DeepGIN: Deep Generative Inpainting Network for Extreme Image Inpainting

Chu-Tak Li[1], Wan-Chi Siu[1], Zhi-Song Liu[2], Li-Wen Wang[1], and Daniel Pak-Kong Lun[1]

[1] Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University
[2] LIX, Ecole Polytechnique, CNRS, IP Paris, France

Abstract. The degree of difficulty in image inpainting depends on the types and sizes of the missing parts. Existing image inpainting approaches usually encounter difficulties in completing the missing parts in the wild with pleasing visual and contextual results as they are trained for either dealing with one specific type of missing patterns (mask) or unilaterally assuming the shapes and/or sizes of the masked areas. We propose a deep generative inpainting network, named DeepGIN, to handle various types of masked images. We design a Spatial Pyramid Dilation (SPD) ResNet block to enable the use of distant features for reconstruction. We also employ Multi-Scale Self-Attention (MSSA) mechanism and Back Projection (BP) technique to enhance our inpainting results. Our DeepGIN outperforms the state-of-the-art approaches generally, including two publicly available datasets (FFHQ and Oxford Buildings), both quantitatively and qualitatively. We also demonstrate that our model is capable of completing masked images in the wild.

Keywords: Image Inpainting, Attention, Back Projection

1 Introduction

Image inpainting (also called image completion) is a task of predicting the values of missing pixels in a corrupted/masked image such that the completed image looks realistic and is semantically close to the reference ground truth even though it does not exist in real-world situations. This task would be useful for repairing corrupted photos or erasing unwanted parts from photos. It could also serve applications of restoration of photos and footage of films, scratch removal, automatic modifications to images and videos, and so forth. Because of the wide-ranging applications, image inpainting has been an overwhelming research topic in the computer vision and graphics communities for decades.

Inspired by the recent success of deep learning approaches at the tasks of image recognition [29,8], image super-resolution [6,19,30], visual place recognition and localization [16,2], image enlightening [27], image synthesis [11,28] and many others, a growing number of CNN based methods of image inpainting [23,32,10,33,17,21,34] have been proposed to fill images with holes in an end-to-end manner. For example, Iizuka et al. [10] employed dilated convolutions

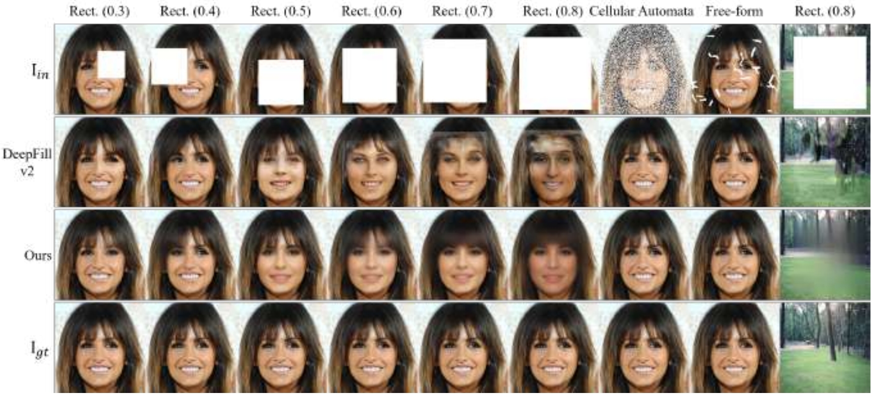


Fig. 1. Degree of difficulty in extreme image inpainting. From top to bottom: the first row shows the input masked images $\mathbf{I}_{in}$ with the corresponding mask described on top of them. Rect. (α) represents a random rectangular mask with the height and width rate of α of each dimension. The randomly generated mask based on cellular automata is introduced in the AIM 2020 Extreme Image Inpainting Challenge [1], and the free-form mask is proposed in DeepFillv2 [34]. The second and third rows are the completed images using DeepFillv2 and our proposed DeepGIN respectively. The last row displays the ground truth images. Please zoom in to see the examples at the 6th and the last column especially

instead of standard convolutions to widen the receptive field at each layer for better conservation of the spatial structure of an image. Yu et al. [33] proposed a two-stage generative network with a contextual attention layer to intentionally consider correlated feature patches at distant spatial locations for coherent estimation of local missing pixels. Liu et al. [17] suggested a partial convolutional layer to identify the non-hole regions at each layer such that the convolutional results are derived only from the valid pixels. However, the effectiveness of these strategies depends highly on the scales and forms of the missing regions as well as the contents of both the valid and invalid pixels as shown in Fig. 1. Based on this observation, we aim for a generalized inpainting network which can complete masked images in the wild.

In this paper, we present a coarse-to-fine Deep Generative Inpainting Network (DeepGIN) which consists of two stages, namely coarse reconstruction stage and refinement stage. Similar to the network design of previous studies [33,21,34], the coarse reconstruction stage is responsible for rough estimation of the missing pixels in an image while the refinement stage is responsible for detailed decoration on the coarse reconstructed image. In order to obtain a realistic and coherent completed image, Spatial Pyramid Dilation (SPD), Multi-Scale Self-Attention (MSSA) and Back Projection (BP) techniques are redesigned and embedded in our proposed network. The main function of SPD is to extensively allow for different receptive fields such that information gathered from both surrounding and distant spatial locations can contribute to the prediction of local missing regions. The concept of SPD is applied to both stages while MSSA and BP are integrated into the refinement stage. The core idea of MSSA is that it takes the self-similarity of the image itself at multiple levels into account for the coherence on the completed image while BP enforces the alignment of the predicted and given pixels in the completed image.

The contributions made in this work are summarized as follows:

- We propose a Spatial Pyramid Dilation (SPD) block to deal with different types of masks with various shapes and sizes.
- We stress the importance of self-similarity to image inpainting and we significantly improve our inpainting results by employing the strategy with Multi-Scale Self-Attention (MSSA).
- We design a Back Projection (BP) strategy for obtaining the inpainting results with better alignment of the generated patterns and the reference ground truth.

2 Related Work

Existing approaches to image inpainting can be classified into two categories, namely conventional and deep learning based methods. PatchMatch [3] is one of the representative conventional methods in which similar patches from a target image or other source images are copied and pasted into the missing regions of the target image. However, it is computationally expensive even a fast approximate nearest-neighbor search algorithm has been adopted to alleviate the problem of costly patch search iterations.

