Signal Processing: Image Communication 87 (2020) 115929

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Generative image inpainting via edge structure and color aware fusion

Hang Shao, Yongxiong Wang *, Yinghua Fu, Zhong Yin

School of Optical Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China

ARTICLE INFO

Keywords:
Deep learning
Image inpainting
Generative adversarial network
Content aware fill
Multi-map fusion

ABSTRACT

Very recently, with the widespread research of deep learning, its achievements are increasingly evident in image inpainting tasks. However, many existing methods fail to effectively reconstruct vivid contents and refine structures. In order to solve this issue, in this paper, a novel two-stage generative adversarial network based on the fusion of edge structures and color aware maps is proposed. In the first-stage network, edges with missing regions are employed to train an edge structure generator. Meanwhile, the input image with missing regions is transformed into a global color feature map after the content aware fill algorithm and a large kernel size Gaussian filtering. In the second-stage network, the image fused from the edge map and the color map is used as a label to guide the network to reconstruct the refined image. Qualitative and quantitative experiments conducted on multiple public datasets demonstrate that the method proposed in this paper has superior performance.

1. Introduction

Image inpainting by filling missing pixels or regions based on their context information to obtain visually realistic results is an important task in the field of computer vision. It is extensively applied in image editing, occlusion object detection, target tracking and intelligent aesthetics optimization. However, due to the complexity and diversity of natural images, inpainting work generally requires not only generating content pixels and texture patterns, but also guaranteeing the visual authenticity and perceptual plausibility of results. Therefore, it remains a challenging issue for images with complex semantics, high-resolution and missing large irregular regions. Early works that relied on computer graphics algorithms are mainly divided into two categories: approaches using pixel expansion or texture matching techniques [1–5] and approaches using large external database-driven techniques [6]. However, it is difficult for these approaches to capture global structures and refine details of images. While the requirement of high-resolution can be met, the semantic information of the inpainting result is inaccurate.

The development of deep learning and convolutional neural networks (CNNs), especially the emergence of generative adversarial networks (GANs), has extremely inspired recent research [7–11]. In the deep network architecture, high-level visual semantic recognition and low-level pixel synthesis are trained together to encourage the network to generate meaningful contents from missing regions. However, over-smoothing, blurring and artifact regions frequently present in inpainting results. To address this issue, some two-stage architecture networks are proposed [12–15]. Yu et al. [12] divide the network into a coarse recovery module and a refine completion module. On this basis, Sagong et al. [13] improve the network to a parallel architecture. However, since ground truth images are used as labels in the initial phase of these methods, high-frequency textures and irrelevant details in ground truth images will mislead the reconstruction process of contents. Nazeri et al. [14] adopt the edge map as the condition for the inpainting network, and there are similar methods in [10,11]. These methods can eliminate high-frequency non-critical textures to a certain extent. However, a lot of meaningful information will be discarded during the edge detection process, which will result in a lack of vividness in the generated content. Ren et al. [15] achieve relatively favorable effects using texture structure images as the guidance. However, structures with similar semantics may have different refine contents, which makes the final inpainting process difficult.

In this paper, a novel two-stage generative network for image inpainting is proposed. The network is composed of an edge structure generation module and a refine content reconstruct module. To reconstruct meaningful contents, we use the fusion image of the edge structure map and the color aware map as the label of the refine content reconstructor. For an input corrupted image, first, we perform the Canny edge detection and a content aware fill (CA) algorithm on it, respectively. Then, in our first-stage network, a complete edge map will be generated based on the corrupted edge. Meanwhile, we perform the global Gaussian blurring with a large kernel size on the color image filled by the CA algorithm. The edge detection can preserve the sharp edge of the image, while the CA algorithm can preserve the color information of the global content. After that, we fuse the edge map and the color map. Then, we use the fusion image as the label

* Corresponding author.
E-mail addresses: 932390809@qq.com (H. Shao), wyxiong@usst.edu.cn (Y. Wang), janeat9902@gmail.com (Y. Fu), yinzhong@usst.edu.cn (Z. Yin).

https://doi.org/10.1016/j.image.2020.115929
Received 15 January 2020; Received in revised form 10 April 2020; Accepted 21 June 2020
Available online 27 June 2020

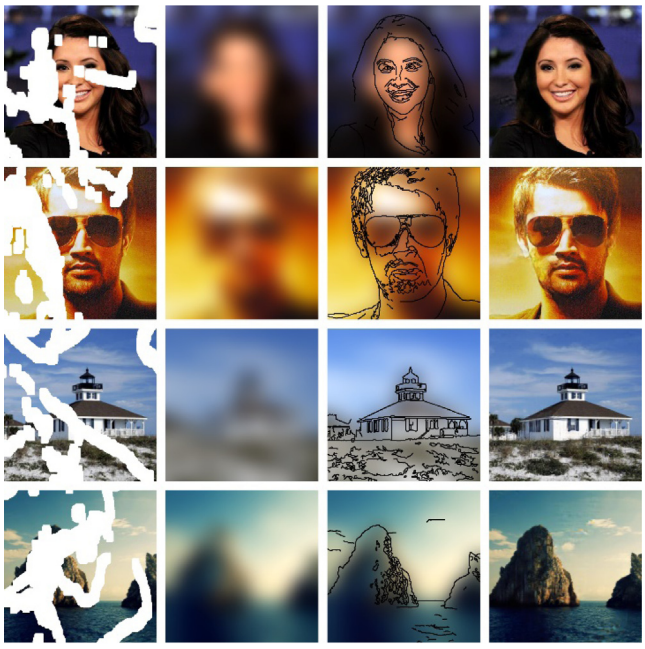

Fig. 1. Illustration of the inpainting task. Given an image with missing regions (a), we employ the CA algorithm and the Gaussian blur to capture its color feature map (b). Then, the color map and the edge map (c) are fused together to guide the result (d) to be more realistic.

