INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 18, Issue: 2, Month: April, Year: 2023
Article Number: 5088, https://doi.org/10.15837/ijccc.2023.2.5088

A Novel Generative Image Inpainting Model with Dense Gated Convolutional Network

Xiaoxuan Ma, Yibo Deng, Lei Zhang, Zhiwen Li

Xiaoxuan Ma*
School of Electrical and Information Engineering
Beijing University of Civil Engineering and Architecture
Beijing 100044, China
*Corresponding author: maxiaoxuan@bucea.edu.cn

Yibo Deng
School of Electrical and Information Engineering
Beijing University of Civil Engineering and Architecture
Beijing 100044, China
210811021003@stu.bucea.edu.cn

Lei Zhang
School of Electrical and Information Engineering
Beijing University of Civil Engineering and Architecture
Beijing 100044, China
lei.zhang@bucea.edu.cn

Zhiwen Li
School of Electrical and Information Engineering
Beijing University of Civil Engineering and Architecture
Beijing 100044, China
2108550020021@stu.bucea.edu.cn

Abstract

Damaged image inpainting is one of the hottest research fields in computer image processing. In view of the problem that existing image inpainting methods lead to an increase in network model parameters and unstable training processes due to the deepening of network layers and the direct connection between layers, as well as the problem that the edge features of damaged areas are discontinuous and the details are blurred in the image inpainting results, a novel generative image inpainting model with dense gated convolutional network(DGCN) by modifying the gated convolutional network structure is proposed in this paper. First, holistically-nested edge detector (HED) is utilized to predict the edge information of the missing areas to assist the subsequent inpainting task to reduce the generation of blur and artifacts. Then, dense connections and indirect connections are added to the generative network to reduce the network parameters while reducing the risk of instability in the training process. Finally, experimental results on the CelebA and Places2 datasets show that the proposed model achieves better inpainting results in terms of PSNR, SSIM and visual effects compared with other classical image inpainting models.

Keywords: Densely connected convolutional networks; gated convolution; image inpainting; generative adversarial networks.

1 Introduction

Image inpainting, also called image completion or hole filling, is used to complete or restore pixel information of missing areas in the image. Its goal is to achieve semantic correctness and visual integrity, that is, correct global semantic structure and fine texture. Image inpainting technology can be used to recover the missing information of images or videos, and the inpainting results can also assist other image processing for subsequent operations. Currently, an increasing number of pictures and videos are captured by cameras. However, the information captured by the camera may be lost due to some factors, such as image damage or blurring, and the use of such images in other visual tasks may lead to some problems. Therefore, sometimes it is necessary to preprocess the acquired images, such as inpainting or completing the missing part of the image or deblurring it.

In the field of image inpainting, there are two main types of inpainting methods, traditional methods based on texture or diffusion and methods based on deep learning. Traditional methods based on texture or diffusion are also known as nondepth learning methods. For example, the nonparametric sampling and texture optimization was proposed for texture synthesis by A. A. Efros [1] and Kwatra, V. et al. [2]. Ding, Sundaresh et al. used nonlocal texture matching and nonlinear filtering for image inpainting [3]. Erkan U, Enginoglu S et al. introduced the similarity of pixel values as prior knowledge [4]. Wang et al. improved the priority function and proposed an exemplar-based image inpainting method using structure-consistent patch matching [5]. Song Y., Yang, C. et al. proposed using context information for image inpainting [6]. These methods can better use the texture information, and the processing speed of the image is also fast, but these traditional methods will produce artifacts or not ideal results for large damaged images. Traditional methods only use shallow feature information, but there is more contextual information and deep features in the image.

