GAIN: Gradient Augmented Inpainting Network for Irregular Holes

Jianfu Zhang
Shanghai Jiao Tong University
c.sis@sjtu.edu.cn

Li Niu*
Shanghai Jiao Tong University
ustcnewly@sjtu.edu.cn

Dexin Yang
Shanghai Jiao Tong University
hadean1998@sjtu.edu.cn

Liwei Kang
Shanghai Jiao Tong University
sjtu-klw@sjtu.edu.cn

Yaoyi Li
Shanghai Jiao Tong University
dsamuel@sjtu.edu.cn

Weijie Zhao
Versa-AI
weijie.zhao@versa-ai.com

Liqing Zhang*
Shanghai Jiao Tong University
zhang-lq@cs.sjtu.edu.cn

ABSTRACT

Image inpainting, which aims to fill the missing holes of the images, is a challenging task because the holes may contain complicated structures or different possible layouts. Deep learning methods have shown promising performance in image inpainting but still, suffer from generating poor-structured artifacts when the holes are large and irregular. Some existing methods use edge inpainting to help image inpainting, with binary edge map obtained from image gradient. However, by only using the binary edge map, these methods discard the rich information in image gradient and thus leave some critical issues (*e.g.*, color discrepancy) unattended. In this paper, we propose **G**radient **A**ugmented **I**npainting **N**etwork (GAIN), which uses image gradient information instead of edge information to facilitate image inpainting. Specifically, we formulate a multi-task learning framework which performs image inpainting and gradient inpainting simultaneously. A novel GAI-Block is designed to encourage the information fusion between the image feature map and the gradient feature map. Moreover, gradient information is also used to determine the filling priority, which can guide the network to construct more plausible semantic structures for the holes. Experimental results on public datasets CelebA-HQ and Places2 show that our proposed method outperforms state-of-the-art methods quantitatively and qualitatively.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**; *Scene understanding*; Image representations.

KEYWORDS

Image inpainting, plausible structural completion, context encoders, adversarial learning, deep learning

*Corresponding authors.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350912

ACM Reference Format:
Jianfu Zhang, Li Niu, Dexin Yang, Liwei Kang, Yaoyi Li, Weijie Zhao, and Liqing Zhang. 2019. GAIN: Gradient Augmented Inpainting Network for Irregular Holes. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350912

1 INTRODUCTION

Image inpainting [5] is a critical task in multimedia applications, especially in the computer vision area. As an imperative technique for image editing tasks, image inpainting aims to fill the missing hole regions (*i.e.*, masked area) of an image based on the background regions (*i.e.*, unmasked area), which can be used for tasks like image or video completion, recovery, distracting objects removing and replacing. However, it is challenging to make the reconstructed hole regions consistent with the background regions semantically for image inpainting models due to the complicated structures and different possible layouts inside the holes.

Traditional methods for image inpainting [3–5, 9, 10] mostly lie into two parts: diffusion-based and exemplar-based. Exemplar-based methods search exemplar patches and paste them to fill the mask. They can produce vivid images but suffer from high time cost for searching exemplar patches. On the other hand, diffusion-based methods reconstruct the current patch with the features around the patch. Compared with exemplar-based methods, diffusion-based methods can produce images with remarkable speed but always generate over-smoothed patches or artifacts. Due to the size limit of the patches, all these methods cannot gather high-level information from the image to provide images with compelling semantic structures consistent with the background regions.

With the help of deep learning methods, diffusion-based methods have been improved significantly. GAN [13] furthermore improve the vividness of the generated images. Many methods based on GAN show promising results for image inpainting tasks [17, 29, 43, 44] compared with methods based on maximum likelihood estimation [8, 46, 47]. Nevertheless, when the mask is too large, or the area around the mask is too complicated, these methods may reconstruct the missing regions with artifacts or irrational structures owing to the limited size of the receptive field. To increase the receptive field, these methods have to increase the number of downsampling layers or use dilation layers with larger rates, which are harmful to generating high-quality images.

