PEPSI : Fast Image Inpainting with Parallel Decoding Network

Min-cheol Sagong[1], Yong-goo Shin[1], Seung-wook Kim[1], Seung Park[1], Sung-jea Ko[2]
Korea university
[1]{mcsagong, ygshin, swkim, spark}@dali.korea.ac.kr
[2]sjko@korea.ac.kr

Abstract

Recently, a generative adversarial network (GAN)-based method employing the coarse-to-fine network with the contextual attention module (CAM) has shown outstanding results in image inpainting. However, this method requires numerous computational resources due to its two-stage process for feature encoding. To solve this problem, in this paper, we present a novel network structure, called PEPSI: parallel extended-decoder path for semantic inpainting. PEPSI can reduce the number of convolution operations by adopting a structure consisting of a single shared encoding network and a parallel decoding network with coarse and inpainting paths. The coarse path produces a preliminary inpainting result with which the encoding network is trained to predict features for the CAM. At the same time, the inpainting path creates a higher-quality inpainting result using refined features reconstructed by the CAM. PEPSI not only reduces the number of convolution operation almost by half as compared to the conventional coarse-to-fine networks but also exhibits superior performance to other models in terms of testing time and qualitative scores.

1. Introduction

Image inpainting techniques have been widely researched [1–3, 5, 6, 10, 13, 16, 17, 19, 21, 25, 26, 28] for removing an unwanted object or synthesizing missing parts of an image in various applications such as photo editing, image-based rendering, and computational photography [15, 20, 28]. Image inpainting methods can be divided into two groups [28]. The diffusion-based and patch-based methods belong to the first group. The diffusion-based method propagates the pixel information from the existing regions of an image, *i.e.* background regions, to the missing regions, *i.e.* hole regions [2,5,19,28]. This method performs well on plain textures and small holes but often fails to fill in the complex hole region such as face and objects with the non-repetitive structures. In contrast to the diffusion-based method, the patch-based method samples patches from the background region and then pastes them into the hole region [22, 28]. Barnes *et al.* [1] proposed a fast approximate nearest neighbor patch search algorithm, called PatchMatch, which has shown notable results for image editing applications including image inpainting. PatchMatch, however, smoothly fills in the hole region without considering the visual semantics or the global structure of an image.

The second group is a generation-based method which applies the deep convolutional neural network (CNN) to predict structures for the hole regions [13, 16, 24]. Thanks to a decade of advances in CNNs, image inpainting methods adopting an encoder-decoder structure have achieved a significant progress [13, 24]. However, these methods often create an image with artifacts such as a blurry image and a distorted image. To cope with this problem, Pathak *et al.* [21] introduced a method called context encoder adopting the generative adversarial network (GAN) [7]. In this method, they utilize a combined loss, the l_2 pixel-wise reconstruction loss and adversarial loss, which helps the networks to generate a more natural image by minimizing the difference between the reference and inpainted image. However, this method has a limitation that it can fill only square holes at the center of an image.

Iizuka *et al.* [10] proposed an improved network structure which can extract features in wider receptive fields by employing the dilated convolution layers to complete hole regions effectively. In addition, they use two sibling discriminators: global and local discriminators. The local discriminator focuses on the inpainted region to distinguish local texture consistency while the global discriminator inspects if the result is coherent in a whole image. Yu *et al.* [28] have extended this work by using the coarse-to-fine network and the contextual attention module (CAM). The CAM learns the relation among background and foreground feature patches by computing the cosine similarity. To collect the background features involved with the missing region, this method needs the features at the missing region encoded from roughly completed images. Thus, they designed two-stage coarse-to-fine networks to produce an intermediate result of a roughly restored image. This

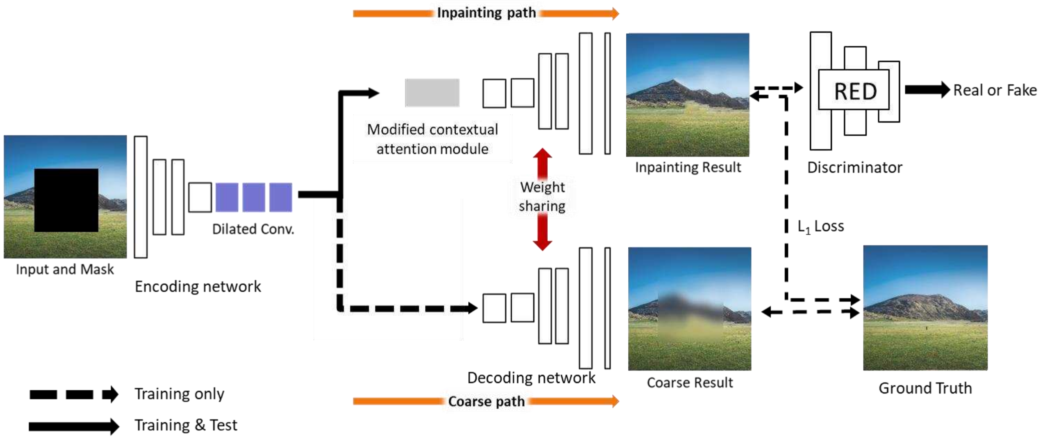


Figure 1. An architecture of PEPSI. The coarse path and inpainting path share their weights to improve each other. The coarse path is trained only with the ℓ_1 reconstruction loss while the inpaiting path is trained with both of ℓ_1 and adversarial loss

method shows a remarkable performance as compared with recent state-of-the-art inpainting methods; however, it requires considerable computational resources due to use of the two-stage network structure.