The other type of image inpainting method is based on deep learning. With the rapid development of artificial intelligence and computers, deep learning has been applied to various fields, such as text detection [7], image detection [8], image classification [9][10], stock price prediction [11][12], and machine translation [13]. To extract more feature information from images, deep learning began to be combined with image inpainting. During this period, an increasing number of network architectures based on deep learning were proposed and achieved certain results. Among the proposed deep learning inpainting algorithms, the architecture of generative adversarial networks (GAN) is widely used for image inpainting. GAN is composed of a generator and discriminator [14], and the inpainting effect of the generator is improved through a game between them. At present, many image inpainting networks based on GAN have been proposed, such as context encoders [15] or multi-column convolutional neural networks [16]. However, most of the previous algorithms are based on regularly damaged areas for inpainting. The inpainting of irregular holes is still difficult. Guilin Liu et al. proposed partial convolution for the inpainting of images with irregular holes and made better use of the effective pixels in the images [17]. Yu, J. et al. proposed a gated convolutional network, which can dynamically learn the features in the image and inpaint the images with a free-form mask [18]. This method improves the inpainting result and increases the interaction between humans and computers, that is, inpainting in the form of a human-guided mask, which is more closely combined with practice. These methods directly inpaint the damaged area of images according to the extracted feature information. The inpainting effect is good for damaged images with simple texture structures, but it is not satisfactory for details with large damaged areas or more complex images. Artifact and blur will appear at the damaged edge of the image. To solve this problem, an edge detector is introduced into the image inpainting network to extract edge information, which is used as prior knowledge for image inpainting. The goal of edge detection is to extract object boundaries and perceive significant edges from natural images. These edges retain the key points of the image and ignore unexpected details [19]. At present, many edge detection algorithms have been proposed, including the traditional Canny detector [20] and those based on the depth convolutional neural network, such as BDCN [19], RCF [21], and holistically-nested edge detection (HED) [22]. For example, M. Arimoto, J. Hara and H. Watanabe used DexiNed [23] to extract edge information and considered the edge connectivity of defects in image inpainting [24]. E2I [25] inpaints the edge information extracted by HED, and then inpaints the damaged area on this basis. To continue to optimize the problem of blurred damaged edges, the prior knowledge of variants is used for image inpainting. Chen et al. [26] proposed a semantic prior-driven fused contextual transformation network for image inpainting and used the semantic prior generator to map the semantic features of ground truth images and the low-level features of broken images to semantic priors. Li et al. [27] proposed a fine inpainting method for incomplete images based on features fusion and two-steps inpainting. Dynamic memory networks (DMN+) are used to fuse the external features and internal features of the incomplete image to generate the incomplete image optimization map. Multi-stage image inpainting network is a research direction at present, which will obtain a better inpainting effect, but the parameters of the network model and the memory of the computer will be greatly improved, which will make the training of the network model more difficult.

To simplify the model, these methods [15][16][17][18] use direct connections between convolution layers. This connection mode will increase the risk of gradient disappearance or overfitting during training. In view of the problems caused by direct connections, He Kaiming et al. proposed a deep residual network (ResNet) and used a special jump connection method to reduce the problem of network gradient disappearance [28], but this method produces a large number of parameters. To extract deeper feature information and improve the effect of inpainting, the depth of the image inpainting model is often deepened. This will bring a huge number of parameters, make model training difficult, and at the same time, the risk of overfitting and gradient disappearance will also increase. Dense connection [29] was proposed to reduce the number of parameters and calculation cost by Huang, G. and Liu, Z. et al.. Pleiss, G. et al. proposed a more efficient dense network to further reduce network memory [30].

In this paper, we propose a novel generative image inpainting model with dense gated convolutional network (DGCN) by modifying the gated convolutional network structure [18]. First, the HED edge detector is introduced to predict the edge information of the missing area to assist the subsequent inpainting task to reduce the generation of artifacts. Then, dense connections and indirect connections are added to the image inpainting network to reduce the risk of instability in the training process while reducing network parameters. Finally, the DGCN model is trained on the CelebA and Places2 datasets and compared with several typical image inpainting networks,such as Contextual Attention (CA) [31], Globally and Locally Consistent Image Completion (GLCIC) [32], EdgeConnect (EC) [33] and original gated convolution (GC) [18]. After testing, the objective evaluation indices and visual effect of the DGCN are all improved when inpainting damaged images. The main contributions of this paper can be summarized as follows:

(1) Dense gated convolutional network (DGCN) is proposed, in which dense connections and indirect connections are used between layers to replace the original direct connections. DGCN can reduce the risk of instability in the training process while reducing network parameters.

