Multistage Attention Network for Image Inpainting

Article *in* Pattern Recognition · May 2020

DOI: 10.1016/j.patcog.2020.107448

Wang Ning
Wuhan University
2 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Jingyuan Li
Wuhan University
3 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Pattern Recognition 106 (2020) 107448

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multistage attention network for image inpainting

Ning Wang[a], Sihan Ma[b], Jingyuan Li[a], Yipeng Zhang[a], Lefei Zhang[a,c,*]

[a] School of Computer Science, Wuhan University, Wuhan, P.R. China
[b] UBTECH Sydney Artificial Intelligence Centre, School of Computer Science, Faculty of Engineering and Information Technologies, The University of Sydney, Darlington, NSW, Australia
[c] State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, P.R. China

ARTICLE INFO

Article history:
Received 22 January 2020
Revised 31 March 2020
Accepted 13 May 2020
Available online 24 May 2020

Keywords:
Image inpainting
Irregular mask
Deep learning
Attention mechanism
Unet-like network

ABSTRACT

Image inpainting refers to the process of restoring the mask regions of damaged images. Existing inpainting algorithms have exhibited outstanding performance on certain inpainting tasks that are focused on recovering small masks or square masks. Tasks that attempt to reconstruct large proportion of damaged images can still be improved. Although many attention-related algorithms have been proposed to solve image inpainting tasks, most of them ignore the requirements to balancing the detail and style level. In this paper, we propose a novel image inpainting method for large-scale irregular masks. We introduce a special multistage attention module that considers structure consistency and detail fineness. The proposed multistage attention module operates in a coarse to-fine manner, where the early stage performs large feature patch swapping and ensures the global consistency in images, and the next stage swaps small patches to refine the texture. Then, we adopt a partial convolution strategy to avoid the misuse of invalid data during convolution. Several losses are combined as the training objective function to generate excellent results with global consistency and exquisite detail. Qualitative and quantitative experiments on the Paris StreetView, CelebA, and Places2 datasets demonstrate the superior performance of the proposed approach compared with state-of-the-art models.

1. Introduction

Image inpainting involves the process of reconstituting damaged regions (or holes) of images in a visually plausible manner [1]. Image inpainting has emerged as a promising technology for various applications, such as recovery of damaged images with scratches, removal of logos, and some other complex applications, including identity obfuscation and eye inpainting [2–4]. This technology provides powerful capabilities for scene understanding and for improving the performance of traditional computer vision problems, such as image editing. Video inpainting tasks [5–7] also have received increasing attention. Deep research on image inpainting has inspired various inpainting algorithms, and these algorithms share the same basic idea. Considering the global consistency of images, the masked areas should have the same style and pattern as the known areas. Thus, inpainting methods usually fill in the missing areas with available information from the background.

Existing methods can be classified into two main categories, namely, traditional [8–16] and learning-based approaches [17–33]. Traditional image inpainting methods consistently fill in damaged regions by propagating several information, such as diffusion coefficients [13] and candidate patches [14,16]. Depending on the contents of propagation, traditional methods can be simply divided into diffusion-based and patch-based methods. Traditional approaches can succeed in restoring some texture details with satisfactory results, but the problem of capturing the global structure remains. These traditional algorithms typically obtain information through mathematical and statistical methods, resulting in the inability to generate novel content. They frequently fail to produce reasonable results when solving the problem of face inpainting.

In recent years, convolutional neural networks (CNNs) have exhibited strong potential in computer vision tasks, especially in image inpainting. The architecture of context encoder (CE) [17] is a simple encoder-decoder pipeline connected by a channel full connection layer. It is the first trainable inpainting algorithm that can provide reasonable contents. Apart from the CE algorithm, some methods [18,19] adopt multiple networks to maintain global and local consistency. In addition to these methods that usually recover the square regions in the center of an image, some learning-based

* Corresponding Author.
E-mail addresses: wang_ning@whu.edu.cn (N. Wang), sima7436@uni.sydney.edu.au (S. Ma), jingyuanli@whu.edu.cn (J. Li), zyp91@whu.edu.cn (Y. Zhang), zhanglefei@whu.edu.cn (L. Zhang).

https://doi.org/10.1016/j.patcog.2020.107448
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

methods are proposed to handle random mask inpainting task. Liu et al. [20] introduce a special convolution strategy to distinguish mask regions from valid regions in the network. Nazeri et al. [21] add structure information to guide inpainting. Although these learning-based methods can generate plausible results for mask regions, they cannot borrow information from remote patches because of the ineffectiveness of CNNs in establishing the correlations in long distant regions.

For damaged images, only searching for missing information from surrounding areas is unreasonable. Yan et al. design the Shift-Net [22] that enhances the generation of fine details by adding a special feature shift in U-Net to capture global semantics. Ren et al. proposed the StructureFlow [23] by applying the appearance flow to construct long-term correlations between mask and valid regions for vivid texture generation. Although these methods can obtain global information to a certain extent, the scope of information searching has limitations. Increasing studies have focused on the attention mechanism to obtain exhaustive global information. Yu et al. [24] use a contextual attention layer to match generated patches with known contextual patches for utilizing the relevant feature at distant locations. Wang et al. [25] adopt multi-scale attention to flexibly utilize background information. In addition to the attention-related algorithms used for regular square mask inpainting tasks, the attention mechanism in irregular mask inpainting task is exploited [27–30]. However, most of the attention-related methods only use the fixed single-scale information. Matching sufficient information with a constant size is difficult because the demands on local and global levels are different from image to image. Thus, these attention-related methods may not obtain good results.

Existing methods have achieved good performance on square masked and small proportion damaged images, and inpainting algorithms specifically for images with large proportions of irregular defects are insufficient. In this paper, we propose a novel image inpainting algorithm for damaged images with large proportion. As previously mentioned, the attention mechanism is an important tool used in image inpainting tasks that can match pixels or patches on deep feature maps to generate good results. However, the patch matching on a fixed single-scale limits the ability to apply this model into different scenes. In response to this phenomenon, we introduce a multistage attention module that can flexibly utilize the deep feature maps in different layers to obtain information with various scales. The proposed multistage attention module can effectively utilize the background information to accurately restore the mask areas. The proposed algorithm adopts partial convolution strategies [20] into U-Net architecture [34] when handling areas of different states, including the known background and unknown foreground regions to avoid the misuse of fake information. Wrong information should be discarded during training because the proposed image inpainting algorithm targets randomly damaged images with large proportion. Therefore, we distinguish the background and foreground regions when calculating convolutional feature maps. A proposed joint loss function combines perceptual and adversarial losses and is mainly generated in accordance with the discriminator and feature extractor. The proposed joint loss function can help in obtaining realistic details. The proposed algorithm can recover the basic texture and structure of the damaged area in an image and obtain exquisite details.

The main contributions of this paper are summarized as follows.

- We propose a multistage attention module in U-Net like architecture. Different from the attention in other image inpainting algorithms, we introduce cascaded attention in the two layers of decoding, to ensure good results with exquisite details after patch swapping processes.
- We aim to solve image inpainting tasks for randomly damaged images with large proportion, and we adopt partial convolution strategy for the hole and valid regions to efficiently avoid the interference of mask areas (the initial value is artificially given) on the generated results.
- We introduce a joint loss function to provide better details and consistent style for inpainting results. The qualitative and quantitative results compared with several state-of-the-art algorithms on the Paris StreetView, Places2, and CelebA datasets demonstrate the superiority of the proposed approach.

The rest of this paper is organized as follows. Section 3 describes the proposed method. Section 4 explains some basic information used in the experiments, including datasets, training settings, and comparison models. Sections 5.1 and 5.2 compare the results. Section 5.3 conducts ablation analysis. Section 6 provides the conclusions.

2. Related works

2.1. Traditional methods

Traditional image inpainting methods found in the literature include diffusion-based and patch-based methods. Diffusion-based inpainting methods usually reconstruct target regions by continuously propagating high-order derivatives of local pixel intensity along isophote lines from the exterior into the damaged region [8,9] or by minimizing high-order partial differential equations of inpainting model [12]. These methods have good performance in maintaining local smoothness and generating geometrical structure. However, they produce blurring results when the holes are large. Patch-based methods can recover slightly large holes by searching the known regions of image at the patch level and copy the best-matching patches into the missing regions. In PatchMatch [14] method, some good patch matches can be found through random sampling, and the natural coherence in the imagery makes it possible to quickly propagate such matches to surrounding areas. Aside from PatchMatch, Ding et al. [16] develop a patch-based image inpainting method that searches the known regions by measuring nonlocal texture similarity and fuses the found candidate patches with α-trimmed mean filter for mask regions.

Patch-based methods solve the problem of diffusion-based methods, that is, the lack of the capability to inpaint large damaged regions. Some methods [10,35] attempt to integrate the diffusion-based and patch-based methods. These methods first decompose an image into two images, which are high-frequency and low-frequency components of the image, respectively. Low-frequency image is restored with diffusion-based method, and high-frequency image is recovered with patch-based method. The two inpainted images are integrated to obtain the final results. These combined methods provide limited performance improvements compared with patch-based patching.

