ISSN: 2688-8645

# Image Inpainting based on Dilated Neighborhood Attention

Kexin Zhang and Hua Huo

Department of Information Engineering, Henan University of Science and Technology, Luoyang, China

## Abstract

**In response to the phenomenon that existing image restoration algorithms have blurry edges, incoherent textures, and lack clarity and delicacy for large-area missing images, a two-stage generative adversarial image restoration algorithm based on dilated neighborhood attention is proposed. This algorithm decouples image restoration into edge structure restoration and texture structure restoration, introduces a dilated neighborhood attention mechanism, and enhances the generator's focus on important information and structures in the image by constructing a residual attention network, thereby improving the perception and utilization of texture details, resulting in more realistic image views and finer texture details. This paper introduces the Binary Cross-Entropy with Logits loss function in the discriminators of the two stages, which can help the discriminator learn how to more effectively identify real and generated images, thus improving the overall network performance. The Ranger21 optimizer is introduced to accelerate learning without affecting generalization, addressing the problem of traditional optimizers systematically staying in poor initial states. The datasets used in this paper are Paris Street View and CelebA-HQ. Comparative experiments with other image restoration methods show that both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) have improved, and the larger the mask area, the more significant the improvement. Experiments prove that the images restored by the proposed algorithm have more reasonable structures and richer details, and the image restoration effect is superior.**

## Keywords

**Network Image Inpainting; Generating Adversarial Networks; Attention Mechanism; Two-stage Framework.**

## 1. Introduction

Image restoration is a technology within the field of computer vision aimed at repairing, restoring, or enhancing damaged, deteriorated, or distorted images. It involves analyzing and processing missing elements, noise, blur, occlusion, or other types of damage in images to reconstruct, restore, or enhance the images' quality and visual content. The goal of image restoration is to restore the appearance and information of the original image as much as possible, making the restored images visually clearer and more accurate while preserving the images' authenticity. This can be achieved by applying various algorithms and techniques, including image interpolation, noise removal, edge repair, texture restoration, and color correction. Image restoration has widespread applications across numerous fields, including digital photography, digital image processing, printing and publishing, medical imaging, and cultural heritage preservation. It can assist in restoring old photos, repairing ancient artworks, improving the quality of medical scan images, and enhancing the visual effects and content of images. The method proposed in this paper renders the edges of the restored images clearer and the filled textures more realistically delicate and clear, thereby enhancing the overall coherence and semantic rationality of the images.

ISSN: 2688-8645

To address the issue of unclear textures and incoherent structures in generated images, inspired by reference [29] and adhering to the principle of edges first, textures second,' we have divided the image restoration process into two stages. In the first stage, an edge-generation network is trained to reconstruct and predict the missing edges of the damaged image. The integrated complete edge map and the damaged image then serve as inputs for the texture generation network in the second stage. The texture generation network fills in the missing areas with color and content based on the edge map and the rest of the image. Both stages of the generation network are based on GANs.

We evaluated our proposed model on the standard datasets Paris StreetView [27] and CelebA [28] and compared its performance with classical and state-of-the-art methods. Our paper has made the following contributions:

•This paper introduces a novel two-stage generative model aimed at resolving the challenges associated with blurry and discontinuous edges, along with the unclear and fine textures observed in existing image restoration algorithms when repairing extensive missing image areas. Constructing a residual attention network within a generative network, coupled with the integration of an expanded neighborhood attention mechanism into each residual block, enables heightened focus on significant image regions. This approach facilitates the improved capture and utilization of meaningful information, thereby augmenting the details and structure of the restored image. Particularly in the context of repairing missing areas, the selective emphasis on crucial contextual information can significantly enhance the realism and coherence of the generated image.

•With a focus on model stability and performance, this paper introduces the Binary Cross-Entropy with Logits loss function into the discriminators of the two-stage model. This loss function, combining sigmoid activation and binary cross-entropy, is well-suited for binary classification problems, aligning with the discriminator's task in GANs of classifying between real and generated samples. This loss function aids the discriminator in learning how to more effectively distinguish between real and generated images, thereby enhancing the overall network performance.

•In order to alleviate the problem that GAN models are difficult to train, we use the Ranger21 optimizer, which improves the training stability, accelerates the convergence speed, enhances the generalization ability, and improves the training efficiency, so as to come to improve the training dynamics of the model, and enhance the performance and efficiency of the model.

## 2. Related Work

### 2.1. Image Inpainting

Traditional image restoration methods can be considered sequence-based approaches, primarily divided into patch-based and diffusion-based methods. Patch-based image restoration methods first generate patches based on existing image information, then employ algorithmic models to select patches that well match the missing areas, subsequently filling these areas to accomplish the restoration task. [1] Efros and Leung proposed a non-parametric texture synthesis restoration method based on Markov random fields [2], which requires substantial time to calculate similarity scores between samples. Building on this, Wei and Levoy [3] reduced computation time by designing a multi-resolution pyramid structure (i.e., the WL algorithm). Ashikhmin [4] improved the WL algorithm [3] from the perspective of increased repair speed; Drori et al.[5]guided the algorithm through iterative complete image restoration using known image training set context. Concurrently, Levin et al [6]. introduced statistical concepts to the restoration task. Criminisi et al. replicated structural and texture information propagation to the missing areas of damaged images, building on the Efros model. Addressing the issue of inaccurate filling orders in the Criminisi algorithm [7], Zhang Shen-Hua et al [8].

ISSN: 2688-8645

introduced curvature and gradient information to obtain a more reliable sample restoration order. Zhao Na et al [9]. employed Markov Random Fields [10] as the matching criterion for the Criminisi algorithm [7] to enhance image texture details. Barnes et al [11]. reduced memory consumption and computational cost during the search process by employing a fast nearest-neighbor algorithm to search for the most similar samples.

