

Unsupervised Holistic Image Generation from Key Local Patches

Donghoon Lee¹, Sangdoo Yun¹, Sungjoon Choi¹, Hwiyeon Yoo¹, Ming-Hsuan Yang², and Songhwai Oh¹

¹Electrical and Computer Engineering and ASRI, Seoul National University, Korea

²Electrical Engineering and Computer Science, University of California at Merced

donghoon.lee@cplab.snu.ac.kr, yunsd101@snu.ac.kr, sungjoon.choi@cplab.snu.ac.kr

hwiyeon.yoo@cplab.snu.ac.kr, mhyang@ucmerced.edu, songhwai@snu.ac.kr

Abstract

We introduce a new problem of generating an image based on a small number of key local patches without any geometric prior. In this work, key local patches are defined as informative regions of the target object or scene. This is a challenging problem since it requires generating realistic images and predicting locations of parts at the same time. We construct adversarial networks to tackle this problem. A generator network generates a fake image as well as a mask based on the encoder-decoder framework. On the other hand, a discriminator network aims to detect fake images. The network is trained with three losses to consider spatial, appearance, and adversarial information. The spatial loss determines whether the locations of predicted parts are correct. Input patches are restored in the output image without much modification due to the appearance loss. The adversarial loss ensures output images are realistic. The proposed network is trained without supervisory signals since no labels of key parts are required. Experimental results on six datasets demonstrate that the proposed algorithm performs favorably on challenging objects and scenes.

1. Introduction

The goal of image generation is to construct images that are as barely distinguishable from target images which may contain general objects, diverse scenes, or human drawings. Synthesized images can contribute to a number of applications such as the image to image translation [6], image super-resolution [11], 3D object modeling [33], unsupervised domain adaptation [13], domain transfer [36], future frame prediction [31], image inpainting [35], image editing [39], and feature recovering of astrophysical images [27].

In this paper, we introduce a new image generation problem, in which a whole image is generated conditioned on parts of an image. The objective of this work, as shown in Figure 1, is to generate an image based on a small number of local patches without geometric priors. This problem is

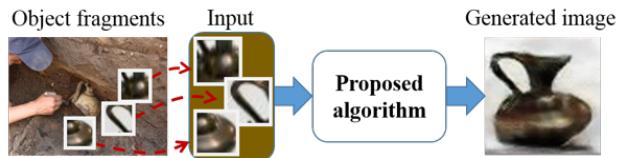


Figure 1: The proposed algorithm is able to synthesize an image from key local patches without geometric priors, e.g., restoring broken pieces of ancient ceramics found in ruins. Convolutional neural networks are trained to predict locations of input patches and generate the entire image based on adversarial learning.

more complicated than conventional image generation tasks as it entails to achieve three goals simultaneously.

First, spatial arrangements of input patches need to be inferred since the data does not contain explicit information about the location. To tackle this issue, we assume that inputs are key local patches which are informative regions of the target image. Therefore, the algorithm should learn the spatial relationship between key parts of an object or scene. Our approach obtains key regions without any supervision such that the whole algorithm is developed within the unsupervised learning framework.

Second, we aim to generate an image while preserving the key local patches. As shown in Figure 1, the appearances input patches are included in the generated image without significant modification. In other words, the inputs are not directly copied to the output image. It allows us to create images more flexibly such that we can combine key patches of different objects as inputs. In such cases, input patches must be deformed by considering each other.

Third and most importantly, the generated image should look closely to a real image in the target category. Unlike the image inpainting problem, which mainly replaces small regions or eliminates minor defects, our goal is to reconstruct a holistic image based on limited appearance information contained in a few patches.

To address the above issues, we adopt the adversarial learning scheme [3] in this work. The generative adversarial network (GAN) contains two networks which are trained based on the min-max game of two players. A generator network typically generates fake images and aims to fool a discriminator, while a discriminator network seeks to distinguish fake images from real images. In our case, the generator network is also responsible for predicting the locations of input patches. Based on the generated image and predicted mask, we design three losses to train the network: a spatial loss, an appearance loss, and an adversarial loss, corresponding to the aforementioned issues, respectively.

While a conventional GAN is trained in an unsupervised manner, some recent methods formulate it in a supervised manner by using labeled information. For example, a GAN is trained with a dataset that have 15 or more joint positions of birds [23]. Such labeling task is labor intensive since GAN-based algorithms need a large amount of training data to achieve high-quality results. In contrast, experiments on six challenging datasets that contain different objects and scenes, such as faces, cars, flowers, ceramics, and waterfalls, demonstrate that the proposed unsupervised algorithm can generate realistic images and predict part locations well. In addition, even if inputs contain parts from different objects, our algorithm is able to generate reasonable images.

The main contributions are as follows. First, we introduce a new problem to render realistic image conditioned on the appearance information of a few key patches. Second, we develop a generative network to jointly predict the mask and the image without supervision to address the defined problem. Third, we propose a novel objective function using additional fake images to strengthen the discriminator network. Finally, we provide new datasets that contain challenging objects and scenes.

2. Related Work

Image Generation. Image generation is an important problem that has been studied extensively in computer vision. With the recent advances in deep convolutional neural networks [10, 29], numerous image generation methods have achieved the state-of-the-art results. Dosovitskiy et al. [2] generate 3D objects by learning transposed convolutional neural networks. In [8], Kingma et al. propose a method based on the variational inference for the stochastic image generation. An attention model is developed by Gregor et al. [4] to generate an image using a recurrent neural network. Recently, the stochastic PixelCNN [30] and PixelRNN [20] are introduced to generate images sequentially.

The generative adversarial network [3] is proposed for generating sharp and realistic images based on two competing networks: a generator and a discriminator. Numerous methods [26, 38] have been proposed to improve the stability of the GAN. Radford et al. [22] propose deep con-

volutional generative adversarial networks (DCGAN) with a set of constraints to generate realistic images effectively. Based on the DCGAN architecture, Wang et al. [32] develop a model to generate the style and structure of indoor scenes (SSGAN), and Liu et al. [13] present a coupled GAN which learns a joint distribution of multi-domain images, such as color and depth images.