Regular Mask. Context Encoder [23] is the first deep learning based inpainting algorithm that employs the framework of Generative Adversarial Networks (GANs) [5] for more realistic image completion. For GAN based image inpainting, a generator is designed for filling the missing regions with semantic awareness and a discriminator is responsible for distinguishing the completed image and the reference ground truth. Based on this setting, the generator and discriminator are alternately optimized to compete against each other, as a result, the completed image given by the generator would be visually and semantically close to the reference ground truth. Specifically, Pathak et al. [23] resize images to 128×128 and assume a 64×64 rectangular center missing region for the task of inpainting. The encoded feature of the image with the center hole is then decoded to reconstruct a 64×64 image for the center hole.

Based on this early work, Yang et al. [32] attached the pre-trained VGG-19 [26] as their proposed texture network to perform the task in a coarse-to-fine manner. The output of the context encoders is then passed to the texture network for local texture refinement to enhance the accuracy of the reconstructed hole content. Iizuka et al. [10] suggested an approach in which two auxiliary discriminators are designed for ensuring both the global and local consistency of the completed image. The global discriminator takes the entire image as input for differentiation between real and completed images while the local discriminator examines only the local area around the filled region. To further alleviate the visual artifacts in the filled region, they also employed Poisson image blending as a simple post-processing. Expanding on this idea and the above-mentioned idea of PatchMatch [3], Yu et al. [33] also adopted a two-stage coarse-to-fine approach with global and local discriminators. A contextual attention layer is proposed and applied to the second refinement network which plays the similar role of the post-processing.

Irregular Mask. For the early stage of deep learning based methods of image inpainting, authors focused on the rectangular types of masks and this assumption limits the effectiveness of these methods in real-world situations. Liu et al. [17] addressed this problem by suggesting a partial convolutional layer, in which a binary mask for indicating the missing regions is automatically updated along with the convolutional operations inside their model for guiding the reconstruction. Nazeri et al. [21] forced an image completion network to generate images with fine-grained details by providing guidance for filling the missing regions with the use of their proposed edge generator. The edge generator is responsible for predicting a full edge map of the masked image. With the estimated edge map as additional information, their trained model can be extended to an interactive image editing tool in which users can sketch the outline of the missing regions to obtain tailor-made completed images. Combining the concept of partial convolution with optional user-guided image inpainting, Yu et al. [34] improved their previous model [33] by proposing gated convolution for free-form image inpainting. They modify the hard-assigned binary mask in partial convolution to a learnable soft-gated convolutional layer. The soft gating layer can be achieved by using convolutional filters with size 1×1 followed by a sigmoid function. However, additional soft gating layers introduce additional parameters and the effectiveness still depends on the scales of the masked areas.

Our work echoes the importance of information given by distant spatial locations and self-similarity of the image itself to image inpainting. We increase the number of receptive fields and apply multi-scale self-attention strategy to handle various types of masks in the wild. Our multi-scale self-attention strategy is derived from the non-local network [29], in which the correlation between features is emphasized and it has been used in image super-resolution [6,18]. For achieving better coherency of the completed images, we also adopt the back projection technique [7,18] to encourage better alignment of the generated and valid pixels. We weight the back projected residual instead of using the parametric back projection blocks as in [7,18] to avoid more additional parameters.

3 Problem Formulation

Let us start to define an input RGB masked image and a binary mask image as $\mathbf{I}_{in} \in \mathbb{R}^{H \times W \times 3}$ and $\mathbf{M} \in \mathbb{R}^{H \times W}$ respectively. The pixel values input to our model are normalized between 0 and 1 and pixels with value 1 in $\mathbf{M}$ represent the masked regions. $\mathbf{I}_{coarse} \in \mathbb{R}^{H \times W \times 3}$ denotes the output of our coarse generator G_1 at the first coarse reconstruction stage. We also define the output of our refinement generator G_2 at the second refinement stage and the reference ground truth image as $\mathbf{I}_{out} \in \mathbb{R}^{H \times W \times 3}$ and $\mathbf{I}_{gt} \in \mathbb{R}^{H \times W \times 3}$ respectively. Note that H and W are the height and width of an input/output image and we fix the input to 256×256 for inpainting. Our objective is straightforward. We would

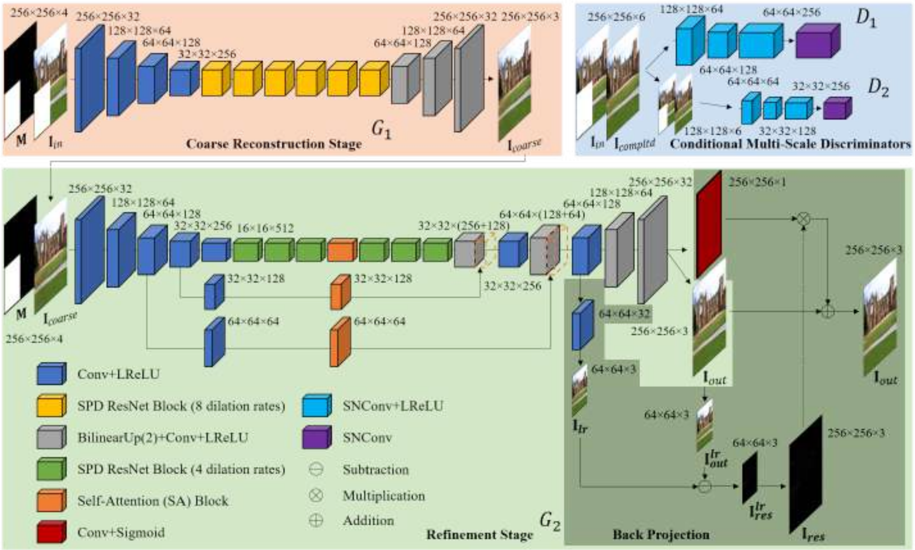
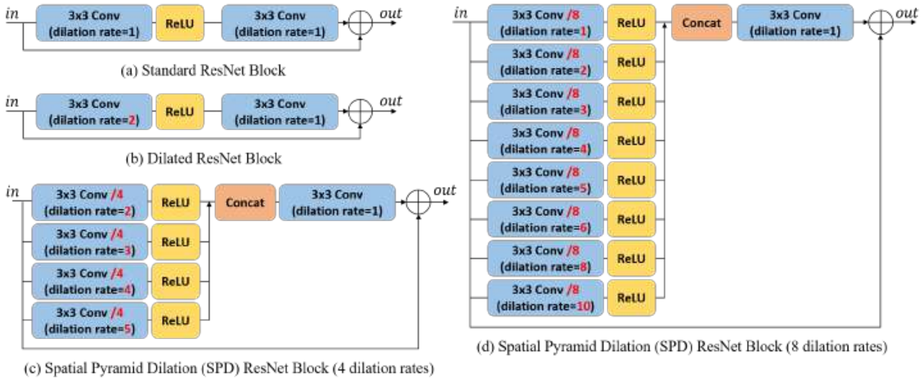