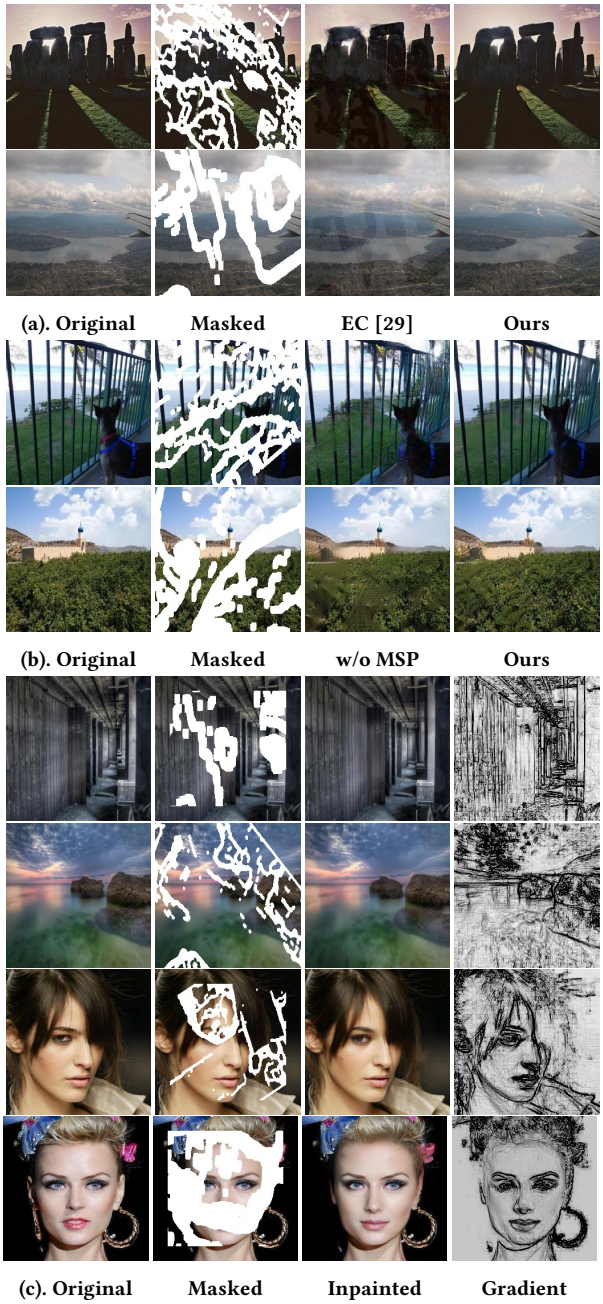

Figure 1: The motivation of our proposed method. (a). Some existing methods like [29] use binary edge map inpainting to guide image inpainting in generating images with better semantic structure, but they are still with color discrepancy; (b). When the mask is too large, the synthesized image may be blurry or with incomplete structure (w/o MSP). We propose our model leveraging image gradient information with well-tailored GAI-Block and mask shrinkage priority (MSP) to handle these two issues. (c). Sample images with high quality generated by our proposed model. For the columns from left to right are the original images, masked images, our image inpainting results, and our gradient inpainting results. For better visualization the values for both gradient and mask are reversed. Best viewed in color and with zoom in.

Some existing methods [29, 39] used edge inpainting to guide image inpainting and made some promising improvement. The edge map, which is obtained from image gradient and translated into a binary map, can define shapes and spaces in the image and help the model hallucinate the structure inside the holes. But the binary edge map discards the fruitful information in the image gradient, leaving some critical issues like color discrepancy unaddressed, leading to the inconsistency between the holes and the background regions.

We propose **G**radient **A**ugmented **I**npainting **N**etwork (GAIN) in this paper which leverages image gradient for image inpainting. Compared with the edge map, the image gradient is natural to acquire without any postprocessing steps. Previous works [23, 27, 31] show that image gradient plays an important role to resolve the color discrepancy issues in image editing and blending. Inspired by these works, we can inpaint images with both superior semantic structures and robust color consistency with the help of image gradient. Specifically, our image inpainting model is trained with an auxiliary gradient inpainting task. Our model is a two-branch style network with GAI-Block, which is well-tailored to encourage the information sharing between image feature map and gradient feature map in the network. Furthermore, we calculate priorities based on gradient feature map and fill the holes in order according to these priorities, to alleviate the harm of dilation layers. Our method can generate images which have plausible semantic structures without blur or artifacts.

The main contributions of our work are listed as follows:

- We propose a one-stage network for image inpainting with gradient inpainting as an auxiliary task. In this network, we design a novel convolution block named GAI block, which facilitates the information fusion between image feature map and gradient feature map.
- We design the mask shrinkage priority based on gradient information to achieve better semantic structure and avoid artifacts caused by dilation layers.
- We show that our method outperforms the state-of-the-art methods on face image dataset CelebA-HQ and scene recognition dataset Places2 qualitatively and quantitatively.

2 RELATED WORKS

Traditional image inpainting. Traditional image inpainting methods can be categorized into two parts: diffusion-based methods and exemplar-based methods. Diffusion-based methods propagate neighboring information from available background regions into the missing holes [3, 5, 12]. However, these methods can only access locally available information and also tend to generate oversmoothing patches. Exemplar-based methods fill in holes by copying information from similar exemplar patches in the background regions or a collection of other candidate images [4, 9, 10, 16]. Compared with diffusion-based methods, these methods can provide images with a plausible structure. Unfortunately, these methods are computationally expensive when computing similarity scores between the holes and candidate regions.

Deep image inpainting. Starting from [30], most deep image inpainting methods train their models by deep feature learning and adversarial learning with a Context Encoder, an auto-encoder

liked structure. Some other auxiliary losses like perceptual loss [18, 40] and Total Variation (TV) loss [18, 24] are introduced to further improve the quality of the synthesized images. Dilated convolution layers [42] was applied to increase the receptive field of the feature map [17]. Similar to exemplar-based methods, many deep learning methods [11, 32, 44] designed attention or swapping layers that can search the exemplar patches in the background feature map for the missing regions in an end-to-end manner. In [17] the adversarial training is extended with both global and local discriminators. Compared with traditional methods, deep learning methods can generate almost realistic images in promising speed.

Deep image inpainting with irregular holes. Most of deep image inpainting methods target at regular rectangle missing regions [17, 30, 36, 44, 45]. However, the holes are often irregular in real-world applications. In this case, the above methods [17, 30, 36, 44, 45] generally cannot perform well. Liu et al. [24] introduced Partial Convolution for image inpainting, where convolution weights are normalized by the mask area of the window to eliminate the effects caused by the irregular missing regions. In [43], Gated Convolution is introduced to split each of the convolution layers into the product of feature filters and gating filters. Gating filters learn attentions on the whole feature map to reduce the effects of irregular holes in an end-to-end manner. The method in [29] showed promising performance for irregular holes inpainting, but it is not specially designed for irregular holes.

Image inpainting with image gradient. Image gradient is calculated by the directional change in an image and widely used in computer vision applications like edge detection [7], image editing [27, 31], and image blending [23]. For image inpainting task, image gradient has also been studied to determine the direction or priority to fill the masked area [6, 9, 34] or calculate patch similarity [1]. All of these works are traditional methods. Our work is the first deep learning method to use image gradient for image inpainting.

Deep image inpainting with edge map. Instead of image gradient, some deep learning methods use edge information to support image inpainting. For example, the result of Holistically-nested Edge Detection (HED) edge detector [38] is used in [43] to guide the network to generate the masked area. In [29], a two-stage model is proposed to hallucinate the edge map obtained by Canny edge detector [7] in the first stage, and the generated edge map can guide filling the masked regions in the second stage. Xiong *et al.* [39] split the whole process into three stages: contour inpainting, edge inpainting, and image inpainting, to achieve a better structure of the generated images. These methods struggle with different choices of edge detectors with different hyper-parameters. Also, compared with image gradient, the binary edge map is ill-suited for adversarial learning and leave some critical issues in image inpainting like color discrepancy unsolved. In contrast, our method utilizes natural gradient information to be fused with image feature information and determine the filling order, contributing to plausible semantic structures for irregular holes.

3 METHODS

We leverage the encoder-decoder framework following an adversarial model [13] to generate the masked area (*i.e.*, holes) for inpainting task, which is similar to most of the methods using deep learning for image inpainting [30, 41, 44]. Let G be the encoder-decoder network, and D be the discriminator network. The overall architecture of G can be found in Figure 2.