Conditional GAN. Conditional GAN approaches [17, 24, 37] are developed to control the image generation process with label information. Mizra et al. [17] propose a class-conditional GAN which uses discrete class labels as the conditional information. The GAN-CLS [24] and StackGAN [37] embed a text describing an image into the conditional GAN to generate an image corresponding to the condition. On the other hand, the GAWWN [23] creates numerous plausible images based on the location of key points or an object bounding box. In these methods, the conditional information, e.g., text, key points, and bounding boxes, is provided in the training data. However, it is labor intensive to label such information since deep generative models require a large amount of training data. In contrast, key patches used in the proposed algorithm are obtained without the necessity of human annotation.

Numerous image conditional models based on GANs have been introduced recently [11, 39, 36, 35, 21, 12, 28, 6]. These methods learn a mapping from the source image to target domain, such as image super-resolution [11], user interactive image manipulation [39], product image generation from a given image [36], image inpainting [35, 21], style transfer [12] and realistic image generation from synthetic image [28]. Isola et al. [6] tackle the image-to-image translation problem including various image conversion examples such as day image to night image, gray image to color image, and sketch image to real image, by utilizing the U-net [25] and GAN. In contrast, the problem addressed in this paper is the holistic image generation based on only a small number of local patches. This challenging problem cannot be addressed by existing image conditional methods as the domain of the source and target images are different.

Unsupervised Image Context Learning. Unsupervised learning of the spatial context in an image [1, 19, 21] has attracted attention to learn rich feature representations without human annotations. Doersch et al. [1] train convolutional neural networks to predict the relative position between two neighboring patches in an image. The neighboring patches are selected from a grid pattern based on the image context. To reduce the ambiguity of the grid in [1], Noroozi et al. [19] divide the image into a large number of tiles, shuffle the tiles, and then learn a convolutional neural network to solve the jigsaw puzzle problem. Pathak et al. [21] address the image inpainting problem which predicts missing pixels in an image, by training a context en-

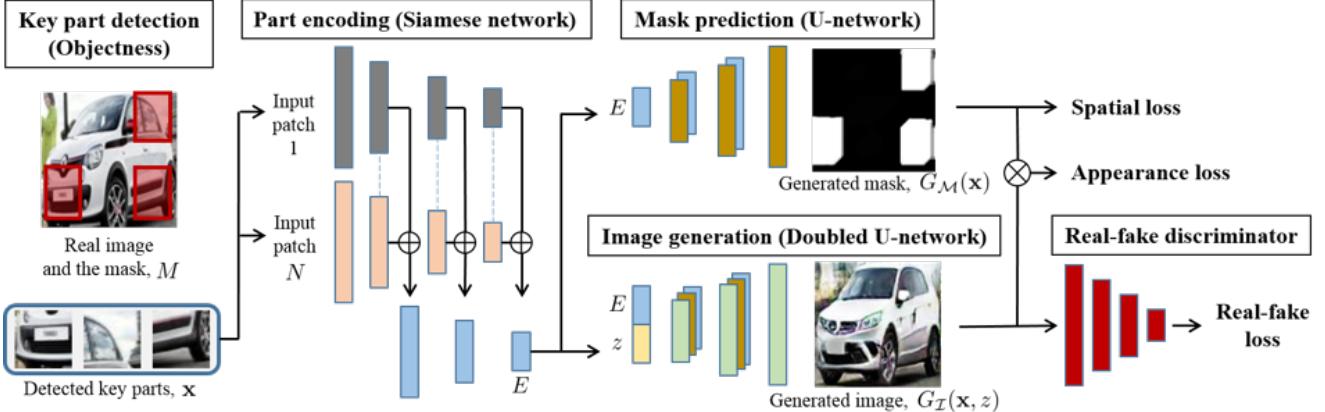


Figure 2: Proposed network architecture. A bar represents a layer in the network. Layers of the same size and the same color have the same convolutional feature maps. Dashed lines in the part encoding network represent shared weights. In addition, E denotes an embedded vector and z is a random noise vector.

coder. Through the spatial context learning, the trained networks are successfully applied to various applications such as object detection, classification and semantic segmentation. However, discriminative models [1, 19] can only infer the spatial arrangement of image patches, and the image inpainting method [21] requires the spatial information of the missing pixels. In contrast, we propose a generative model which is capable of not only inferring the spatial arrangement of input patches but also generating the entire image.

3. Proposed Algorithm

Figure 2 shows the structure of the proposed network for image generation from a few patches. It is developed based on the concept of adversarial learning, where a generator and a discriminator compete with each other [3]. However, in the proposed network, the generator has two outputs: the predicted mask and generated image. Let G_M be a mapping from N observed images $\mathbf{x} = \{x_1, \dots, x_N\}$ to a mask M , $G_M : \mathbf{x} \rightarrow M$.¹ Also let G_I be a mapping from \mathbf{x} and a random noise vector z to an output image y , $G_I : \{\mathbf{x}, z\} \rightarrow y$. These mappings are performed based on three networks: a part encoding network, a mask prediction network, and an image generation network. The discriminator D is based on a convolutional neural network which aims to distinguish the real image from the image generated by G_I .

We use three losses to train the network. The first loss is the spatial loss \mathcal{L}_S . It compares the inferred mask and real mask which represents the cropped region of the input patches. The second loss is the appearance loss \mathcal{L}_A , which maintains input key patches in the generated image without much modification. The third loss is the adversarial loss \mathcal{L}_R to distinguish fake and real images. The whole network is

¹ Here, \mathbf{x} is a set of image patches resized to the same width and height suitable for the proposed network and N is the number of image patches in \mathbf{x} .