Fig. 2. Architecture of our proposed model for image inpainting. Our proposed model consists of two generators and two discriminators. The coarse generator G_1 at Coarse Reconstruction Stage and the second refinement generator G_2 at Refinement Stage constitute our DeepGIN which is used in both training and testing. The two discriminators D_1 and D_2 located within Conditional Multi-Scale Discriminators area are only used in training as an auxiliary network for generative adversarial training

like to complete $\mathbf{I}_{in}$ conditioned on $\mathbf{M}$ and produce a completed image $\mathbf{I}_{out}$ ($\mathbf{I}_{compltd}$) which should be both visually and semantically close to the reference ground truth $\mathbf{I}_{gt}$. $\mathbf{I}_{compltd}$ is the same as $\mathbf{I}_{out}$ except the valid pixels are directly replaced by the ground truth. We propose a coarse-to-fine network trained under the framework of generative adversarial learning with training data $\{\mathbf{I}_{in}, \mathbf{M}, \mathbf{I}_{gt}\}$ where $\mathbf{M}$ is randomly generated with arbitrary sizes and shapes. Generator G_1 takes $\mathbf{I}_{in}$ and $\mathbf{M}$ as input and generates $\mathbf{I}_{coarse}$ as output. Subsequently, we feed $\mathbf{I}_{coarse}$ and $\mathbf{M}$ to generator G_2 to obtain the completed image $\mathbf{I}_{out}$ ($\mathbf{I}_{compltd}$).

4 Approach

Our proposed Deep Generative Inpainting Network (DeepGIN) consists of two stages as shown in Fig. 2, a coarse reconstruction stage and a refinement stage. The first coarse generator $G_1(\mathbf{I}_{in}, \mathbf{M})$ is trained to roughly reconstruct the masked regions and gives $\mathbf{I}_{coarse}$. The second refinement generator $G_2(\mathbf{I}_{coarse}, \mathbf{M})$ is trained to exquisitely decorate the coarse prediction with details and textures, and eventually forms the completed image $\mathbf{I}_{out}$ ($\mathbf{I}_{compltd}$). For our discriminators, motivated by SN-GANs [20,34] and multi-scale discriminators [11,28], we modify and employ two SN-GAN based discriminators $D(\mathbf{I}_{in}, \mathbf{I}_{compltd})$ which operate at two image scales, 256×256 and 128×128 respectively, to encourage better details and textures of local reconstructed patterns at different scales. Details of our network architecture and learning are shown below.


in
3x3 Conv (dilation rate=1)
ReLU
3x3 Conv (dilation rate=1)
out
in
3x3 Conv (dilation rate=2)
ReLU
3x3 Conv (dilation rate=1)
out
in
3x3 Conv /4 (dilation rate=2)
3x3 Conv /4 (dilation rate=3)
3x3 Conv /4 (dilation rate=4)
3x3 Conv /4 (dilation rate=5)
ReLU
Concat
3x3 Conv (dilation rate=1)
out
in
3x3 Conv /8 (dilation rate=1)
3x3 Conv /8 (dilation rate=2)
3x3 Conv /8 (dilation rate=3)
3x3 Conv /8 (dilation rate=4)
3x3 Conv /8 (dilation rate=5)
3x3 Conv /8 (dilation rate=6)
3x3 Conv /8 (dilation rate=8)
3x3 Conv /8 (dilation rate=10)
ReLU
Concat
3x3 Conv (dilation rate=1)
out

(a) Standard ResNet Block

(b) Dilated ResNet Block

(c) Spatial Pyramid Dilation (SPD) ResNet Block (4 dilation rates)

(d) Spatial Pyramid Dilation (SPD) ResNet Block (8 dilation rates)

Fig. 3. Variations of ResNet Block. From top to bottom, left to right: (a) Standard ResNet block [8], (b) Dilated ResNet block used in [10,33,21] which adopts a dilation rate of 2 of the first convolutional layer, (c) The proposed SPD ResNet block with 4 dilation rates and (d) 8 dilation rates. To avoid additional parameters, we split the number of input feature channels into equal parts according to the number of dilation rates employed. As shown in (c), if 4 dilation rates are used, the output channel size of the first convolutions equals a quarter of the input channel size

4.1 Network Architecture

Coarse Reconstruction Stage. Recall that G_1 is our coarse generator and it is responsible for rough estimation of the missing pixels in a masked image. Referring to the previous section, we concatenate $(\mathbf{I}_{in}, \mathbf{M}) \in \mathbb{R}^{H \times W \times (3+1)}$ as the input to G_1 and then obtain the coarse image $\mathbf{I}_{coarse}$. G_1 follows an encoder-decoder structure. As the scales of the masked regions are randomly determined, we propose a Spatial Pyramid Dilation (SPD) ResNet block with various dilation rates to enlarge the receptive fields such that information given by distant spatial locations can be included for reconstruction. Our SPD ResNet block is an improved version of the original ResNet block [8] as shown in Fig. 3, and in total, 6 SPD ResNet blocks with 8 different dilation rates are used at this stage.