(2) HED edge detector is introduced in the DGCN preprocessing stage to predict the edge information of the missing area to assist the subsequent inpainting task. The extracted edge information is used as the prior knowledge of image inpainting, which can optimize the inpainting effect.

(3) Compared with several typical image inpainting algorithms, the inpainting results and visual effects of our proposed model on the CelebA and Places2 datasets have been improved. Compared with the gated convolutional network, the DGCN improves the PSNR and SSIM by 10.16% and 1.09%, respectively, and optimizes the structural texture details of the inpainted area.

2 Related Works

To improve the inpainting effect of the network and reduce the risk of overfitting or gradient disappearance that may occur in the training process of the gated convolutional network or most other image inpainting networks, the architecture of the gated convolutional network is modified in this paper. Dense connections and indirect connections are introduced into the inpainting network to replace the traditional direct connection mode. Dense connection mode not only solves the problem of unstable training, but also reduces the network parameters. The principles of gated convolution and dense connection are analyzed in detail in 2.1 and 2.2.

2.1 Gated Convolution

Gated convolution is a special partial convolution [17] proposed by Jiahui Yu et al.[18]. It can show different sensitivities to different feature information in images. Prior to this, many methods have been proposed for regular hole filling, such as Wang, Y. et al.' s multi-branch convolution [16], which introduced a three-branch structure network to inpaint images containing regular masks (damaged area). However, with the development of images, an increasing number of images with irregularly damaged areas need to be inpainted, and it is still difficult to inpaint irregularly damaged areas. At the same time, there is much feature information on the edges of damaged areas, so how to properly handle the edge information of irregular holes is also a difficult problem. In previous models [15][16], ordinary convolution is used to extract feature information. Traditional ordinary convolution regards valid and invalid feature information with the same weight. As a result, invalid information will affect the inpainting effect.

Partial convolution is proposed to solve the problem that traditional ordinary convolution cannot make full use of invalid and valid features inside and outside the mask (damaged area). Valid elements (unmasked pixels) and missing pixels (masked pixels) are processed in different ways. The convolution layers of partial convolution only convolve and normalize the valid pixels that meet the conditions and then automatically update the mask according to the mask updating rules until all values in the mask are 1. The specific principle of partial convolution is as follows.

For hole filling, the input of each layer is composed of valid pixels or features outside holes and invalid ones in masked regions. According to this principle, partial convolution treats the valid pixels as **1** and the invalid pixels as **0**. The output value of partial convolution layers can be calculated by Formula (1):

$$\mathbf{x}' = \begin{cases} \mathbf{W}^T(\mathbf{X} \odot \mathbf{M})\frac{sum(\mathbf{1})}{sum(\mathbf{M})} + \mathbf{b} & if sum(M) > 0 \\ 0 & otherwise \end{cases} \tag{1}$$

where $\mathbf{W}$ is the convolution operation of the convolution filter and b is the corresponding bias. $\mathbf{X}$ is the characteristic value or pixel value of the current convolution (sliding window), and $\mathbf{M}$ is the corresponding binary mask. $\odot$ represents the multiplication of corresponding position elements, and $\mathbf{1}$ is the total 1 tensor with the same shape as $\mathbf{M}$. According to Formula (1), the output value only depends on the unmasked input. The scale factor sum $(\mathbf{1})$/sum $(\mathbf{M})$ takes an appropriate range to adjust the variation of the valid (unmasked) input. After the partial convolution process, the mask is updated as shown in Formula (2):

$$m' = \begin{cases} 1 & if sum(\mathbf{M}) > 0 \\ 0 & otherwise \end{cases} \tag{2}$$

According to Formula (2), the updating rule of the mask is as follows. If at least one valid input value is output in the convolution process, the position is marked as valid.