of the second-stage network and obtain the globally generated image. Then, we transplant the region of the generated image corresponding to the missing region of the corrupted image into the corrupted image. Finally, we perform a progressive degradation (PD) operator on the transplanted boundary to increase the consistency of boundary pixels. Unlike the StructureFlow method [15] which applies the relative total variation measure (RTV) model [16] to extract image structure features and remove high-frequency textures, we use edge and color fusion images as the guidance. In the StructureFlow, first, for images with large irregular missing regions, it is much more difficult to recover three-channel structures based on smooth color maps than to recover binary edges based on edge maps in a single-stage network. Second, mistakes made by the first-stage network can easily mislead results of the second-stage network. Third, limitations of the RTV algorithm itself cause many important details to be ignored when applied to image inpainting tasks. Instead, our method focuses more on restoring visually realistic global contents, as shown in Fig. 1. At the same time, compared with the EdgeConnect method [14] and the progressive reconstruction network [11] which are only guided by edge maps, our network takes into account both edges and colors, so the inpainting result is more vivid. Moreover, for the method proposed by Xiong et al. [10] which applies a foreground salient detection to extract edges of foreground objects. Their network needs to invoke three additional multi-layer encoder–decoder modules, which will greatly increase the cost of training and testing. Furthermore, general salient detection algorithms are not suitable for extracting global edge features of images. Using such methods for image inpainting will cause many useful background information to be lost. Moreover, we use a contextual attention U-Net architecture as the backbone of the first-stage network generator to recover missing edges more accurately. The discriminator in the first-stage employs the VGG-19 architecture, and an improved patch matching loss function is introduced. For the generator of the second-stage network, a U-Net architecture with dilated convolutional layers is used to ensure that color information can be captured more comprehensively and contents can be restored more finely. For the second-stage network discriminator, the VGG-16 architecture is adopted, and we introduce an improved joint loss based on the adversarial loss, which includes perceptual loss, novel color reconstruction (CR) loss and structural similarity (SSIM) loss.

We evaluate the proposed method on multiple publicly available datasets: Places2 [17], CelebA [18], Facade [19], Paris StreetView [20] and Oxford Building [21]. We compare the performance of our model qualitatively and quantitatively with existing state-of-the-art methods. In addition, we conduct ablation studies to verify our hypotheses and modifications. The primary contributions of our paper are summarized as follows:

(1) A novel edge structure and color aware fusion label is introduced into the two-stage generative adversarial network to guide image inpainting more intelligently.
(2) Improved joint loss functions are introduced to train the multi-stage model more effectively.
(3) A pixel progressive degradation operator is designed to increase the consistency of the boundary and eliminate artifacts of the final image.
(4) Experiments on multiple publicly available datasets demonstrate that our method can obtain competitive inpainting effects.

2. Related work

2.1. Pixel filling

In natural scenes, when facing some objects that are partially occluded or not totally visible, our perception system can naturally reconstruct their hidden parts. This is due to the ability of "amodal completion" [22] in the human visual system. Therefore, even as early as the classical era, artists used brushes to restore corrupted paintings [23]. In recent years, with the development of digital image processing technology, researchers began to apply matrix theories and algorithms to various attempts to solve image disocclusion problems. Efros et al. [1] propose a non-parametric texture synthesis method based on the Markov random field model, which is an early representative of exemplar-based inpainting approaches. Similarly, there is the Graph-Cut method of Kwatra et al. [3]. However, these greedy algorithm-based methods are easily affected by the filling proceed. Bertalmio et al. [2] design a pixel expansion method based on the Navier–Stokes equation to avoid being restricted by the above factor. On this basis, Criminisi et al. [4] improve the sampling model to better preserve the structure information. However, these approaches typically merely fill small or tiny regions in the image, such as stains, lines and scratches. In contrast, the PatchMatch based on the random search algorithm proposed by Barnes et al. [5] can fill larger holes and edit structured images. However, these approaches rely on low-level pixels. They are ineffective for complex structures or high-level semantics, and they cannot generate new objects that do not exist in the original image. To compensate these shortcomings of patch-based approaches, Hays et al. [6] propose an image inpainting method invoking large external databases. They assume that regions surrounded by similar contexts may also have cognate contents. They search the external database for the image most similar to the input corrupted image. Then, they copy corresponding regions from the matching image and transplant these regions into the target image. However, when there are no suitable samples in the database, the approach will lead to inpainting errors.

2.2. Image generation

The CNN is trained on ImageNet [24], a large dataset with more than 14 million labeled images, has been proven to be robust in learning high-level visual semantic features of target images. The stacked deep convolutional network with vast amounts of hidden layers can be trained by massive data to capture the mapping of nonlinear complex relationships between different samples. This is quite in conformity

with our scheme to implement semantic inpainting based on the image content.

Pathak et al. [7] propose a context encoder network based on a encoder–decoder CNN, where the encoder contingent is a series of layer-by-layer sub-sampling, and the decoder contingent is the opposite to the former. By training with constraints of the adversarial loss and Euclidean distance, new pixels can be generated from locally missing regions. Yang et al. [8] take the low-resolution generation result of the context encoder network as content constraints and use a multi-scale neural network to increase image texture details. Iizuka et al. [25] split the adversarial loss into a global adversarial loss and a local adversarial loss to ensure that the generated region are semantically coherent with the original region. Yan et al. [26] synchronize patch copying and pixel filling by adding a special shift connection layer to the U-Net. It turns out that these methods can generate plausible new contents in highly structured images, such as scenes, objects, buildings, landscapes and portraits. Although Lempitsky et al. [27] prove that the hidden variable deep prior network constructed with random weights is also suitable for image inpainting tasks. However, this kind of network that can achieve image inpainting without using massive images for pre-training should be classified as a novel exemplar-based method that uses convolution kernel sampling instead of mainstream deep learning. Besides, it is similar to most exemplar-based methods, which is only suitable for image enhancement. For images with large missing regions, the effect of the deep prior method is far less robust than GAN. At the same time, in fields such as image style translation [19], image interactive editing [28], image colorization [29], image domain translation [30] and image super-resolution reconstruction [31], GAN-based approaches are also widely implanted.

3. The proposed approach

The framework of the proposed generative inpainting network is shown in Fig. 2. Our network consists of two stages: the edge structure generation network G_e and the refine content reconstruct network G_c. These two stages follow the deep adversarial network, that is, the backbone of each stage has a generator and a discriminator.

3.1. Edge structure generator

The main challenge of image inpainting tasks is to generate visually realistic contents for missing regions. Therefore, we adopt strategies of "coarse first, refine behind" [12] and "line first, color behind" [14]. First, an edge structure generation module is designed to recover the global edge of the corrupted image. Let I_{gt} be the ground truth image, and the grayscale map of I_{gt} is denoted by I_{gray}. Then, we perform a Gaussian filter with 3×3 kernel size and the Canny edge detection on I_{gray} to obtain the edge map E_{gt}. The Canny is a standard edge detection operator. Compared with the Sobel operator and the Laplacian operator, the anti-noise ability and robustness of the Canny operator are tested. Therefore, these series of operations can effectively remove the high-frequency noise of corrupted images while maintaining sharp edges, and obtain the objective representation that can reflect the image structure.