Refinement Stage. Generator G_2 is designed for refinement of $\mathbf{I}_{coarse}$ and it is similar to generator G_1. At this stage, we have 6 SPD ResNet blocks with 4 different dilation rates and a Self-Attention (SA) block in between at the middle layers. Apart from the SPD ResNet block, Multi-Scale Self-Attention (MSSA) blocks [29,18] are used for self-similarity consideration. The SA block used in this paper is exactly the same as the one proposed in [29]. One similarity between the SA block and the contextual attention layer [33,34] is that they both have the concept of self-similarity which is useful for amending the reconstructed patterns based on the remaining ground truth in a masked image. We apply MSSA instead of single scale SA to enhance the coherency of the completed image $\mathbf{I}_{out}$ by attending on the self-similarity of the image itself at three different scales, namely 16×16, 32×32 and 64×64 as shown in Fig. 2. To avoid an excessive increase in additional parameters, we simply use standard convolutional layers to reduce the channel size before connecting to the SA blocks. The idea of Back

Projection (BP) [7,18] is also redesigned and it is used at the last decoding process of this stage (see the shaded Back Projection region in Fig. 2). At the layer with spatial size of 64×64, we output a low-resolution (LR) completed image $\mathbf{I}_{lr}$ and perform BP with $\mathbf{I}_{out}$. By learning to weight the BP residual and adding it back to update $\mathbf{I}_{out}$, the generated patterns can have better alignments with the reference ground truth and hence $\mathbf{I}_{out}$ looks more coherent.

Conditional Multi-Scale Discriminators. Two discriminators D_1 and D_2 at two input scales (i.e. 256×256 and 128×128) are trained together with the generators to stimulate details of the filled regions. Combining the idea of multi-scale discriminators [28] with SN-GANs [20] and PatchGAN [11,34], our $D_1(\mathbf{I}_{in}, \mathbf{I})$ and $D_2(\mathbf{I}_{in}, \mathbf{I})$ take the concatenation result of two RGB images as input ($\mathbf{I}$ is either $\mathbf{I}_{compltd}$ or $\mathbf{I}_{gt}$, recall that $\mathbf{I}_{compltd}$ is the same as $\mathbf{I}_{out}$ except the valid pixels are directly replaced by the ground truth) and output a set of feature maps with size of $H/2^2 \times W/2^2 \times c$ where c represents the number of feature maps. Note that each value on these output feature maps represents a local region in the input image at two different scales. By training D_1 and D_2 to discriminate between real and fake local regions, $\mathbf{I}_{out}$ would gradually be close to its reference ground truth $\mathbf{I}_{gt}$ in terms of both appearance and semantic similarity. For achieving stable generative adversarial learning, we employ the spectral normalization layer described in [20] after each convolutional layer in D_1 and D_2.

4.2 Network Learning

We design our loss function based on consideration to both quantitative accuracy and visual quality of the completed images. Our loss function consists of five major terms, namely (i) a *L1 loss* to ensure the pixel-wise reconstruction accuracy especially if using quantitative evaluation metrics such as PSNR and mean L1 error to evaluate the completed images; (ii) an *adversarial loss* to urge the distribution of the completed images to be close to the distribution of the real images; (iii) the *feature perceptual loss* used in [12] that encourages each completed image and its reference ground truth image to have similar feature representations as computed by a well-trained network with good generalization like VGG-19 [26]; (iv) the *style loss* [4] to emphasize the style similarity such as textures and colors between completed images and real images; and (v) the *total variation loss* used as a regularizer in [12] to guarantee the smoothness in the completed images by penalizing its visual artifacts or discontinuities.

L1 Loss. Our *L1 loss* is derived from three image pairs, namely $\mathbf{I}_{coarse}$ and $\mathbf{I}_{gt}$; $\mathbf{I}_{out}$ and $\mathbf{I}_{gt}$; and $\mathbf{I}_{lr}$ and $\mathbf{I}_{gt}^{lr}$. Note that $\mathbf{I}_{gt}^{lr}$ is obtained by down-sampling $\mathbf{I}_{gt}$ by 4 times. We sum the L1-norm distances of these three image pairs and define our *L1 loss*, $\mathcal{L}_{L1}$, as follows:

$$\mathcal{L}_{L1} = \lambda_{hole}\mathcal{L}_{hole} + \mathcal{L}_{valid} \tag{1}$$

where $\mathcal{L}_{hole}$ and $\mathcal{L}_{valid}$ are the sums of the distances which are calculated only from the missing pixels and the valid pixels respectively. λ_{hole} is a weight to the pixel-wise loss within the missing regions.

Adversarial Loss. For generative adversarial learning, our discriminators are trained to rightly distinguish $\mathbf{I}_{compltd}$ from $\mathbf{I}_{gt}$ while our generators strive to cheat the discriminators of incorrect classification. We employ the hinge loss to train our model, $\mathcal{L}_{adv,G}$ and $\mathcal{L}_{adv,D}$ are computed as:

$$\mathcal{L}_{adv,G} = -\mathbb{E}_{\mathbf{I}_{in}\sim\mathbb{P}_i}[D_1(\mathbf{I}_{in}, \mathbf{I}_{compltd})] - \mathbb{E}_{\mathbf{I}_{in}\sim\mathbb{P}_i}[D_2(\mathbf{I}_{in}, \mathbf{I}_{compltd})] \tag{2}$$

$$\mathcal{L}_{adv,D} = \mathbb{E}_{\mathbf{I}_{in}\sim\mathbb{P}_i}\left[\sum_{d=1}^{2}[\mathrm{ReLU}(1 - D_d(\mathbf{I}_{in}, \mathbf{I}_{gt})) + \mathrm{ReLU}(1 + D_d(\mathbf{I}_{in}, \mathbf{I}_{compltd}))]\right] \tag{3}$$

where $\mathbb{P}_i$ represents the data distribution of $\mathbf{I}_{in}$, ReLU is the rectified linear unit defined as $f(x) = \max(0, x)$.