Although patch-based methods can generate more reasonable results for large holes than diffusion-based methods, patch-based methods may generate unsuitable results, such as disconnected lines or broken edges when the background information is complex. Traditional methods cannot produce novel image contents for complex inpainting regions involving intricate structures, such as faces, because they assume that pixel information of missing regions can be found somewhere in the background regions. Traditional methods are limited by the available image information, causing the high-level semantics or global structure of the image cannot be captured.

2.2. Learning-based methods

With the rapid development of machine learning, many approaches based on CNNs have been proposed to cope with the abovementioned limitations by the support of large-scale training data.

2.2.1. Inpainting for rectangular areas

Pathak et al. [17] propose the CE algorithm. This algorithm can provide reasonable contents for filling damaged regions at the semantic level by passing the represented features of the input image. However, this method consistently produces semantically feasible but blurry results. Iizuka et al. [18] propose a globally and locally consistent image completion (GLCIC) architecture based on the CE algorithm that consists of completion network, global context discriminator, and local context discriminator to ensure global consistency and locally plausible details. Although GLCIC methods can capture globally consistent structure with locally plausible details, the sharpness level of details needs improvement. MNPS [19] combines the content network and texture network to obtain constraint information at holistic and local levels. MNPS introduces special neural features, such as neural style transfer [36], to generate realistic texture information. Similarly, the CA method [24] combines two networks with contextual attention layer to restore damaged regions in a coarse-to-fine manner. However, MNPS and CA are time consuming. Yan et al. [22] introduce a shift-connection layer into the benchmark U-Net named Shift-Net for filling in damaged regions to quickly generate reasonable content. With the help of shift-connection layer, encoder features of valid regions are shifted to missing regions as extra constraints.

However, these abovementioned methods usually focus on square mask region that frequently appears in the center of images. Some approaches [19,22,24] attempt to solve certain specific tasks with irregular masks, such as object removal. However, they do not conduct extensive experiments on complete datasets to verify the power of restoring irregularly damaged areas.

2.2.2. Inpainting for random areas

Some methods [20,21,26–28,30] have been proposed to solve this task for randomly damaged images. Mask regions in this task are distributed in random areas. Liu et al. [20] first replace convolutional layers with partial convolution (PConv) and mask updates to generate output by only utilizing valid inputs. With the similar idea to PConv, Yu et al. [27] propose a gated convolution (GatedConv) method that provides a learnable dynamic feature selection mechanism for each channel at each spatial location and automatically updates the mask on the basis of data. Ma et al. [26] add a region-wise convolution similar to [20] into a coarse-to-fine network and adopt nonlocal operation to deal with the differences and correlation between intact and damaged areas.

In addition to these methods improved using new convolutional manners, some methods have considered the contour/edge information to restore the damaged areas. Xiong et al. [30] propose a foreground-aware method that first detects and completes the contour of saliency objects in foreground and then uses the predicted contour to guide the inpainting stage. However, this method consists of excessively many networks, making it difficult to save computing resources. Nazeri et al. [21] design a more concise architecture called EdgeConnect than the foreground-aware method. EdgeConnect has the similar inpainting process of the foreground-aware method and is composed of an edge generator and an image completion network to reduce training time. Li et al. [28] adopt four visual structure reconstruction layers to restore visual structure using a progressive strategy.

2.3. Attention mechanism in inpainting

Attention mechanism, which has been widely used in high-level computer vision tasks, seeks for the most significant parts for result generation and enhances the performance of segmentation, reidentification, tracking, captioning, and question answering algorithms [37–42]. Attention mechanism is an important tool used in image inpainting that can help learn related features from the background. CA [24] innovatively adds a contextual attention module to its coarse-to-fine architecture that focuses on relevant feature patches at any location to improve inpainting results. Wang et al. [25] develop a multi-scale attention module in the MUSICAL architecture to flexibly utilize the background content. Pluralistic image completion (PIC) method [31] adopts a self-attention layer that utilizes short+long term context information to ensure appearance consistency. Xie et al. [29] design learnable bidirectional attention maps for handling irregular mask regions. Liu et al. [32] use a coherent semantic attention layer in refinement network to ensure semantic relevance between swapped features. Zeng et al. [33] apply attention mechanism to build an attention transfer network for utilizing high-level semantic information to reconstruct low-level image features. These image inpainting algorithms demonstrate the effectiveness of attention mechanism. However, most methods adopt attention mechanism with single fixed-scale while swapping patches that may cause inaccurate results. To solve this problem, we introduce a multistage attention module to flexibly utilize multi-scale information in this paper.

3. Proposed approach

In this section, we briefly describe of the proposed model architecture. The convolutional strategy and multistage attention are introduced. The loss function of the proposed method is provided. The overall architecture is shown in Fig. 1.

3.1. Model architecture

The proposed method adopts a U-Net like architecture as our backbone, which has been widely used in inpainting models [34]. U-Net like architectures use skip connections in each feature scale and have demonstrated superiority in retaining low-level information. Therefore, the network parameters can focus on the damaged parts of images and ignore the known regions of damaged images. In this study, we adopt the U-Net as the baseline but with some modifications.

Considering that the target of our methods is to inpaint on irregularly damaged images, conventional convolution strategy is suboptimal because it fails to consider the mask boundary shape and experiences fitting into the changing shapes of the hole boundary. To avoid this issue, we use partial convolution layers as the basic operator that renormalizes the feature value after the weighted sum operation.

Then, we introduce a multistage attention scheme to the third-to-last and fourth-to-last layers in the decoding stage of the network, which performs patch swapping in two different image scales and leads to accurate results.

In addition to the generator based on U-Net, we use a pre-trained VGG-16 feature extractor [43] and a fully convolutional patch discriminator in the proposed model [27]. Consequently, perceptual losses and patch-based adversarial loss are generated to ensure the images' global consistencies and the quality of filled details in irregular holes.

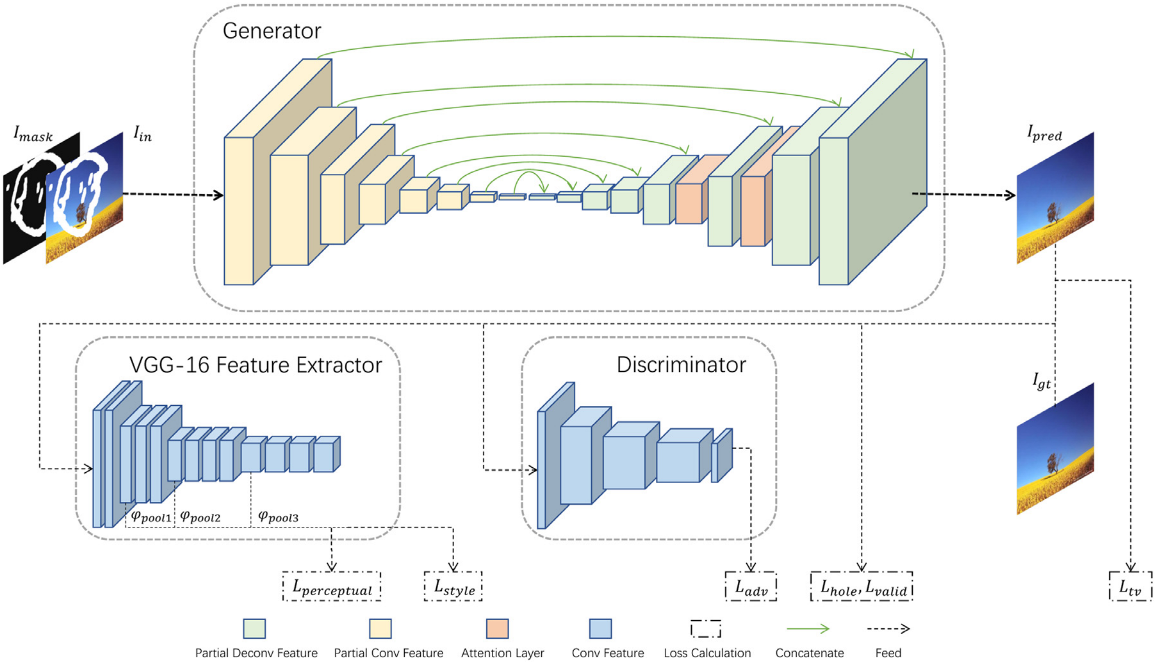

Fig. 1. Overall model architecture. We adopt a multistage attention module by adding two pixelwise attention layers in the decoding stage. The input of the proposed U-net like generator is the damaged image I_{in} and corresponding mask image II_{mask}. Several losses are generated by feeding predicted image I_{pred} and ground truth I_{gt} in the VGG-16 feature extractor and discriminator.

3.2. Convolutional strategy

For image inpainting issue, the mask regions are usually restored by imitating the pattern and style of the background regions. However, the information that can be used to learn the unknown regions during inpainting is relatively scarce when the proportion of defective regions is large. As previously mentioned, the generated results will be seriously affected by the fallacious information from the mask areas when the entire image is placed in the convolutional network. Therefore, a partial convolution strategy is needed [20] to distinguish the damaged and undamaged regions.