Figure 3: Examples of detected key patches on faces [14], vehicles [9], flowers [18], and waterfall scenes. Three regions with top scores from the EdgeBox algorithm are shown in red boxes after pruning candidates of an extreme size or aspect ratio.

trained by the following min-max game:

$$\min_{G_M, G_I} \max_D \mathcal{L}_R(G_I, D) + \lambda_1 \mathcal{L}_S(G_M) + \lambda_2 \mathcal{L}_A(G_M, G_I), \quad (1)$$

where λ_1 and λ_2 are weights for the spatial loss and the appearance loss, respectively.

3.1. Key Part Detection

Key patches are defined as informative local regions to generate the entire image. For example, when generating a face image, patches of eyes and a nose are more informative than those of the forehead and cheeks. Therefore, it would be better for the key patches to contain important parts that can describe objects in a target class. However, it is not desirable to manually fix the categories of key patches since objects in different classes are composed of different parts. To address this issue, we use the objectness score from the Edgebox algorithm [40] to detect key patches. It can detect key patches of objects in general classes in an unsupervised manner. In addition, we discard detected patches with extreme sizes or aspect ratios. Figure 3 shows examples

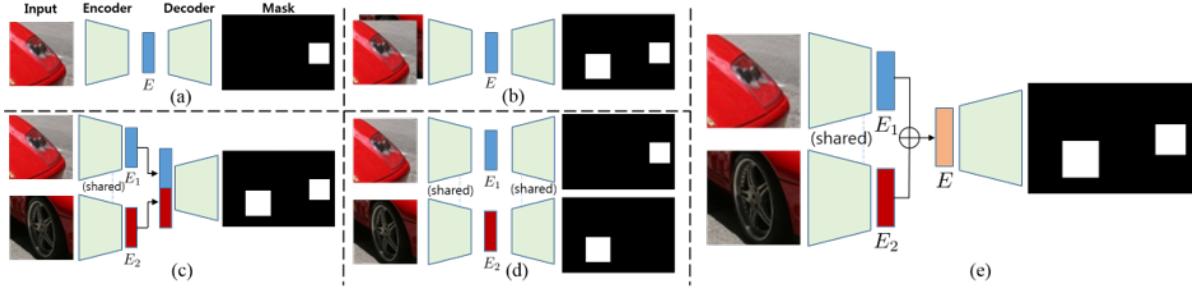


Figure 4: Different structures of networks to predict a mask from input patches. We choose (e) as our encoder-decoder model.

of detected key patches from various objects and scenes. Overall, the detected regions from these object classes are fairly informative. We sort candidate regions by the objectness score and feed the top N patches to the proposed network. In addition, the training images and corresponding key patches are augmented using a random left-right flip with the equal probability.

3.2. Part Encoding Network

The structure of the generator is based on the encoder-decoder network [5]. It uses convolutional layers as an encoder to reduce the dimension of the input data until the bottleneck layer. Then, transposed convolutional layers upsample the embedded vector to its original size. For the case with a single input, the network has a simple structure as shown in Figure 4(a). For the case with multiple inputs as considered in the proposed network, there are many possible structures. We examine four cases in this work.

The first network is shown in Figure 4(b), which uses depth-concatenation of multiple patches. This is a straightforward extension of the single input case. However, it is not suitable for the task considered in this work. Regardless of the order of input patches, the same mask should be generated when the patches have the same appearance. Therefore, the embedded vector E must be the same for all different orderings of inputs. Nevertheless, the concatenation causes the network to depend on the ordering, while key patches have an arbitrary order since they are sorted by the objectness score. In this case, the part encoding network cannot learn proper filters. The same issue arises in the model in Figure 4(c). On the other hand, there are different issues with the network in Figure 4(d). While it can solve the ordering issue, it predicts a mask of each input independently, which is not desirable as we aim to predict masks jointly. The network should consider the appearance of both input patches to predict positions. To address the above issues, we propose to use the network in Figure 4(e). It encodes multiple patches based on a Siamese-style network and summarizes all results in a single descriptor by the summation, i.e., $E = E_1 + \dots + E_N$. Due to the commutative property, we can predict a mask jointly, even if inputs

have an arbitrary order. In addition to the final bottleneck layer, we use all convolutional feature maps in the part encoding network to construct U-net [25] style architectures as shown in Figure 2.

3.3. Mask Prediction Network

The U-net is an encoder-decoder network that has skip connections between i -th encoding layer and $(L - i)$ -th decoding layer, where L is the total number of layers. It directly feeds the information from an encoding layer to its corresponding decoding layer. Therefore, combining the U-net and a generation network is effective when the input and output share the same semantic [6]. In this work, the shared semantic of input patches and the output mask is the target image.

We pose the mask prediction as a regression problem. Based on the embedded part vector E , we use transposed convolutional layers with a fractional stride [22] to upsample the data. The output mask has the same size as the target image and has a value between 0 and 1 at each pixel. Therefore, we use the sigmoid activation function at the last layer. The detailed configurations are presented in Table 1.

The spatial loss, \mathcal{L}_S , is defined as follows:

$$\mathcal{L}_S(G_M) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x}), M \sim p_{data}(M)} [\|G_M(\mathbf{x}) - M\|_1]. \quad (2)$$

We note that other types of losses, such as the l_2 -norm, or more complicated network structures, such as GAN, have been evaluated for mask prediction, and similar results are achieved by these alternative options.

3.4. Image Generation Network

We propose a double U-net structure for the image generation task as shown in Figure 2. It has skip connections from both the part encoding network and mask generation network. In this way, the image generation network can communicate with other networks. This is critical since the generated image should consider the appearance and locations of input patches. Figure 5 shows generated images with and without the skip connections. It shows that the proposed network improves the quality of generated images. In addition, it helps to preserve the appearances of input patches.

Table 1: Details of each network. # Filter is the number of filters. BN is the batch normalization. Conv denotes a convolutional layer. F-Conv denotes a transposed convolutional layer that uses the fractional-stride.