Perceptual Loss. Let ϕ be the well-trained loss network, VGG-19 [26], and $\phi_l^{\mathbf{I}}$ be the activation maps of the l^{th} layer of the network ϕ given an image $\mathbf{I}$. We choose five layers of the pre-trained VGG-19, namely *conv*1_1, *conv*2_1, *conv*3_1, *conv*4_1, and *conv*5_1 for computing this loss. Our $\mathcal{L}_{perceptual}$ is calculated as:

$$\mathcal{L}_{perceptual} = \sum_{l=1}^{L} \frac{\|\phi_l^{\mathbf{I}_{out}} - \phi_l^{\mathbf{I}_{gt}}\|_1}{N_{\phi_l^{\mathbf{I}_{gt}}}} + \sum_{l=1}^{L} \frac{\|\phi_l^{\mathbf{I}_{compltd}} - \phi_l^{\mathbf{I}_{gt}}\|_1}{N_{\phi_l^{\mathbf{I}_{gt}}}} \tag{4}$$

where $N_{\phi_l^{\mathbf{I}_{gt}}}$ indicates the number of elements in $\phi_l^{\mathbf{I}_{gt}}$ and L equals 5 as five layers are used. Here, we compute the L1-norm distance between the high-level feature representations of $\mathbf{I}_{out}$, $\mathbf{I}_{compltd}$ and $\mathbf{I}_{gt}$ given by the network ϕ.

Style Loss. Let $(\phi_l^{\mathbf{I}})^\top(\phi_l^{\mathbf{I}})$ be the Gram matrix [4] which computes the feature correlations between each activation map of the l^{th} layer of ϕ given $\mathbf{I}$, and this is also called auto-correlation matrix. We then calculate the *style loss* ($\mathcal{L}_{style}$) using $\mathbf{I}_{out}$, $\mathbf{I}_{compltd}$ and $\mathbf{I}_{gt}$ as:

$$\mathcal{L}_{style} = \sum_{\mathbf{I}}^{\mathbf{I}_{out},\mathbf{I}_{compltd}} \sum_{l=1}^{L} \frac{1}{C_l C_l} \left\| \frac{1}{C_l H_l W_l}((\phi_l^{\mathbf{I}})^\top(\phi_l^{\mathbf{I}}) - (\phi_l^{\mathbf{I}_{gt}})^\top(\phi_l^{\mathbf{I}_{gt}})) \right\|_1 \tag{5}$$

where C_l denotes the number of activation maps of the l^{th} layer of ϕ. H_l and W_l are the height and width of each activation map of the l^{th} layer of ϕ. Note that we use the same five layers of the VGG-19 as mentioned for this loss as well.

Total Variation (TV) Loss. We also adopt the total variation regularization to ensure the smoothness in $\mathbf{I}_{compltd}$.

$$\mathcal{L}_{tv} = \sum_{x,y}^{H-1,W} \frac{\|\mathbf{I}_{compltd}^{x+1,y} - \mathbf{I}_{compltd}^{x,y}\|_1}{N_{\mathbf{I}_{compltd}}^{row}} + \sum_{x,y}^{H,W-1} \frac{\|\mathbf{I}_{compltd}^{x,y+1} - \mathbf{I}_{compltd}^{x,y}\|_1}{N_{\mathbf{I}_{compltd}}^{col}} \tag{6}$$

where H and W are the height and width of $\mathbf{I}_{compltd}$. $N_{\mathbf{I}_{compltd}}^{row}$ and $N_{\mathbf{I}_{compltd}}^{col}$ are the number of pixels in $\mathbf{I}_{compltd}$ except for the last row and the last column respectively.

Total Loss. Our total loss function for the generators is the weighted sum of the five major loss terms:

$$\mathcal{L}_{total} = \mathcal{L}_{L1} + \lambda_{adv}\mathcal{L}_{adv,G} + \lambda_{perceptual}\mathcal{L}_{perceptual} + \lambda_{style}\mathcal{L}_{style} + \lambda_{tv}\mathcal{L}_{tv} \quad (7)$$

where λ_{adv}, $\lambda_{perceptual}$, λ_{style}, and λ_{tv} are the hyper-parameters which indicate the significance of each term.

5 Experimental Work

We have participated in the AIM 2020 Extreme Image Inpainting Challenge [1] of the ECCV 2020 (please find in our github page for details and qualitative results of the challenge). In designing our proposed model, we take reference to the networks in [12,11,28]. We have attached our improved SPD ResNet block to our DeepGIN. We have also modified and applied the ideas of MSSA and BP in our proposed model. Inspired by ESRGAN [30], we remove all batch normalization layers in the model to smooth out the related visual artifacts. We have used discriminators at two different scales which share the same architecture. Also, we have adjusted the number of layers of each discriminator and applied spectral normalization layers [20,34] after the convolutional layers for training stability.

5.1 Training Procedure

Random Mask Generation. Three different types of masks are used in our training. The first type is a rectangular mask with the height and width between 30-70% of each dimension [1,23,32,10,33]. The second type is the free-form mask proposed in [34]. The third type of masks is introduced in the AIM 2020 Image Inpainting Challenge [1], for which masks are randomly generated based on cellular automata. During training, each mask was randomly generated and we applied the three types of masks to each training image to get three different masked images. We observed that this can balance the three types of masks to achieve more stable training.

Training Batch Formation. As the size of training images could be very diverse, we resized all training images to the size of 512×512 and adopted a subsampling method [25] to randomly select a sub-image with size of 256×256. We then apply the random mask generation as stated above to obtain three masked images. Therefore, each training image becomes three training images. We set a batch size of 4 and this means that there are 12 training images in a batch.

Two-Stage Training. Our training process is divided into two stages, namely a warm-up stage and then the main stage. First, we trained only the generators by using the *L1 loss* for 10 epochs. We used the initialization method mentioned in [30], using a smaller initialization for ease of training a very deep network. The trained model at the warm-up stage was used as an initialization for the main stage. This *L1*-oriented pre-trained model provides a reasonable initial point for training GANs, for which a balance between quantitative accuracy of the reconstruction and visual quality of the output is required. For the main stage, we trained the generators alternately with the discriminators for 100 epochs. We used Adam [15] with momentum 0.5 for both stages. The initial learning rates for generators and discriminators were set to 0.0001 and 0.0004 respectively. We trained them for 10 epochs with the initial rates and linearly decayed the rates to zero over the last 90 epochs. The hyper-parameters of the loss terms in Eq. (1) and Eq. (7) were set to λ_{hole} = 5.0, λ_{adv} = 0.001, $\lambda_{perceptual}$ = 0.05, λ_{style} = 80.0, and λ_{tv} = 0.1. We developed our model using Pytorch 1.5.0 [22] and trained it on two NVIDIA GeForce RTX 2080Ti GPUs.