Diffusion-based methods refer to the use of partial differential equations in mathematics or physics to deduce image information of unknown areas based on the boundary information of the missing areas, with the deduction process propagating from the periphery of the missing area towards its center [12]. Bertalmio et al. [13] proposed a restoration method based on isophotes and diffusion principles capable of filling missing areas from any direction. Shen et al. [14] introduced a digital restoration method combining the TV denoising model with PDEs. Telea et al. [15] built on this by proposing a Fast Marching Method (FMM), which enables quicker implementation of PDE-based algorithms for rapid propagation of image information. Due to a lack of understanding of high-level semantic image content, traditional restoration methods struggle with complex structural images or excessively large masked areas, failing to produce results that are semantically consistent and visually plausible with the original image.

In recent years, with the profound and effective research of deep learning in image processing tasks, numerous researchers have begun to explore deep learning-based image restoration techniques, attempting to incorporate various advanced techniques to achieve image restoration and proposing a multitude of restoration methods. Pathak et al. proposed an unsupervised feature semantic restoration method based on contextual information, known as context encoders (CE) [16], capable of generating content for any region of an image. Addressing the issue of inconsistent overall structure in CE [16], Iizuka et al. [17] introduced a context-aware local discriminator for generating repaired images with globally and locally consistent semantics. Cao et al. [18] employed an encoder-decoder to learn sketch tensor spaces for reliable prediction of an image's overall structure while also incorporating gated convolutions[19] and efficient attention modules into the network. Wang et al. [20] utilized a multi-column structure to decompose images into components with different receptive fields and feature resolutions, predicting global and local structural feature information at various scales. Liu et al. [21] introduced an interactive encoder-decoder network that employs a multi-scale approach to jointly repair image structure and texture information and integrates a bilateral propagation activation function to balance the consistency of image structure and texture features. Sagong et al. [22] proposed an image restoration method consisting of a shared encoding network and parallel decoding networks to reduce computational costs and testing time. Guo et al. [23] employed the U-Net concept [7] to propose a full-resolution residual network for the incremental repair of irregularly missing areas. GANs were introduced by Goodfellow et al. [1] in 2014, and in 2017, Isola et al. [24] proposed a model named PatchGAN, which applies Conditional Generative Adversarial Networks (CGANs) to the field of image restoration. Addressing the issue of large-area image losses, Yeh et al. [25] proposed deep generative models (DGMs). Yu et al. [19] introduced a GAN network based on gated convolution, named SN-PatchGAN, designed for targeted feature extraction in images to minimize the interference of invalid pixels. Nazeri et al. [26] combined edge prior information with the PatchGAN [24] restoration model to predict image edge information, thereby guiding the image restoration process.

## 2.2. Attention Model

An attention mechanism is a model structure that mimics the human attention mechanism and is used to enhance the attention of a neural network to the input data, improving the performance and generalization of the model. In the field of deep learning, attention mechanisms are often used to process sequential data (e.g., natural language, audio, video, etc.)

ISSN: 2688-8645

or image data so that the network can more effectively learn and understand the key information in the input data.

The basic idea of the attention mechanism is that the network should not treat all input elements equally in a generalized manner when processing input data, but rather pay attention selectively according to their importance. At each step or layer, the network dynamically learns to assign attention weights based on the characteristics of the input data, enabling the network to focus more on the parts that are meaningful to the task at hand. The attention mechanism consists of the following key steps:

(1) Calculating Attentional Weights. At each step or layer, the network computes the attentional weights of each element in the input data by learning. These weights are usually obtained by somehow weighting and summing the features of the input data, with higher weights indicating that the corresponding input elements are more important.

(2) Weighted input features. The features of the input data are weighted and summed according to the calculated attention weights. This allows the network to pay more attention to the information that is important for the current task in the subsequent processing and reduces the interference of irrelevant information.

(3) Application of Attention Weighted Features. Attention weighted features obtained by weighted summation are fed into the next layer or next step of the network for further processing. The attention mechanism can be applied in various network structures such as Recurrent Neural Networks (RNN), Transformer, Convolutional Neural Networks (CNN), etc.

The Dilated Neighborhood Attention (DiNA) mechanism chosen in this paper enables the network to capture a wider range of contextual information by expanding the model's sensory field, thus enabling the model to better understand the global structure and details of the image.

### 2.3. Optimizer

The optimizer refers to the process of adjusting the parameters of the loss function during deep learning backpropagation, aiming to iteratively update them in the appropriate direction to gradually minimize the loss function and converge towards its global minimum.

Optimization algorithms have undergone development, including SGD, SGDM, AdaGrad, RMSProp, AdaDelta, Adam, Nadam, and others. Among these, SGD struggles to converge, and it encounters challenges when faced with local optima or saddle points, resulting in gradients becoming 0 and halting parameter updates. SGDM alleviates the issue of SGD encountering local optima with gradients of 0 by incorporating momentum factors. However, even with momentum, SGDM may oscillate in local optima when the valleys are deep. The second-order momentum of AdaGrad continuously increases, causing the learning rate to gradually decrease to 0, prematurely terminating the training process. RMSProp and AdaDelta address the issue of accumulating second-order momentum excessively, preventing premature termination of the training process.

Most previously published papers have focused on incremental enhancements to existing optimization algorithms, often introducing them as new optimizers rather than combinable algorithms. Ranger21 is a novel optimizer that integrates AdamW with eight components, resulting in notable enhancements such as improved validation accuracy, accelerated training speed, smoother training curves, and the capability to train ResNet50 on ImageNet2012 without batch normalization layers, addressing the issue of AdamW frequently remaining in an unfavorable initial state.

## 3. Approach

The image inpainting network proposed in this article consists of two stages: an edge generation network and a texture generation network. This structure decouples the entire

ISSN: 2688-8645

image inpainting task into two subtasks: edge inpainting and texture inpainting, which helps to stabilize training and expand receptive fields. Both stages follow the generation of adversarial models. Each stage consists of a generator and a discriminator. The generator is an encoder-decoder structure, where the encoder down samples the image twice, followed by a residual attention network consisting of 8 residual blocks with an expanded neighborhood attention mechanism. The decoder finally up samples the image back to its original size. The overall repair framework is shown in Figure 1.