The generator of the edge structure network adopts a contextual attention U-Net [32], which is composed of a series of sub-sampling encoders and up-sampling decoders. To effectively capture features, we set the convolution kernel to 3×3, the sampling step size to 2, and the activation function to rectified linear unit (ReLU). The input of the generator G_e is the masked edge map $\tilde{E}_{mask}$

$$\tilde{E}_{mask} = E_{gt} \odot (1 - M) \tag{1}$$

where M represents the binary mask of the input corrupted image (1 represents the missing region, and 0 represents the background region), and $\odot$ represents Hadamard product. The processing of the generator G_e is expressed as

$$E_{pred} = G_e\left(\tilde{E}_{mask}, M\right) \tag{2}$$

where E_{pred} is the predicted edge map.

The discriminator of the edge structure network applies the VGG-19 which has been pre-trained on ImageNet. We use E_{gt} and E_{pred} as the input of the discriminator D_e to differentiate whether the edge map is real. In the network training process, the objective function is composed of three parts: adversarial loss, ℓ_1 loss and patch matching loss. The adversarial loss $\mathcal{L}^e_{adv}$ is defined as

$$\mathcal{L}^e_{adv} = \mathbb{E}\left[\log\left(1 - D_e\left(G_e\left(\tilde{E}_{mask}, M\right)\right)\right)\right] + \mathbb{E}\left[\log D_e\left(E_{gt}\right)\right] \tag{3}$$

For the ground truth image, the discrimination result is real, and for the generated image, the discrimination result is fake. After that, the generator and the discriminator each update parameters. These processes are similar to finding the Nash equilibrium point in a zero-sum non-cooperative game. ℓ_1 loss $\mathcal{L}^e_{\ell_1}$ is defined as the mean absolute error between the ground truth image and the predicted image

$$\mathcal{L}^e_{\ell_1} = \left\| E_{pred} - E_{gt} \right\|_1 \tag{4}$$

The patch matching loss $\mathcal{L}^e_{PM}$ is similar to the function proposed by Johnson et al. [33], which stabilizes the training process by comparing activation function maps in intermediate layers of the discriminator. $\mathcal{L}^e_{PM}$ is defined as

$$\mathcal{L}^e_{PM} = \mathbb{E}\left[\sum_i \frac{1}{N_i}\left\| D^{(i)}_e\left(E_{gt}\right) - D^{(i)}_e(E_{pred})\right\|_1\right] \tag{5}$$

where i is the number of discriminator convolutional layers, $D^{(i)}_e$ is the activation function map of the ith layer of the discriminator (in our architecture, $D^{(i)}_e$ means ReLU1_1, ReLU2_1, ReLU3_1, ReLU4_1 and ReLU5_1 in the VGG-19), and N_i is the number of elements in the ith activation function map of the discriminator. The joint objective function of the edge structure network is

$$\min_{G_e}\max_{D_e} \mathcal{L}^e = \lambda^e_{adv}\mathcal{L}^e_{adv} + \lambda^e_{\ell_1}\mathcal{L}^e_{\ell_1} + \lambda^e_{PM}\mathcal{L}^e_{PM} \tag{6}$$

where λ^e_{adv}, $\lambda^e_{\ell_1}$ and λ^e_{PM} are hyperparameters. After our experiments, λ^e_{adv}, $\lambda^e_{\ell_1}$ and λ^e_{PM} are set to 1, 5 and 10, respectively.

3.2. Refine content reconstructor

The refine content reconstructor uses the fusion image of the edge map and the color map as the label, as shown in Fig. 3. In this process, the edge structure map E_{comp} is stitched by E_{pred} and E_{gt}, which is expressed as

$$E_{comp} = E_{pred} \odot M + E_{gt} \odot (1 - M) \tag{7}$$

Meanwhile, for the corrupted image $\tilde{I}_{mask} = I_{gt} \odot (1 - M)$, relevant research [12] proves that the existing context information is essential for subsequent works. To make ample utilize of the existing context information, first, we fill the corrupted image. Considering the computational complexity, we use a texture-based filling method to replace the structure-aware appearance flow model that needs training [15]. However, traditional texture-based filling methods also face many limitations. For example, the receptive field of the Gaussian sampling-based method is too small to be suitable for large region filling; the noise resistance of the partial differential-based method is relatively poor; the visual coherence of the radial basis-based method will be reduced. Therefore, the CA algorithm based on the PatchMatch [5] is used in this paper to fill the corrupted image $\tilde{I}_{mask}$, and the filled image is expressed as F_{CA}. The CA algorithm utilizes randomly sampled pixels from the existing region to the fill missing region, which can validly eliminate boundary artifacts. However, the CA algorithm is difficult to capture semantics. If the second-stage network directly uses the filled image

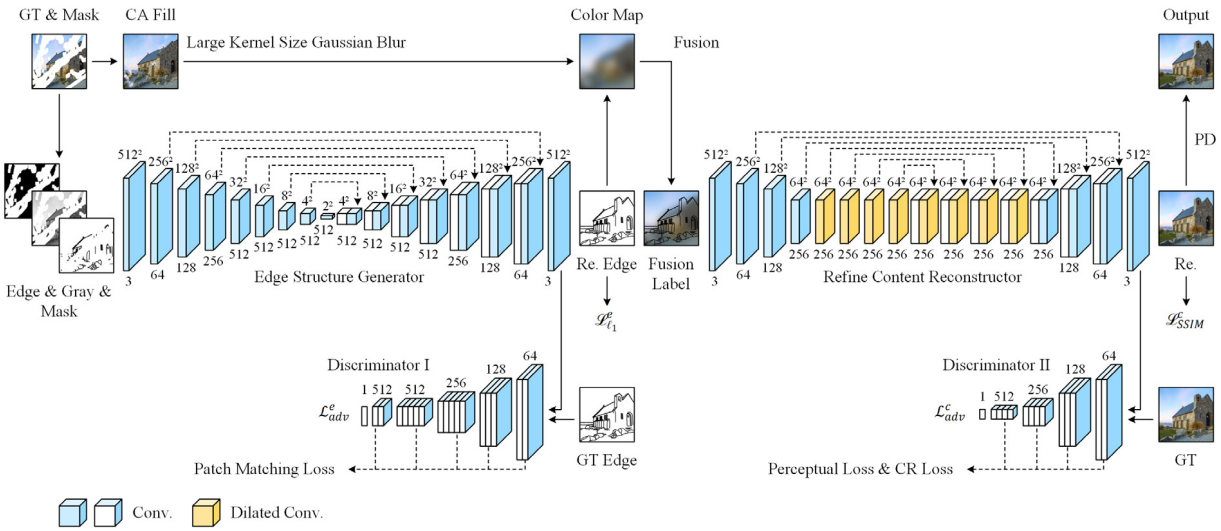

Fig. 2. The framework of the proposed network. First, an edge structure generator is used to perform global edge completion. Then, a refine content reconstructor is adopted to generate images with realistic details. We employ the fusion image of the edge map and the color map as the label of the refine content reconstructor. We design a progressive degradation operator in the output stage to eliminate pixel artifacts.