However, if we simply regard all position information as valid pixels and invalid pixels (i.e., 0 and 1, respectively), we will ignore much image information, such as edge information. Moreover, if it is extended to user-guided image inpainting, that is, the user provides sparse sketches in the masked area, it is difficult to determine these pixel positions as valid or invalid.

Yu, Jiahui et al. proposed a dynamic updating mechanism for gated convolution [18], which can automatically learn the mask, and the mask may have different values according to whether the current position in the input image is masked, even in the deep layers. As a dynamic learning version of the partial convolution kernel, gated convolution can process images containing user directed masks differently. For gated convolution, there are not only 0 and 1 (valid pixels and invalid pixels, respectively) but also 0-1 range values as weights to better handle the impact of different masks on image information.

The output calculation of gated convolution is shown in Formula (3):

$$\begin{gathered} \text{Gating}_{y,x} = \sum\sum W_g \cdot I \\ \text{Feature}_{y,x} = \sum\sum W_f \cdot I \\ O_{y,x} = \emptyset\left(\text{ Feature}_{y,x}\right) \odot \sigma\left(\text{Gating}_{y,x}\right) \end{gathered} \tag{3}$$

In Formula (3), the output value of gated convolution is multiplied by the output value of two standard convolution layers element by element, where σ is a sigmoid function, so the gated output value is between 0 and 1. $\emptyset$ can be any active function (such as ReLU [34], ELU and LeakyReLU).W_g and W_f are two different convolution filters. Gated convolution learns the dynamic feature selection mechanism of each channel and each spatial position. In this way, different pixel information in different input images can be automatically learned, and the weights of different pixel masks are no longer 0 and 1, but values in the range of 0 to 1. The specific updating rules of partial convolution and gated convolution are shown in Figure 1. In Figure 1(left), the mask is updated based on the rules. If the area covered by the convolution kernel has any useful pixels during the convolution process, the location mark of the area after the convolution operation is useful, that is, the area mask is 1, as shown in Formula (2). Figure 1(right) shows the updating mechanism of gated convolution. Sigmoid is used as the soft gating switch to weight the output of the current convolution layer by 0-1 before it is input to the next convolution layer, so that the convolution kernel can better learn the useful pixels in the feature information.