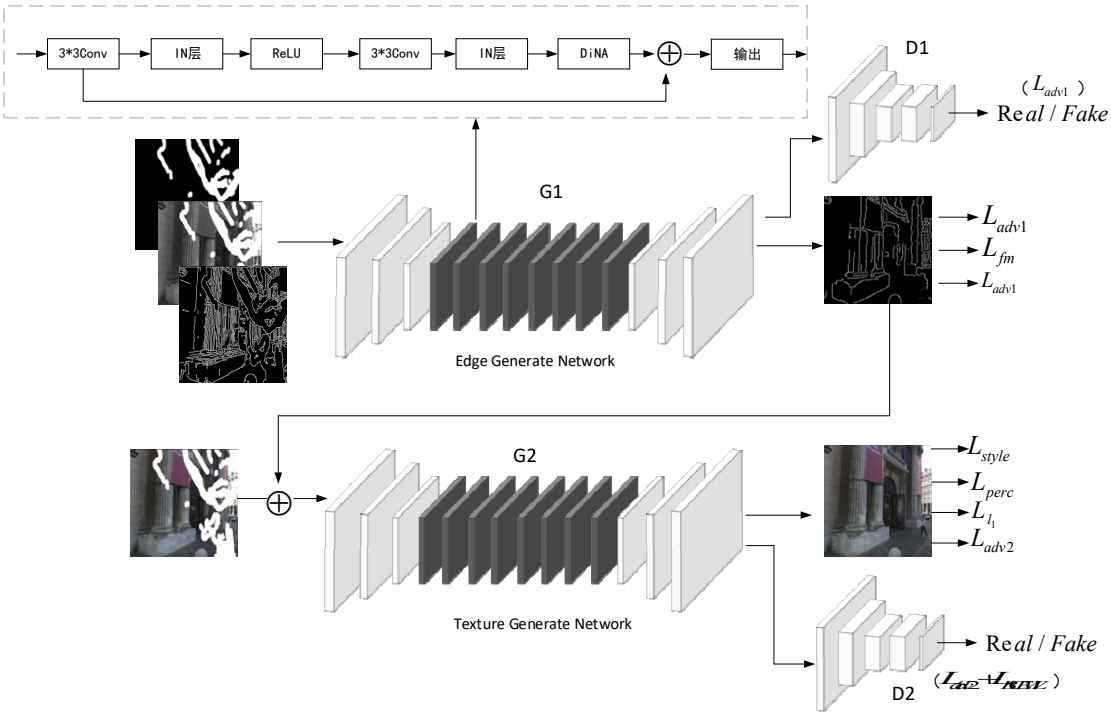


**Figure 1.** Overall framework diagram

## 3.1. Edge Generation Network

The network's generator employs an encoder-decoder architecture. Initially, the encoder conducts a mirror filling operation on the input image to preserve the image edges. Subsequently, two down-sampling operations are executed to extract the primary features and global structure, projecting them into the latent space. The residual attention network comprises eight residual blocks incorporating dilated neighborhood attention. These residual blocks utilize dilated convolution with a dilation factor of 2. Integration of an attention mechanism into the residual blocks enables the network to focus more on crucial image regions, enhancing the capture and utilization of meaningful information and thereby improving the details and structure of the generated image. This is particularly beneficial when restoring missing areas, as it enables selective attention to significant contextual information, thereby enhancing the realism and coherence of the generated images. Eventually, the decoder reconstructs the feature map from the latent space, progressively restoring its resolution and details via up-sampling and convolution operations. The decoder produces the final output image, ensuring its resolution matches that of the input image.

$I_{gt}$ represents a real image, $I_{gray}$ represents a grayscale image with a mask, $C_{gt}$ represents an edge image of a real image, then the edge image with a mask is represented as:

$$\tilde{C} = C_{gt} \odot (1 - M) \tag{1}$$

ISSN: 2688-8645

$C_{pred}$ is edge map representing the predicted defect area, $M$ is mask representation, $\odot$ is Hadamard plot

The output of the edge generator can be represented as:

$$C_{pred} = G_1\left(I_{gray}, M, \tilde{C}\right) \tag{2}$$

The discriminator network structure utilizes convolutional layers and LeakyReLU activation functions to extract image features while gradually reducing spatial resolution. The final layer outputs a single channel for binary classification tasks, enabling discrimination between the generated repaired image and the real image, thereby encouraging the generator to produce more realistic repairs.

Using $C_{gt}$ and $C_{pred}$ as inputs to the edge generation network discriminator $D_1$ the training objectives of the edge generation network include adversarial loss $L_{adv_1}$ and feature matching loss $L_{fm}$ .

## 3.2. Texture Generation Network

The structure of the texture generation network is similar to that of the edge generation network, which transforms color images with masks. $\tilde{I}_{gt} = I_{gt} \odot (1 - M)$ ,The complete edge map $C_{comp}$ serves as the input for the texture generation network, ultimately predicting the complete image $I_{pred}$ , The complete edge map $C_{comp}$ is constructed by combining the output $G_1$ edge map $C_{pred}$ of the missing area $\tilde{C}$ is output by the edge generation network:

$$C_{comp} = C_{pred} + \tilde{C} \tag{3}$$

The output of the texture generation network can be represented as:

$$I_{pred} = G_2\left(C_{comp}, \tilde{I}_{gt}\right) \tag{4}$$

This stage is trained using a joint loss consisting of $L_1$ , adversarial loss $L_{adv}$ , perceptual loss $L_{per}$ , and style loss $L_{style}$ .