Partial convolution can be described in three parts, namely, mask convolution, feature renormalization, and mask updating. We define X_{in} as the feature map of input image and M_{in} as the corresponding feature map of binary mask image. W denotes the weights of convolution filter, and b represents the corresponding bias. We use M to denote the sum of M_{in} and E to denote the sum of a feature map, which has the same shape as M_{in} but with each element of one. Elementwise multiplication can be denoted as $\odot$. Then, partial convolution is expressed in Eqs. (1), (2) and (3):

$$F_{conv} = W^T(X_{in} \odot M_{in}) \tag{1}$$

$$F_{out} = \begin{cases} F_{conv} f_s(M) + b, & \text{if } M > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$M' = f_u(M) \tag{3}$$

Output feature map F_{out} is calculated by multiplying feature map F_{conv} with scaling factor $f_s(M)$ and adding bias b. $f_s(M)$ in Eq. (2) denotes the scaling factor and is defined as follows:

$$f_s(M) = \begin{cases} \frac{E}{M}, & \text{if } M > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We use M to denote the mask distribution in M_{in} at specific sampling grid τ. M is the sum of sampling region's values in M_{in}. E represents the size of sampling grid τ. It is equivalent to the sum of values of sampling grid τ, and every element in sampling grid τ is equal to one. The size of sampling grid depends on the setting of convolution layer. In accordance with the definition of $f_s(M)$, output feature map F_{out} is generated on the basis of the unmasked regions. For different undamaged regions (valid information), we can apply scaling factor $\frac{E}{M}$ to renormalize the feature accordingly.

After mask convolution and feature renormalization, we update the mask. We mark the position as valid when the input region contains at least one valid value. We define function $f_u(M)$ for mask updating as follows:

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

The partial convolution strategy is displayed in Fig. 2. In the inpainting process, we need to place two images, namely, damaged image I_{in} that needs to be restored and mask image I_{mask} that provides the locations of the damaged regions in I_{in}, in the generator. M_{in} is the feature map of I_{mask}, which has the corresponding mask regions of I_{in}'s feature map X_{in}. The values of binary mask image are 0s and 1s, representing the unmasked and masked regions, respectively.

Partial convolution goes through mask convolution, feature renormalization, and mask updating. With such a convolutional strategy, we can make adjustments in accordance with the varying condition of masks during convolution. When dealing with image inpainting tasks with a large proportion of mask areas, we can avoid the misuse of fake information and generate results close to realistic images.

3.3. Multistage attention module

Attention modules have been widely used in image inpainting tasks [24,25,29,31] because they allow nonlocal feature generating processes in a convolutional architecture that can initially capture local information.

In image inpainting task, we aim to recover the texture and structure of image and generate fine details. A previous work [25] indicated that generating attentive feature in different scales

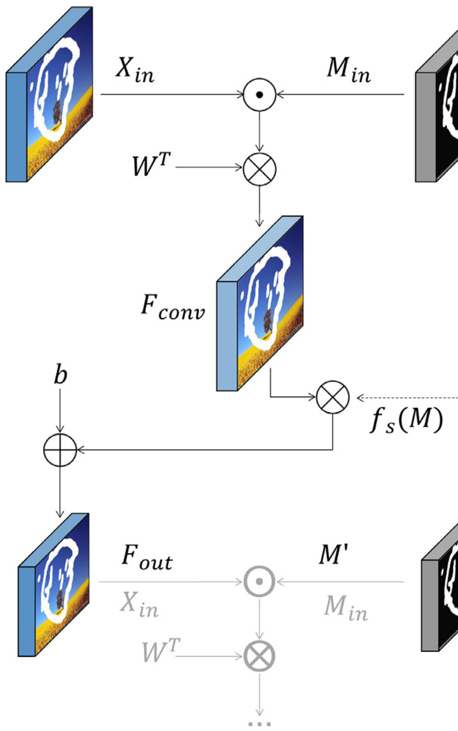

Fig. 2. Partial convolution strategy. The partial convolution strategy can be described in three parts, namely, mask convolution, feature normalization, and mask updating. In mask convolution, we obtain F_{conv}. After feature normalization, we generate output feature map F_{out}. M' is the updated mask feature map. F_{out} and M' can serve as the inputs for the next layer.

can effectively enhance the quality of feature map. However, this idea is infeasible for irregular hole cases because the irregular mask shape leads to 1) difficulty in background patch extraction (i.e., the invalid area in extracted patches causes artifacts in image) and 2) problems in patch matching (i.e., the invalid area leads to ineffective similarity calculation). To fit the multi-scale idea into the cases where the masks shapes are irregular, we perform pixelwise patch swapping in different decoding layers of the network. This type of cascaded attention mechanism is equivalent to that in the original work [25] because pixels in previous parts of the decoding stage will be decoded into large patches in the following layers, without being influenced by the unpredictable hole boundary. In this paper, we call the proposed attention mechanism as multistage attention.

Undamaged regions are defined as background, and mask regions are defined as foreground. A foreground that is consistent with the background should be generated by borrowing features from it. As shown in Fig. 3, given an input feature map ϕ_{in}, we first extract pixelwise patches from the background and reshape them as convolutional filters. We measure the similarity of patches in background $\{b_{x',y'}\}$ and foreground $\{f_{x,y}\}$. We match them by calculating the normalized inner product. Similarity $s_{x,y,x',y'}$ can be expressed as follows:

$$s_{x,y,x',y'} = \langle \frac{f_{x,y}}{||f_{x,y}||}, \frac{b_{x',y'}}{||b_{x',y'}||} \rangle \tag{6}$$

After the matching of patches, we apply a SoftMax function to calculate the weight, that is, attention score $s^*_{x,y,x',y'}$, of each pixel. λ is a constant value.

$$s^*_{x,y,x',y'} = f^{softmax}_{x',y'}(\lambda s_{x,y,x',y'}) \tag{7}$$

For the attention of background and foreground, we aim to maintain the coherency that the corresponding patch from background is likely to have an equal shift when a patch from foreground has a shift. Therefore, we conduct a left-right shift and a top-down shift with kernel size of k to propagate the attention score as follows:

$$s'_{x,y,x',y'} = \sum_{i,j\in\{-k,\ldots,k\}} s^*_{x+i,y+j,x'+i,y'+j} \tag{8}$$

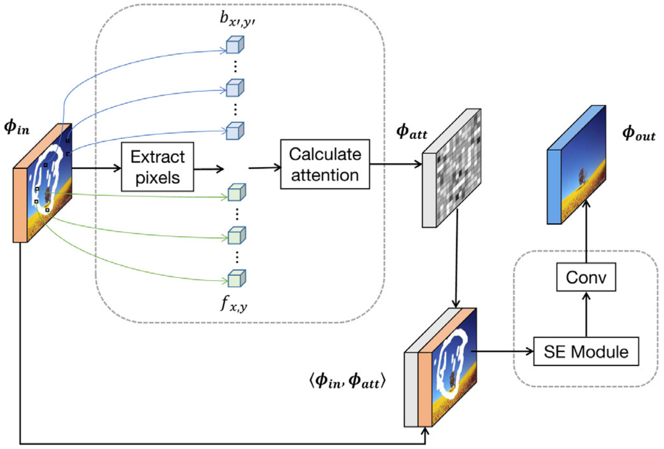

Fig. 3. Pixelwise Attention. Given input feature map ϕ_{in}, our pixelwise attention can extract pixelwise patches from both background and foreground. Then we can calculate the attention scores ϕ_{att}. The concatenated feature map of input and generated feature map is denoted by $\langle\phi_{in}, \phi_{att}\rangle$, and it is sent into SE Module to enhance the weight of useful features. Finally, pixelwise convolution is adopt to ensure that the channel number is as the same as the original channel number. To achieve the idea of multi-scale, we add the pixelwise attention in different decoding layers.

The calculation of attention score is implemented as convolutional calculation. We use background patches $\{b_{x',y'}\}$ previously extracted as filters to restore the foreground through deconvolutional calculation. Then, input feature maps ϕ_{in} and generated feature maps ϕ_{att} are concatenated and denoted by $\langle\phi_{in}, \phi_{att}\rangle$.

The concatenated feature maps include the attentive and original features. They will be sent to a convolution operator in the following layers of the network. However, one issue is that the importance of the attention feature map is variant for different images. This condition indicates that the feature merging process should be treated in an adaptive manner, where the original and attentive features should be given different weights. Thus, we borrow the idea of a Squeeze-and-Excitation (SE) module [44], where different channels emphasize differently. SE is denoted as $f_{SE}()$. The SE Module is shown in Fig. 4. The pipeline of the module can be decomposed into three parts, namely, 1) Squeeze that compresses feature maps through spatial dimensions, 2) Excitation that generates weights for each channel, and 3) Reweight operation, where the outputs of excitation are regarded as the importance of each feature channel, and they are weighted to the previous features channel by channel-wise multiplication. In particular, the squeeze operation uses a spatial global average pooling to embed the global information in the following manner:

$$z_c = \frac{1}{H \times W}\sum_{i}^{H}\sum_{j}^{W} u_{i,j} \tag{9}$$

where H and W are the height and width of input feature map $u_{i,j}$, respectively.