5.2 Training Data

ADE20K Dataset. We trained our model on the subset of ADE20K dataset [36,37] for participating in the AIM challenge [1]. This dataset is collected for scene parsing and understanding, in which it contains images from various scene categories. The subset is provided by the organizers of the challenge and it consists of 10,330 training images with diverse resolutions roughly, from 256×256 to 3648×2736. We took around two and a half days for training on this dataset.

CelebA-HQ Dataset. Beyond the ADE20K dataset, we also trained our model on the CelebA-HQ dataset [13] that contains 30K high-quality face images with a standard size of 1024×1024. We randomly split this dataset into two groups, 27,000 images for training and 3,000 images for testing. This required approximately 6 days to train our model on this dataset.

6 Analysis of Experimental Results

We have thoroughly evaluated our proposed model. We first provide evidence in our model analysis to show the effectiveness of our suggested strategies for using Spatial Pyramid Dilation (SPD) ResNet block, Multi-Scale Self-Attention (MSSA), and Back Projection (BP). We then compare our model with state-of-the-art approaches, namely DeepFillv1 [33] and DeepFillv2 [34], which are known to have a good generalization. We demonstrate that our model is able to handle images in the wild by testing it on two publicly available datasets, namely Flickr-Faces-HQ (FFHQ) dataset [14] and The Oxford Buildings (Oxford) dataset [24]. Related materials are available at: `https://github.com/rlct1/DeepGIN`.

6.1 Model Analysis

We first evaluate the effectiveness of the three proposed strategies, namely SPD, MSSA, and BP. Refer to the proposed architecture as shown in Fig. 2, our baselines are denoted as StdResBlk (Coarse only, using only the Coarse Reconstruction Stage) and StdResBlk (a conventional ResNet for inpainting), for which all SA blocks and BP branch are eliminated and all SPD ResNet blocks are replaced by standard ResNet blocks (see Fig. 3(a)). DilatedResBlk or SPDResBlk represents StdResBlk with standard ResNet blocks replaced by Dilated or SPD ResNet blocks. Please refer to Fig. 3(b),(c), and (d). SA or MSSA indicates whether single SA block or MSSA is used and the use of BP is denoted as BP. We conducted the model analysis on CelebA-HQ dataset [13] using the 3K testing images. Note that the testing images were randomly masked by the three types of masks and the same set of masked images was used for each variation of our model. During testing, for images with size larger than 256×256, we divided

Table 1. Model analysis of our proposed model on CelebA-HQ dataset. The best results are in **bold** typeface

Variations of our model	Number of parameters	PSNR	SSIM	L1 err.(%)	FID	LPIPS
StdResBlk (Coarse only)	8.168M	31.55	0.925	4.690	23.824	0.182
StdResBlk	40.850M	31.34	0.923	4.710	19.436	0.191
StdResBlk-SA	41.376M	31.60	0.925	4.510	18.239	0.180
StdResBlk-MSSA	42.892M	32.66	0.933	4.067	12.843	0.148
DilatedResBlk-MSSA	42.892M	32.71	0.933	4.034	12.548	0.149
SPDResBlk-MSSA	42.892M	32.88	0.935	3.884	12.335	0.143
SPDResBlk-MSSA-BP	42.930M	**33.26**	**0.939**	**3.666**	**11.424**	**0.132**

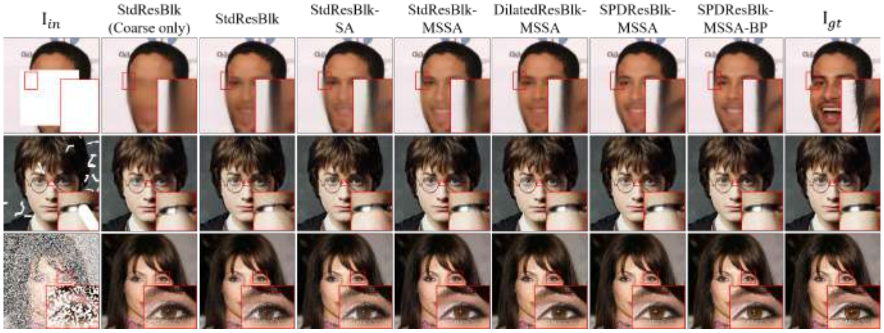


Fig. 4. Comparisons of test results of the variations of our model on CelebA-HQ dataset. Three different types of masked images are displayed from top to bottom. The first and the last columns show $\mathbf{I}_{in}$ and $\mathbf{I}_{gt}$ respectively. The variations of our model are indicated on top of Fig.4. Our full model (the 8th column), SPDResBlk-MSSA-BP (GAN based), provides high quality results with both the best similarity and visual quality to the ground truth images. Please zoom in for a better view

the input into a number of 256×256 sub-images using the sub-sampling method [25] and obtained the completed sub-images. We then regrouped the sub-images to form the completed image by using the reverse sub-sampling method. We finally replaced the valid pixels by the ground truth.

Quantitative Comparisons. As the lack of good quantitative evaluation metric for inpainting [33,17,34], we report several numerical metrics which are commonly used in image manipulation, namely PSNR, SSIM [31], mean L1 error, Fréchet Inception Distance (FID) [9], and Learned Perceptual Image Patch Similarity (LPIPS) [35], for a comprehensive analysis of the performance. The results are listed in Table 1 and higher PSNR, SSIM and smaller L1 err. mean better pixel-wise reconstruction accuracy. FID and LPIPS are also used to estimate the visual quality of the output, the smaller the better. It is obvious that our full model, SPDResBlk-MSSA-BP, gives the best performance on these numerical metrics. The employment of MSSA brings an 1.06 dB increase in PSNR compared to StdResBlk-SA. This reflects the importance of multi-scale self-similarity to inpainting. Our SPD ResNet blocks and the adoption of BP also bring about 0.22 dB and 0.38 dB improvement in PSNR respectively.