## 3.3. Residual Attention Network

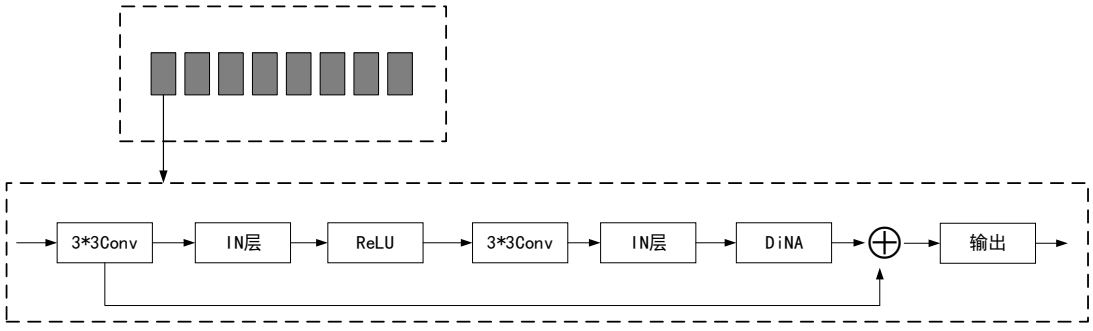


**Figure 2.** Residual attention network

To extract image features under mask occlusion, this paper constructs a residual attention network based on ResBlock, known for its strong performance. The network comprises 8 residual blocks, each enhanced with an expanded neighborhood attention mechanism. The residual attention unit is formed by the DiNA module, and the residual blocks employ dilated convolution with a dilation factor of 2. The constructed residual attention network enhances

ISSN: 2688-8645

the model's ability to generalize to occluded regions and facilitates more effective feature extraction. The residual attention network is depicted in Figure 2.

### 3.3.1. Attention Mechanism DiNA Module

Dilated Neighborhood Attention (DiNA) is a state-of-the-art attention mechanism designed to improve the performance of deep learning models especially in image processing and vision tasks. It enables the network to capture a wider range of contextual information by expanding the model's receptive field. Because it allows the model to better understand the global structure and details of an image, this mechanism is particularly useful for tasks such as image inpainting. Traditional attention mechanisms, such as Self-Attention or local attention in Convolutional Neural Networks (CNNs), typically focus on local regions or relatively small neighborhoods of the input data. In contrast, DiNA increases the size of the receptive field through a dilation operation so that the model can take into account contextual information from farther away while maintaining computational efficiency. The overall structure of the DiNA module is shown in Figure 3.

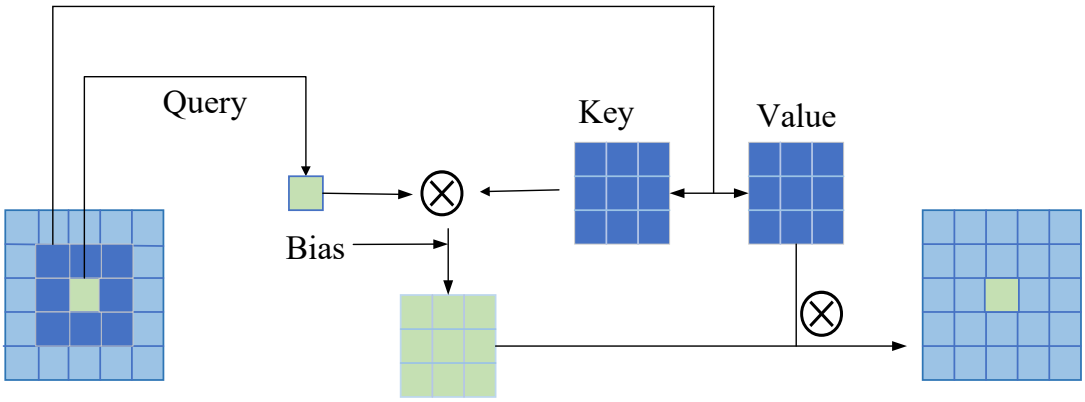


**Figure 3.** DiNA attention module

DiNA is calculated as follows:

For input $X \in R^{n\times d}$, given an inflated value $\delta$, we define the neighborhood attention weight $A_i^{(k,\delta)}$ of the $i$ token with neighborhood size $k$ can be expressed as.

$$A_i^{(k,\delta)} = \begin{bmatrix} Q_i K^T_{\rho_1^\delta(i)} + B_{(i,\rho_1^\delta(i))} \\ Q_i K^T_{\rho_2^\delta(i)} + B_{(i,\rho_2^\delta(i))} \\ \vdots \\ Q_i K^T_{\rho_k^\delta(i)} + B_{(i,\rho_k^\delta(i))} \end{bmatrix} \tag{5}$$

$B(i,j)$ is the relative positional deviation between any two tokens $i$ and $j$, $\rho_j(i)$ represents the $j$ nearest neighbor token of token, $Q_i$ is the query vector for the $i$ token, $K$ is the key vector of $k$ nearest neighbor tokens.

$V_i^{(k,\delta)}$ is the k-nearest v of the $i$ token, which can be expressed as:

$$V_i^{(k,\delta)} = \begin{bmatrix} V^T_{\rho_1^\delta(i)} & V^T_{\rho_2^\delta(i)} & \cdots & V^T_{\rho_k^\delta(i)} \end{bmatrix}^T \tag{6}$$

The neighborhood attention output of the $i$ token with a neighborhood size of $k$ is defined as:

ISSN: 2688-8645

$$DiNA_k^{\delta}(i) = soft\max\left(\frac{A_i^{(k,\delta)}}{\sqrt{d_k}}\right)V_i^{(k,\delta)} \tag{7}$$

Figures 4 present comparisons between fully connected layers, convolutions, and dilated convolutional receptive fields, as well as among the receptive fields of the self-attention mechanism, neighborhood attention mechanism (NA), and dilated neighborhood attention mechanism (DiNA). The illustrations demonstrate that the DiNA module achieves a superior receptive field without significantly increasing computational complexity.