Then we extract the information from embedding vector z_c and transform it as a weighting vector, where its value range is [0,1]. This process is implemented using a simple multilayer perceptron. This process can be expressed as follows:

$$w_c = \sigma(W_2\psi(W_1, z)) \tag{10}$$

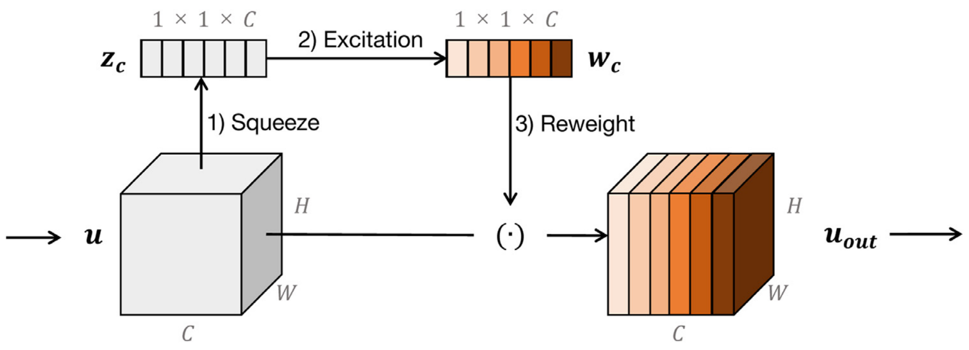

Fig. 4. SE Module. The SE Module includes three steps: 1) Squeeze, 2) Excitation, and 3) Reweight. In Squeeze part, feature map of input u is compressed to get channel-wise responses z_c, which is actually the global average pooling. Then the weights w_c of feature maps in C channels are learned through fully connected layers and non-linear layers in Excitation part. It is the most important segment in SE Module. Finally, channel-wise multiplication is adopt to get the reweighted the feature map u_{out}.

where W_1, W_2, indicate the weights, and ψ denotes the ReLU activation function. σ is the sigmoid function that transforms the vector into a weight vector w_c. The channel number of w_c should be the same as that of the original feature map of the layer. We perform channel-wise multiplication between the original feature map and the weight vector when the weight vector is obtained, where the importance of each channel is adjusted as follows:

$$u_{out} = u \odot w_c \tag{11}$$

We use $f_{SE}()$ to represent the SE network, and we feed concatenated feature maps $\langle \phi_{in}, \phi_{att} \rangle$ in SENet to selectively enhance the weight of useful features. We describe the output of SE module as $f_{SE}(\langle \phi_{in}, \phi_{att} \rangle)$. SENet can adaptively generate various weights on the basis of background information. Then, we apply the generated weights to the corresponding channels.

We need to maintain the channel of outputs and inputs equal to propagate the attention to the next layer. Therefore, we merge all feature maps and shrink the channel numbers to the original channel number using pixelwise convolution $f_{Conv}()$. The final output of the module can be denoted as:

$$\phi_{out} = f_{Conv}(f_{SE}(\langle \phi_{in}, \phi_{att} \rangle)) \tag{12}$$

We add the attention module to the third-to-last and fourth-to-last layers in the decoding stage to consider the computing resources and quality of inpainted results. The feature maps of the attention from fourth-to-last layer ϕ^1_{out} have the size of 32 × 32, and the corresponding results from third-to-last layer ϕ^2_{out} have the size of 64 × 64. The cascaded attention modules can be realized under the condition where the masked area is irregular.

3.4. Loss function

Similar to the abovementioned design concept, we consider the global consistency and exquisite detail while constructing the loss function. The loss network consists of two parts, namely, a 16-layer VGG network [43] pretrained on ImageNet [45] and a discriminator similar to the counterpart in GatedConv [27]. The loss function is composed of six parts, namely, the perceptual and style losses for global consistency, the adversarial loss for delicate details, the hole and valid losses for content similarity, and the total variation loss for spatial smoothness.

Perceptual Loss: We must ensure the similarity of high-level structures to maintain the structure information of the global image. Therefore, similar feature representations to ground truth are needed rather than the pixels matching between them. We calculate perceptual loss ($L_{perceptual}$) by feeding generated image (I_{pred}) and ground truth (I_{gt}) in the VGG-16 feature extractor. Then, feature maps $\phi^{gt}_{pool_i}$ and $\phi^{pred}_{pool_i}$ produced from $pool_i$ are compared accordingly. $pool_i$ indicates the i^{th} layer of VGG-16. The perceptual loss can be written as follows:

$$L_{perceptual} = \sum_{i=1}^{N} \frac{1}{H_i W_i C_i} |\phi^{gt}_{pool_i} - \phi^{pred}_{pool_i}|_1 \tag{13}$$

Style Loss: Perceptual loss helps to obtain high-level structure and avoids the deviation of the generated image from the ground truth in content. However, we still need the ability to preserve style consistency, such as color and patterns. To achieve this goal, we add style loss (L_{style}) to the loss function. Similar to $L_{perceptual}$, we need to input I_{pred} and I_{gt} into VGG-16. The difference is that we no longer compare the feature maps generated in VGG-16, and $\phi^{style}_{pool_i}$ is defined in advance. $\phi^{style}_{pool_i}$ is the product of feature maps multiplied by its transpose.

$$\phi^{style}_{pool_i} = \phi_{pool_i} \phi^{T}_{pool_i} \tag{14}$$

We obtain the style loss by comparing $\phi^{style}_{pool_i}$ of the generated image and ground truth. With the help of style loss, the proposed model can obtain the color and overall style information from the background.

$$L_{style} = \sum_{i=1}^{N} \frac{1}{C_i * C_i} \left| \frac{1}{H_i W_i C_i} (\phi^{style_{gt}}_{pool_i} - \phi^{style_{pred}}_{pool_i}) \right|_1 \tag{15}$$

In the equations of $L_{perceptual}$ and L_{style}, H_i, W_i, and C_i represent the height, weight, and channels size of the i^{th} feature map, respectively. N refers to the number of feature maps generated by the VGG-16 feature extractor.

Adversarial Loss: $L_{perceptual}$ and L_{style} are the combination of image semantic and feature information. Thus, the proposed model can generate plausible content with similar pattern in structure and style to that of ground truth. However, the generated content has no advantage in the level of detail. Thus, we introduce adversarial loss L_{adv}, to our loss function. $L_{perceptual}$ and L_{style} are produced through VGG-16, and adversarial loss L_{adv}, is generated through another part of our loss network, that is, a fully convolutional discriminator.

Most existing studies, where their mask is a single square area, use an extra discriminator to judge the quality of generated images. The masks may exist anywhere in the images because our inpainting task aims at randomly damaged images. Therefore, we use a full convolutional network as discriminator D in the proposed method. Different from the existing methods, the layers of our discriminator D are shallow, thereby saving computing resources. The outputs of the discriminator no longer represent the entire image but refer to different areas of an image. In this way, each of the discrimination results represents the local features of image, which is reasonable for inpainting tasks that contain irregular masks. The set of local discrimination results covers all regions of the image. Thus, additional global discriminator is unnecessary.

We use I_{mask} to represent the input mask of generator G and I_{in} to represent the input damaged image of G. Then, the output of generator G is expressed as $G(I_{in}, I_{mask})$, which is also the result of our inpainting algorithm (I_{pred}). Evidently, we aim to obtain a result similar to the ground truth. Thus, we aim to discriminate I_{pred} as true when this result is sent to discriminator D. However, the basic work of discriminator D is to identify whether each input is real. In summary, our adversarial loss L_{adv} can be expressed by combining generator G and discriminator D as follows:

$$L_{adv} = \max_{G} \min_{D} BCE_T[D(I_{gt})] + BCE_F[D(I_{pred})] \tag{16}$$

$BCE(x)$ refers to binary cross entropy, and the only difference between $BCE_T(x)$ and $BCE_F(x)$ is the value of labels.

$$BCE(x) = \begin{cases} BCELoss(x, 1.0), & \text{if } BCE_T(x) \\ BCELoss(x, 0.0), & \text{if } BCE_F(x) \end{cases} \tag{17}$$

The binary cross entropy loss is expressed as follows:

$$BCELoss(x, y) = \sum_{i=1}^{N} -W_i[y_i \cdot \log x_i + (1 - y_i) \cdot \log(1 - x_i)] \tag{18}$$

where N denotes the batch size of input, and W_i is a manual rescaling weight, which is undefined in the proposed method.

We can control the level of sharpness and detail of inpainting results using different weights of adversarial loss.

Total Variation Loss: L_{style} considers the global consistency, but still has the issue on checkboard artifact. In the proposed model, total variation loss (L_{tv}) is introduced to enhance the spatial smoothness of the generated image. We can obtain harmonious results by computing L_{tv} of damaged areas. $m_{x,y}$ refers to the pixel value of position (x, y).

$$L_{tv} = \mathbb{E}[\sum_{x,y}(\left|m_{x,y} - m_{x+1,y}\right|_1 + \left|m_{x,y} - m_{x,y+1}\right|_1)] \tag{19}$$

Hole Loss and Valid Loss: L_{hole} and L_{valid} are the L1 regularization between generated image I_{pred} and ground truth I_{gt} of hole (damaged) and valid (undamaged) regions. Then, L_{hole} and L_{valid} can be expressed as follows:

$$L_{hole} = \mathbb{E}[\sum_{i,j,k} \left|p_{i,j,k}^{gt\_hole} - p_{i,j,k}^{pred\_hole}\right|_1] \tag{20}$$

$$L_{valid} = \mathbb{E}[\sum_{i,j,k} \left|p_{i,j,k}^{gt\_valid} - p_{i,j,k}^{pred\_valid}\right|_1] \tag{21}$$

In the equations of L_{hole} and L_{valid}, i, j, and k refer to the location of height, weight, and channel of the generated image/ground truth, respectively. p^{gt_hole} represents the pixel value from the hole regions in I_{gt}, and p^{pred_valid} represents the pixel value from the valid regions in I_{pred}.