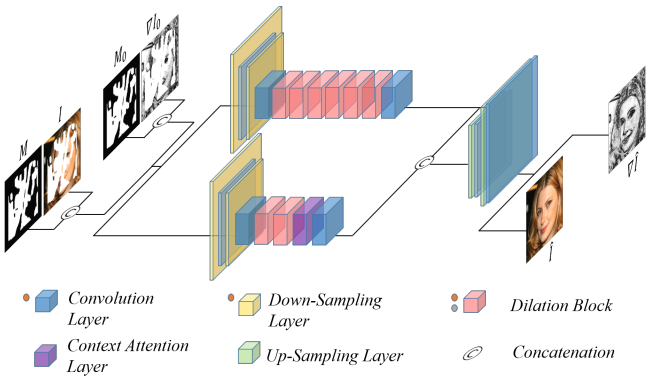

Figure 2: The overview of the generator for our proposed model. The inputs are the mask, masked image, and masked gradient map. The outputs are the filled image and gradient map. Convolution layer, down-sampling layer, and dilation block are replaced with GAI-Block (marked with orange dots). Mask shrinkage priority is applied in dilation block (marked with grey dots).

To inpaint the missing regions with visual-coherent structure and context, following the network structure [44], our generator G consists of two branches, in which the first branch focuses on reconstructing the image structure while the second branch focuses on reconstructing the image content. The input and output image sizes are both 256×256. The first branch focuses on image structure, which downsamples the input image twice using two convolution layers with stride 2 followed by six dilation blocks. Each dilation block contains four convolution layers with dilation factors of $(2, 4, 8, 16)$, which means that for a feature map with a size of 64×64, two dilation blocks will make sure that each position of the feature map has a receptive field as large as the whole feature map. The second branch focuses on image content, which downsamples the input image twice using two convolution layers with stride 2 followed by two dilation blocks and a context attention block [44]. The context attention block matches the generated feature in the holes with the features of the known regions, and then selectively assigns the features from the known regions to the holes according to the matching scores. The context attention block will improve the quality of the synthetic images, and its details can be found in [44]. The output feature maps of two branches are concatenated and then upsampled twice using bilinear upsampling layers.

We leverage the image gradient to help inpaint the image. The detailed definition of image gradient will be introduced in Section 3.1. Not only optimizing the network for image inpainting task, but we also choose to optimize the gradient inpainting task as an auxiliary task to assist in image inpainting (in Section 3.2). We input the network with the mask, masked image, and masked gradient map and output the reconstructed images and gradient map. We design GAI-Block (in Section 3.3) to supersede the original convolution layers, including the down-sampling layers and those in dilation blocks, which leverages gradient feature map to reconstruct the image. This block will facilitate the network to gather gradient features and fuse them with image features to generate images with

a compelling structure. Furthermore, we define and apply the priority (in Section 3.4) for mask shrinkage in the network propagation process, which will alleviate the misleading of the generated image structure of network propagation. Note that GAI-block is applied to convolution layers, downsampling layers, and dilation blocks, while mask shrinkage priority is applied to the first four dilation blocks in the first branch to help the network generate plausible images with better structure.

3.1 Definitions

Let $\mathbf{I}$ be the groundtruth image and $\mathbf{M}$ be the mask where the available pixels are marked as 1 (*i.e.*, unmasked area) while the missing pixels in the holes are 0 (*i.e.*, masked area). We define $\nabla\mathbf{I}$ as the image gradient map of $\mathbf{I}$. Mathematically, the image gradient is calculated by the derivatives in the horizontal and vertical directions of the image. We define:

$$\nabla\mathbf{I} = (\nabla\mathbf{I}_x, \nabla\mathbf{I}_y) = (\frac{\partial\mathbf{I}}{\partial x}, \frac{\partial\mathbf{I}}{\partial y}). \tag{1}$$

Where $\frac{\partial\mathbf{I}}{\partial x}$ and $\frac{\partial\mathbf{I}}{\partial y}$ are the derivative with respect to horizontal and vertical direction. We use finite differences to approximate these derivative, which can be represented by 3×3 convolution form:

$$\nabla\mathbf{I} = (\nabla\mathbf{I}_x, \nabla\mathbf{I}_y) = \left(\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} * \mathbf{I}, \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \mathbf{I}\right), \tag{2}$$

where $*$ denotes convolution operation. We further define $\rho = arctan(\frac{\nabla\mathbf{I}_y}{\nabla\mathbf{I}_x})$ as the gradient direction and $\sqrt{\nabla\mathbf{I}_x^2 + \nabla\mathbf{I}_y^2}$ as the gradient magnitude. Specifically, before we calculate the gradient of an image, we convert the image to grayscale. For simplicity, in this paper, we will use gradient magnitude to visualize the image gradient and reverse the color for better visualization.

Image gradients are robust under different lighting or camera properties [35], making image gradient a very robust feature. The most common use of image gradient is edge detection [7, 19] because pixels with large gradient values are highly possible to be edge pixels. Edge detectors will search the pixels with local maximum gradients and trace the direction of the gradients.

In our proposed method, we inpaint the image and gradient map at the same time. We build up a convolutional neural network G with N blocks (*i.e.*, , layers, or combinations of layers) following the encoder-decoder structure. We define $\mathbf{I}_1 = \mathbf{I} \odot \mathbf{M}$, $\mathbf{M}_1 = \mathbf{M}$. Here $\odot$ means element-wise multiplication. For the initial input of gradient for the network $\nabla\mathbf{I}_1$, we dilate the size of $\mathbf{M}$ to $\mathbf{M}_0$ by 1 and take $\nabla\mathbf{I}_0 = \nabla\mathbf{I} \odot \mathbf{M}_0$ to avoid unreasonable use of the information inside the masked area. Then $\nabla\mathbf{I}_1$ is obtained by applying partial convolution $Pconv(\nabla\mathbf{I}_0, \mathbf{M}_0)$ to propagate the gradient information and shrink the mask to $\mathbf{M}_1$. The detail of partial convolution and mask shrinkage can be found in Section 3.3. For each type of the block, we introduce the function G that defines how the block changes the input to the output. For the k-th block in the network, we have:

$$(\hat{\mathbf{I}}_{k+1}, \hat{\mathbf{M}}_{k+1}, \nabla\hat{\mathbf{I}}_{k+1}) = G_k(\mathbf{I}_k, \mathbf{M}_k, \nabla\mathbf{I}_k). \tag{3}$$