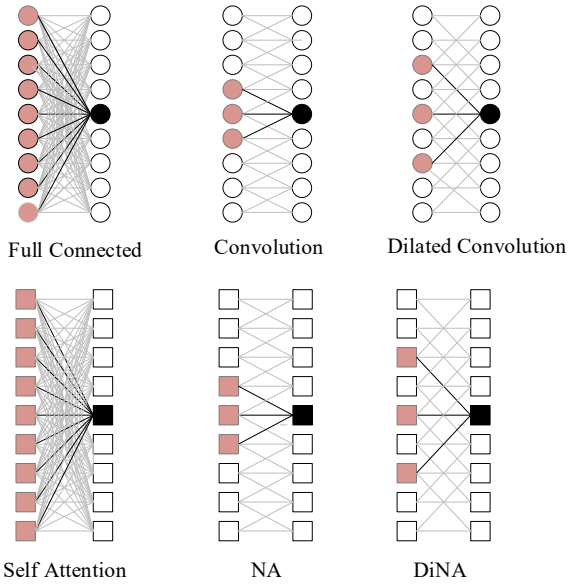


**Figure 4.** Comparison of receptive fields

## 3.4. Ranger21 Optimizer

The Ranger21 deep learning optimizer employed in this paper integrates the following optimization principles: based on AdamW, adaptive gradient clipping, gradient centralization, positive-negative momentum, weight decay with regularization, stable weight decay, linear learning rate warm-up, and an exploratory learning rate scheduler.

AdamW typically results in lower training losses and testing errors for the model. Additionally, AdamW demonstrates better generalization performance even when the same training loss is fixed, which is another reason why it serves as the core foundation.

The pseudo-code for the Ranger21 optimizer is as follows Table 1:

## 3.5. Construction of Loss Function

### 3.5.1. The Loss of Edge Generative Networks

The overall loss of the edge generation network is represented as:

$$\min_{G_1}\max_{D_1} L_{G_1} = \min_{G_1}\left(\lambda_{adv,1}\max_{D_1}\left(L_{adv_1}\right) + \lambda_{FM}L_{FM}\right) \tag{8}$$

Where $\lambda_{adv_1}$ and $\lambda_{fm}$ are regularization parameters.

The adversarial loss ( $L_{adv_1}$ ) of the edge generation network is represented as:

$$L_{adv_1} = \mathrm{E}_{(C_{gt},I_{gray})}\left[\log D_1\left(C_{gt},I_{gray}\right)\right] + E_{I_{gray}}\log\left[1 - D_1\left(C_{pred},I_{gray}\right)\right] \tag{9}$$

**ISSN: 2688-8645**

**Table 1.** The pseudo-code for the Ranger21 optimizer.

Algorithm 1: The pseudo-code for the Ranger21 optimizer

Require: $f(\theta)$ :objective function Require: $\theta_0$ : initial parameter vector Require: $\eta$ :learning rate Require: $\lambda$ :weight decay (default:)

Require: $\beta_0, \beta_1, \beta_2, \beta_{lookahead}$ :decay rates

Require: $\alpha$ , $\alpha_{clipping}$ :epsilon for numerical stability

Require: $T_{clipping}$ :threshold for adaptive gradient clipping

Require: $k_{lookahead}$ :frequency of the update

Require: $t_{\max}$ :number of iterations

Require: $t_{warmup}$ :number of learning rate warm-up iterations

Require: $t_{warmdown}$ :number of learning rate warm-down iterations

1. $m_0, v_0, v_{\max} \leftarrow 0, 0, 0$
2. $l_0 \leftarrow \theta_0$
3. **for** $t \leftarrow 1$ to $t_{\max}$ **do**
4. $g_t \leftarrow \nabla f_t(\theta_{t-1})$
5. **for** $r \in rows(g_t)$ **do**
6. **if** $\frac{\left\| g_t^r \right\|}{\max\left( \left\| \theta_t^r \right\|, \alpha_{clipping} \right)} > T_{clipping}$ **then**
7. $g_t^r \leftarrow T_{clipping} \frac{\max\left( \left\| \theta_t^r \right\|, \alpha_{clipping} \right)}{\left\| g_t^r \right\|} g_t^r$
8. **end if**
9. **end for**
10. $g_t = g_t - mean(g_t)$
11. $m_t \leftarrow \beta_1^2 m_{t-2} + \left(1 - \beta_1^2\right) g_t$
12. $\hat{m}_t \leftarrow \left( \left(1 + \beta_0\right) m_t - \beta_0 m_{t-1} \right) / \left(1 - \beta_1^t\right)$
13. $v_t \leftarrow \beta_2 v_{t-1} + \left(1 - \beta_2\right) g_t^2$
14. $v_{\max} \leftarrow \max\left(v_t, v_{\max}\right)$
15. $\hat{v}_t \leftarrow v_{\max} / \left(1 - \beta_2^t\right)$
16. $u_t \leftarrow \hat{m}_t / \left( \sqrt{\left(1 + \beta_0\right)^2 + \beta_0^2} \left( \sqrt{\hat{v}_t} + \alpha \right) \right)$
17. $\eta_t = \min\left( 1, \max\left( \frac{1 - \beta_2}{2} \bullet t, \frac{t}{t_{warmup}} \right), \frac{t_{\max} - t}{t_{warmup}} \right) \eta$
18. $d_t = \frac{\eta_t}{\sqrt{mean\left(\hat{v}_t\right)}} \lambda \left( 1 - \frac{1}{\left\| \theta_{t-1} \right\|} \right) \theta_{t-1}$
19. $\theta_t \leftarrow \theta_{t-1} - \eta_t u_t - \eta_t d_t$
20. **if** $t \% k_{lookahead} == 0$ **then**
21. $l_{t/k} \leftarrow \beta_{lookahead} l_{t/k-1} + \left(1 - \beta_{lookahead}\right) \theta_t$
22. $\theta_t \leftarrow l_{t/k}$
23. **end if**
24. **end for**
25. **return** $\theta_t$

ISSN: 2688-8645

The feature matching loss ( $L_{fm}$ ) is defined as:

$$L_{fm} = \mathrm{E}\left[\sum_{i=1}^{n}\frac{1}{N_i}\left\|D_1^{(i)}\left(C_{gt}\right)-D_1^{(i)}\left(C_{pred}\right)\right\|_1\right] \tag{10}$$

Where $n$ is the last convolutional layer of the discriminator( $D_1$ ), the number of elements( $N_i$ ) in the $i$ activation layer, and the activation in the $i$ layer of the discriminator( $D_1^{(i)}$ ).