(a) Details of the {part encoding, discriminator} network						(b) Details of the {mask prediction, image generation} network					
Layer	# Filter	Filter Size	Stride	Pad	BN	Layer	# Filter	Filter Size	Stride	Pad	BN
Conv. 1	64	$5 \times 5 \times 3$	2	2	\times	Conv. 1	$4 \times 4 \times 1024$	$1 \times 1 \times \{100, 200\}$	1	0	\circ
Conv. 2	128	$5 \times 5 \times 64$	2	2	\circ	F-Conv. 2	1024	$5 \times 5 \times 512$	1/2	-	\circ
Conv. 3	256	$5 \times 5 \times 128$	2	2	\circ	F-Conv. 3	512	$5 \times 5 \times 256$	1/2	-	\circ
Conv. 4	512	$5 \times 5 \times 256$	2	2	\circ	F-Conv. 4	256	$5 \times 5 \times 128$	1/2	-	\circ
Conv. 5	1024	$5 \times 5 \times 512$	2	2	\circ	F-Conv. 5	128	$5 \times 5 \times 64$	1/2	-	\circ
Conv. 6	{100,1}	$1 \times 1 \times 1024$	1	0	{ \circ, \times }	F-Conv. 6	64	$5 \times 5 \times \{1, 3\}$	1/2	-	\times

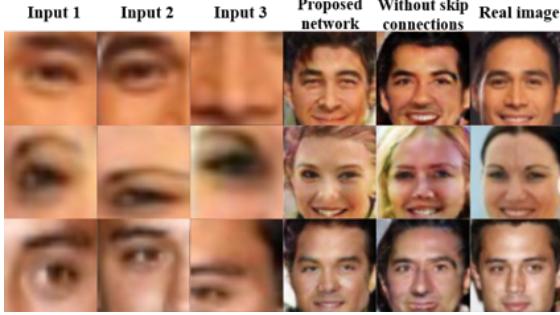


Figure 5: Sample image generation results on the CelebA dataset. Images are generated based on the network in Figure 2. Generated images are sharper and realistic with the skip connections.

Based on the generated image and predicted mask, we define the appearance loss \mathcal{L}_A as follows:

$$\begin{aligned} \mathcal{L}_A(G_M, G_I) = & \mathbb{E}_{\mathbf{x}, y \sim p_{data}(\mathbf{x}, y), z \sim p_z(z), M \sim p_{data}(M)} \\ & [\|G_I(\mathbf{x}, z) \otimes G_M(\mathbf{x}) - y \otimes M\|_1], \end{aligned} \quad (3)$$

where \otimes is an element-wise product.

3.5. Real-Fake Discriminator Network

A simple discriminator can be trained to distinguish real images from fake images. However, it has been shown that a naive discriminator may cause artifacts [28] or network collapses during training [16]. To address this issue, we propose a new objective function as follows:

$$\begin{aligned} \mathcal{L}_R(G_I, D) = & \mathbb{E}_{y \sim p_{data}(y)} [\log D(y)] + \\ & \mathbb{E}_{\mathbf{x}, y, y' \sim p_{data}(\mathbf{x}, y, y'), z \sim p_z(z), M \sim p_{data}(M)} \\ & [\log(1 - D(G_I(\mathbf{x}, z)))] + \\ & \log(1 - D(M \otimes G_I(\mathbf{x}, z) + (1 - M) \otimes y)) + \\ & \log(1 - D((1 - M) \otimes G_I(\mathbf{x}, z) + M \otimes y)) + \\ & \log(1 - D(M \otimes y' + (1 - M) \otimes y)) + \\ & \log(1 - D((1 - M) \otimes y' + M \otimes y)), \end{aligned} \quad (4)$$

where y' is a real image randomly selected from the outside of the current mini-batch. When the real image y is com-

bined with the generated image $G_I(\mathbf{x}, z)$ (line 4-5 in (4)), it should be treated as a fake image as it partially contains the fake image. When two different real images y and y' are combined (line 6-7 in (4)), it is also a fake image although both images are real. It not only enriches training data but also strengthens discriminator by feeding difficult examples.

4. Experiments

For all experiments, images are resized to the minimum length of 128 pixels on the width or height. All key part candidates are obtained using the Edgebox algorithm [40]. We reject candidate boxes that are larger than 25% or smaller than 5% of the image size unless otherwise stated. After that, the non-maximum suppression is applied to remove candidates that are too close with each other. Finally, the image and top N candidates are resized to the target size, $128 \times 128 \times 3$ pixels for the CompCars dataset or $64 \times 64 \times 3$ pixels for other datasets, and fed to the network. The λ_1 and λ_2 are decreased from 10^{-2} to 10^{-4} as the epoch increases.

Table 1 shows detailed description of the proposed network for $128 \times 128 \times 3$ pixels image. The input parts are encoded into a 100-dimensional vector E . A mask is predicted using E , while an image is generated based on a 200-dimensional vector which is a concatenation of E and a 100-dimensional random noise vector z . The part encoding network uses the leaky ReLU [15] with a slope of 0.2 as an activation function. The discriminator uses the same leaky ReLU except for the last layer which uses a sigmoid function. The mask prediction and image generation networks use ReLU except for the last layer which uses a sigmoid function and tanh, respectively. The filters in the network are initialized with a zero mean Gaussian distribution with a standard deviation of 0.02.