Fig. 3. Illustration of the fusion label in this paper. Given an image (a) with missing regions (b), we fill the image with the CA algorithm (c). Then, we use a large kernel size Gaussian blur to obtain the global color map (d) of the filled image. After that, the color map is fused with the edge map (e) generated by the first-stage network to obtain the label (f) of the second-stage network.

F_{CA} as the label, the inaccurate region will mislead the final output. Therefore, a large kernel size Gaussian blur is introduced to filter F_{CA} and obtain the global color map F_{color}. Taking experimental images of 512×512 in this paper as the example, we set the Gaussian kernel size to 49×49. Although the semantic of the region filled by the CA algorithm is not accurate. However, since the color difference between the missing region and the surrounding region is not large, and there is blessing from the aware metric of the PatchMatch. Therefore, above operations can accurately reconstruct the global color of the image. Then, we fuse F_{color} and the binary edge map E_{comp} with the same resolution (1 represents the edge map and 0 represents the background region) to obtain the fusion image I_{fuse}

$$I_{fuse} = F_{color} \odot \left(1 - E_{comp}\right) \tag{8}$$

The generator of the refine content reconstruction module adopts the U-Net as the backbone. In order to expand the receptive field, we replace eight standard convolutional layers in the U-Net with dilated convolutional layers. Moreover, in experiments of this paper, we set the convolution kernel to 3×3, the step of sub-sampling layers and up-sampling layers to 2, the dilation rate to 2, and the activation function to ReLU. For the generator G_c, we take I_{fuse} as the input and the refine content image I_{pred} predicted by the model as

$$I_{pred} = G_c\left(I_{fuse}\right) \tag{9}$$

The discriminator of the refine content reconstruction module applies the VGG-16 which has been pre-trained on ImageNet. For the discriminator D_c, we use I_{gt} and I_{fuse} as the input and introduce a joint function which includes adversarial loss, perceptual loss, CR loss and SSIM loss. The adversarial loss $\mathcal{L}^c_{adv}$ is

$$\begin{aligned}\mathcal{L}^c_{adv} &= \mathbb{E}\left[\log\left(1 - D_c\left(G_c\left(I_{fuse}\right)\right)\right)\right] \\ &\quad + \mathbb{E}\left[\log D_c\left(I_{gt}\right)\right]\end{aligned} \tag{10}$$

The perceptual loss is similar to the parch matching loss in the edge structure generation network, it is expressed as $\mathcal{L}^c_{perc}$

$$\mathcal{L}^c_{perc} = \mathbb{E}\left[\sum_j \frac{1}{N_j}\left\|D^{(j)}_c\left(I_{gt}\right) - D^{(j)}_c\left(I_{fuse}\right)\right\|_1\right] \tag{11}$$

where j is the number of discriminator convolutional layers, $D^{(j)}_c$ is the activation function map of the jth layer of the discriminator (in our paper, $D^{(j)}_c$ means ReLU1_1, ReLU2_1, ReLU3_1 and ReLU4_1 in the VGG-16, respectively), and N_j is the number of elements in the jth activation function map of the discriminator. Meanwhile, these activation function maps are also used to calculate the CR loss, which is the covariance between activation function maps. Sajjadi et al. [34] prove that the similar architecture can effectively eliminate checkerboard artifacts in results. Given a characteristic map with the size of $H_k \times W_k \times C_k$, the CR loss $\mathcal{L}^c_{CR}$ is expressed as

$$\mathcal{L}^c_{CR} = \mathbb{E}\left[\left\|T_k\left(I_{gt}\right) - T_k\left(I_{fuse}\right)\right\|_1\right] \tag{12}$$

where T_k is the Gram matrix of $C_k \times C_k$ constructed according to the activation function map $D^{(k)}_c$. In addition, we apply the SSIM loss to

replace the ℓ_1 loss used in traditional methods. The SSIM loss $\mathcal{L}^c_{SSIM}$ is expressed as

$$\mathcal{L}^c_{SSIM} = 1 - \frac{\left[2\mu_g\mu_f + (0.01\varepsilon)^2\right]\left[2\sigma_{gf} + (0.03\varepsilon)^2\right]}{\left[\mu_g^2 + \mu_f^2 + (0.01\varepsilon)^2\right]\left[\sigma_g^2 + \sigma_f^2 + (0.03\varepsilon)^2\right]} \tag{13}$$

where μ_g is the average value of I_{gt}, μ_f is the average value of I_{fuse}, σ_{gf} is the covariance of I_{gt} and I_{fuse}, σ_g^2 is the variance of I_{gt}, σ_f^2 is the variance of I_{fuse}, and ε is the dynamic range of image pixels. Our joint loss function is

$$\min_{G_c}\max_{D_c}\mathcal{L}^c = \lambda^c_{adv}\mathcal{L}^c_{adv} + \lambda^c_{perc}\mathcal{L}^c_{perc} + \lambda^c_{SSIM}\mathcal{L}^c_{SSIM} + \lambda^c_{CR}\mathcal{L}^c_{CR} \tag{14}$$

where λ^c_{adv}, λ^c_{perc}, λ^c_{CR} and λ^c_{SSIM} are hyperparameters. For our experiments, λ^c_{adv}, λ^c_{perc}, λ^c_{SSIM}, and λ^c_{CR} are set to 1, 1, 2 and 200, respectively.

After getting the predicted color image I_{pred}, we design a progressive degradation operator to process it and obtain the final result image $\hat{I}$

$$\hat{I} = I_{pred} \odot \tilde{M} + I_{gt} \odot \left(1 - \tilde{M}\right) \tag{15}$$

where $\tilde{M}$ is the matrix obtained by performing the PD operator processing on the binary mask matrix M. $\tilde{M}$ is expressed as

$$\tilde{M} = \sum_{\varphi=1}^{\Phi} \operatorname{argmax}\left(\tilde{M}_p(x)\right) \tag{16}$$

where p represents the point in the mask matrix M, x represents the pixel value of the point p, Φ is the processing region width of the PD operator, which is set to 10 in our paper, and φ is the number of iterations. $\tilde{M}_p(x)$ is the sign function defined as

$$\tilde{M}_p(x) = \begin{cases} \dfrac{\Phi - \varphi}{\Phi}, & \text{if } y > x \\ x, & \text{else} \end{cases} \tag{17}$$

where y represents pixel values of the remaining 8 points in a 3×3 pane centered on p. This operator can effectively eliminate boundary artifacts and perfectly increase the pixel coordination at the missing region boundary.

