	57	67.4

E. Running Time

The running time contains the training time and testing time. The training time refers to the time consumption of one image with the resolution 256 × 256 during the training process, and the testing time refers to the time of restoring a damaged image. According to Table VI, we can observe that the PConv method has the fastest speed in both the training and testing process. The RN method is fast in training while it doesn't perform well in testing. The EC method needs the longest time for training due to its two-stage architecture, which should be trained separately. But the EC method is fast when testing. The PRVS method and the proposed DSNet method have a similar speed at training and testing. Although these methods have different speeds, they can satisfy the requirements of real-time inpainting. What is more, we design a novel VMC for flexibly information learning in DSNet, whose code is implemented by Pytorch. In contrast, the convolution stage of other models is usually implemented by the more low-level programming language, which is faster than Pytorch. Thus, the proposed DSNet has the potential to realize the inpaint process with less time consumption by optimizing the code.

VI. Discussions and Future Work

This paper proposes a novel DSNet for image inpainting. The proposed DSNet contains two important modules, namely, the VMC and RCN modules, which are based on the dynamic selection mechanism and region-wise approach. The VMC module is devised to solve the problem in the convolution stage. The VMC module dynamically selects sampling locations based on the information in feature maps for flexible learning. The RCN module is designed for the normalization stage. The RCN module can adaptively learn weights to utilize BN, IN, and LN simultaneously. The promising performance of the DSNet image inpainting model is demonstrated on some publicly available datasets. The proposed DSNet outperforms all state-of-the-art comparison methods.

Although the proposed DSNet is efficient in image inpainting, it still has weaknesses. Firstly, the proposed method designs new convolution and normalization operations without optimization. Thus, the processing time can be reduced with nice optimization. Secondly, the proposed method cannot recover the original information for largely damaged regions. The proposed method can provide help for image editing tasks like face editing. Still, it cannot contribute to specific tasks that need to recover the right image, like face restoration of suspects.

We believe that dynamically utilizing information in the convolution and normalization phases is significant to future work involving image inpainting or editing. We hope this work paves the way for further development of inpainting algorithms.

References

[1] Y. Jo and J. Park, "SC-FEGAN: Face editing generative adversarial network with User's sketch and color," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1745–1753.

[2] Q. Sun, L. Ma, S. Joon Oh, L. V. Gool, B. Schiele, and M. Fritz, "Natural and effective obfuscation by head inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5050–5059.

[3] R. Uittenbogaard, C. Sebastian, J. Vijverberg, B. Boom, D. M. Gavrila, and P. H. N. de With, "Privacy protection in street-view panoramas using depth and multi-view imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10581–10590.

[4] M. Lan, Y. Zhang, L. Zhang, and B. Du, "Global context based automatic road segmentation via dilated convolutional neural network," *Inf. Sci.*, vol. 535, pp. 156–171, Oct. 2020.

[5] Levin, Zomet, and Weiss, "Learning how to inpaint from global image statistics," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 305–312.

[6] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, Aug. 2001.

[7] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. SIGGRAPH*, 2000, pp. 417–424.

[8] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang, "Image compression with edge-based inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1273–1287, Oct. 2007.

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

[10] D. Ding, S. Ram, and J. J. Rodriguez, "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1705–1719, Apr. 2019.

[11] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *Proc. ACM SIGGRAPH Papers SIGGRAPH*, 2003, pp. 303–312.

[12] 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 Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[13] 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 Recognit. (CVPR)*, Jul. 2017, pp. 6882–6890.

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

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

[16] W. Xiong *et al.*, "Foreground-aware image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5840–5848.

[17] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6721–6729.

[18] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "EdgeConnect: Structure guided image inpainting using edge prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3265–3274.

[19] J. Li, F. He, L. Zhang, B. Du, and D. Tao, "Progressive reconstruction of visual structure for image inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5962–5971.

[20] L. Song *et al.*, "Unsupervised domain adaptive re-identification: Theory and practice," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107173.

[21] G. 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. (ECCV)*, Sep. 2018, pp. 85–100.