For the whole generator, we have:

$$(\hat{\mathbf{I}}, \hat{\mathbf{M}}, \nabla\hat{\mathbf{I}}) = G_N(G_{N-1}(\cdots G_1(\mathbf{I}_1, \mathbf{M}_1, \nabla\mathbf{I}_1))). \tag{4}$$

Here $\hat{\mathbf{I}}$ and $\nabla\hat{\mathbf{I}}$ are the reconstructed image and gradient map which should be as close as possible to $\mathbf{I}$ and $\nabla\mathbf{I}$. $\mathbf{M}$ should equal to $\mathbf{1}$ (we define $\mathbf{1}$ and $\mathbf{0}$ as the matrices that only contain 1 or 0) which means all the pixels of the image are fully reconstructed.

3.2 Objective Functions

We have two different objective functions, $l1$-loss and GAN loss, for both image and gradient inpainting to reach our goal. $l1$-loss calculates the distance between the predict image (*resp.*, gradient map) and the groundtruth image (*resp.*, gradient map). The $l1$-loss for image can be defined as:

$$L_{l1}^{I} = \left\|\mathbf{I} - \hat{\mathbf{I}}\right\|_1. \tag{5}$$

Similarly we can get the $l1$-loss $L_{l1}^{\nabla\mathbf{I}}$ for the gradient. $l1$-loss can help the generator to synthesize images which are close but coarse compared with the groundtruth image. Many previous works [17, 30, 43, 44] have combined GAN loss [13] and $l1$-loss to generate plausible images and here we are also using these two losses. We train two discriminator networks D^I and D^S for image and gradient map. We choose WGAN [2] with hinge loss and the GAN loss for an image can be defined as:

$$\begin{aligned} L_D^I =& \mathbb{E}_{\mathbf{I}\sim p_{Idata}}\left(max\left(0, 1 - D^I(\mathbf{I})\right)\right) + \mathbb{E}_{\hat{\mathbf{I}}\sim p_{\hat{\mathbf{I}}}}\left(max\left(0, 1 + D^I(\hat{\mathbf{I}})\right)\right), \\ L_G^I =& -\mathbb{E}_{\hat{\mathbf{I}}\sim p_{\hat{\mathbf{I}}}}\left(D^I(\hat{\mathbf{I}})\right). \end{aligned} \tag{6}$$

Where $D(x)$ denotes the output from D, and p_x denotes the distribution of x. Similarly we can get the $l1$-loss $L_D^{\nabla\mathbf{I}}$ and $L_G^{\nabla\mathbf{I}}$ for the gradient.

GAN [13] learns to generate a data distribution of interest (*i.e.*, $p_{\hat{I}}$), while the discriminative network distinguishes candidates produced by the generator from the true data distribution (*i.e.*, $p_{\mathbf{I}data}$). The generative network's training objective is to increase the error rate of the discriminative network (*i.e.*, "fool" the discriminator network by producing novel candidates that the discriminator thinks are not synthesized (*i.e.*, are part of the true data distribution). Some existing methods alleviate edge map to guide image inpainting also use GAN for edge inpainting. They take the binary edge map as the true data distribution and soft generated edge maps as the synthesized distribution. It is ill-suited because the discriminator can distinguish the candidates by the values. Compared with the binary edge map, the image gradient map is more suitable for GAN.

The overall loss function for the generator G of our proposed method is:

$$L = L_G^I + \lambda_{l1}L_{l1}^I + \lambda_{\nabla\mathbf{I}}(L_G^{\nabla\mathbf{I}} + \lambda_{l1}L_{l1}^{\nabla\mathbf{I}}). \tag{7}$$

Here $\lambda_{\nabla\mathbf{I}}$ and λ_{l1} are the parameters to balance between $l1$-loss and GAN loss. We choose $\lambda_{\nabla\mathbf{I}} = 0.1$ and $\lambda_{l1} = 100$ in the training process of our proposed method.

3.3 GAI-Block

We replace the vanilla convolution layer with a more complicated but well-designed block, called GAI-Block, which gathers gradient information from the current feature map and propagate this information to the masked area. The block contains two flows: image feature flow and gradient feature flow. In each block, partial

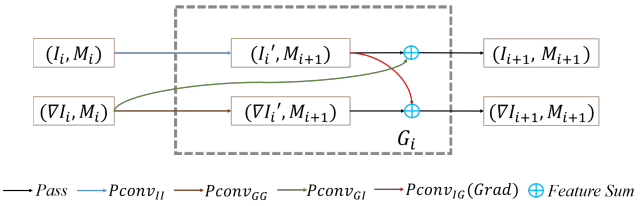

Figure 3: An illustration of the GAI-Block G_i. Here feature sum means element-wise sum for the image feature map and gradient feature map.

convolution and gradient filter are applied to propagate the information within each flow and share information across two flows. The detailed structure can be found in Figure 3. Before we introduce this block, let us review the two key modules in this block: partial convolution [24] and gradient filter.

Partial Convolution. We apply partial convolution in GAI-Block to propagate information from background regions to the holes and pass the information from gradient feature flow to image feature flow. Let $\mathbf{W}$ be the convolution filter weights, and b be the corresponding bias. $\mathbf{X}$ are the feature values for the current convolution window, and $\mathbf{M}_x$ is the mask with binary values for the same window. We first state how the mask changes during the partial convolution. The mask m will be updated as:

$$m' = \begin{cases} 1 & \text{if } sum(\mathbf{M}_x) > 0, \\ 0 & otherwise. \end{cases} \tag{8}$$

Here m' is the value for the mask corresponding to $\mathbf{M}_x$, $sum(\cdot)$ calculates the sum of the elements. All the pixels with at least one available pixel in the window before the convolution will be unmasked pixel after applying partial convolution. As we can see that after each partial convolution, some of the values for the mask $\mathbf{M}$ will change from 0 to 1, which means that the masked area is shrunken. We call this process as *mask shrinkage*.