In this paper, we propose a novel parallel network called PEPSI: parallel extended-decoder path for semantic inpainting, which has the small number of operation by employing a single-stage encoder-decoder network to solve this problem. As shown in Figure 1, PEPSI extracts features via a single encoding network and generates a high-quality inpainted result via a single decoding network. To make a single shared encoding network handle two different tasks, feature generation both for a roughly completed result and for a high-quality result, we propose a joint learning method using a parallel decoding network which has coarse and inpainting paths. The coarse path produces a roughly completed result with which the encoding network is trained to predict features for the CAM. At the same time, the inpainting path creates a higher-quality inpainting result using the refined features reconstructed by the CAM. We also modify the CAM to use Euclidean distance instead of the cosine similarity to learn the relationship of patches more suitably.

We conduct extensive experiments to demonstrate that our method outperforms conventional methods on various dataset such as Celeb-a [11, 18], place2 [30], and Imagenet [14]. We use both of the random square mask and freeform mask mimicking human brushings. The experimental results indicate that the proposed method not only exhibits superior performance compared to the conventional ones but also significantly reduces the computational time.

In summary, in this paper we present:

- A novel generative network that improves the inpainting performance while reducing the number of computational resources by unifying cascade network of the coarse-to-fine network and modifying the CAM.
- A novel discriminator distinguishing image regions separately which is more suitable in real user applications.

2. Preliminaries

2.1. Generative adversarial networks

The GAN was first introduced by Goodfellow *et al.* [7] for the image generation. In the GAN, two networks are simultaneously trained: a generative network, G, is trained to create a new image which is indistinguishable from real images, whereas a discriminative network, D is trained to differentiate between real and generated images. This relation can be considered as a two-player min-max game in which G and D compete. To this end, the G (D) tries to minimize (maximize) the loss function, *i.e.* adversarial loss, as follows:

$$\min_G \max_D E_{x\sim P_{\text{data}}(x)}[\log D(x)] + E_{z\sim P_{z(z)}}[\log(1-D(G(z)))], \tag{1}$$

where z and x denote a random noise vector and a real image sampled from the noise $P_z(z)$ and real data distribution $P_{\text{data}}(x)$, respectively. Recently, the GAN have been applied to several semantic inpainting techniques [10, 21, 28] in order to complete the hole region naturally.

2.2. Coarse-to-fine network

Yu *et al.* [27, 28] proposed a novel image inpainting framework consisting of two networks: the coarse network and the refinement network. This two-stage network, called a coarse-to-fine network, performs a couple of tasks separately. First, it produces an initial coarse prediction with the coarse network, and refines the results by extracting features from the roughly filled prediction with the refinement network. To produce a higher-quality image inpainting with the generative network, the network should understand the

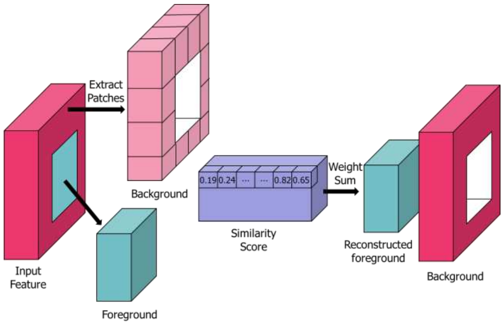


Figure 2. The illustration of the CAM. The conventional CAM reconstructs foreground patches by measuring the cosine similarities with background patches. In contrast, the modified CAM uses the Euclidean distance to compute similarity scores.

relation between background and hole regions. The refinement network learns the relation by using the CAM, which computes the cosine similarity between those regions. As shown in Figure 2, the CAM first divides features into a target foreground and its surrounding background and extracts 3 × 3 patches. Then, the similarity score $s_{(x,y),(x',y')}$ between the foreground patch at (x, y), $f_{x,y}$, and the background patch at (x', y'), $b_{x',y'}$, can be obtained as

$$s_{(x,y),(x',y')} = \left\langle \frac{f_{x,y}}{\|f_{x,y}\|}, \frac{b_{x',y'}}{\|b_{x',y'}\|} \right\rangle, \tag{2}$$

$$s^*_{(x,y),(x',y')} = \text{softmax}(\lambda s_{(x,y),(x',y')}), \tag{3}$$

where λ is a hyper-parameter for scaled *softmax*. By using $(s^*_{(x,y),(x',y')})$ as weights, the CAM reconstructs features of foreground regions by a weighted sum of background patches to learn the relation between them. The coarse-to-fine network shows outstanding a performance among state-of-the-arts image inpainting techniques.