4. Experiments

4.1. Implementation details

In order to verify the scientificity and effectiveness of the proposed method, we trained and validated our network on five publicly datasets: MIT Places2 [17], CUHK CelebA [18], CMP Facade [19], Paris StreetView [20] and Oxford Building [21]. Places2 contains 365 independent scene categories, and images in each category are unevenly distributed to mimic their different frequencies of occurrence in nature scenes. For Places2, we train our network with 1.8 million images on its standard dataset and 6.2 million images on its challenge dataset. CelebA is a highly structured face dataset with 202,599 images. Paris StreetView and Oxford Building are building datasets. Images in Oxford Building have the high-resolution of more than 1024×1024. We use CelebA, Paris StreetView and Oxford Building as the verification of our network in different scenarios. Facade is often used in image-to-image translation tasks, and we mainly invoke it for ablation studies. In addition, we use the irregular mask dataset provided by Liu et al. [35] to simulate the missing region of the corrupted image. The mask dataset provides 12,000 irregular mask images and classifies them based on the size of the entire image missing region.

We use the fusion image of the edge map and the color map as the label for our inpainting network. In Section 4.3, we discuss different results when training the network with different labels, and the impact of different Canny detection thresholds and Gaussian kernel parameters on label images and inpainting results.

For the training process of our model, we divide the process into two steps. First, we use edge maps to train the edge structure generation network. Then, we use fusion images to train the refine content reconstruct network. During these processes, our network uses 512×512 images as inputs, the batch size is set to 8, the optimizer is Adam [36], and the initial learning rate is set to 10^{-4}. During the testing phase, our network can process and reconstruct input images end-to-end. In addition, Miyato et al. [37] shows that the spectral normalization can stabilize the training process by reducing the weight matrix according to their respective largest singular values, thereby it can effectively limit the Lipschitz constant of the deep network to one. Both generators and discriminators can benefit from the spectral normalization by suppressing sudden changes in parameters and gradient values. Therefore, we apply the spectral normalization to the generator and the discriminator in our first-stage network. However, for our second-stage network, if the spectral normalization is embraced, the training time will be greatly increased. Nazeri et al. [14] prove that as the number of entries in the loss function increases, the network becomes stricter. Therefore, in our refine content reconstruct network, we do not employ the spectral normalization.

4.2. Comparative experiments

We apply our network proposed in this paper to perform quantitative and qualitative comparison experiments with several existing state-of-the-art methods, including PatchMatch-based content filling method [5], globally and locally consistent network (GL) [25], contextual attention network (GCA) [12], partial convolutional network (PConv) [35], EdgeConnect method [14] and StructureFlow method [15].

Since the current lack of objective evaluation indicators for image inpainting tasks, in order to measure inpainting results as justice as possible, we use two types of metrics: distortion metrics and perception metrics. For distortion metrics, we use mean absolute error (MAE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) as evaluation indicators. MAE represents the deviation of corresponding positions between the ground truth image and the completed image. PSNR is employed to measure the quality of the inpainting result. SSIM compares the similarity index between the ground truth image and the completed image in terms of the luminance, contrast and structure. For perception metrics, we apply the pre-trained Inception V3 model to the extract feature of the ground truth image and the generated image when calculating FID (Fréchet Inception Distance) score [38]. We perform statistical calculations on 18,250 validation images of Places2 dataset. Results are shown in Table 1. It can be seen that for PSNR indicator, our model has achieved a competitive advantage. For SSIM indicator, our model is only slightly lower than the StructureFlow when the mask occupies 20%–40% of the image frame, but the overall of our model remains at a high level. For MAE and FID, our model also demonstrates favorable performance.

Meanwhile, we make a series of qualitative comparisons with above methods. Experimental results are shown in Fig. 4. It can be seen that PatchMatch-based method cannot accurately capture image global semantics. There are artifacts in results of GL and PConv, which means that these two methods may be difficult to balance textures and structures in the generated result. EdgeConnect can capture and restore certain image edges, but for the color information, we fail to reproduce it effectively. StructureFlow can recover the image global structure, but results are not fine enough. While, our method is more competitive.

Moreover, we also compare our label with the label of the StructureFlow in Fig. 5. It can be seen that our method can focus more on the detail. Therefore, our network can balance the structure and the texture well, and can obtain more realistic and vivid results.