Qualitative Comparisons. Fig. 4 shows the comparisons of the variations of our model on CelebA-HQ dataset. Without the second refinement stage, the completed images lack for facial details like the first example of the 2nd column

Table 2. Comparisons of DeepFillv1 [33] and DeepFillv2 [34] on both FFHQ and Oxford datasets with two sets of masked images. One set only contains the rectangular masks while another set includes all the three types of masks. Our DeepGIN is denoted as Ours (i.e. the full model, SPDResBlk-MSSA-BP in the previous section). (OS) and (256) mean that the testing images are with the original sizes and size of 256×256 respectively. The best results are in **bold** typeface

Method	PSNR	SSIM	L1 err.(%)	FID	LPIPS
Flickr-Faces-HQ Dataset (FFHQ), random rectangular masks					
DeepFillv1 (OS)	20.22	0.872	16.523	97.630	0.173
DeepFillv2 (OS)	20.95	0.903	14.607	92.070	0.170
Ours (OS)	**26.05**	**0.923**	**7.183**	**20.849**	**0.137**
DeepFillv1 (256)	21.55	0.836	13.631	26.276	0.144
DeepFillv2 (256)	22.52	0.845	12.029	**19.336**	**0.128**
Ours (256)	**24.36**	**0.867**	**9.797**	37.577	0.142
The Oxford Buildings Dataset (Oxford), random rectangular masks					
DeepFillv1 (OS)	19.20	0.767	20.322	67.193	0.187
DeepFillv2 (OS)	18.58	0.766	21.204	77.636	0.192
Ours (OS)	**21.92**	**0.861**	**12.067**	**63.744**	**0.170**
DeepFillv1 (256)	19.49	0.795	16.851	**58.588**	**0.169**
DeepFillv2 (256)	18.88	0.789	18.308	66.615	0.174
Ours (256)	**21.90**	**0.819**	**12.995**	74.866	0.185
Flickr-Faces-HQ Dataset (FFHQ), random three types of masks					
DeepFillv1 (OS)	25.12	0.839	11.363	64.534	0.232
DeepFillv2 (OS)	29.70	0.912	7.994	36.940	0.188
Ours (OS)	**32.36**	**0.929**	**4.071**	**14.327**	**0.156**
DeepFillv1 (256)	22.87	0.683	16.812	80.952	0.310
DeepFillv2 (256)	22.75	0.716	17.472	75.555	0.293
Ours (256)	**24.71**	**0.760**	**13.417**	**64.542**	**0.274**
The Oxford Buildings Dataset (Oxford), random three types of masks					
DeepFillv1 (OS)	21.48	0.741	23.460	61.958	0.237
DeepFillv2 (OS)	24.68	0.802	19.195	38.315	**0.179**
Ours (OS)	**27.57**	**0.871**	**7.268**	**38.016**	0.191
DeepFillv1 (256)	21.64	0.686	18.835	81.009	0.284
DeepFillv2 (256)	20.80	0.702	20.687	82.671	0.266
Ours (256)	**23.60**	**0.744**	**14.659**	**79.927**	**0.265**

in Fig. 4. It can also be observed that the use of MSSA greatly enhances the visual quality as compared to the two which are without SA block and with only a single SA block (see the 3rd and 4th columns). Apart from this, with the SPD ResNet blocks and BP technique, the completed images are with better color coherency and alignment of the generated features. For example, see the spectacle frames in the 2nd row.

6.2 Comparison With Previous Works

In order to test the generalization of our model, we compare our best model against some state-of-the-art approaches, DeepFillv1 [33] and DeepFillv2 [34], on the two publicly available datasets, FFHQ [14] and Oxford Buildings [24]. It is worth noting that both DeepFillv1 and v2 are known to have good generalization for dealing with images in the wild. We directly used their provided pre-trained models[3] for comparison. The FFHQ dataset is similar to the CelebA-HQ dataset and it contains 70K high-quality face images at 1024×1024 resolution. We randomly selected 1,000 images for the testing on this dataset. For the Oxford dataset, it consists of 5,062 images of Oxford landmarks with a wide variety of styles. The images include buildings, suburban areas, halls, people, etc. We also randomly selected 523 testing images on this dataset for comparison.

[3] `https://github.com/JiahuiYu/generative_inpainting`

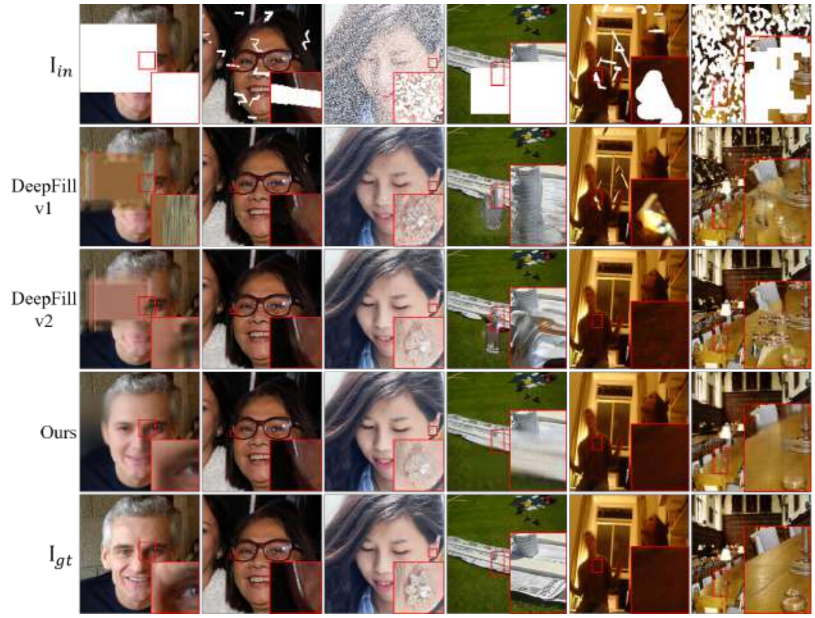


Fig. 5. Comparisons of test results on FFHQ and Oxford Buildings datasets. Each column shows an example of the test results. From top to bottom: the first row displays various masked input images from both datasets. The second to the fourth rows show the completed images by DeepFillv1, v2 and our DeepGIN. The reference ground truth images are also provided at the last row. Zoom in for a better view

Similarly, testing images were randomly masked by the three types of masks. For DeepFillv1 and v2, the authors divided an image into a number of grids to perform inpainting and indicated that their models were trained with images of resolution 256×256. Note also that DeepFillv1 was trained only for the rectangular types of masks. For fair comparison, we also conducted experiments in which testing images were randomly masked only by the rectangular masks.