3. Proposed Method

As described in Section 2.2, the CAM learns where to borrow or copy the feature information from known background feature patches to generate missing feature patches by computing the between-patch similarity. Thus, it is necessary to extract features from a roughly completed image, *i.e.*, the coarse result. The refinement network without the coarse result shows worse results than the full coarse-to-fine network as shown in Table 1 and Figure 3 (these results were obtained by training the refinement network using raw masked images as an input). This means that, if the coarse feature of the hole region is not encoded well, the CAM produces the missing features using unrelated feature patches, yielding contaminated results as shown in Figure 3(d). In other words, the coarse-to-fine network must pass through a two-stage encoder-decoder network which needs massive computational complexity, especially on high-resolution images.

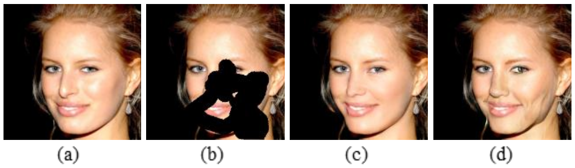


Figure 3. The toy example about coarse network. (a) The ground truth (b) The masked input image (c) The result from the coarse-to-fine network (d) The result without the coarse result.

	Square mask		Free-form mask		Time
	PSNR	SSIM	PSNR	SSIM	
GatedConv [27]	24.67	0.8949	27.78	0.9252	21.39ms
GatedConv *	23.50	0.8822	16.35	0.9098	14.28ms

Table 1. The toy example about coarse network. * means a model without coarse results.

3.1. Architecture of PEPSI

As shown in Figure 1, our proposed network, PEPSI, unifies the two-stage cascade network of the coarse-to-fine network into a single-stage encoder-decoder network. The PEPSI consists of a single shared encoding network and a parallel decoding network which has coarse and inpainting paths. The encoding network is jointly learned for extracting features from input images with hole regions as well as completing the missing features without the coarse result. As listed in Table 2, our feature encoding network is composed of a series of 3 × 3 convolution layers. In this network, we use a 5 × 5 kernel in the first convolution layer to fully exploit the latent information in the input image. In addition, we employ the dilated convolution layers with different dilation rate in the last four convolution layers to extract the features with large receptive fields.

Table 3 shows a detailed architecture of the decoding network. In the proposed method, a parallel decoding network consists of two sibling paths: coarse and inpainting paths. The coarse path attempts to produce a roughly completed result from the encoded feature map. On the other hand, taking the encoded features as an input, the inpainting path first reconstructs the feature map by using the CAM. Then, the reconstructed feature map is decoded to generate a higher-quality inpainting result. By sharing the weight parameters of the two paths, we attempt to regularize the inpainting path of the decoding network. Moreover, two different paths employ the same encoded feature map as their input, and thus they compel the single encoder to generate valuable features for two different image generation tasks. Note that we employ only the inpainting path during tests, which substantially reduces the computational resources. In the proposed method, the coarse path is trained with the reconstruction L_1 loss explicitly, whereas the inpainting path is trained with the L_1 loss as well as the GAN losses. More detailed information will be described in Section 3.4.

Similar to traditional image inpainting networks [10, 21,

Type	Kernel	Dilation	Stride	Outputs
Convolution	5×5	1	1×1	32
Convolution	3×3	1	2×2	64
Convolution	3×3	1	1×1	64
Convolution	3×3	1	2×2	128
Convolution	3×3	1	1×1	128
Convolution	3×3	1	2×2	256
Dialated convolution	3×3	2	1×1	256
Dialated convolution	3×3	4	1×1	256
Dialated convolution	3×3	8	1×1	256
Dialated convolution	3×3	16	1×1	256

Table 2. Detail architecture of encoding network.

Type	Kernel	Dilation	Stride	Outputs
Convolution $\times 2$	3×3	1	1×1	128
Nearest Neighbor ($\times 2 \uparrow$)	-	-	-	-
Convolution $\times 2$	3×3	1	1×1	64
Nearest Neighbor ($\times 2 \uparrow$)	-	-	-	-
Convolution $\times 2$	3×3	1	1×1	32
Nearest Neighbor ($\times 2 \uparrow$)	-	-	-	-
Convolution $\times 2$	3×3	1	1×1	16
Convolution (Output)	3×3	1	1×1	3

Table 3. Detail architecture of decoding network. The output layer consists of a convolution layer clipped value to the [-1, 1].

27, 28], the proposed network takes a masked image and a binary mask indicating the background regions as input pairs. The masked image includes holes with the variable numbers, sizes, shapes, and locations randomly sampled during every iteration. In terms of layer implementations, we use mirror padding for all convolution layers and employ the exponential liner unit (ELU) [4] as an activation function instead of ReLU except the last layer. Also, we utilize $[-1, 1]$ normalized image with 256×256 pixels as an input image of the network, and generate an output image with the same resolution by clipping the output values into $[-1, 1]$ instead of using $\tanh$ functions.