We train the network with a learning rate of 0.0002. As the epoch increases, we decrease λ_1 and λ_2 in (1). With this training strategy, the network focuses on predicting a mask in the beginning, while it becomes more important to generate realistic images in the end. The mini-batch size is 64, and the momentum of the Adam optimizer [7] is set to 0.5. During training, we first update the discriminator network and then update the generator network twice. More results

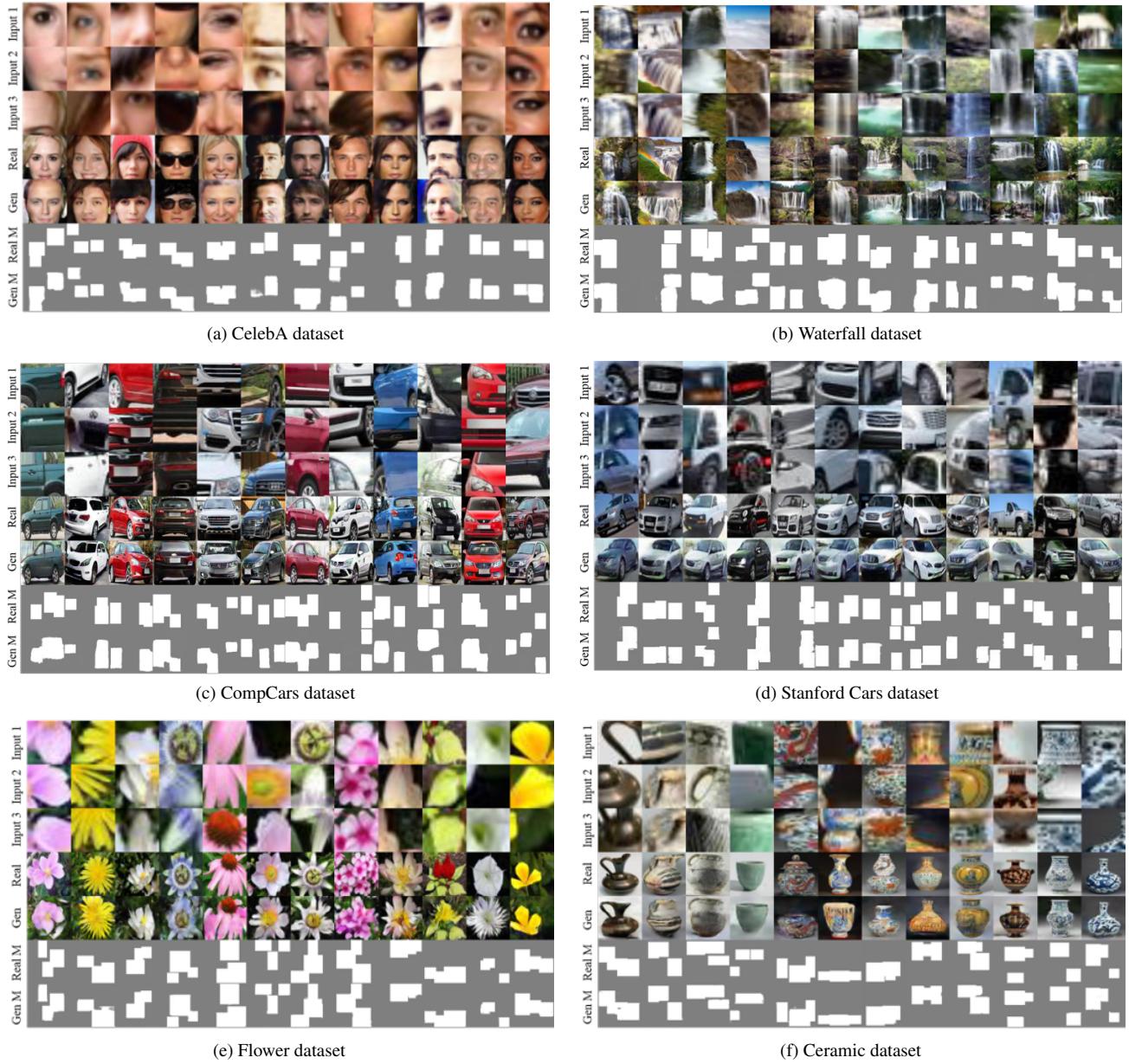


Figure 6: Examples of generated masks and images on six datasets. The generated images for each class are shown in 12 columns. Three key local patches (Input 1, Input 2, and Input 3) from a real image (Real). The key parts are top-3 regions in terms of the objectness score. Given inputs, images (Gen) and masks (Gen M) are generated. Real M is the ground truth mask.

are available in the supplementary material. All the source code and datasets will be made available to the public.

4.1. Datasets

The CelebA dataset [14] contains 202,599 celebrity images with large pose variations and background clutters (see Figure 6(a)). There are 10,177 identities with various attributes, such as eyeglasses, hat, mustache, and facial ex-

pressions. We use aligned and cropped face images of 108×108 pixels. The network is trained for 25 epochs.

The flower dataset [18] consists of 102 flower categories (see Figure 6(e)). There is a total of 8,189 images, and each class has between 40 and 258 images. The images contain large variations in the scale, pose, and lighting condition. We train the network for 800 epochs.

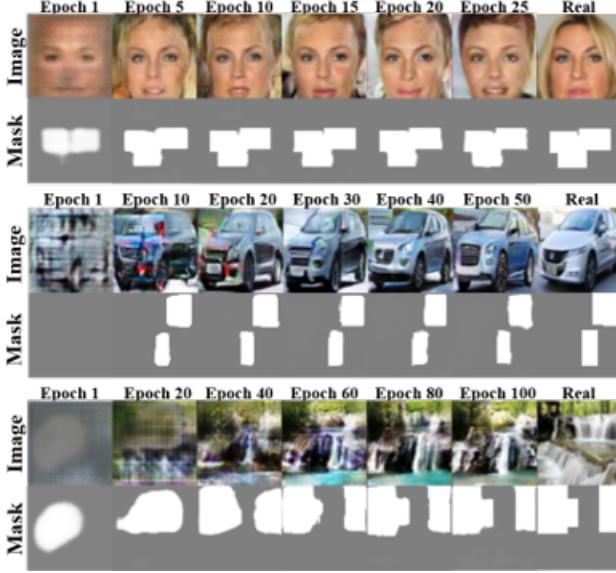


Figure 7: Sample generated masks and images at different epochs.