Quantitative Comparisons. Table 2 shows the comparisons with DeepFillv1 and v2 on the two datasets with two sets of masked images. It is clear that our model outperforms DeepFillv1 and v2 in all the experiments on the two datasets in terms of the pixel-wise reconstruction accuracy. Our model achieves better PSNR compared with DeepFillv1 and v2 in the range of 1.84∼5.1 dB and offers better SSIM and L1 error. For FID and LPIPS, we attain better performance on the testing images with the original sizes. For the testing images with size of 256×256 and masked by random rectangular masks, we are also comparable to the other two approaches.

Qualitative Comparisons. Fig. 5 displays the test results on both FFHQ and Oxford datasets. It can be seen that DeepFillvl and v2 fail to achieve satisfactory visual quality on the large rectangular masks as shown in the first and fourth columns in Fig. 5. For the other two types of masked images, our model also provides the completed images with better color and content coherency. Note that our model tends to produce blurry images and the reason is that our model

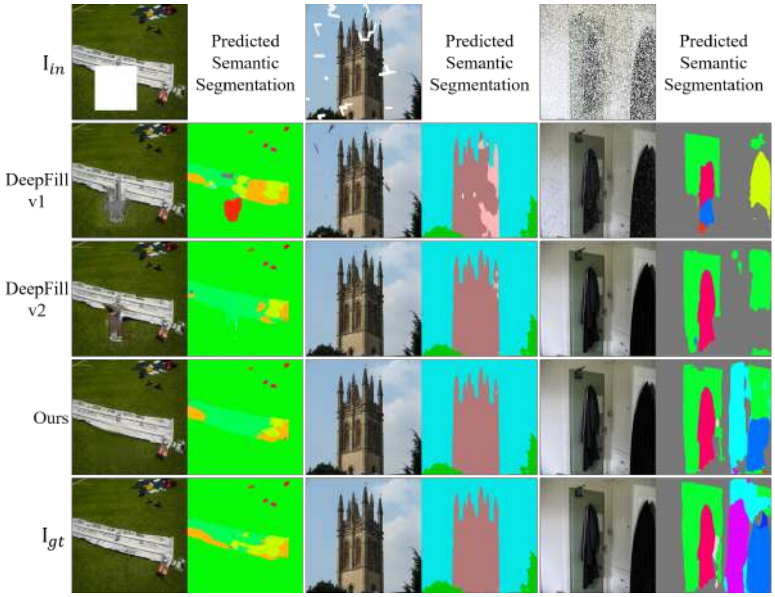

I_{in}
Predicted Semantic Segmentation
Predicted Semantic Segmentation
Predicted Semantic Segmentation
DeepFill v1
DeepFill v2
Ours
I_{gt}

Fig. 6. Visualizations of predicted semantic segmentation test results on Oxford Buildings dataset. 2^{nd} to 4^{th} rows show the completed images by different methods and the corresponding predicted semantic segmentation obtained using the trained network [37]. The ground truth images are also attached to the last row for reference. Please zoom in for a better view

was trained to be more PSNR-oriented than the DeepFillv1 and v2. We seek a balance between the pixel-wise accuracy and the visual quality to avoid some strange generated patterns like the completed image by DeepFillv2 of the last example in Fig. 5. To show that our model offers better pixel-wise accuracy, we provide the predicted semantic segmentation test results in Fig. 6. It is obvious that our results are semantically closer to $\mathbf{I}_{gt}$ than that of the other two methods, see for example, the intersection of the newspaper and the lawn in Fig. 6.

7 Conclusions

We have presented a deep generative inpainting network, called DeepGIN. Unlike the existing works, we propose a Spatial Pyramid Dilation (SPD) ResNet block to include more receptive fields for utilizing information given by distant spatial locations. This is important to inpainting especially when the masked regions are too large to be filled. We also enhance the significance of self-similarity consideration, hence we employ Multi-Scale Self-Attention (MSSA) strategy to enhance our performance. Furthermore, Back Projection (BP) is strategically used to improve the alignment of the generated and valid pixels. We have achieved performance better than the state-of-the-art image inpainting. This research work participated in the AIM 2020 Extreme Image Inpainting Challenge, which requires the right balance of pixel-wise reconstruction accuracy and visual quality. We believe that our DeepGIN is able to achieve the right balance and we encourage scholars in the field to give more attention in this direction.

References

1. AIM2020: Aim 2020 extreme image inpainting challenge: Methods and results. In: ECCV Workshops (2020)
2. Anoosheh, A., Sattler, T., Timofte, R., Pollefeys, M., Gool, L.V.: Night-to-day image translation for retrieval-based localization. 2019 International Conference on Robotics and Automation (ICRA) (May 2019). https://doi.org/10.1109/icra.2019.8794387
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3), 24 (2009)
4. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
6. Gu, S., Danelljan, M., Timofte, R., Haris, M., Akita, K., Shakhnarovic, G., Ukita, N., Michelini, P.N., Chen, W., Liu, H., et al.: Aim 2019 challenge on image extreme super-resolution: Methods and results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3556–3564. IEEE (2019)
7. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1664–1673 (2018)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in neural information processing systems. pp. 6626–6637 (2017)
10. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Trans. Graph. **36**(4) (Jul 2017). https://doi.org/10.1145/3072959.3073659
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
12. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
13. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
14. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Li, C., Siu, W., Lun, D.P.K.: Vision-based place recognition using convnet features and temporal correlation between consecutive frames. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). pp. 3062–3067 (2019)
17. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 85–100 (2018)

18. Liu, Z.S., Wang, L.W., Li, C.T., Siu, W.C., Chan, Y.L.: Image super-resolution via attention based back projection networks. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3517–3525. IEEE (2019)
19. Lugmayr, A., Danelljan, M., Timofte, R.: Ntire 2020 challenge on real-world image super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 494–495 (2020)
20. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
21. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. pp. 8026–8037 (2019)
23. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
24. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: 2007 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)
25. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Wang, L.W., Liu, Z.S., Siu, W.C., Lun, D.P.: Lightening network for low-light image enhancement. IEEE Transactions on Image Processing (2020). https://doi.org/10.1109/TIP.2020.3008396
28. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
29. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
30. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
32. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6721–6729 (2017)
33. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)

34. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4471–4480 (2019)
35. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
36. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
37. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal on Computer Vision (2018)