3.2. Modified CAM

The conventional CAM [28] measures similarity scores by applying the cosine similarity. However, normalizing the feature patch vector in (2) can distort the semantic feature representation. Thus, we propose a modified CAM which directly measures distance similarity scores $(d_{(x,y),(x',y')})$ using the Euclidean distance. It is more suitable for a reconstruction because the Euclidean distance considers not only an angle between two vectors of feature patches but also magnitudes of them. Since the distance similarity scores are hard to be applied *softmax* having the output range of $[0, \infty)$, we define the truncated distance similarity scores, $\widetilde{d}_{(x,y),(x',y')}$, as

$$\widetilde{d}_{(x,y),(x',y')} = \tanh\,(-(\frac{d_{(x,y),(x',y')} - m(d_{(x,y),(x',y')})}{\sigma(d_{(x,y),(x',y')})})), \tag{4}$$

where

$$d_{(x,y),(x',y')} = \|f_{x,y} - b_{x',y'}\|. \tag{5}$$

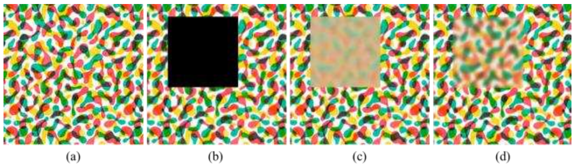


Figure 4. A comparison of the image reconstruction between the cosine similarity and the truncated distance similarity: (a) The original image, (b) masked image, (c) image reconstructed by using the cosine similarity and (d) image reconstructed by using the truncated distance similarity.

	square mask		free-form mask	
	PSNR	SSIM	PSNR	SSIM
Cosine similarity	25.16	0.8950	27.95	0.9218
Euclidean distance	25.57	0.9007	28.59	0.9293

Table 4. Comparison of the performance between the cosine similarity and the Euclidean distance applying on the PEPSI.

In (4), the truncated distance similarity score has limited values within $[-1, 1]$. It operates like a threshold which sorts out the distance score less than the mean value because $\tanh$ function changes rapidly across zero. It means that the truncated distance similarity score helps to divide background patches into two groups which are related to the foreground patch and not. We perform toy examples comparing the cosine similarity and the truncated distance similarity. We reconstruct the hole region by the weighted sum of existing image patches where the weights are obtained by using the cosine similarity scores or the truncated distance similarity scores. As can be seen in Figure 4, reconstruction applying the truncated distance similarity can collect more similar patches than the cosine similarity. Furthermore, we evaluate the results between PEPSI with conventional and modified CAMs to confirm the improvement of the modified CAM. As shown in Table 4, the modified CAM increases the performance as compared to the conventional CAM, which means that the modified CAM is more suitable to express the relationship between background and hole regions.

Similar to the conventional CAM, modified one also weigh them with scaled *softmax* and reconstruct the foreground patch by a weighted sum of background patches at last. Consequently, it supports the module to reconstruct foreground patches from a related patch vector group.

3.3. Region Ensemble Discriminator(RED)

PEPSI is learned based on the GAN, which consists of the generator and discriminator. In [27, 28], the conventional global and local discriminators aim at not only coherence in a whole image but also the local texture of hole region. However, the local discriminator can handle only the hole region with the fixed size of the square shape, while holes can be appeared with arbitrary locations, shapes, and sizes in real applications. Thus, it is hard to employ the

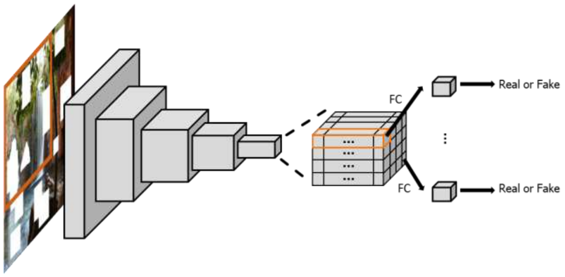


Figure 5. The overview of the RED. The RED aims to classify hole regions which may appear any region with any sizes in an image.

local discriminator to train the inpainting network for the irregular hole. To cope with this problem, we unify global and local discriminators into the region ensemble discriminator (RED), which is inspired by the region ensemble network [8] detecting a target object that appears anywhere in images by handling multiple feature regions individually. As depicted in Figure 5, the RED divides the feature of the last layer as a pixel-wise block and differentiates each feature is real or fake individually by fully-connected layers. Since the RED tries to classify each feature block which has different receptive fields separately, it assumes different image regions are real or fake individually. In contrast to the local discriminator, the RED can handle the various hole regions that may appear anywhere in images of any sizes. A detailed architecture of the RED is listed in Table 5.

3.4. Loss function

To train PEPSI, we jointly optimize two different paths: the inpainting path and the coarse path. For the inpainting path, we adopt the GAN [7] optimization framework in (1), which is described in Section 2.1. It is well known that, in the original GAN [7], the gradient of the generator can easily disappear, which yields unsatisfactory results of generated images. To solve this problem, motivated by [29], we employ the adversarial loss functions of the generator and the discriminator using the hinge loss and the spectral normalization, which are defined as

$$L_G = -E_{x\sim P_{X_\text{i}}}[D(x)], \tag{6}$$

$$\begin{aligned} L_D = {} & E_{x\sim P_Y}[\min(0, -1 + D(x))] \\ & -E_{x\sim P_{X_\text{i}}}[\min(0, -1 - D(x))], \end{aligned} \tag{7}$$

where P_{X_i} and P_Y denote the data distributions of inpainting results and input images. Since goal of image inpainting is not only to generate the natural hole filling but also to restore the missing part of the original image accurately, we add a strong constraint using ℓ_1 norm to (6) as follows:

$$L_G = \frac{\lambda_\text{i}}{N}\sum_{n=1}^{N}\|X_\text{i}^{(n)} - Y^{(n)}\|_1 - \lambda_\text{adv}E_{x\sim P_{X_\text{i}}}[D(x)], \tag{8}$$