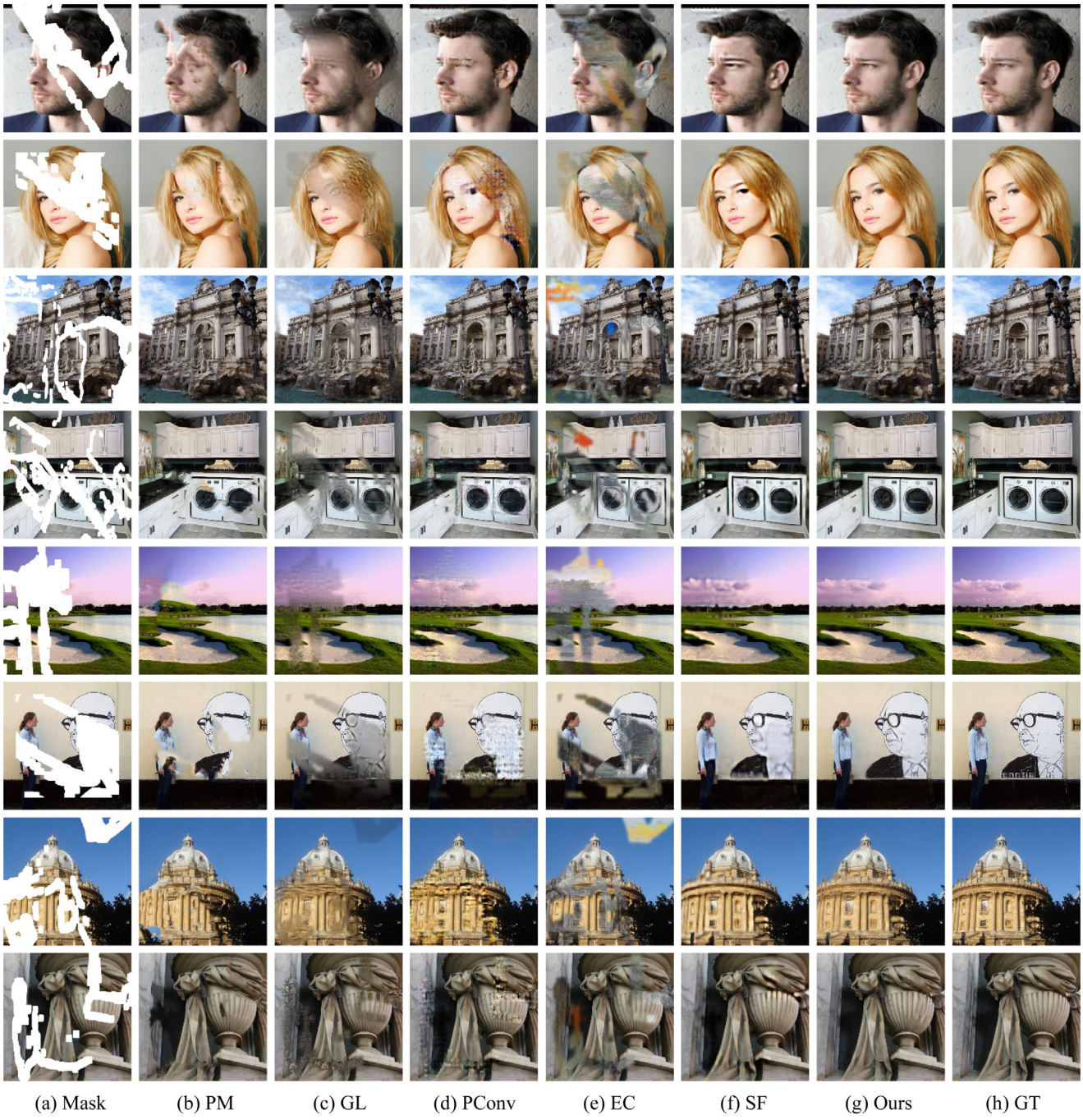

Fig. 4. Qualitative comparisons with existing state-of-the-art methods. (From top to bottom) Input corrupted images (a), results of PatchMatch (PM) [5] (b), results of GL [25] (c), results of PConv [35] (d), results of EdgeConnect (EC) [14] (e), results of StructureFlow (SF) [15] (f), results of our method (g), and ground truth images (GT) (h).

4.3. Ablation studies

In this section, we analyze the contribution of each map in the fusion label to the final inpainting performance from two perspectives: color information and edge structure.

First, we assume that the color information is important for image inpainting tasks. Therefore, we use the color map as one of components of the fusion label. In order to test this hypothesis, we remove the filling process of the CA algorithm and the fusion label, and only using the edge structure map generated by the first-stage network as the guidance for the second-stage network. Results are shown in Fig. 6(b). It can be seen that the absence of the color information will cause some deviations or errors between the restored color and the original color. After that, we turn our attention to the edge structure information. We believe that the edge map can effectively represent the image structure. To verify this conjecture, we use ground truth images as the label to train our network. Results are shown in Fig. 6(c). It can be seen that if the edge information is not used, the effect of the restored image will decrease. Relevant quantitative results are shown in Table 2.

However, how to accurately obtain the information of colors and edges is the key to the research problem. We find that inpainting becomes difficult if too many edges are retained, because the edge structure generator need to recover more information. In addition, the same problem exists in selecting the kernel size of the Gaussian blur. If the kernel size is set too small, more artifacts will generated in the inpainting result. For the extraction of the edge information, we use δ to represent the controlled threshold of the Canny algorithm. We use the edge map detected from $\delta = 50$, $\delta = 100$ and $\delta = 150$

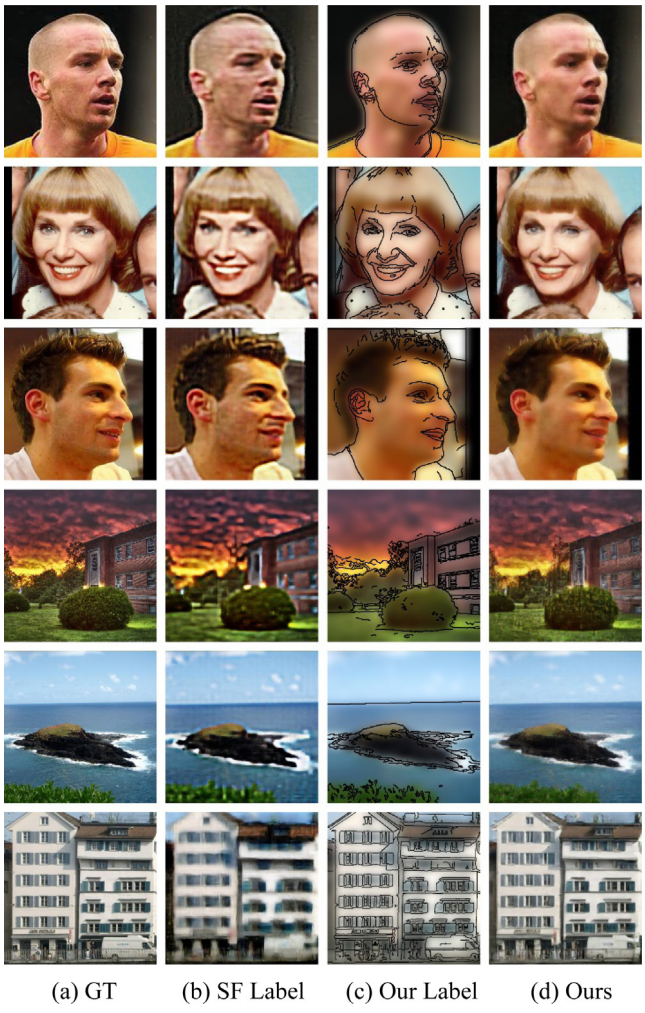

Fig. 5. Visual comparisons between StructureFlow [15] and our method. (From left to right) Ground truth images, global generated results of StructureFlow, our fusion labels, and global generated results of our method.

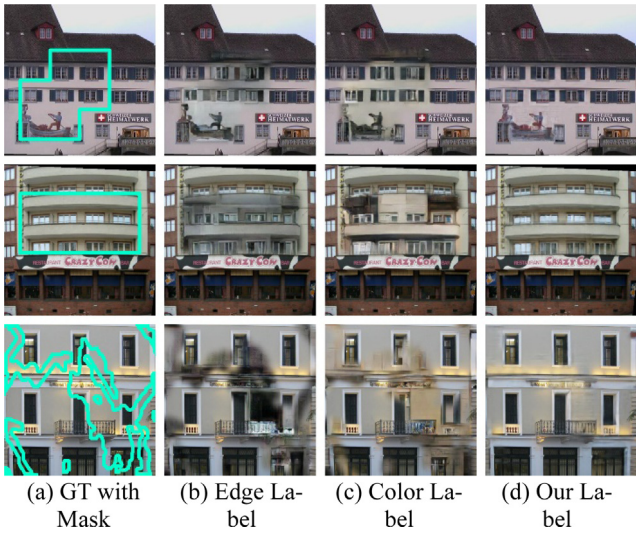

Fig. 6. Visual verifications of the color aware map and the edge structure information for inpainting results. (From left to right) Ground truth images (a), results with edge structure maps as labels (c), results with color smoothing maps as labels, and results with our fusion images as labels (d).