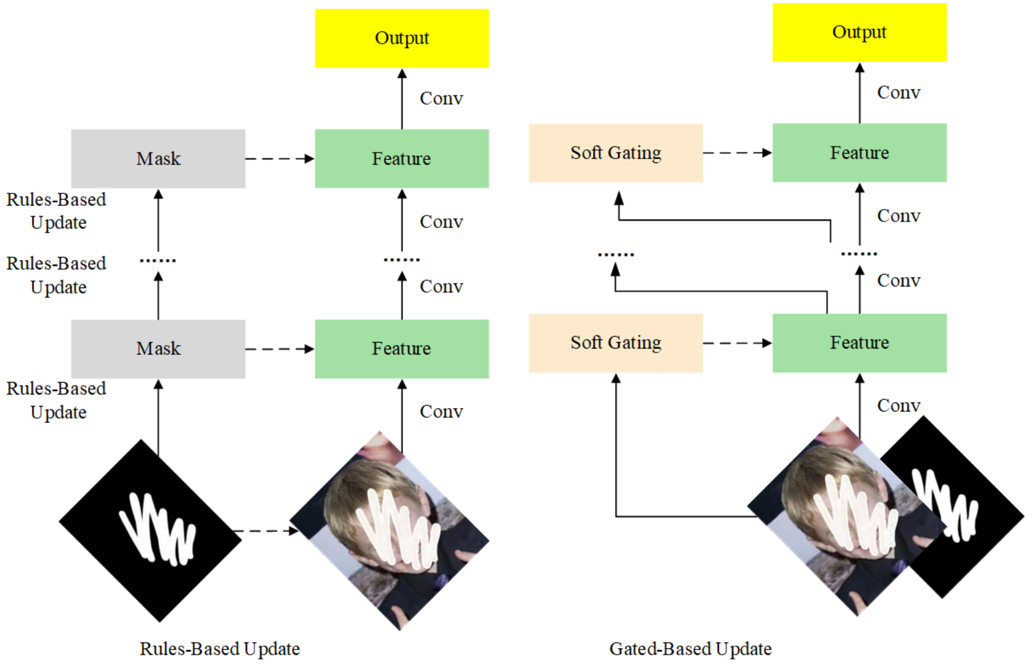

Figure 1: Update rules of partial convolution (left) and gated convolution (right).

2.2 Dense Connection

The direct connection method [15][16][17][18] is usually used between the layers of most image inpainting network convolution layers, which can simplify the network structure and facilitate the construction of the network model. However, with the development of artificial intelligence, to obtain more image feature information, the layers of the network are often deepened because of the deeper the layers of the network, the stronger the network learning ability. However, with the deepening of the number of network layers, the network parameters begin to increase by huge multiples. The increase in network parameters often leads to an increase in computer memory and network training time, and may also increase the risk of network overfitting or gradient disappearance.

To solve the above problems caused by the increase in parameters due to the increase in network layers, He Kaiming et al. [28] proposed a deep residual learning network (ResNet) for image inpainting and made great progress. The core of the ResNet model is to establish a jump connection between the previous layer and the next layer, which is conducive to the back-propagation of the gradient in the training process, to train a deeper network. The principle of jump connection is shown in Figure 2. Jump connection adds the features of the previous layer directly to the next layer, which makes the network easy to optimize and can improve the accuracy rate by increasing a considerable depth while reducing the loss of feature information in the transmission process. The advantage of this method is that it can reduce the possibility of network degradation and alleviate the problem of gradient disappearance caused by the increase in network depth in deep neural networks.

Huang, G. et al. [29] proposed a new network called the Densely Connected Convolutional Network (DenseNet). The dense connection structure used in this network can further reduce the risk of gradient disappearance during network training. A short connection is established between the upper layer and the lower layer. On the premise of ensuring the maximum information transmission between network layers, all layers are directly connected so that the input of the lower layer contains all the outputs of the previous layers. At the same time, the parameter K is set to make the output of each dense block a fixed value. The result is that although the network occupies a large amount of computer memory, the network parameters are greatly reduced compared with ResNet. DenseNet establishes a close connection between all the front and back layers. Another characteristic is that feature reuse is accomplished through feature connection on the channel. These features enable DenseNet to achieve better performance than ResNet with fewer parameters and computing costs.

In the convolution process, the output of layer L of the traditional convolution network is shown in Formula (4):

$$x_l = H_l(x_{l-1}) \tag{4}$$

For ResNet, the output of layer L is shown in Formula (5):

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{5}$$

For Densenet, the output of layer L is shown in Formula (6):

$$x_l = H_l\left([x_0, x_1, \ldots, x_{l-1}]\right) \tag{6}$$

where $H_l()$ represents a combined operation, which may include a series of batch normalization [36], ReLU, pooling and convolution operations.

In the convolution process, the output of the lower layer of the traditional convolution network is only related to the upper layer, and the output information of the lower layer connected by the residuals in Formula (5) is not only related to the feature information extracted by the convolution operation of the upper layer, but also includes the features of the upper layer. For dense connections as shown in Formula (6), the output of layer L in the dense network is related to the output of all previous layers, and the number of output channels of each dense layer is K, which can greatly reduce the number of parameters of the network model. Jump connection mechanism and dense connection mechanism are shown in Figure 2.