Type	Kernel	Stride	Outputs
Convolution	5 × 5	2 × 2	64
Convolution	5 × 5	2 × 2	128
Convolution	5 × 5	2 × 2	256
Convolution	5 × 5	2 × 2	256
Convolution	5 × 5	2 × 2	256
Convolution	5 × 5	2 × 2	512
FC	1 × 1	1 × 1	1

Table 5. Detailed architecture of RED. After each convolution layer, except last one, there is a leaky-ReLU as the activation function. Every layer is normalized by a spectral normalization. The fully-connected layer is applied to every pixel-wise feature block.

where $X_\text{i}^{(n)}$ and $Y^{(n)}$ are the nth image pair of the generated image via the inpainting path and its corresponding original input image in a mini-batch, respectively, N is the number of image pairs in a mini-batch, and λ_i and λ_adv are hyper-parameters to balance between two loss terms. As mentioned in Section 3.3, we respectively average the adversarial losses of each feature elements in the last layer of a discriminator in (7).

The role of the coarse path loss is to complete the missing features properly for the CAM. Thus, we optimize the following simple l_1 loss function defined as

$$L_C = \frac{1}{N}\sum_{n=1}^{N}\|X_\text{c}^{(n)} - Y^{(n)}\|_1, \tag{9}$$

where $X_\text{c}^{(n)}$ and $Y^{(n)}$ are the nth image pair of the generated image via the coarse path and its corresponding original input image in a mini-batch, respectively. Finally, we define the total loss function of the generative network of PEPSI as follows:

$$L_\text{total} = L_G + \lambda_\text{c}(1 - \frac{k}{k_\text{max}})L_C, \tag{10}$$

where λ_c is a hyper-parameter controlling the contributions from each loss term, and k and k_max represent the iteration of the learning procedure and the maximum number of iterations, respectively. In the proposed method, as the training progresses, we slightly reduce the weights of the coarse path loss for the decoding network to focus on the image reconstruction process.

4. Experiments

4.1. Implementation details

Free-Form Mask As shown in Figure 7(b), traditional methods [10, 21, 28] usually adopt the regular mask (*e.g.* hole region with rectangular shape) during the training procedure. Thus, the network trained with regular mask often yields visual artifacts such as color discrepancy and blurriness when the hole region has irregular shapes. To cope with this problem, Yu *et al.* [27] adopt the free-form

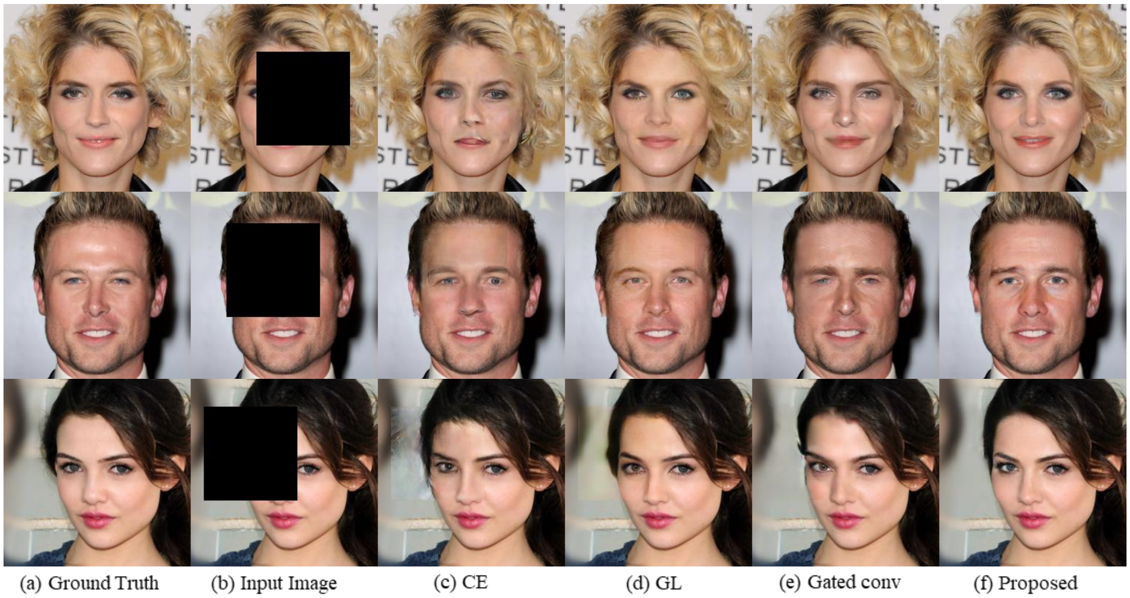


Figure 6. Comparison of our method and conventional methods on randomly square masked CelebA-HQ datasets. (a) The ground truth (b) The input image of the network (c) Results of the Context Encoder [21] (d) Results of the Globally-Locally [10] (e) Results of the gated convolution [27] (f) Results of the proposed method