Overall Loss: The overall loss function of the proposed algorithm is expressed as follows:

$$\begin{aligned} L_{overall} = {} & \lambda_{perceptual} L_{perceptual} + \lambda_{style} L_{style} \\ & + \lambda_{adv} L_{adv} + \lambda_{tv} L_{tv} + \lambda_{hole} L_{hole} + \lambda_{valid} L_{valid} \end{aligned} \tag{22}$$

3.5. Network architecture

The detailed architecture and hyperparameters of our generator are shown in Tables 1 and 2. The architecture of the discriminator is shown in Table 3. We use Input to represent the source of the input of each layer. In_S and Out_S indicate the sizes of the feature maps before and after processing, respectively. Ori indicates the size of the original input image. K, S, and P denote the kernel size, stride, and padding of operators, respectively. In_C and Out_C denote the channel number of the input feature and output feature maps, respectively. BN indicates whether batch normalization layer is adopted after each operator, and SN indicates whether spectral normalization layer is used. Act_Func indicates the nonlinear function after the layer. Operator represents the operation of each layer.

Table 1
Architecture of the generator.

Generator Architecture				
Type	Input	BN	Act_Func	Operator
Encode_1	Img_Masked	F	ReLU	PConv
Encode_2	F_Enc1	T	ReLU	PConv
Encode_3	F_Enc2	T	ReLU	PConv
Encode_4	F_Enc3	T	ReLU	PConv
Encode_5	F_Enc4	T	ReLU	PConv
Encode_6	F_Enc5	T	ReLU	PConv
Encode_7	F_Enc6	T	ReLU	PConv
Encode_8	F_Enc7	T	ReLU	PConv
Upsampling	F_Enc8			Nearest
Decode_8	cat(F_Up,F_Enc7)	T	Leaky_ReLU	PConv+Nearest
Decode_7	cat(F_Dec8,F_Enc6)	T	Leaky_ReLU	PConv+Nearest
Decode_6	cat(F_Dec7,F_Enc5)	T	Leaky_ReLU	PConv+Nearest
Decode_5	cat(F_Dec6,F_Enc4)	T	Leaky_ReLU	PConv+Nearest
Decode_4	cat(F_Dec5,F_Enc3)	T	Leaky_ReLU	PConv
Attention_1	F_Dec4			Attention+Nearest
Decode_3	cat(F_Att1,F_Enc2)	T	Leaky_ReLU	PConv
Attention_2	F_Dec3			Attention+Nearest
Decode_2	cat(F_Att2,F_Enc1)	T	Leaky_ReLU	PConv+Nearest
Decode_1	cat(F_Dec2,F_Input)	F	None	PConv

Table 2
Hyperparameters of the generator.

Generator Hyperparameters							
Type	In_S	Out_S	K	S	P	In_C	Out_C
Encode_1	Ori	Ori/2	7	2	3	3	64
Encode_2	Ori/2	Ori/4	5	2	2	64	128
Encode_3	Ori/4	Ori/8	5	2	2	128	256
Encode_4	Ori/8	Ori/16	3	2	1	256	512
Encode_5	Ori/16	Ori/32	3	2	1	512	512
Encode_6	Ori/32	Ori/64	3	2	1	512	512
Encode_7	Ori/64	Ori/128	3	2	1	512	512
Encode_8	Ori/128	Ori/256	3	2	1	512	512
Upsampling	Ori/256	Ori/128				512	512
Decode_8	Ori/128	Ori/64	3	1	1	512+512	512
Decode_7	Ori/64	Ori/32	3	1	1	512+512	512
Decode_6	Ori/32	Ori/16	3	1	1	512+512	512
Decode_5	Ori/16	Ori/8	3	1	1	512+512	512
Decode_4	Ori/8	Ori/8	3	1	1	512+256	256
Attention_1	Ori/8	Ori/4				256	256
Decode_3	Ori/4	Ori/4	3	1	1	256+128	128
Attention_2	Ori/4	Ori/2				128	128
Decode_2	Ori/2	Ori	3	1	1	128+64	64
Decode_1	Ori	Ori	3	1	1	64+3	3

4. Experiments

4.1. Datasets

Experiments are conducted on three datasets commonly adopted in image inpainting literature, namely, Paris StreetView [46], CelebA [47], and Places2 [48]. We also adopt an external mask dataset [20] to simulate the damaged areas. We use the same dataset setting for our experiment and comparison experiments.

Paris StreetView: It includes 14,900 training images and 100 test images, and they are collected from the street views of Paris. This dataset contains a large amount of buildings in Paris and structure information, such as windows and doors.

CelebFaces Attributes: CelebA is a large-scale facial dataset containing more than 200,000 celebrity images. For this dataset, we adopt 182,637 images for training (the entire sets of training

Table 3
Architecture of the discriminator.

Discriminator Architecture										
Type	Input	In_S	Out_S	K	S	P	In_C	Out_C	SN	Act_Func
Conv1	Img_Pred	Ori	Ori/2	4	2	1	3	64	T	Leaky_ReLU
Conv2	F_conv1	Ori/2	Ori/4	4	2	1	64	128	T	Leaky_ReLU
Conv3	F_conv2	Ori/4	Ori/8	4	2	1	128	256	T	Leaky_ReLU
Conv4	F_conv3	Ori/8	Ori/8	4	1	1	256	512	T	Leaky_ReLU
Conv5	F_conv4	Ori/8	Ori/8	4	1	1	512	1	T	Sigmoid

and validation images) and 2000 for testing (selected from 19,962 testing images).

Places2: Places365-Standard contains 1.6 million training images from 365 scene categories. We select six scene categories from it as training data, including Butte, Canyon, Field, Synagogue, Tundra, and Valley. The validation sets corresponding to the six scenarios are used as testing data. Places2 has a total of 196,981 training images and 700 testing images.

Masks: The irregular mask dataset is collected from the work of PConv. Irregular masks are augmented by introducing four rotations, namely, 0°, 90°, 180°, and 270°.

4.2. Training settings

We resize each of the training images to 256 × 256 and train our networks with the batch size of six. In particular, we randomly crop an image for Paris StreetView and Places2 with the target size of 256 × 256 as input by resizing the minimum width or length to 350. For CelebA, we need the information of faces, which is located at the center of images to be preserved after cropping. Thus, we resize each training image to make its minimal length/width be 256 and crop a subimage of size 256 × 256 at the center as input to the proposed model. For the hyperparameters, we use 0.05 for $\lambda_{perceptual}$, 120 for λ_{style}, 0.1 for λ_{adv}, 0.1 for λ_{tv}, 1 for λ_{valid}, and 6 for λ_{hole}. We empirically choose them for our approach on the basis of experimental observations. All the experiments are conducted on Python with Ubuntu 17.10 system, i7-6800K 3.40 GHz CPU, and 12G NVIDIA Titan Xp GPU.

4.3. Comparison models

We compare the proposed algorithm with some state-of-the-art image inpainting algorithms. We train these models with the same settings as our experiments. The models used for comparison include the following:

Pluralistic Image Completion [31]: The PIC approach proposes a parallel pipeline model based on probabilistically principled framework to maintain sample diversity. A short+long term attention layer is introduced to improve semantic consistency.

Partial Convolutional U-Net [20]: PConv is a classic method that allows irregular inpainting. This method uses a specially devised PConv U-Net to make the model aware of the mask shape and devises a loss function to take the place of conventional adversarial loss.

EdgeConnect [21]: EdgeConnect is composed of an edge generator that which can provide edge information in missing regions and an image completion network for recovering the damaged areas with known color and texture information and the generated edges.

5. Results

The performance of the proposed method is evaluated in three phases. In the first phase, qualitative comparison results are presented. In the second phase, we present the quantitative comparison results. We perform ablation analysis on three topics in the third phase.