Then partial convolution in [24] is defined as:

$$x' = \begin{cases} \frac{a}{sum(\mathbf{M}_x)} \mathbf{W} * (\mathbf{X} \odot \mathbf{M}_x) + b & \text{if} m' = 1, \\ 0 & otherwise, \end{cases} \tag{9}$$

in which a is the area of convolution window, x' is the convolution output of $\mathbf{X}$.

Based on eq. (8) and (9), we define $(\mathbf{X}', \mathbf{M}') = Pconv_i(\mathbf{X}, \mathbf{M}; \boldsymbol{\theta}_i)$ as a partial convolution layer, which outputs a new feature map $\mathbf{X}'$ and an updated mask $\mathbf{M}'$ with each entry being x' in eq. (9) and m' in eq. (8) respectively. For ease of description, we additionally define the mask shrinkage function $\omega_i(\mathbf{M}; \boldsymbol{\theta}_i)$ for $Pconv_i$.

Gradient filter. Gradient filter is applied in GAI-Block to calculate the gradient of the image feature map and fuse it with the gradient feature map. Similar to the definition of the image gradient in eq. (2), we define the gradient feature map of image feature map X as $\nabla\mathbf{X}$. Different from image gradient map $\nabla\mathbf{I}$ which is calculated based on the grayscale image converted from $\mathbf{I}$, gradient feature map will calculate gradients on each channel of the feature map X and output a feature gradient map $\nabla\mathbf{X}$ with doubled number of channels.

When calculating the gradient of the unmasked feature map, the gradient on the mask boundary should be excluded because the masked feature map is unknown. So the available region of gradient feature map should be the mask $\mathbf{M}$ dilated by 1. The inverse process of mask dilation by 1, *i.e.*, mask erosion by 1, could be implemented by convolution with kernel size 3, stride 1, and dilation rate 1. We use $\omega_g(\cdot)$ to denote mask erosion and $\omega_g^{-1}(\cdot)$ to denote mask dilation. Then, the available region of the gradient feature map will be $\omega_g^{-1}(\mathbf{M})$ after applying the gradient filter. We use $Grad(\mathbf{X}, \mathbf{M}) = (\nabla\mathbf{X}, \omega_g^{-1}(\mathbf{M}))$ to indicate the process of calculating the gradient feature map $\nabla\mathbf{X}$ with dilated mask $\omega_g^{-1}(\mathbf{M})$.

GAI-Block. The overall structure of GAI-Block is shown in Figure 3. We calculate the image feature map and gradient feature map, which share information with each other.

For the block G_i with input of $(\mathbf{I}_i, \nabla\mathbf{I}_i, \mathbf{M}_i)$, we split them into image feature map with mask $(\mathbf{I}_i, \mathbf{M}_i)$ and gradient feature map with mask $(\nabla\mathbf{I}_i, \mathbf{M}_i)$. We first apply two partial convolution layers to $(\mathbf{I}_i, \mathbf{M}_i)$ and $(\nabla\mathbf{I}_i, \mathbf{M}_i)$, yielding an intermediate image feature map with mask $(\mathbf{I}_i', \omega_i(\mathbf{M}_i))$ and an intermediate gradient feature map with updated mask $(\nabla\mathbf{I}_i', \omega_i(\mathbf{M}_i))$

$$\begin{aligned} (\mathbf{I}_i', \mathbf{M}_{i+1}) &= Pconv_{II}(\mathbf{I}_i, \mathbf{M}_i), \\ (\nabla\mathbf{I}_i', \mathbf{M}_{i+1}) &= Pconv_{GG}(\nabla\mathbf{I}_i, \mathbf{M}_i), \end{aligned} \tag{10}$$

in which we have $\mathbf{M}_{i+1} = \omega_i(\mathbf{M}_i)$. After obtaining another intermediate image feature map with mask $Pconv_{GI}((\nabla\mathbf{I}_i, \mathbf{M}_i))$ and gradient feature map with mask $Pconv_{IG}(Grad(\mathbf{I}_i', \mathbf{M}_{i+1}))$, we sum two intermediate image feature maps and two intermediate gradient feature maps separately while the mask remains unchanged:

$$\begin{aligned} (\mathbf{I}_{i+1}, \mathbf{M}_{i+1}) &= (\mathbf{I}_i', \mathbf{M}_{i+1}) + Pconv_{GI}(\nabla\mathbf{I}_i, \mathbf{M}_i), \\ (\nabla\mathbf{I}_{i+1}, \mathbf{M}_{i+1}) &= (\nabla\mathbf{I}_i', \mathbf{M}_{i+1}) + Pconv_{IG}(Grad(\mathbf{I}_i', \mathbf{M}_{i+1})). \end{aligned} \tag{11}$$

The design of GAI-block could ensure that the masks of two inputs for summation are identical. Note that $Pconv_{GG}$, $Pconv_{II}$ and $Pconv_{GI}$ in each block G_i have the same hyper-parameters (*i.e.*, kernel size, stride, dilation) determined by G_i yet different convolution filter weights. $Pconv_{IG}$ has fixed kernel size 3, stride 1, and dilation rate 1. All the convolution kernel weights are different across different blocks.

In previous image blending and editing works [23, 27, 31], the benefit of utilizing gradient information has also been demonstrated. Compared with partial convolution [24], GAI-Block will help the network get better color consistency between the holes and the background regions, especially when the network is combined with gradient inpainting. We conjecture that with the help of the GAI-Block, gradient information is stitched into image feature information, which can help the model learn stable features from unmasked regions and propagate the gradient information to the entire image.

3.4 Mask Shrinkage Priority

Partial convolution provides a great idea of isolating the features of the unmasked regions from the masked regions. On the one hand, the kernel of the convolution layers cannot be too large in case the mask shrinks too fast. Partial convolution will be less effective due to the lack of available contextual information for the patches on the mask border when the kernel size is too large. On the other hand, the kernel size of the convolution layers cannot be too small,

Figure 4: An example of applying the mask shrinkage priority. The leftmost image is the original image, then from left to right is how the mask (white area) shrinks and turns into the rightmost image which is fully synthesized by our proposed model.

because small kernel size leads to the small receptive field of each pixel, which will hinder the reconstruction of a better structure.