There are two car datasets [34, 9] used in this paper. The CompCars dataset [34] includes images from two scenarios: the web-nature and surveillance-nature (see Figure 6(c)). The web-nature data contains 136,726 images of 1,716 car models, and the surveillance-nature data contains 50,000 images. The network is trained for 50 epochs. The Stanford Cars dataset [9] contains 16,185 images of 196 classes of cars (see Figure 6(d)). They have different lighting conditions and camera angles. Furthermore, a wide range of colors and shapes, e.g., sedans, SUVs, convertibles, trucks, are included. The network is trained for 400 epochs.

The waterfall dataset consists of 15,323 images taken from various viewpoints (see Figure 6(b)). It has different types of waterfalls as images are collected from the internet. It also includes other objects such as trees, rocks, sky, and ground, as images are obtained from natural scenes. For this dataset, we allow tall candidate boxes, in which the maximum height is 70% of the image height, to catch long water streams. The network is trained for 100 epochs.

The ceramic dataset is made up of 9,311 side-view images (see Figure 6(f)). Images of both Eastern-style and Western-style potteries are collected from the internet. The network is trained for 800 epochs.

4.2. Image Generation Results

Figure 7 shows generation results as the training epoch is increased. At the start, the network focuses on predicting a good mask. As the epoch is increased, input parts become sharper. At the end of the epoch, the network concentrates on generating realistic images. In the case of the CelebA dataset, it is relatively easy to find the mask since the im-



Figure 8: For each generated image in the green box, nearest neighbors in the corresponding training dataset are displayed.

ages of this dataset are aligned. On the other hand, for other datasets, it takes more epochs to find a good mask. The results show that the masked regions have similar appearances while other regions are changed in a way to make realistic holistic images.

Figure 6 shows image generation results of different object classes. Each input has three key patches from a real image and we show both generated and original ones for visual comparisons. For all datasets, which contain challenging objects and scenes, the proposed algorithm is able to generate realistic images. The subject of the generated face images using the CelebA dataset in Figure 6(a) may have different gender (column 1 and 2), wear a new beanie or sunglasses (column 3 and 4), and become older, chubby, and with new hairstyles (column 5-8). Even when the input key patches are concentrated on the left or right sides, the proposed algorithm can generate realistic images (column 9 and 10). In the CompCars dataset, the shape of car images is mainly generated based on the direction of tire wheels, head lights, and windows. For some cases, such as column 2 in Figure 6(c), input patches can be from both left or right directions and the generation results can be flipped. It demonstrates that the proposed algorithm is flexible since the correspondence between the generated mask and input patches, e.g., the left part of the mask corresponds to the left wheel patch, is not needed. Due to the small number of training samples compared to the CompCars dataset, the results of the Stanford Cars dataset are less sharp but still realistic. For the waterfall dataset, the network learns how to draw a new water stream (column 1), a spray from the waterfall (column 3), or other objects such as rock, grass, and puddles (column 10). In addition, the proposed algorithm can help restoring broken pieces of ceramics found in ancient ruins (see Figure 6(f)).

Figure 8 shows nearest neighbors of generated images. We measure the Euclidean distance between the generated image and images in the training set to define neighbors. The generated images are visually similar to real images in

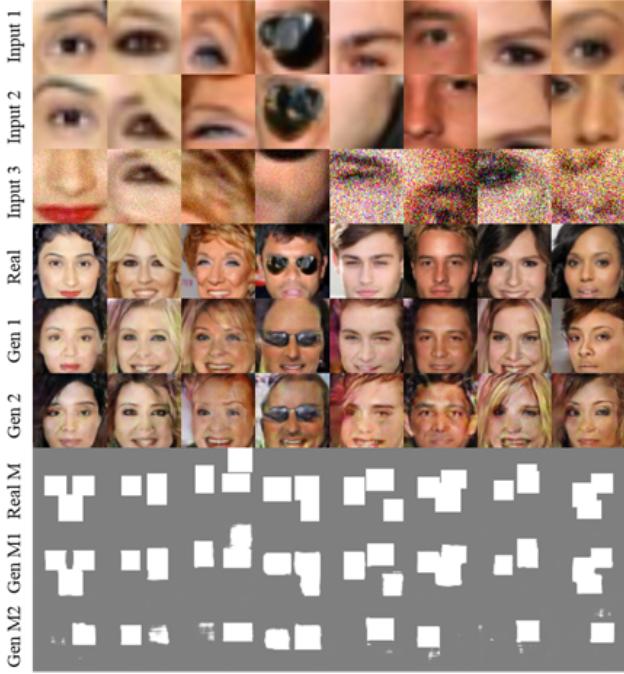


Figure 9: Examples of generated results when the input image contains noises. We add a Gaussian noise at each pixel of Input 3. Gen 1 and Gen M1 are generated without noises. Gen 2 and Gen M2 are generated with noises.

the training set, but have clear differences.

Figure 9 shows the results when input patches are degraded by noises. We apply the mean zero Gaussian noise at each pixel of the third input patch with the standard deviation of 0.1 (column 1-4) and 0.5 (column 5-8). The results show that the proposed algorithm is able to deal with certain amount of noise when generating realistic images.

Figure 10 shows generated images and masks when input patches are obtained from different persons. The results show that the proposed algorithm can handle a wide scope of input patch variations. For example, inputs contain different skin colors in the first column. In this case, it is not desirable to exactly preserve inputs since it will generate a face image with two different skin colors. The proposed algorithm generates an image with a reasonable skin color as well as the overall shape. Other cases include with or without sunglasses (column 2), different skin textures (column 3), hairstyle variations (column 4 and 5), and various expressions and orientations. Despite large variations, the proposed algorithm is able to generate realistic images.