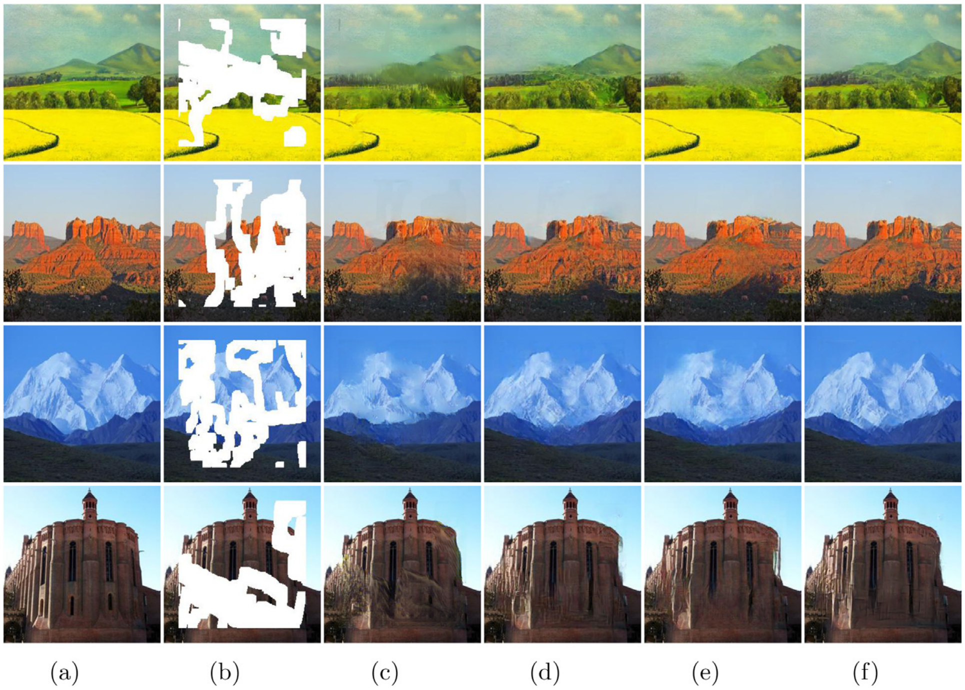

Fig. 5. Qualitative comparisons on Places2 dataset. From left to right are: (a) Ground Truth, (b) Input, (c) PIC, (d) PConv, (e) EdgeConnect, (f) Ours.

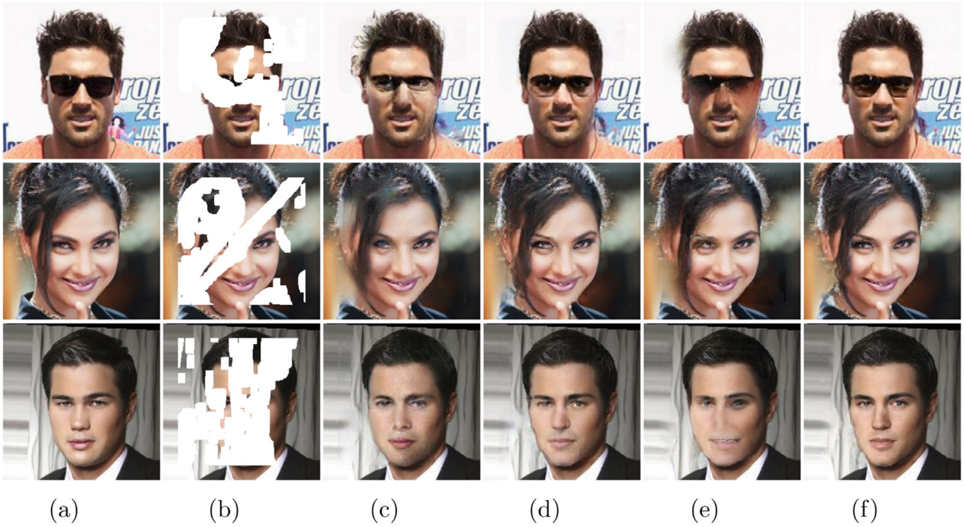

Fig. 6. Qualitative comcelebaons on CelebA dataset. From left to right are: (a) Ground Truth, (b) Input, (c) PIC, (d) PConv, (e) EdgeConnect, (f) Ours.

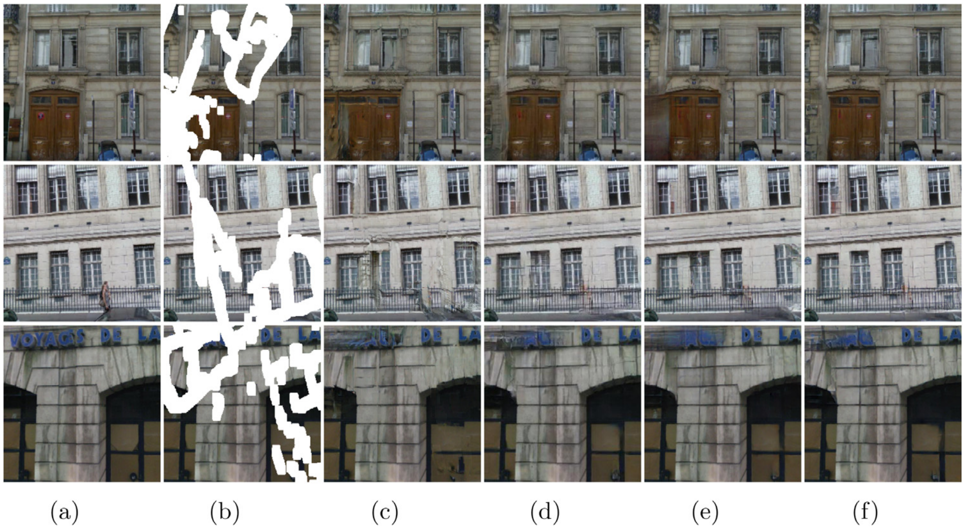

Fig. 7. Qualitative comparisons on Paris StreetView dataset. From left to right are: (a) Ground Truth, (b) Input, (c) PIC, (d) PConv, (e) EdgeConnect, (f) Ours.

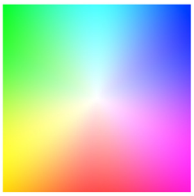

Fig. 8. Color map of attention. We use different colors to indicate the location of attention.

5.1. Qualitative analysis

Figs. 5, 6, and 7 present the inpainting results on Places2, CelebA, and Paris StreetView, respectively. Then, we discuss the qualitative performance of the proposed method and compare it with some of the state-of-the-art methods. The qualitative results show that the generated contents tend to be blurry although PIC can provide multiple and diverse results for inpainting tasks. As shown in Fig. 5, the PIC method can obtain plausible predicted results but many areas are blurry. PConv is effective in dealing with irregular masks, but the details in the restored regions are not as delicate as the background. EdgeConnect can successfully recover the mask regions when they are sparse because of its excellent ability to recover the edge of damaged areas. However, EdgeConnect fails to restore the missing edges and cannot generate plausible results when the mask regions are immense. The results of EdgeConnect in Paris StreetView dataset are extremely similar to the ground truth, showing its power in structure recovery, but the quality of results in CelebA are insufficient. The inpainted results of PIC, PConv, and EdgeConnect cannot obtain the equivalent level of detail compared with our results. The proposed method considers the global consistency and exquisite detail and performs better in visually generating realistic results with fine details and consistent textures.

5.2. Quantitative analysis

We adopt three quantitative measures, namely, peak-signal-to-noise ratio (PSNR), structure similarity index measure (SSIM), and mean $l1$ loss. PSNR, SSIM, and $l1$ loss are widely used metrics in image inpainting tasks, such as PConv [20], Shift-Net [22], and EdgeConnect [21]. In image inpainting task, we usually aim to generate results similar to ground truth. Although PSNR does not cor-

Table 4
Numerical comparison on three datasets. *Higher is better. †Lower is better.

Dataset	Metric	Method	Mask Ratio		
			10%-20%	30%-40%	50%-60%
Paris StreetView	SSIM*	PIC [31]	0.9071	0.7881	0.6393
		PConv [20]	0.8937	0.8042	0.6814
		EdgeConnect [21]	**0.9330**	0.8351	**0.7111**
		Ours	0.9327	**0.8355**	0.7061
	PSNR*	PIC [31]	28.76	23.70	19.61
		PConv [20]	29.75	25.41	21.97
		EdgeConnect [21]	30.69	25.35	21.79
		Ours	**31.08**	**25.81**	**22.17**
	Mean $l_1^\dagger$	PIC [31]	0.01115	0.02921	0.05866
		PConv [20]	0.01819	0.02924	0.04652
		EdgeConnect [21]	**0.00862**	0.02284	0.04284
		Ours	0.00890	**0.02212**	**0.04102**
CelebA	SSIM*	PIC [31]	0.9380	0.8437	0.7085
		PConv [20]	0.9261	0.8533	0.7431
		EdgeConnect [21]	0.9469	0.8605	0.7335
		Ours	**0.9529**	**0.8744**	**0.7599**
	PSNR*	PIC [31]	30.65	24.76	19.55
		PConv [20]	31.43	26.41	21.83
		EdgeConnect [21]	31.17	25.38	20.23
		Ours	**32.42**	**26.67**	**21.95**
	Mean $l_1^\dagger$	PIC [31]	0.00812	0.02282	0.05264
		PConv [20]	0.01310	0.02277	0.04209
		EdgeConnect [21]	0.00732	0.02049	0.04827
		Ours	**0.00630**	**0.01743**	**0.03774**
Places2	SSIM*	PIC [31]	0.8916	0.7521	0.6027
		PConv [20]	0.9085	0.7783	0.6267
		EdgeConnect [21]	0.9081	0.7796	0.6352
		Ours	**0.9101**	**0.7852**	**0.6359**
	PSNR*	PIC [31]	27.54	22.65	18.89
		PConv [20]	28.58	23.69	20.20
		EdgeConnect [21]	28.41	23.63	20.18
		Ours	**28.84**	**23.96**	**20.41**
	Mean $l_1^\dagger$	PIC [31]	0.01299	0.03284	0.06266
		PConv [20]	**0.01237**	0.02986	0.05435
		EdgeConnect [21]	0.01250	0.02999	0.05465
		Ours	0.01245	**0.02863**	**0.05232**

relate well with perceptual/subjective image quality assessment by humans, it reflects the difference between the inpainted result and ground truth.