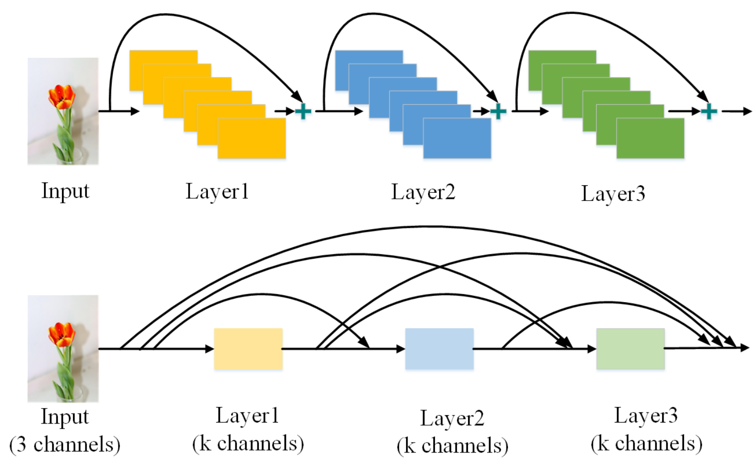

Figure 2: The top is the short connection mechanism of ResNet, where + represents the element-level addition operation; the bottom is the dense connection mechanism of DenseNet, where the arrow represents the channel-level connection operation.

The advantages of this special connection structure of DenseNet are as follows:

(1) It can make the transmission of features and gradients more effective, and the network is easier to train.

(2) It strengthens the transmission of features and makes more effective use of features.

(3) Due to the existence of the parameter K, the number of parameters is reduced to a certain extent.

3 The Proposed Method

Gated convolution can achieve a good inpainting effect in tests, but the convolution layers are directly connected between layers, which will increase the parameters of the model and increase the risk of overfitting or gradient disappearance in the training process. At the same time, the inpainting of damaged areas will cause artifacts and other problems. To solve these shortcomings or problems, based on the gated convolution network architecture, in this paper, dense connections and indirect connections are used to replace the direct connections between layers, and holistically-nested edge detector (HED) is introduced to extract the edges of damaged images for subsequent image inpainting. The modified model is called the dense gated convolutional network (DGCN). Section 3.1 mainly introduces the architecture of the modified generative image inpainting network model, Section3.2 shows the process of image inpainting, Section 3.3 introduces the application details of introducing the HED edge detector, and Section 3.4 introduces the structure of the core dense layer and dense blocks of the DGCN, as well as the details of the interlayer indirect connection.

3.1 Dense gated convolution network architecture

The complete model architecture of the dense gated convolutional network (DGCN) for image inpainting proposed in this paper is shown in Figure 3.