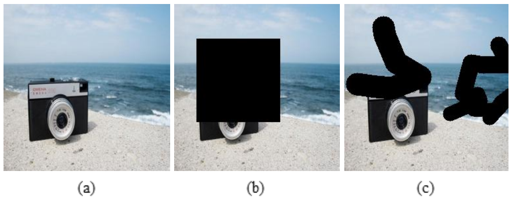


Figure 7. Examples of (a) the original image, (b) its square masked image, and (c) the free-form masked image

mask algorithm during the training procedure, which automatically generates multiple random free-form holes as depicted in Figure 7(c). In particular, this algorithm produces the free-form mask by drawing multiple different lines and erasing pixels closer than an arbitrary distance from these lines. For a fair comparison, we adopt the same free-form mask generation algorithm to our method.

Training Procedure PEPSI is trained for one million iterations using a batch size of 8 in an end-to-end manner. We perform optimization using the ADAM optimizer [12], which is a stochastic optimization method with adaptive estimation of moments. The parameters of Adam optimizer, β_1 and β_2, are set to 0.5 and 0.9, respectively. Inspired by [9], we employ two-timescale update rule (TTUR) where the learning rates of the discriminator and generator are 0.0004 and 0.0001, respectively. In addition, we reduce the learning rate as 1/10 after 0.9 million iterations. The hyper-parameters in our model are set to $\lambda_{\mathrm{i}} = 10$, $\lambda_{\mathrm{c}} = 5$, and $\lambda_{\mathrm{adv}} = 0.1$. Our experiments were conducted on CPU Intel(R) Xeon(R) CPU E3-1245 v5 and GPU TITAN X (Pascal), and implemented in TensorFlow v1.8.

4.2. Performance Evaluation

For our experiments, we use the CelebA-HQ [11, 18], ImageNet [14], and Place2 [30] datasets which consist of human faces, things, and various scenes, respectively. In the CelebA-HQ dataset, we randomly sample the 27,000 images as a training set and 3,000 ones as a test set. We also train the network with all the images in the ImageNet dataset and test it on Place2 dataset to measures the performance of trained deep learning models on other datasets to confirm the generalization ability of PEPSI. In addition, to demonstrate the superiority of PEPSI, we compare its qualitative, quantitative, and operation speed results with those of the conventional generative methods: CE [21], GL [10], GCA [28], and GatedConv [27].

Qualitative Comparison We compare the qualitative performance of PEPSI with the conventional methods using the image masked with the free-form mask as well as that with the squared mask. The conventional methods are implemented by following the training procedure in each paper. As shown in Figures 6 and 8, CE [21] and GL [10] show obvious visual artifacts including blurred or distorted images in the masked region, especially on the free-form mask. Although GatedConv [27] shows a fine performance, it shows lack of relevance between hole and background regions such as symmetry of eyes. In contrast to the conventional methods, PEPSI shows visually pleasing results and high relevance between hole and background regions.

Moreover, we show the real application of PEPSI by testing on the challenging datasets, ImageNet and Place2 datasets. We compare PEPSI with GatedConv and the widely available non-generative method, PatchMatch [1],

Method	Square mask			Free-form mask			Time (ms)
	PSNR		SSIM	PSNR		SSIM	
	Local	Global		Local	Global		
CE [21]	17.7	23.7	0.872	9.7	16.3	0.794	**5.8**
GL [10]	19.4	25.0	0.896	15.1	21.5	0.843	39.4
GCA [28]	19.0	24.9	0.898	12.4	18.9	0.798	22.5
GatedConv [27]	18.7	24.7	0.895	21.2	27.8	0.925	21.4
GatedConv *	17.5	23.5	0.882	19.8	26.4	0.910	14.3
PEPSI(Ours)	**19.5**	**25.6**	**0.901**	**22.0**	**28.6**	**0.929**	9.2
PEPSI *	19.2	25.2	0.894	21.6	28.2	0.923	

Table 6. Results of global and local PSNR, SSIM and operation time with both of square and free-formed masks on CelebA-HQ dataset. * means a model without coarse results.