We provide the quantitative results on three datasets with mask ratio (0.1, 0.2], (0.3, 0.4], and (0.5, 0.6] in Table 4. In CelebA dataset, the proposed method achieves the best results for three indicators. In Places2 dataset, we also obtain the best performance, except for the mean l_1 loss of the PConv method at mask ratio of 10%-20%. The proposed method obtains better results with the increase in mask ratio. In Paris StreetView, the proposed method has better results compared with PIC and PConv methods. However, SSIM and mean l_1 loss are sometime poor compared with the corresponding results of EdgeConnect. The images in Paris StreetView have sufficient structure information. Thus, inpainting the damaged regions with EdgeConnect is effective when they are relatively small. However, the propose method can obtain better results with the increase in damaged regions. The proposed model performs well in three measures, especially when the mask ratio is high, thereby showing the robustness of the proposed algorithm.

5.3. Ablation analysis

5.3.1. Visualization of multistage attention

One of the main contributions of the proposed model is the multistage attention module. To present the effectiveness of our multistage attention, the attention maps of ϕ_{att}^1 and ϕ_{att}^2 are visualized using different colors to show the relative location of the most interesting background region for each pixel in the foreground. We use the color map in Fig. 8, where pink indicates that the pixel appears on the bottom-right region, and green indicates that it appears on the top-left area. The visualization results of attention maps are displayed in Fig. 9.

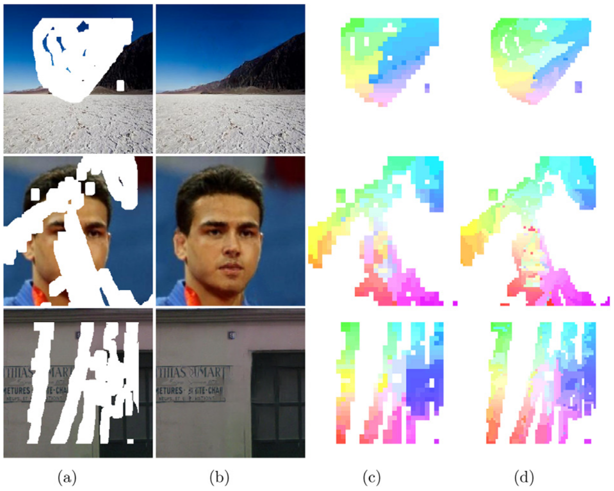

Fig. 9. Visualization of attention maps. From left to right are: (a) Input Image, (b) Generated Image, (c) Attention Map of 32 × 32, (d) Attention Map of 64 × 64.

The size of attention map ϕ_{att}^1 is 32 × 32 and that of ϕ_{att}^2 is 64 × 64. For the missing areas, our pixelwise attention first searches in surrounding valid areas that may have the most related distribution. Therefore, ϕ_{att}^1 and ϕ_{att}^2 have extremely similar distributions of color globally. Subsequently, our pixelwise attention can borrow pixels from remote regions when the nearest valid region cannot provide reasonable results. Taking the second case in Fig. 9 as an example, the attention maps of the nose position include several different information from many locations. We observe a similar phenomenon in the other test images. The experimental results suggest that our attention can learn useful information locally and globally. In particular, ϕ_{att}^1 can obtain the structure and texture information of missing regions, and ϕ_{att}^2 obtains many delicate information of details. On this basis, we can effectively acquire global semantic with the proposed multistage attention module, which is appropriate for mask inpainting task with large proportion. We can semantically generate reasonable and fine-detailed results with the proposed multistage attention module.

5.3.2. Effect of loss function

We set a series of experiments by changing the hyperparameters ($\lambda_{perceptual}$, λ_{style}, and λ_{adv}) to identify the role of perceptual loss ($L_{perceptual}$), style loss (L_{style}), and adversarial loss (L_{adv}). The experimental results are shown in Fig. 10. In Figs. 10(c-e), we set λ_{adv} to 0, 0.001, and 1, respectively. The results of $\lambda_{adv} = 0$ and $\lambda_{adv} = 0.001$ are insufficiently realistic. The structures in the images of $\lambda_{adv} = 1$ tend to be distorted. Compared with the three sets of experiments, the results generated by our original settings ($\lambda_{adv} = 0.1$) are more natural. In Fig. 10(f), we remove the perceptual and style losses. Without the perceptual and style losses, the inpainted results are less semantically plausible and much structural information cannot be recovered.

5.3.3. Speed

We implement the proposed algorithm on Python with Ubuntu 17.10 system, and all the experiments are conducted with i7-6800K 3.40 GHz CPU and 12G NVIDIA Titan Xp GPU. The execution time comparison of the proposed method with the other methods is shown in Table 5. The PIC method is the fastest. However, its inpainted results contain many blurry regions and are poor. The PConv method is approximately twice as fast as the proposed

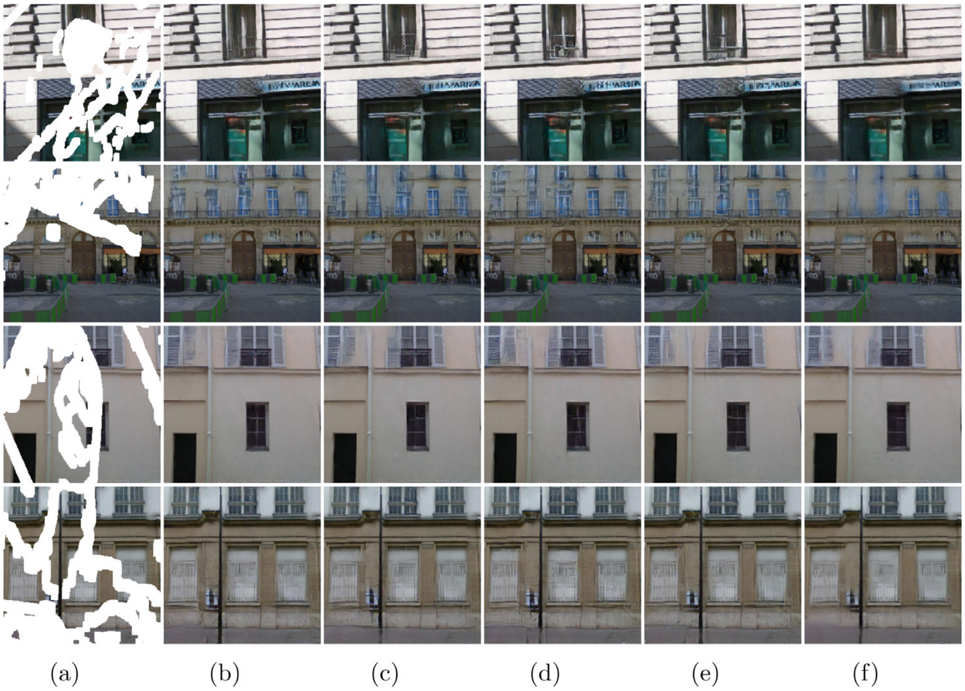

Fig. 10. Effect of the adversarial loss. From left to right are: (a) Input Image, (b) Ours ($\lambda_{adv} = 0.1$), (c) $\lambda_{adv} = 0$, (d) $\lambda_{adv} = 0.001$, (e) $\lambda_{adv} = 1$, (f) $\lambda_{perceptual} = 0$ and $\lambda_{style} = 0$.

Table 5
Run time of comparison methods. Run time refers to the training time for a single image.

Algorithm	Run Time (ms)
PIC [31]	6.3
PConv [20]	32.75
EdgeConnect [21]	56.5+84
Ours	67.5

method, but PConv cannot produce reasonable and exquisite results similar to the proposed method. EdgeConnect, which is a two-stage method, first needs to train the edge generator and the image completion network. Thus, the training time of EdgeConnect includes 56.5 ms of edge training time and 84 ms of inpainting training time, with a total of approximately 140.5 ms per image. EdgeConnect takes approximately twice as long to run as the proposed method.

6. Conclusion

This paper proposes a novel multistage attention network for image inpainting. Partial convolution strategy helps avoid the misuse of fake information by making adjustments during convolution. The multistage attention module realizes the multi-scale idea for irregular mask regions by adding cascaded attention. We can generate realistic results by combining the partial convolution strategy and multistage attention with the proposed joint loss function to maintain the global style of background regions and generate delicate details. Exhaustive experiments indicate that the proposed model outperforms state-of-the-art methods in generating realistic and fine-detailed results.

As an improvement of region-wise convolution inpainting method, this work has an important value for image inpainting. The embedding of multistage attention module into deep neural network provides a feasible research idea to other image processing techniques, such as image denoising. Although the proposed method performs well, it still has weaknesses. The proposed image inpainting model may create artifacts in some cases. This finding is mainly because batch normalization misuses invalid information from mask regions that may influence the inpainted results. To solve this problem, we will introduce region-wise normalization into deep neural network in our future study to avoid interference of mask regions during normalization.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant 61771349, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170, and the Natural Science Foundation of Hubei Province under Grant 2018CFB432. The numerical calculations in this paper had been supported by the supercomputing system in the Supercomputing Center of Wuhan University.