Figure 11 shows failure cases of the proposed algorithm. It is difficult to generate images when detected key input patches include less informative regions (column 1 and 2) or rare cases (column 3). In addition, when input patches have conflicting information, e.g., the same nose-mouth patches

that have different orientations, the proposed algorithm is not able to generate realistic images (column 4, 5, and 6). Furthermore, it becomes complicated when the inputs are low-quality patches (column 7 and 8). We note these issues can be alleviated with additional pre-processing modules.

5. Conclusions

We introduce a new problem of generating images based on local patches without geometric priors. Local patches are obtained using the objectness score to retain informative parts of the target image in an unsupervised manner. We propose a generative network to render realistic images from local patches. The part encoding network embeds multiple input patches using a Siamese-style convolutional neural network. Transposed convolutional layers with skip connections from the encoding network are used to predict a mask and generate an image. The discriminator network aims to classify the generated image and the real image. The whole network is trained using the spatial, appearance, and adversarial losses. Extensive experiments show that the proposed network can generate realistic images of challenging objects and scenes. As humans can visualize a whole scene with a few visual cues, the proposed network can generate realistic images based on given unordered image patches.

References

- [1] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. of the IEEE International Conference on Computer Vision*, 2015. [2](#), [3](#)
- [2] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. [2](#)
- [3] 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*, 2014. [2](#), [3](#)
- [4] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. [2](#)
- [5] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. [4](#)
- [6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. [1](#), [2](#), [4](#)
- [7] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#)
- [8] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [2](#)
- [9] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proc. of the IEEE International Conference on Computer Vision Workshops*, 2013. [3](#), [7](#)



Figure 10: Results of the proposed algorithm on the CelebA dataset when input patches are came from other images. Input 1 and Input 2 are patches from Real 1. Input 3 is a local region of Real 2. Given inputs, the proposed algorithm generates the image (Gen) and mask (Gen M).



Figure 11: Examples of failure cases of the proposed algorithm.

- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 2
- [11] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016. 1, 2
- [12] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. of the European Conference on Computer Vision*, 2016. 2
- [13] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2016. 1, 2
- [14] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. of International Conference on Computer Vision*, 2015. 3, 6
- [15] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of the International Conference on Machine Learning*, 2013. 5
- [16] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016. 5
- [17] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2
- [18] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proc. of the IEEE Conference on Computer Vision, Graphics & Image Processing*, 2008. 3, 6
- [19] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. of the European Conference on Computer Vision*, 2016. 2, 3
- [20] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2
- [21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3
- [22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 4
- [23] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Advances In Neural Information Processing Systems*, 2016. 2
- [24] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *Proc. of the European Conference on Computer Vision*, 2016. 2

- Proc. of the International Conference on Machine Learning*, 2016. 2
- [25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 2, 4
- [26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016. 2
- [27] K. Schawinski, C. Zhang, H. Zhang, L. Fowler, and G. K. Santhanam. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Monthly Notices of the Royal Astronomical Society: Letters*, 467(1):L110–L114, 2017. 1
- [28] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2016. 2, 5
- [29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2
- [30] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixel-cnn decoders. In *Advances in Neural Information Processing Systems*, 2016. 2
- [31] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, 2016. 1
- [32] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *Proc. of the European Conference on Computer Vision*, 2016. 2
- [33] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, 2016. 1
- [34] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 7
- [35] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016. 1, 2
- [36] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-level domain transfer. In *Proc. of the European Conference on Computer Vision*, 2016. 1, 2
- [37] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016. 2
- [38] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2
- [39] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *Proc. of the European Conference on Computer Vision*, 2016. 1, 2
- [40] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proc. of the European Conference on Computer Vision*, 2014. 3, 5

6. Supplementary Material

6.1. Image Generation from Parts of Different Cars

Figure 12 shows generated images and masks when input patches are from different cars. Overall, the proposed algorithm generates reasonable images despite large variations of input patches.

6.2. Image Generation from a Different Number of Patches

In the manuscript, we show image generation with three local patches using the proposed algorithm. Figure 13 shows generated images based on two local patches. The results show that the network can be trained with different number of input patches.

6.3. Image Generation using an Alternative Objective Function

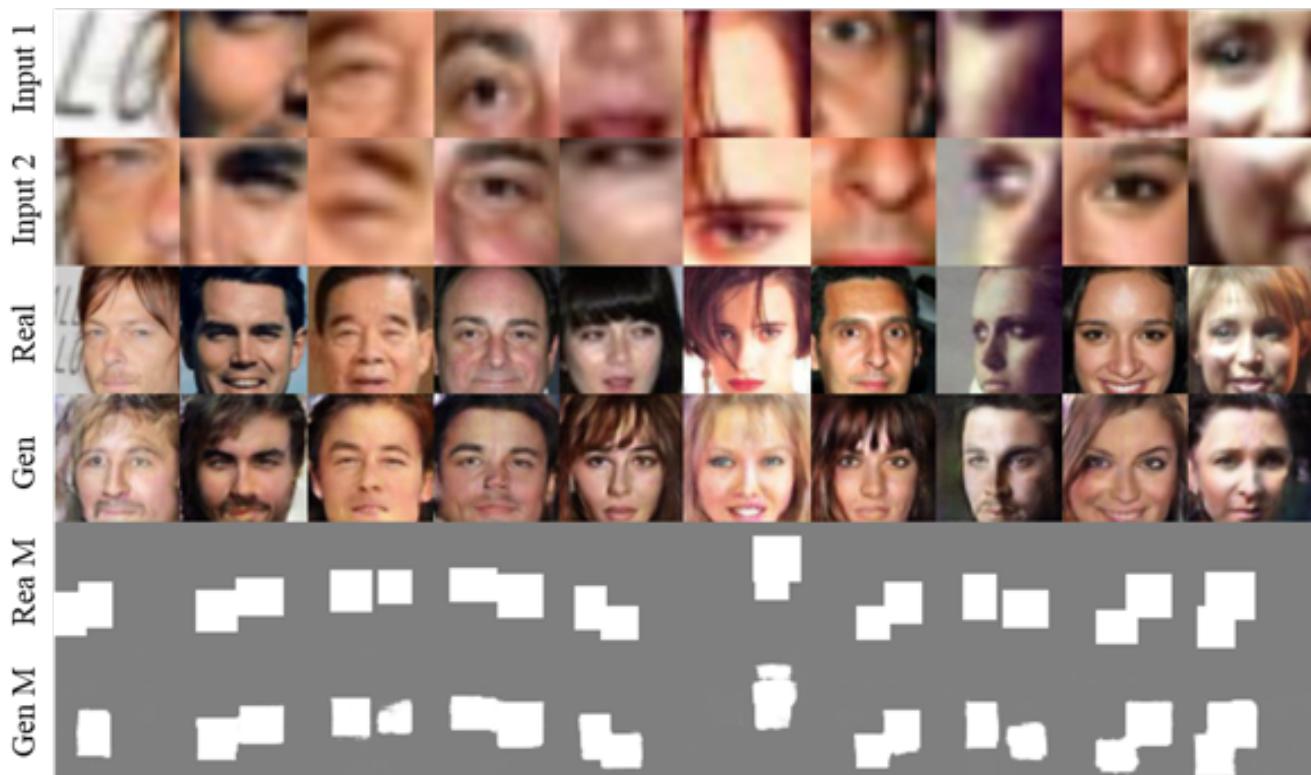
In order to demonstrate the effectiveness of (4) in the paper, we show generation results in Figure 14 using the following objective function:

$$\mathcal{L}_R(G_{\mathcal{I}}, D) = \mathbb{E}_{y \sim p_{data}(y)}[\log D(y)] + \mathbb{E}_{\mathbf{x}, y, y' \sim p_{data}(\mathbf{x}, y, y'), z \sim p_z(z), M \sim p_{data}(M)}[\log(1 - D(G_{\mathcal{I}}(\mathbf{x}, z)))] \quad (5)$$

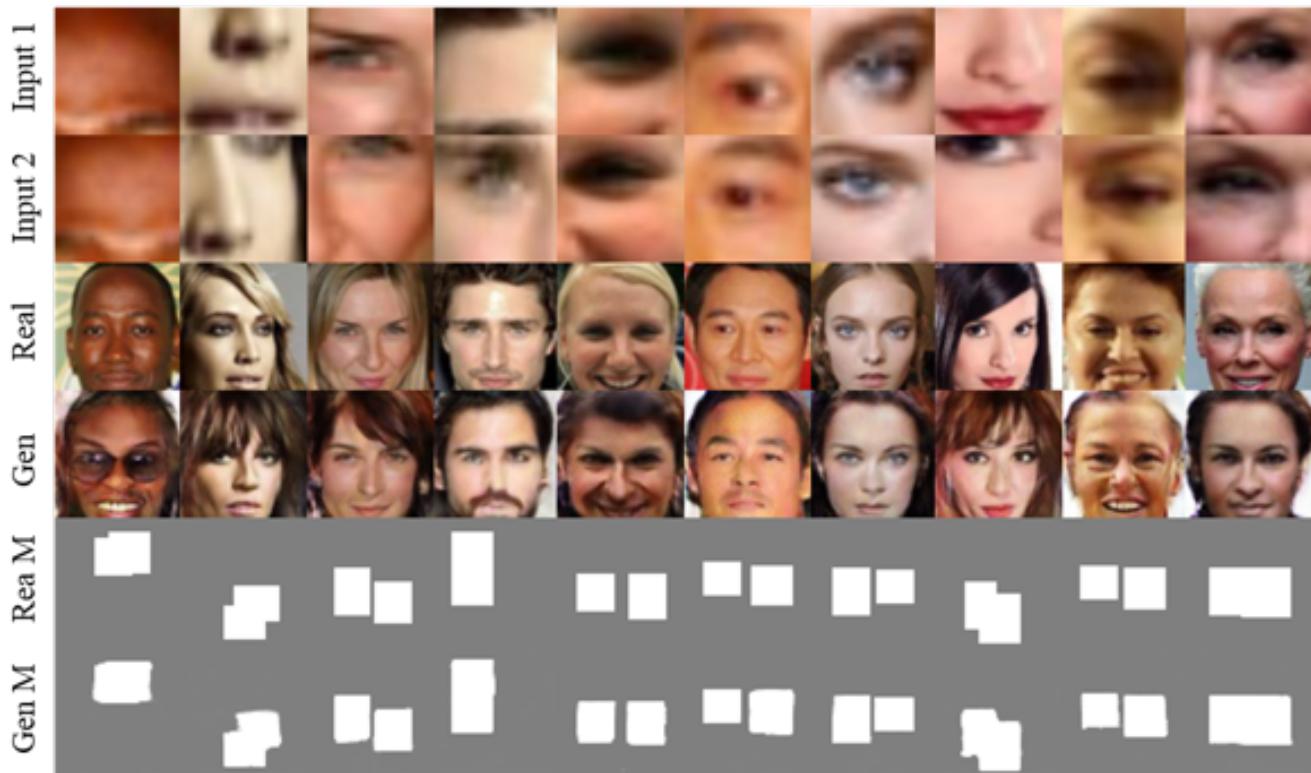
Both results are obtained after 25 epochs. The results show that generated images with (5) are less realistic compared to the results of (4) in the paper.



Figure 12: Results of the proposed algorithm on the CompCars dataset when input patches are from different cars. Input 1 and Input 2 are patches from Real 1. Input 3 is a local region of Real 2. Given inputs, the proposed algorithm generates the image (Gen) and mask (Gen M). The size of the generated image is of 128×128 pixels.



(a)



(b)



Figure 13: Image generation results with two input patches. Input 1 and 2 are local patches from the image Real.



(a)



(b)

Figure 14: Image generation results on the CelebA dataset. Gen 1 and GenM1 are generated by (5). Gen 2 and GenM2 are obtained using (4) in the paper.