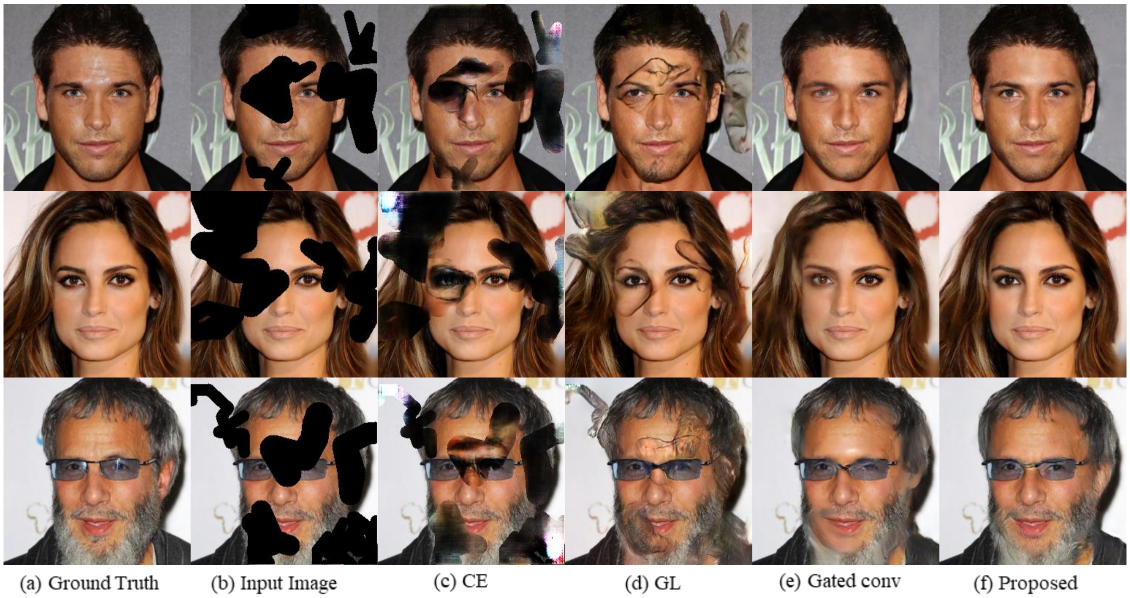
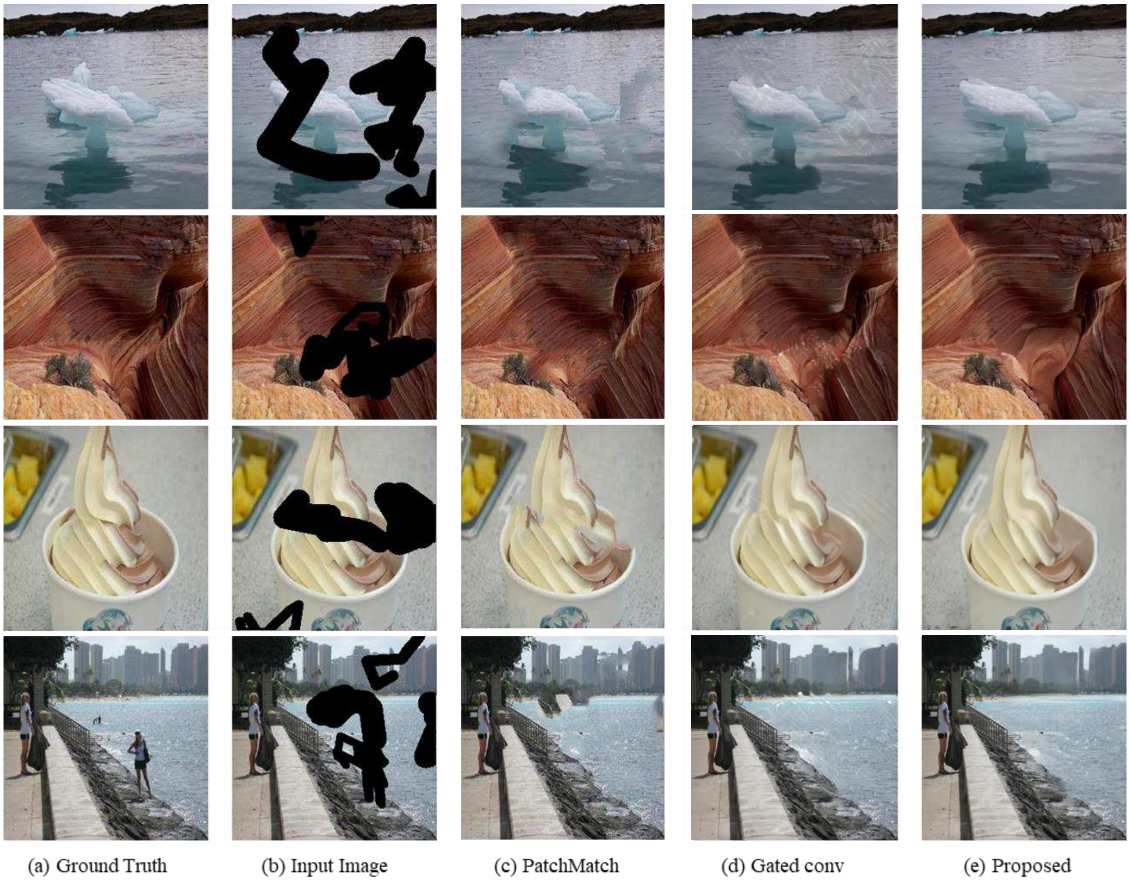


Figure 8. Comparison of our method and conventional methods on free-form masked CelebA-HQ datasets. (a) The ground truth (b) The input image of the network (c)Results of the Context Encoder [21] (d) Results of the Globally-Locally [10] (e) Results of the GatedConv [27] (f) Results of the proposed method

on the Place2 dataset with 256 × 256 image resolution. As depicted in Figure 9, PatchMatch shows visually poor performance especially on the edge of images because it fills the hole region without understands of the contexts of scenes. GatedConv generates more realistic results without color discrepancy or edge distortion but still produces wrong textures. In contrary, PEPSI generates the most natural images without artifacts or distortion on various contents and complex scenes for real applications.