In [25], several downsampling layers are applied to enlarge the receptive field. However, it might cause the synthesized images to have checkerboard artifacts. Alternatively, many methods use dilation convolution [29, 39, 43, 44] to get a larger receptive field, which is similar to using a larger kernel but with a less computational cost. Therefore, our method also employs dilation blocks. However, dilation convolutions will also be harmful for partial convolution because the mask shrinks too fast, which will generate artifacts for the missing regions.

To keep the large receptive field and avoid fast mask shrinkage, we get inspired by previous exemplar-based methods [9, 22], which assign priorities for the pixels and fill them in order according to the assigned priorities. We also assign priorities and only allow the mask shrinkage for the pixels with higher priorities. Our designed priority also leverages the gradient information. Specifically, the priority for pixel x is defined as:

$$p_x = sum(\mathbf{M}_x) \times (n_x \cdot \nabla \mathbf{I}_x^{\perp}), \tag{12}$$

where p_x is the priority at pixel x, $sum(\cdot)$ and $\mathbf{M}_x$ are defined as the same in eq. 9, n_x is the normal vector of the mask border and $\nabla \mathbf{I}_x^{\perp}$ is the normal vector of gradient. Specifically, $\nabla \mathbf{I}_x^{\perp}$ is calculated by the normal vector for the channel-wise mean of the gradient feature map $\nabla \mathbf{I}$ in GAI-Block. n_x is the normal vector for the mask border, which is calculated by the gradient of the mask. Note that the mask only contains binary values; if we directly calculate the normal vector, there will be only 9 possible normal vectors. Hence, we first calculate the 3×3 windowed average of the mask centered at x then calculate the n_x based on this smoothed mask.

This priority is the product of two parts: the first part is $sum(\mathbf{M}_x)$, which calculates the number of pixels that are available for the current pixel. It is natural to fill the pixels with more contextual knowledge first so that the pixels with higher $sum(\mathbf{M}_x)$ should have higher priorities; the other part is $n_x \cdot \nabla \mathbf{I}_x^{\perp}$, which calculate the overlapping ratio of the normal vector of gradient and the mask border. The pixels with a higher value of $n_x \cdot \nabla \mathbf{I}_x^{\perp}$ are more likely to be from informative regions such as the edges on the mask border.

After using p_i^{max} to denote the highest priority for the feature map generated by the block G_i, we will only allow the pixels with priorities higher than $\delta \cdot p_i^{max}$, in which δ is a hyper-parameter to control the speed of mask shrinkage and we set $\delta = 0.4$ for our model. Similar to eq. (8), we have:

$$m' = \begin{cases} 1 & \text{if } sum(\mathbf{M}_x) > 0 \text{ and } p_x \geq \delta \cdot p_i^{max}, \\ 0 & otherwise. \end{cases} \tag{13}$$

Here we only update the mask shrinkage process, the feature propagation process in eq. (9) will remain the same. In fact, (9) is a special case of (13) when $\delta = 0$.

Note that $p_x = 0$ for these pixels are not on the mask border, which means x is in $\mathbf{M}_x$ or $sum(\mathbf{M}_x) = 0$. In Figure 1, we show an example comparison between partial convolution (**w/o MSP**) and our proposed mask shrinkage method. As we can see, the image generated by partial convolution may contain artifacts inside the holes. While with the mask shrinkage, we can fill the holes more consistently with the background.

In Figure 4, we show an example process of mask shrinkage with the priority we defined. The masked area with higher structural knowledge (*e.g.* borders or lines for the small house, the texture of the leaves) and contextual knowledge (mask border) are filled first. With this priority, our proposed model fills the holes with both perfect structure and plausible context.

3.5 Implement Details

Our proposed model is implemented in TensorFlow. The network is trained using 256×256 images with a batch size of 8. The model is optimized by Adam optimizer [21]. Learning rate is set to be 10^{-4}. The model is trained with 10^6 iterations. For discriminators, we use gradient penalty [14] and spectral normalization layers [28] to make the training of GAN fast and stable. The number of channels for the output of the first convolution block is 64 for I and 128 for $\nabla \mathbf{I}$. The number of channels will be doubled after each downsampling layer and reduced by half after each upsampling layer.

4 EXPERIMENTS

4.1 Datasets, Masks and Evaluation Metrics

Our proposed model is evaluated on the datasets CelebA-HQ [20, 26], Places2 [48]. Following the evaluation in [29], we evaluate all the methods on 5 sets that each of them contains 10000 images randomly sampled from the test set of each dataset, and the results are averaged. For training, we follow the standard training split of the datasets. Results are compared against the current state-of-the-art methods, both qualitatively and quantitatively. In our experiments, we focus on irregular image masks, which are provided by Liu *et al.* [24]. Irregular masks are augmented by introducing four rotations and a horizontal reflection for each mask. They are classified based on their sizes relative to the entire image in increments of 10% (*e.g.* 0-10%, 10-20%, etc.) with the maximum ratio being 60%.

For quantitative comparison, we adopt the following four evaluation metrics: relative $l1$, Structural Similarity (SSIM) [37], Peak Signal-to-Noise Ratio (PSNR), and Frechet Inception Distance (FID) [15]. The first three metrics assume pixel-wise independence, which may assign favorable scores to perceptually inaccurate results.

Table 1: The results of our proposed model and other state-of-the-art methods on dataset Places2. The best result of each row is in boldface. † Results are taken from [24].