References

[1] C. Guillemot, O. Le Meur, Image inpainting: overview and recent advances, IEEE Signal Process. Mag. 31 (1) (2013) 127–144.
[2] Q. Sun, L. Ma, S. Joon Oh, L. Van Gool, B. Schiele, M. Fritz, Natural and effective obfuscation by head inpainting, in: Proc. IEEE Comput. Vision Pattern Recognit., 2018, pp. 5050–5059.
[3] B. Dolhansky, C. Canton Ferrer, Eye in-painting with exemplar generative adversarial networks, in: Proc. IEEE Comput. Vision Pattern Recognit., 2018, pp. 7902–7911.
[4] S. Wang, S. Tsai, Automatic image authentication and recovery using fractal code embedding and image inpainting, Pattern Recognit. 41 (2) (2008) 701–712.
[5] W. Hu, D. Tao, W. Zhang, Y. Xie, Y. Yang, The twist tensor nuclear norm for video completion, IEEE Trans. Neural Netw. Learn. Syst. 28 (12) (2017) 2961–2973.
[6] J. Han, X. Ji, X. Hu, D. Zhu, K. Li, X. Jiang, G. Cui, L. Guo, T. Liu, Representing and retrieving video shots in human-centric brain imaging space, IEEE Trans. Image Process. 22 (7) (2013) 2723–2736.
[7] NewsonAlasdair and Almansa, Andrés and Fradet, Matthieu and Gousseau, Yann and Pérez, Patrick, Video inpainting of complex scenes, SIAM J. Imag. Sci. 7 (4) (2014) 1993–2019.
[8] M. Bertalmio, A.L. Bertozzi, G. Sapiro, Navier-stokes, fluid dynamics, and image and video inpainting, in: Proc. IEEE Comput. Vision Pattern Recognit., 2001, pp. I355–I362.

[9] F. Li, L. Pi, T. Zeng, Explicit coherence enhancing filter with spatial adaptive elliptical kernel, IEEE Signal Process. Lett. 19 (9) (2012) 555–558.

[10] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, IEEE Trans. Image Process. 12 (8) (2003) 882–889.

[11] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, B. Du, Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding, Pattern Recognit. 48 (10) (2015) 3102–3112.

[12] T.F. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions, J. Vis. Commun. Image Represent. 12 (4) (2001) 436–449.

[13] BallesterColoma and Bertalmio, Marcelo and Caselles, Vicent and Sapiro, Guillermo and Verdera, Joan, Filling-in by joint interpolation of vector fields and gray levels, IEEE Trans. Image Process. 10 (8) (2001) 1200–1211.

[14] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, Patchmatch: a randomized correspondence algorithm for structural image editing, ACM Trans. Graph. 28 (3) (2009) 24.

[15] D. Ding, S. Ram, J.J. Rodriguez, Perceptually aware image inpainting, Pattern Recognit. 83 (2018) 174–184.

[16] D. Ding, S. Ram, J.J. Rodriguez, Image inpainting using nonlocal texture matching and nonlinear filtering, IEEE Trans. Image Process. 28 (4) (2019) 1705–1719.

[17] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: feature learning by inpainting, in: Proc. IEEE Comput. Vision Pattern Recognit., 2016, pp. 2536–2544.

[18] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Trans. Graph. 36 (4) (2017) 107.

[19] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, H. Li, High-resolution image inpainting using multi-scale neural patch synthesis, in: Proc. IEEE Comput. Vision Pattern Recognit., 2017, pp. 6721–6729.

[20] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, B. Catanzaro, Image inpainting for irregular holes using partial convolutions, in: Proc. Eur. Conf. Comput. Vision, 2018, pp. 85–100.

[21] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, M. Ebrahimi, Edgeconnect: Structure guided image inpainting using edge prediction, in: Proc. IEEE Int. Conf. Comput. Vision Workshops, 2019.

[22] Z. Yan, X. Li, M. Li, W. Zuo, S. Shan, Shift-net: Image inpainting via deep feature rearrangement, in: Proc. Eur. Conf. Comput. Vision, 2018, pp. 3–19.

[23] Y. Ren, X. Yu, R. Zhang, T.H. Li, S. Liu, G. Li, Structureflow: Image inpainting via structure-aware appearance flow, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 181–190.

[24] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Generative image inpainting with contextual attention, in: Proc. IEEE Comput. Vision Pattern Recognit., 2018, pp. 5505–5514.

[25] N. Wang, J. Li, L. Zhang, B. Du, Musical: Multi-scale image contextual attention learning for inpainting, in: Proc. Int. Joint Conf. Artif. Intell., 2019, pp. 3748–3754.

[26] Y. Ma, X. Liu, S. Bai, L. Wang, D. He, A. Liu, Coarse-to-fine image inpainting via region-wise convolutions and non-local correlation, in: Proc. Int. Joint Conf. Artif. Intell., 2019, pp. 3123–3129.

[27] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Free-form image inpainting with gated convolution, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 4471–4480.

[28] J. Li, F. He, L. Zhang, B. Du, D. Tao, Progressive reconstruction of visual structure for image inpainting, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 5962–5971.

[29] C. Xie, S. Liu, C. Li, M.-M. Cheng, W. Zuo, X. Liu, S. Wen, E. Ding, Image inpainting with learnable bidirectional attention maps, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 8858–8867.

[30] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, J. Luo, Foreground-aware image inpainting, in: Proc. IEEE Comput. Vision Pattern Recognit., 2019, pp. 5840–5848.

[31] C. Zheng, T.-J. Cham, J. Cai, Pluralistic image completion, in: Proc. IEEE Comput. Vision Pattern Recognit., 2019, pp. 1438–1447.

[32] H. Liu, B. Jiang, Y. Xiao, C. Yang, Coherent semantic attention for image inpainting, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 4170–4179.

[33] Y. Zeng, J. Fu, H. Chao, B. Guo, Learning pyramid-context encoder network for high-quality image inpainting, in: Proc. IEEE Comput. Vision Pattern Recognit., 2019, pp. 1486–1494.

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

[35] H. Grossauer, A combined PDE and texture synthesis approach to inpainting, in: Proc. Eur. Conf. Comput. Vision, 2004, pp. 214–224.

[36] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: Proc. Eur. Conf. Comput. Vision, 2016, pp. 694–711.

[37] L. Huang, W. Wang, J. Chen, X.-Y. Wei, Attention on attention for image captioning, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 4634–4643.

[38] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, Ccnet: criss-cross attention for semantic segmentation, in: Proc. IEEE Int. Conf. Comput. Vision, 2019, pp. 603–612.

[39] W. Liu, D. Xu, I.W. Tsang, W. Zhang, Metric learning for multi-output tasks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2) (2018) 408–422.

[40] F. Yang, K. Yan, S. Lu, H. Jia, X. Xie, W. Gao, Attention driven person re-identification, Pattern Recognit 86 (2019) 143–155.

[41] L. Song, C. Wang, L. Zhang, B. Du, Q. Zhang, C. Huang, X. Wang, Unsupervised domain adaptive re-identification: theory and practice, Pattern Recognit. 102 (2020) 107173(1–11).

[42] B. Chen, P. Li, C. Sun, D. Wang, G. Yang, H. Lu, Multi attention module for visual tracking, Pattern Recognit. 87 (2019) 80–93.

[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. Int. Conf. Learn. Represent., 2015.

[44] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proc. Comput. Vision Pattern Recognit., 2018, pp. 7132–7141.

[45] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, in: Proc. IEEE Comput. Vision Pattern Recognit., 2009, pp. 248–255.

[46] C. Doersch, S. Singh, A. Gupta, J. Sivic, A.A. Efros, What makes paris look like paris? ACM Trans. Graph. 31 (4) (2012) 101:1–101:9.

[47] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proc. IEEE Int. Conf. Comput. Vision, 2015, pp. 3730–3738.

[48] B. Zhou, À. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: a 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1452–1464.

Ning Wang 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 in the School of Computer Science, Wuhan University, China. Her research interests include deep learning and computer vision.

Sihan Ma received the B.S. degree in Software Engineering from the School of Computer Science, Wuhan University, China, in 2018. She is currently pursuing the MPhil degree in the UBTECH Sydney Artificial Intelligence Center, School of Computer Science, Faculty of Engineering and Information Technologies, The University of Sydney, Australia. Her research interests include deep learning and computer vision.

Jingyuan Li will receive the B.S. degree in Computer Science and Technology from the School of Computer Science, Wuhan University, China, in 2020. His research interests include deep learning and computer vision.

Yipeng Zhang received the B.S. degree in electronic engineering from the School of Electronic Information, Wuhan University, Wuhan, China, and the master's degree in electrical engineering from Syracuse University, Syracuse, NY, USA. He is currently pursuing the Ph.D. degree in the School of Computer Science, Wuhan University. His current research interests include deep learning, architecture-optimization on the FPGA, as well as the machine learning-oriented processor and accelerator.

Lefei Zhang received the B.S. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2008 and 2013, respectively. He is currently a professor with the School of Computer Science, Wuhan University. His research interests include pattern recognition, image processing, and remote sensing. Dr. Zhang is a reviewer of more than 30 international journals, including the IEEE TPAMI, TIP and TGRS.