### 3.5.2. The Loss of Texture Generative Networks

The overall loss of the generator in the texture generation network can be expressed as:

$$L_{G_2} = \lambda_{L_1}L_1 + \lambda_{adv_2}L_{adv_2} + \lambda_{per}L_{per} + \lambda_{style}L_{style} \tag{11}$$

The adversarial loss ( $L_{adv_2}$ ) is defined as:

$$L_{adv,2} = \mathrm{E}_{(I_{gt},C_{comp})}\left[\log D_2\left(I_{gt},C_{comp}\right)\right] + \mathrm{E}_{C_{comp}}\log\left[1-D_2\left(I_{pred},C_{comp}\right)\right] \tag{12}$$

The perceived loss ( $L_{per}$ ) penalizes results that deviate from labels in perception by defining distance measures between pre-trained network feature maps. The formula for this loss is as follows:

$$L_{per} = \mathrm{E}\left[\sum_{i}\frac{1}{N_i}\left\|\phi_i\left(I_{gt}\right)-\phi_i\left(I_{pred}\right)\right\|_1\right] \tag{13}$$

Where $\phi_i$ is the feature map of the i layer of the preprocessing network.

Style loss can effectively solve the "chessboard" artifacts caused by transposed convolutional layers. For a given size feature map ( $C_j \times H_j \times W_j$ ), the formula for calculating style loss is:

$$L_{style} = \mathrm{E}\left[\left\|G_j^{\phi}\left(\tilde{I}_{pred}\right)-G_j^{\phi}\left(\tilde{I}_{gt}\right)\right\|_1\right] \tag{14}$$

Among them, $G_j^{\phi}$ is the Gram matrix ( $C_j \times C_j$ ) constructed from feature maps ( $\phi_i$ ).

The overall loss of the discriminator is expressed as:

$$L_{D_2} = \lambda_{adv_2}L_{adv_2} + \lambda_{BCE}L_{BCE} \tag{15}$$

Within the GAN architecture, the primary role of the discriminator is to differentiate between the images generated by the generator and real images. Therefore, the discriminator requires an effective approach to measure and optimize its discriminative capability. The binary cross-entropy loss ( $L_{BCEWL}$ ) with logits is a suitable choice for this task, as it combines the sigmoid activation function with binary cross-entropy loss. This loss function computes the disparity between the predicted probability output by the discriminator and the real label. Through this process, the discriminator learns to more accurately discern between real and generated images. The binary cross entropy loss with Logits is represented as:

$$L_{BCEWL} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(\sigma(x_i)) + (1-y_i)\cdot log(1-\sigma(x_i))] \tag{16}$$

Among them, $N$ is the number of samples in the batch, $y_i$ is the true label of sample $i$ , $y_i \in \{0,1\}$ and $\sigma(x_i)$ are the sigmoid function applied to the original output( $x_i$ ).

ISSN: 2688-8645

# 4. Experiment

## 4.1. Experimental Setup

In this study, experiments were conducted on the Paris Street View [27] dataset and the CelebA-HQ [28] dataset to validate the performance of the model. The Paris Street View dataset comprises 14,900 images in the training set and 100 images in the test set, covering various architectural images collected from 12 cities, primarily featuring Parisian street scenes and suburbs. The CelebA-HQ dataset is a public dataset containing 30,000 facial images. For this study, the first 28,000 images were selected as the training set, and the remaining 2,000 images were used as the test set. The mask dataset utilized irregular masks[32], including 12,000 masks of various sizes and shapes. This study categorized the dataset into six levels based on the mask area size, namely 0-10%, 10%-20%, 20%-30%, 30%-40%, 40%-50%, and 50%-60%. During training, image restoration was performed according to different mask ratios, with all images and masks input at a size of $256\times256$.

The deep learning framework used in the experiment is Python 1.12.0, the software environment is Python 3.9, and the operating system is Linux. The hardware environment is a NVIDIA 3090-24G GPU. The optimization method adopts Ranger21, with a minimum batch size setting of 4 and an initial learning rate of 1e-4 for the generator. When training the edge generation network, this article first uses the Canny edge detection algorithm with a standard deviation of 2 for Gaussian smoothing filters to generate edges of non defective area images. The sensitivity of the detector is controlled by the standard deviation of the Gaussian smoothing filter, which is set to 2.

## 4.2. Evaluation Method

This study evaluates the restored images using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), the two most commonly used assessment methods in the field of image restoration.

PSNR is used to reflect the quality of the repaired image, with higher values indicating higher quality of the restored image. The formula is as follows:

$$PSNR = 10\log_{10}\left(\frac{MAX_i^2}{\sqrt{MSE}}\right) = 20\log_{10}\left(\frac{MAX_i}{\sqrt{MSE}}\right) \tag{17}$$

Among them, $MAX_i$ represents the maximum value of the color of the image points, if the sampling point is a bit. $MAX_i = 2^a - 1$ and MSE stands for mean square error.

SSIM evaluates the structural similarity between the original image and the repaired image from three aspects: brightness, structure, and contrast. The range of SSIM is [0,1], and the closer the structure of the two images is, the closer their values are to 1. The definition is as follows:

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_2\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{18}$$

Among them, $\mu_x$ and $\mu_y$ represent the pixel mean of the original image and the repaired image, $\sigma_{xy}$ represent covariance, $c_1$ and $c_2$ are constants.