Table 1
Evaluation results of PatchMatch [5], GL [25], GCA [12], PConv [35], EdgeConnect [14], StructureFlow [15] and our method on Places2 dataset [17]. Since we failed to achieve an effective reproduction of EdgeConnect, we cite [15] for reporting on its results.

PSNR			
Mask	0%–20%	20%–40%	40%–60%
PatchMatch	29.67	23.08	18.02
GL	24.36	20.16	16.67
GCA	27.15	20.00	16.91
PConv	31.03	23.67	19.74
EdgeConnect	29.97	23.32	19.64
StructureFlow	32.03	25.22	21.09
Ours	**33.83**	**26.79**	**22.92**
SSIM			
Mask	0%–20%	20%–40%	40%–60%
PatchMatch	0.8726	0.7100	0.5027
GL	0.8280	0.6935	0.5450
GCA	0.9269	0.7613	0.5718
PConv	0.9070	0.7310	0.5325
EdgeConnect	0.9603	0.8600	0.6916
StructureFlow	0.9738	**0.9026**	0.7561
Ours	**0.9754**	0.8832	**0.7718**
MAE			
Mask	0%–20%	20%–40%	40%–60%
PatchMatch	0.0149	0.0329	0.0662
GL	0.0245	0.0493	0.0792
GCA	0.0205	0.0469	0.0743
PConv	**0.0147**	0.0325	0.0649
EdgeConnect	–	–	–
StructureFlow	0.0151	0.0327	0.0650
Ours	0.0161	**0.0318**	**0.0626**
FID			
Mask	0%–20%	20%–40%	40%–60%
PatchMatch	–	–	–
GL	–	–	–
GCA	4.8586	18.4190	37.9432
PConv	–	–	–
EdgeConnect	3.0097	7.2635	**19.0003**
StructureFlow	**2.9420**	7.0354	22.3803
Ours	3.0016	**7.0194**	22.9347

Table 2
Evaluation results of ablation studies for labels. We provide the statistical information for three models: the model trained with edge structure maps as labels (i.e. w/o color aware); the model trained with color smoothing maps as labels (i.e. w/o edge structure); our full model (i.e. fusion label).

	PSNR	SSIM
	CelebA	
w/o Color Aware Fill	17.99	0.7998
w/o Edge structure	19.26	0.8156
Fusion label	**29.16**	**0.9235**
	Facade	
w/o Color Aware Fill	21.73	0.8290
w/o Edge structure	23.25	0.8145
Fusion label	**31.41**	**0.9221**

to train the network, respectively. We find that the most satisfactory result is obtained when $\delta = 100$. For the extraction of the color information, we use ω to represent the controlled kernel size of the Gaussian blur. We use $\omega = 29$, $\omega = 49$, $\omega = 99$ and $\omega = 199$ to train the network, respectively. Finally, we find that when $\omega = 49$, we obtain the favorable result. Relevant quantitative results are listed in Table 3, and visualization results are shown in Fig. 7.

In this paper, we design a progressive degradation operator to increase the boundary pixel consistency of the inpainting image. In order to verify the effectiveness of the PD operator, we show the inpainting results with and without the PD in Fig. 8. It can be seen

Table 3
Evaluation results of ablation studies for Canny detection parameters and Gaussian kernel parameters. We provide the statistical information for six models in Paris StreetView and Oxford Building: the model trained with $\delta = 150$, $\delta = 50$, $\omega = 29$, $\omega = 99$, $\omega = 199$ and our parameters.

	PSNR	SSIM
$\delta = 150$	26.24	0.8236
$\delta = 50$	27.29	0.8291
$\omega = 29$	26.68	0.8287
$\omega = 99$	27.04	0.8287
$\omega = 199$	27.58	0.8329
Our parameters	**29.73**	**0.9011**

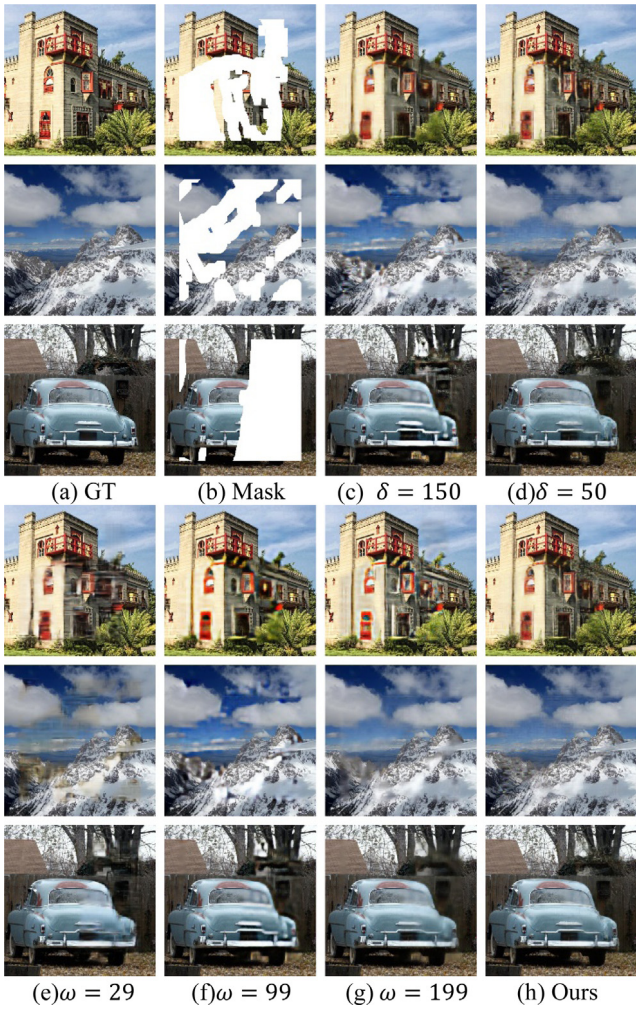

Fig. 7. Influence of parameters δ in the Canny detection and ω in the Gaussian blur. Given images (a) with mask (b), inpainting results for $\delta = 150$ (c), $\delta = 50$ (d), $\omega = 29$ (e), $\omega = 99$ (f), $\omega = 199$ (g) and parameters in our paper (h).

that our operator can effectively increase the pixel consistency of the boundary.