Metric	Mask	GL [17]	CA [44]	EC [29]	PC † [24]	Ours
$l1$(%)	0-10%	0.87	0.96	0.56	**0.49**	0.59
	10-20%	2.24	2.15	1.42	**1.18**	**1.18**
	20-30%	4.17	3.87	2.56	2.07	**2.04**
	30-40%	6.29	5.76	3.84	3.19	**3.04**
	40-50%	8.48	7.78	5.33	4.37	**4.26**
	50-60%	10.46	10.12	7.68	6.45	**6.42**
SSIM	0-10%	0.962	0.968	0.982	0.946	**0.989**
	10-20%	0.897	0.912	0.950	0.867	**0.967**
	20-30%	0.804	0.830	0.900	0.775	**0.931**
	30-40%	0.703	0.740	0.839	0.681	**0.883**
	40-50%	0.607	0.647	0.765	0.583	**0.819**
	50-60%	0.511	0.536	0.638	0.468	**0.694**
PSNR	0-10%	29.67	31.23	33.58	33.75	**35.97**
	10-20%	24.47	25.37	28.06	27.71	**30.26**
	20-30%	21.12	21.81	24.85	24.54	**26.79**
	30-40%	18.93	19.55	22.67	22.01	**24.38**
	40-50%	17.41	17.95	20.88	20.34	**22.37**
	50-60%	16.52	16.47	18.79	18.21	**19.89**
FID	0-10%	3.09	1.48	0.75	-	**0.65**
	10-20%	9.39	4.83	2.25	-	**2.04**
	20-30%	20.72	11.94	5.08	-	**4.72**
	30-40%	35.65	22.11	9.50	-	**8.70**
	40-50%	50.92	33.99	16.34	-	**14.28**
	50-60%	53.29	41.78	29.22	-	**24.94**

While FID measures the Wasserstein-2 distance between the feature representations of real and inpainted images using a pre-trained Inception-V3 model [33], which is a popular perceptual metric to evaluate the quality of synthesized images.

4.2 Quantitative Results

In Table 1 and 2, we report the results of our method and the state-of-the-art baselines on scene recognition dataset Places2 [48] and facial expression dataset CelebA-HQ [26]. We select four state-of-the-art deep image inpainting methods: Global and Local (GL) [17], Contextual Attention (CA) [44], EdgeConnect (EC) [29], and Partial Convolution (PC) [24]. For CA and EC on both datasets and GL on the Places2 dataset, we obtain their results using their officially released source code or trained model for a fair comparison. Note that GL does not release the trained model for the CelebA-HQ dataset or training code, so we use the results of GL on the CelebA-HQ dataset reported in [29]. For PC, neither the trained models on two datasets nor the source code is available, so we simply copy their reported results on the Places2 dataset in [24].

As we can see, our proposed method outperforms all state-of-the-art methods except for relative $l1$ when the mask is tiny. Especially when the holes in the images are large, our method shows superior performance.

4.3 Qualitative Results

In Figure 6 and 7, we show the qualitative comparisons between our methods and other state-of-the-art methods on Places2 and CelebA-HQ dataset (see Supplementary for more results).

Table 2: The results of our proposed model and other state-of-the-art methods on dataset CelebA-HQ. The best result of each row is in boldface. † Results are taken from [29].

Metric	Mask	GL† [17]	CA [44]	EC [29]	Ours
$l1$(%)	0-10%	0.91	0.70	**0.33**	0.48
	10-20%	2.53	1.45	**0.86**	**0.86**
	20-30%	4.67	2.54	1.64	**1.49**
	30-40%	6.95	3.84	2.55	**2.18**
	40-50%	9.18	5.43	3.67	**3.03**
	50-60%	11.21	7.88	5.79	**4.58**
SSIM	0-10%	0.947	0.983	0.992	**0.994**
	10-20%	0.865	0.954	0.977	**0.982**
	20-30%	0.773	0.910	0.951	**0.963**
	30-40%	0.689	0.855	0.916	**0.939**
	40-50%	0.609	0.788	0.868	**0.906**
	50-60%	0.560	0.683	0.770	**0.836**
PSNR	0-10%	30.24	33.81	37.47	**38.19**
	10-20%	24.09	28.52	31.88	**32.92**
	20-30%	20.71	25.08	28.31	**29.53**
	30-40%	18.50	22.64	25.81	**27.17**
	40-50%	17.09	20.62	23.67	**25.13**
	50-60%	16.24	18.43	20.82	**22.49**
FID	0-10%	16.84	1.00	0.24	**0.22**
	10-20%	58.74	3.64	0.81	**0.64**
	20-30%	102.97	9.27	2.02	**1.43**
	30-40%	136.47	17.76	3.69	**2.54**
	40-50%	163.95	30.58	6.48	**4.24**
	50-60%	167.07	43.90	12.74	**7.44**

As we can see, for Places2 dataset, CA and GL fail to synthesize the images with reasonable structure and consistent color. EC performs better, but may also have some small failures like that in the second row. EC also provide results with some observable color discrepancy. Compared with EC, which are using edge information to guide inpainting, our method can provide images without watermark like color discrepancy in the masked area. Thus, we conjecture that gradient information might be more helpful for image inpainting compared with edge information.

For CelebA-HQ dataset, all of the other methods fill the holes with artifacts and fail to generate realistic faces. While our proposed method provides plausible faces even when the masks are large. In Figure 7 we show our results on face dataset CelebA-HQ. We can see that our method can provide plausible images with a stable structure of the faces.

4.4 Ablation Studies

In this section, we conduct ablation studies for our proposed method and compare with the following special cases: 1) **full**: our full-fledged model; 2) **w/o grad**: remove the gradient inpainting task by setting $\lambda_{\nabla I} = 0$ in eq. (7); 3) **edge**: replace the gradient inpainting task with the edge inpainting task, in which the edge map is generated by Canny edge detector; 4) **w/o MSP**: without using mask shrinkage priority by setting $\delta = 0$ in eq. (13); 5) **w/o GAI-Block**: replace GAI-Block with vanilla partial convolution by removing the gradient feature map $\nabla \mathbf{I}_k$ in eq. (3) and calculating the mask shrinkage priority based on the gradient of image feature map; and 6) **w/o CA**: remove the branch containing the context attention

Figure 5: Effects of mask shrinkage with different choices of threshold δ.

Figure 6: The visual comparison results on Places2.

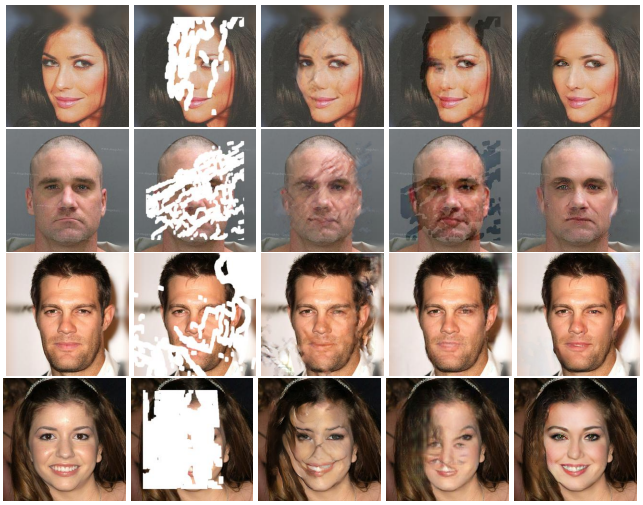

Figure 7: The visual comparison results on CelebA-HQ.

layer. We evaluate our special cases on the Places2 dataset with six groups of mask ratios and report the averaged results over six groups based on four evaluation metrics. The other experimental settings including hyper-parameters are the same as in Section 4.2.