## 4.3. Comparative Experiment

This paper compares the proposed model with representative image restoration models such as EdgeConnect[26], GC[19], MADF[29] to validate the effectiveness of our model. Figure 5 displays the restoration effects of our method versus these advanced models on the Paris Street View [27] dataset and CelebA-HQ dataset[28] under various proportions of irregular missing

ISSN: 2688-8645

areas. The first column shows the original images, the second column displays images with masks, the third column presents the restoration effects by GC [19], the fourth column by EC [26], the fifth column shows the results of MADF[29], and the last column the outcomes of our method. It can be observed that when the missing area is small, all methods can recover the basic structure.

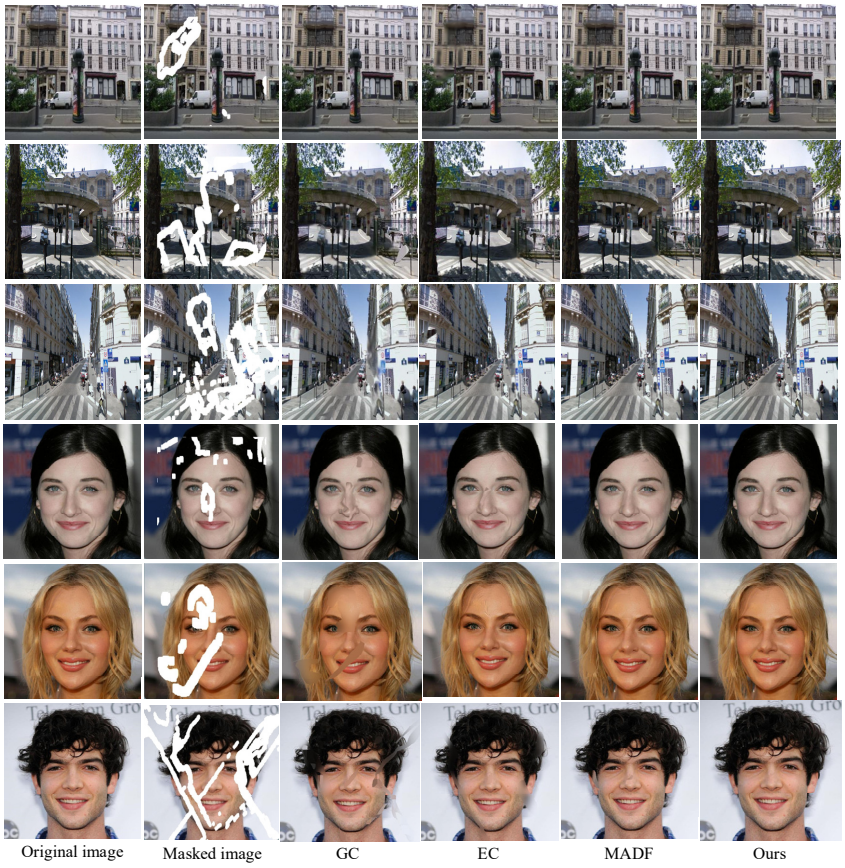


**Figure 5.** Experimental Comparison Chart

**Table 1.** Comparison of metrics across models on the Paris Street View dataset.

| index | Masks | 1-10% | 10-20% | 20-30% | 30-40% | 40-50% | 50-60% |
|---|---|---|---|---|---|---|---|
| PSNR | GC | 34.45 | 29.62 | 25.53 | 24.06 | 22.20 | 20.73 |
| | EC | 35.82 | 30.52 | 27.07 | 24.92 | 23.81 | 21.67 |
| | MADF | 36.36 | 30.71 | 27.63 | 25.42 | 23.50 | 21.29 |
| | Ours | 36.94 | 31.34 | 28.09 | 25.89 | 24.53 | 22.25 |
| SSIM | GC | 0.980 | 0.892 | 0.835 | 0.766 | 0.692 | 0.637 |
| | EC | 0.973 | 0.933 | 0.889 | 0.814 | 0.774 | 0.699 |
| | MADF | 0.979 | 0.933 | 0.891 | 0.833 | 0.767 | 0.693 |
| | Ours | 0.982 | 0.949 | 0.917 | 0.863 | 0.815 | 0.737 |

ISSN: 2688-8645

In order to objectively compare the performance of the algorithm proposed in this article with the comparison algorithm, the structural similarity (SSIM) and peak signal-to-noise ratio (PSNR) of various algorithms were compared. The performance on the Paris Street View [27] dataset is shown in Tables 1, while the performance on the CelebA-HQ [28] dataset is shown in Tables 2. From the data, it can be seen that the method proposed in this article is superior to other methods.

**Table 2.** Comparison of metrics across models on the CelebA-HQ dataset.

| index | Masks | 1-10% | 10-20% | 20-30% | 30-40% | 40-50% | 50-60% |
|---|---|---|---|---|---|---|---|
| PSNR | GC | 35.83 | 29.32 | 27.09 | 24.94 | 22.41 | 19.18 |
| | EC | 36.31 | 31.64 | 28.15 | 26.07 | 23.86 | 21.43 |
| | MADF | 38.47 | 32.26 | 29.53 | 26.13 | 24.32 | 22.47 |
| | Ours | 38.86 | 33.02 | 30.35 | 27.22 | 25.73 | 23.53 |
| SSIM | GC | 0.982 | 0.935 | 0.903 | 0.836 | 0.757 | 0.643 |
| | EC | 0.984 | 0.958 | 0.923 | 0.877 | 0.833 | 0.762 |
| | MADF | 0.986 | 0.978 | 0.959 | 0.923 | 0.885 | 0.825 |
| | Ours | 0.990 | 0.982 | 0.962 | 0.937 | 0.892 | 0.841 |

## 5. Conclusion

To address the issues of unclear textures, blurred structures, and lack of coherence in images generated by existing image restoration algorithms, this paper introduces an image restoration algorithm based on dilated neighborhood attention. The algorithm employs a two-stage restoration model that adheres to the "structure-first, texture-next" principle, initially restoring the overall structure of the image and then using that structure as a prior condition for texture restoration. A residual attention network is constructed using dilated convolution blocks and the Dilated Neighborhood Attention (DiNA) module, which effectively reduces computational complexity and enhances the capture of fine structures. In the discriminator of the texture generation network, a binary cross-entropy loss function with logits is used. To mitigate the instability of GAN model training and slow model convergence, the Ranger21 optimizer was selected, considering its impact on neural network performance.