5. Conclusion

In this paper, we propose an effective two-stage generative network for image inpainting tasks. The inpainting process is divided into two sub-tasks: the fusion label generation network and the refine content reconstruction network. We verify experimentally that the fusion image of the edge structure map and the color aware map can represent the global structure information of the image well, and it plays important roles in inpainting tasks. Compared with existing state-of-the-art methods, our method can obtain competitive results. Furthermore, the generative network not only is very powerful in the image high-frequency detail inpainting, but also provides a powerful prior in terms of semantics and global structures. It can be useful for other applications such as denoising, retargeting, super-resolution and view/time interpolation. However, when the image scene is too complex, our approach may still introduce discontinuities or artifacts. In addition, the computing speed and the hardware capacity are still the bottleneck of our algorithm. Our goal is to solve or overcome these problems in the future work.

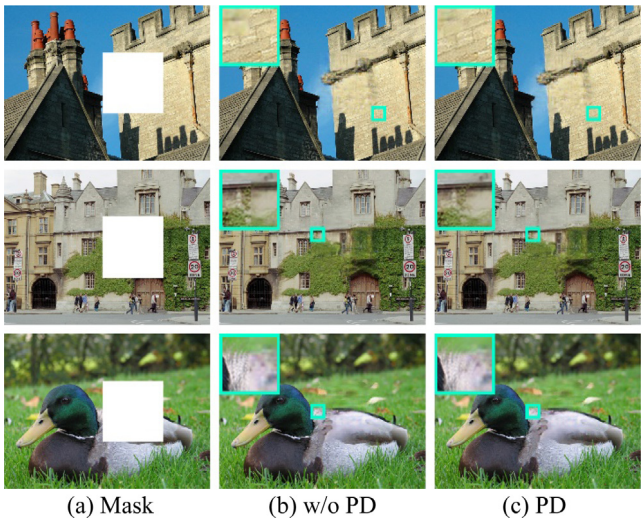

Fig. 8. Influence of our progressive degradation operator. (From left to right) Input corrupted images (a), results without our progressive degradation operator (w/o PD) (b), and results with our progressive degradation operator (PD) (c).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the National Natural Science Foundation of China under Grant 61673276.

References

[1] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling, in: IEEE International Conference on Computer Vision, 1999, pp. 1033–1038.
[2] M. Bertalmio, A.L. Bertozzi, G. Sapiro, Navier–Stokes, fluid dynamics, and image and video inpainting, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. 355–362.
[3] V. Kwatra, A. Schödl, I. Essa, G. Turk, A. Bobick, Graphcut textures: Image and video synthesis using graph cuts, ACM Trans. Graph. 22 (3) (2003) 277–286.
[4] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212.
[5] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, PatchMatch: A randomized correspondence algorithm for structural image editing, ACM Trans. Graph. 28 (3) (2009) 24–33.
[6] J. Hays, A.A. Efros, Scene completion using millions of photographs, ACM Trans. Graph. 26 (3) (2007) 4–10.
[7] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[8] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, H. Li, High-resolution image inpainting using multi-scale neural patch synthesis, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6721–6729.

[9] Q. Liu, S. Li, J. Xiao, M. Zhang, Multi-filters guided low-rank tensor coding for image inpainting, Signal Process., Image Commun. 73 (2019) 70–83.
[10] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, J. Luo, Foreground-aware image inpainting, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5840–5848.
[11] J. Li, F. He, L. Zhang, B. Du, D. Tao, Progressive reconstruction of visual structure for image inpainting, in: IEEE International Conference on Computer Vision, 2019, pp. 5962–5971.
[12] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Generative image inpainting with contextual attention, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5505–5514.
[13] M.C. Sagong, Y.G. Shin, S.W. Kim, S. Park, S.J. Ko, PEPSI: Fast image inpainting with parallel decoding network, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11360–11368.
[14] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, M. Ebrahimi, EdgeConnect: Generative image inpainting with adversarial edge learning, 2019, arXiv preprint arXiv:1901.00212.
[15] Y. Ren, X. Yu, R. Zhang, T.H. Li, S. Liu, G. Li, StructureFlow: Image inpainting via structure-aware appearance flow, in: IEEE International Conference on Computer Vision, 2019, pp. 181–190.
[16] L. Xu, Q. Yan, Y. Xia, J. Jia, Structure extraction from texture via relative total variation, ACM Trans. Graph. 31 (6) (2012) 139–148.
[17] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2017) 1452–1464.
[18] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.
[19] P. Isola, J.Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[20] C. Doersch, S. Singh, A. Gupta, J. Sivic, A.A. Efros, What makes Paris look like Paris? Commun. ACM 58 (12) (2015) 103–110.
[21] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[22] S. Masnou, J.M. Morel, Level lines based disocclusion, in: International Conference on Image Processing, 1998, pp. 259–263.
[23] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Annual Conference on Computer Graphics and Interactive Techniques, 2000, pp. 417–424.
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[25] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Trans. Graph. 36 (4) (2017) 107–120.
[26] Z. Yan, X. Li, M. Li, W. Zuo, S. Shan, Shift-Net: Image inpainting via deep feature rearrangement, in: European Conference on Computer Vision, 2018, pp. 1–17.
[27] V. Lempitsky, A. Vedaldi, D. Ulyanov, Deep image prior, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.
[28] B. Dolhansky, C.C. Ferrer, Eye in-painting with exemplar generative adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7902–7911.
[29] K. Uruma, K. Konishi, T. Takahashi, T. Furukawa, Colorization-based image coding using graph Fourier transform, Signal Process., Image Commun. 74 (2019) 266–279.
[30] S. Milz, M. Simon, K. Fischer, M. Popperl, Points2Pix: 3D point-cloud to image translation using conditional generative adversarial networks, 2019, arXiv preprint arXiv:1901.09280.
[31] D.F. Noor, Y. Li, Z. Li, S. Bhattacharyya, G. York, Multi-scale gradient image super-resolution for preserving SIFT key points in low-resolution images, Signal Process., Image Commun. 78 (2019) 236–245.
[32] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[33] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European Conference on Computer Vision, 2016, pp. 694–711.
[34] M.S. Sajjadi, B. Scholkopf, M. Hirsch, EnhanceNet: Single image super-resolution through automated texture synthesis, in: IEEE International Conference on Computer Vision, 2017, pp. 4491–4500.
[35] G. Liu, F.A. Reda, K.J. Shih, T.C. Wang, A. Tao, B. Catanzaro, Image inpainting for irregular holes using partial convolutions, in: European Conference on Computer Vision, 2018, pp. 85–100.
[36] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[37] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, 2018, arXiv preprint arXiv:1802.05957.
[38] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in: Advances in Neural Information Processing Systems, 2017, pp. 6626–6637.