[22] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4471–4480.

[23] T. Yu *et al.*, "Region normalization for image inpainting," in *Proc. Assoc. Advan. Artif. Intell. (AAAI)*, 2020, pp. 12733–12740.

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

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention MICCAI*, 2015, pp. 234–241.

[26] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[27] P. Luo, R. Zhang, J. Ren, Z. Peng, and J. Li, "Switchable normalization for learning-to-normalize deep representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. Aug. 30, 2019, doi: 10.1109/TPAMI.2019.2932062.

[28] 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, Aug. 2003.

[29] D. Ding, S. Ram, and J. J. Rodriguez, "Perceptually aware image inpainting," *Pattern Recognit.*, vol. 83, pp. 174–184, Nov. 2018.

[30] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, "Shift-Net: Image inpainting via deep feature rearrangement," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[31] L. Shen, R. Hong, H. Zhang, H. Zhang, and M. Wang, "Single-shot semantic image inpainting with densely connected generative networks," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1861–1869.

[32] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "StructureFlow: Image inpainting via structure-aware appearance flow," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 181–190.

[33] N. Wang, J. Li, L. Zhang, and B. Du, "MUSICAL: Multi-scale image contextual attention learning for inpainting," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3748–3754.

[34] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1486–1494.

[35] C. Xie *et al.*, "Image inpainting with learnable bidirectional attention maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8857–8866.

[36] H. Liu, B. Jiang, Y. Xiao, and C. Yang, "Coherent semantic attention for image inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4169–4178.

[37] J. Zhang *et al.*, "GAIN: Gradient augmented inpainting network for irregular holes," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1870–1878.

[38] H. Zhang, Z. Hu, C. Luo, W. Zuo, and M. Wang, "Semantic image inpainting with progressive generative networks," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1939–1947.

[39] Z. Guo, Z. Chen, T. Yu, J. Chen, and S. Liu, "Progressive image inpainting with full-resolution residual network," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2496–2504.

[40] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7757–7765.

[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[42] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: http://arxiv.org/abs/1607.08022

[43] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450

[44] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[45] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 901–909.

[46] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 2558–2567.

[47] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 2017–2025.

[48] N. Wang, S. Ma, J. Li, Y. Zhang, and L. Zhang, "Multistage attention network for image inpainting," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107448.

[49] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 694–711.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[52] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–9, Aug. 2012.

[53] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[54] B. 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, Jun. 2018.

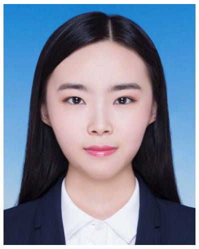
[55] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 6626–6637.

[56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[57] M. S. Rad, B. Bozorgtabar, U.-V. Marti, M. Basler, H. K. Ekenel, and J.-P. Thiran, "SROBB: Targeted perceptual loss for single image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2710–2719.

[58] J. Gu, Y. Shen, and B. Zhou, "Image processing using multi-code GAN prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3012–3021.

[59] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7508–7517.



Ning Wang (Student Member, IEEE) received the B.S. degree in information security from the School of Cyber Science and Engineering, Wuhan University, China, in 2019. She is currently pursuing the M.S. degree with the School of Computer Science, Wuhan University. Her research interests include deep learning and computer vision.

Yipeng Zhang (Student Member, IEEE) received the B.S. degree from the School of Electronic Information, Wuhan University, in 2014, and the master's degree from Syracuse University in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University. His research interests include machine learning, AI SoC, and computer-aided design.

Lefei Zhang (Senior Member, IEEE) received the B.S. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

He was a Big Data Institute Visitor with the Department of Statistical Science, University College London, in 2016, and a Hong Kong Scholar with the Department of Computing, The Hong Kong Polytechnic University, in 2017. He is currently a Professor with the School of Computer Science, Wuhan University. His research interests include pattern recognition, image processing, and remote sensing. He serves as an Associate Editor for *Pattern Recognition* and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.