The results of our special cases are reported in Table 3, from which we can see that our **full** method outperforms **w/o grad**, **edge**, and **w/o GAI-Block**, especially based on FID. These results demonstrate the benefit of gradient inpainting task as well as the information fusion between gradient feature map and image feature map. Our **full** method is also better than **w/o CA**, which shows

Table 3: The results of ablative studies for Places2 dataset. The best result of each column is in boldface.

Metric	$l1$	SSIM	PSNR	FID
full	2.92	**0.881**	**26.61**	**9.22**
w/o grad	2.93	0.875	26.38	10.36
edge	2.95	0.880	26.55	9.89
w/o MSP	**2.89**	0.870	26.55	11.31
w/o GAI-Block	3.18	0.852	25.80	13.89
w/o CA	3.02	0.880	25.90	10.37

that it is helpful to use two branches to focus on reconstructing image content and image structure separately.

By comparing **w/o MSP** with **full**, The $l1$, SSIM, and PSNR results of **w/o MSP** are close to **full** while the FID results drop sharply, which proves that mask shrinkage priority can significantly improve the overall quality of the generated images.

In Figure 5, we show the impact of different thresholds δ in eq. (8). It can be seen that the generated image is blurry when δ is small while the holes are filled with artifacts when δ is large. In our experiments, we use $\delta = 0.4$ as the default value, which can generally achieve satisfactory performance.

5 CONCLUSION

In this paper, we have proposed a novel image inpainting framework GAIN, which uses image gradient information instead of edge information to facilitate image inpainting. In this framework, we have formulated a multi-task learning framework which performs image inpainting and gradient inpainting simultaneously. We design novel GAI-Block to encourage the information sharing between the gradient map and feature map. Gradient map is also used to determine the filling priorities of pixels to guide the mask shrinkage process. Experimental results on public datasets CelebA-HQ and Places2 show that our proposed model can fill the holes with plausible semantic structure and consistent color to the background regions.

ACKNOWLEDGMENTS

The work was supported in part by the National Basic Research Program of China (Grant No. 2015CB856004), the Key Basic Research Program of Shanghai Municipality, China (15JC1400103), and Startup Fund for Youngman Research at SJTU (WF220403041).

REFERENCES

[1] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. 2011. A variational framework for exemplar-based image inpainting. *International journal of computer vision* 93, 3 (2011), 319–347.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

[3] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. 2000. Filling-in by joint interpolation of vector fields and gray levels. (2000).

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, Vol. 28. ACM, 24.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 417–424.

[6] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. 2003. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing* 12, 8 (2003), 882–889.

[7] John Canny. 1987. A computational approach to edge detection. In *Readings in computer vision*. Elsevier, 184–203.

[8] Tony F Chan and Jianhong Shen. 2005. Variational image inpainting. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 58, 5 (2005), 579–619.

[9] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing* 13, 9 (2004), 1200–1212.

[10] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. 2012. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.* 31, 4 (2012), 82–1.

[11] Brian Dolhansky and Cristian Canton Ferrer. 2018. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7902–7911.

[12] Selim Esedoglu and Jianhong Shen. 2002. Digital inpainting based on the Mumford–Shah–Euler image model. *European Journal of Applied Mathematics* 13, 4 (2002), 353–370.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*. 5767–5777.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. 6626–6637.

[16] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)* 33, 4 (2014), 129.

[17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 107.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*. Springer, 694–711.

[19] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. 1988. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits* 23, 2 (1988), 358–367.

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Olivier Le Meur, Josselin Gautier, and Christine Guillemot. 2011. Examplar-based inpainting based on local geometry. In *2011 18th IEEE international conference on image processing*. IEEE, 3401–3404.

[23] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. 2004. Seamless image stitching in the gradient domain. In *European Conference on Computer Vision*. Springer, 377–389.

[24] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 85–100.

[25] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 85–100.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*. 3730–3738.

[27] James McCann and Nancy S Pollard. 2008. Real-time gradient-domain painting. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 93.

[28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

[29] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. 2019. EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning. *arXiv preprint arXiv:1901.00212* (2019).

[30] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.

[31] Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Transactions on graphics (TOG)* 22, 3 (2003), 313–318.

[32] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. 2018. Contextual-based image inpainting: Infer, match, and translate. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 3–19.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[34] Alexandru Telea. 2004. An image inpainting technique based on the fast marching method. *Journal of graphics tools* 9, 1 (2004), 23–34.

[35] Jack Tumblin, Amit Agrawal, and Ramesh Raskar. 2005. Why I want a gradient camera. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, 103–110.

[36] Huy V Vo, Ngoc QK Duong, and Patrick Pérez. 2018. Structural inpainting. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1948–1956.

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[38] Saining Xie and Zhuowen Tu. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*. 1395–1403.

[39] Wei Xiong, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. 2019. Foreground-aware Image Inpainting. *arXiv preprint arXiv:1901.05945* (2019).

[40] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. 2017. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6721–6729.

[41] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. 2017. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5485–5493.

[42] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122* (2015).

[43] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589* (2018).

[44] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5505–5514.

[45] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. 2018. Semantic Image Inpainting with Progressive Generative Networks. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1939–1947.

[46] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. 2015. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence* 37, 9 (2015), 1751–1763.

[47] Qibin Zhao, Guoxu Zhou, Liqing Zhang, Andrzej Cichocki, and Shun-Ichi Amari. 2015. Bayesian robust tensor factorization for incomplete multiway data. *IEEE transactions on neural networks and learning systems* 27, 4 (2015), 736–748.

[48] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence* 40, 6 (2018), 1452–1464.