Experimental results have demonstrated that, in comparison to conventional mainstream restoration techniques, the method introduced in this manuscript is more effective for structural reconstruction in complex scenarios and can significantly reduce the incidence of edge blurring during the restoration process. For images restored using the method proposed in this manuscript, texture quality is also superior.

## References

[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[2] Efros A A, Leung T K. Texture synthesis by non-parametric sampling[C]//Proceedings of the seventh IEEE international conference on computer vision. IEEE, 1999, 2: 1033-1038.

[3] Ashikhmin M. Synthesizing natural textures[C]//Proceedings of the 2001 symposium on Interactive 3D graphics. 2001: 217-226.

[4] Wei L Y, Levoy M. Fast texture synthesis using tree-structured vector quantization[C]//Proceedings of the 27th annual conference on Computer graphics and interactive techniques. 2000: 479-488.

[5] Drori I, Cohen-Or D, Yeshurun H. Fragment-based image completion[M]//ACM SIGGRAPH 2003 Papers. 2003: 303-312.

[6] Zomet. Learning how to inpaint from global image statistics[C]//Proceedings Ninth IEEE international conference on computer vision. IEEE, 2003: 305-312 vol. 1.

[7] Criminisi A, Pérez P, Toyama K. Region filling and object removal by exemplar-based image inpainting[J]. IEEE Transactions on image processing, 2004, 13(9): 1200-1212.

[8] ZHANG S H, WANG K G, ZHU X. Improved Criminisi algorithm constrained by local feature[J]. Computer Engineering and Applications, 2014, 50(8): 127-13.

[9] Zhao N, WANG H Q, Wu M. Criminisi digital inpainting algorithm based on Markov random field matching criterion[J]. Journal of Frontiers of Computer Science and Technology, 2017, 11(7): 1150-1158.

[10] CROSS G R, JAIN A K. Markov random field texture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983(1): 25-39.

[11] BARNES C, SHECHTMAN E, FINKELSTEIN A, et al. PatchMatch: a randomized correspondence algorithm for structural image editing[J]. ACM Transactions on Graphics, 2009, 28(3): 24.

[12] Guillemot C, Le Meur O. Image inpainting: Overview and recent advances[J]. IEEE signal processing magazine, 2013, 31(1): 127-144.

[13] Bertalmio M, Sapiro G, Caselles V, et al. Image inpainting[C]//Proceedings of the 27th annual conference on Computer graphics and interactive techniques. 2000: 417-424.

[14] Shen J, Chan T F. Mathematical models for local nontexture inpaintings[J]. SIAM Journal on Applied Mathematics, 2002, 62(3): 1019-1043.

[15] Telea A. An image inpainting technique based on the fast marching method[J]. Journal of graphics tools, 2004, 9(1): 23-34.

[16] PATHAK D, KRÄHENBÜHL P, DONAHUE J, et al. Context encoders: feature learning by inpainting [C] // Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Jun 27-30, 2016. Washington: IEEE Computer Society, 2016: 2536-2544.

[17] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. ACM Transactions on Graphics, 36(4), 2017.

[18] CAO C, FU Y. Learning a sketch tensor space for image inpainting of man- made scenes [C]// Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, Oct 10-17, 2021. Piscataway: IEEE, 2021: 14509- 14518.

[19] YU J, LIN Z, YANG J, et al. Free-form image inpainting with gated convolution[C]//Proceedings of the 2019 IEEE/ CVF International Conference on Computer Vision, Seoul, Oct 27-Nov 3, 2019. Piscataway: IEEE, 2019: 4470-4479.

[20] WANG Y, TAO X, QI X, et al. Image inpainting via generative multi-column convolutional neural networks[C]//Proceedings of the Annual Conference on Neural Information Processing Systems 2018, Montréal, Dec 3-8, 2018: 329-338.

[21] LIU H, JIANG B, SONG Y, et al. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations [C]//LNCS 12347: Proceedings of the 16th European Conference on Computer Vision, Glasgow, Aug 23- 28, 2020. Cham: Springer, 2020: 725-741.

[22] SAGONG M, SHIN Y, KIM S, et al. PEPSI: fast image inpainting with parallel decoding network [C]// Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Los Angeles, Jun 15-21, 2019. Piscataway: IEEE, 2019: 11360-11368.

[23] GUO Z Y, CHEN Z B, YU T, et al. Progressive image inpainting with full-resolution residual network [C] // Proceedings of the 27th ACM International Conference on Multimedia, Nice, Oct 21-25, 2019. New York: ACM, 2019: 2496-2504.

[24] Isola P, Zhu J Y, Zhou T H, et al. Image-to-image translation with conditional adversarial networks [C] //2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 5967-5976.

[25] Yeh R A, Chen C, Lim T Y, et al. Semantic image inpainting with deep generative models[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6882-6890.

[26] Nazeri K, Ng E, Joseph T, et al. Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv 2019[J]. arXiv preprint arXiv:1901.00212, 2020.

[27] Doersch C, Singh S, Gupta A, et al. What makes paris look like paris?[J]. Communications of the ACM, 2015, 58(12): 103-110.

[28] Karras T, Aila T, Laine S, et al. Progressive growing of gans for improved quality, stability, and variation[J]. arXiv preprint arXiv:1710.10196, 2017.

[29] Zhu M, He D, Li X, et al. Image inpainting by end-to-end cascaded refinement with mask awareness [J]. IEEE Trans on Image Processing, 2021, 30: 4855-4866.