Quantitative Comparison We evaluate a performance of the proposed and conventional methods by measuring the peak signal-to-noise ratio (PSNR) of the local and global regions, *i.e.* the hole region and the whole image, and the structural similarity (SSIM) [23]. Table 6 provides the comprehensive performance benchmarks between PEPSI and conventional ones [10, 21, 27, 28] on CelebA-HQ datasets [11]. As shown in Table 6, CE [21], GL [10], and [28] effectively fill the hole region with a square shape, but they could not complete the hole region with an irregular shape. Since these methods mainly focus on filling the holes with a rectangular shape, they could not generalize well on the free-form masks. Note that GL [10] shows a competitive PSNR value with the PEPSI only in the local region of the square mask since it applies a image blending technique as the post-processing. However, this post-processing yields blurred results as shown in Figure 6(d) and needs more computation time. GatedConv [27] shows fine performance on both of square and free-form holes, but also needs much computation time. Contrary to the conventional methods, PEPSI can complete any shape of the hole region, while reducing the operation time significantly.

For further study, we conduct an experiment in which the models, GatedConv and PEPSI, are trained without using the coarse results, *i.e.* GatedConv without using the coarse network and PEPSI without using the coarse path learning. As shown in Table 6 (models without coarse results



(a) Ground Truth (b) Input Image (c) PatchMatch (d) Gated conv (e) Proposed

Figure 9. Comparison of our method and conventional methods on Place2 datasets. (a) The ground truth (b) The input image of the network (c) Results of the non-generative method, PatchMatch (d) Results of the GatedConv [27] (e) Results of the proposed method

Mask	Method	PSNR		SSIM
		Local	Global	
Square	GatedConv [27]	14.2	20.3	0.818
	PEPSI(Ours)	15.2	21.2	0.832
Free-form	GatedConv [27]	17.4	24.0	0.875
	PEPSI(Ours)	18.2	24.8	0.882

Table 7. Results of global and local PSNR and SSIM on the Places2 dataset.

are denoted by *), even without the coarse results, PEPSI shows better results compared to the full model of GatedConv thanks to the modified CAM and RED. With the coarse path learning, PEPSI exhibits the better than PEPSI without using coarse results in terms of all the Quantitative metrics, which indicates that the coarse path drives the encoding network to produce missing features properly for the CAM. In other words, the single-stage network structure of PEPSI can overcome the limitation of the two-stage coarse-to-fine network through a parallel learning scheme.

To demonstrate the generalization ability of PEPSI, we conduct another experiment using the challenging datasets, ImageNet [14], and Place2 [30] datasets. Table 7 shows the experimental results of the test using the input image with the resolution of 256×256. We compare the performance of PEPSI with GatedConv [27], which exhibits superior performance compared to other conventional methods in Celeb-A dataset. As shown in Table 7, PEPSI achieves better performance than GatedConv on Place2 dataset, which indicates that the PEPSI can consistently generate the high-quality results from various contents and complex scenes either.

5. Conclusion

In this paper, a novel image inpainting method, called PEPSI, has been proposed. As shown in the experimental results, the proposed method not only achieves superior performance as compared to conventional ones, but also significantly reduces the operation time by redesigning unifying two-stage coarse-to-fine network into an efficient single-stage network structure and adopting an effective joint learning scheme for training the proposed architecture. Therefore, it is expected that PEPSI can be widely employed in various applications including image generation, style transfer, and image editing. Further improvements can be achieved by reducing the parameters of the network, which helps to be applied to restricted hardware systems.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24, 2009. 1, 6

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. 1

[3] N. Cai, Z. Su, Z. Lin, H. Wang, Z. Yang, and B. W.-K. Ling. Blind inpainting using the fully convolutional neural network. *The Visual Computer*, 33(2):249–261, 2017. 1

[4] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 4

[5] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 1

[6] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard. Image inpainting through neural networks hallucinations. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*, pages 1–5. Ieee, 2016. 1

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2, 5

[8] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang. Region ensemble network: Improving convolutional network for hand pose estimation. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 4512–4516. IEEE, 2017. 5

[9] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 1, 2, 3, 5, 6, 7

[11] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2, 6, 7

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[13] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014. 1

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2, 6, 8

[15] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *European Conference on Computer Vision*, pages 377–389. Springer, 2004. 1

[16] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016. 1

[17] H. Li, G. Li, L. Lin, and Y. Yu. Context-aware semantic inpainting. *arXiv preprint arXiv:1712.07778*, 2017. 1

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 2, 6

[19] H. Noori, S. Saryazdi, and H. Nezamabadi-Pour. A convolution based image inpainting. In *1st International Conference on Communication and Engineering*, 2010. 1

[20] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 702–711. IEEE, 2017. 1

[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 1, 2, 3, 5, 6, 7

[22] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[24] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014. 1

[25] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017. 1

[26] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017. 1

[27] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2, 3, 4, 5, 6, 7, 8

[28] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018. 1, 2, 3, 4, 5, 6, 7

[29] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 5

[30] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2018. 2, 6, 8