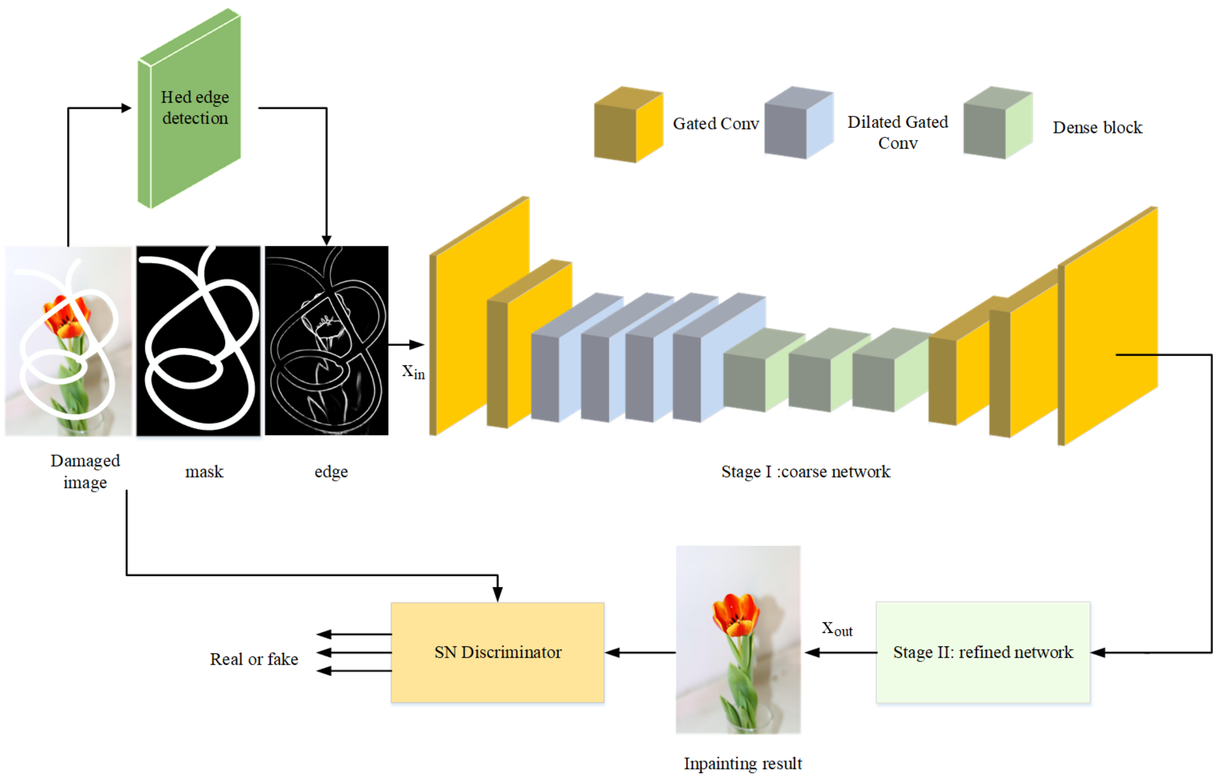

Figure 3: Inpainting Network Architecture of DGCN

DGCN is a multipart image inpainting network architecture based on GAN. The complete model consists of three parts: preprocessing module, generator and discriminator. The preprocessing module is mainly composed of an HED edge detector to extract the edge of the damaged image region. The generator is composed of a coarse generation network and a refined network. The coarse generation network is used to generate the general information of the damaged area, and the refined network focuses on rebuilding the detailed texture. The discriminator is mainly responsible for judging whether the image inpainted by the generator is consistent with the image in the training sets. Consistency is true, and inconsistency is false. The discriminator and generator play each other to improve the inpainting performance of the generator.

In the model architecture shown in Figure 3, this paper modifies the preprocessing stage and generation part of the model based on the original gated convolution network model. In the input stage of images, to reduce the artifacts or edge blurring caused by direct inpainting of damaged areas, the pretrained HED edge detector is introduced for edge extraction, and the extracted image edge information is used as prior knowledge for subsequent or downstream image inpainting. In stage I of the coarse network, in order to reduce the parameters of the network and the risk of overfitting or gradient disappearance in the process of network training, the dense connection is introduced, which not only reduces the number of parameters of the network but also realizes the reuse of features. Meanwhile, the sequence of the network layers is adjusted.

In Figure 3, stage I mainly consists of subsampled, dilated convolution, multiple dense blocks, and upsampling. The upsampling and subsampled are mainly used for the extraction and inpainting of feature information in images to facilitate the operation of the next layer. The dilated convolution with different dilated ratios is used for the extraction of features at different locations. The dilated ratios here are set to 2, 4, 8, and 16 to provide different receptive fields for the extraction of local and global information. Among them, dense blocks can extract more feature information of damaged images while greatly reducing the number of network parameters and feature reuse.

3.2 Image inpainting process

A schematic diagram of the inpainting process of damaged images is shown in Figure 4.

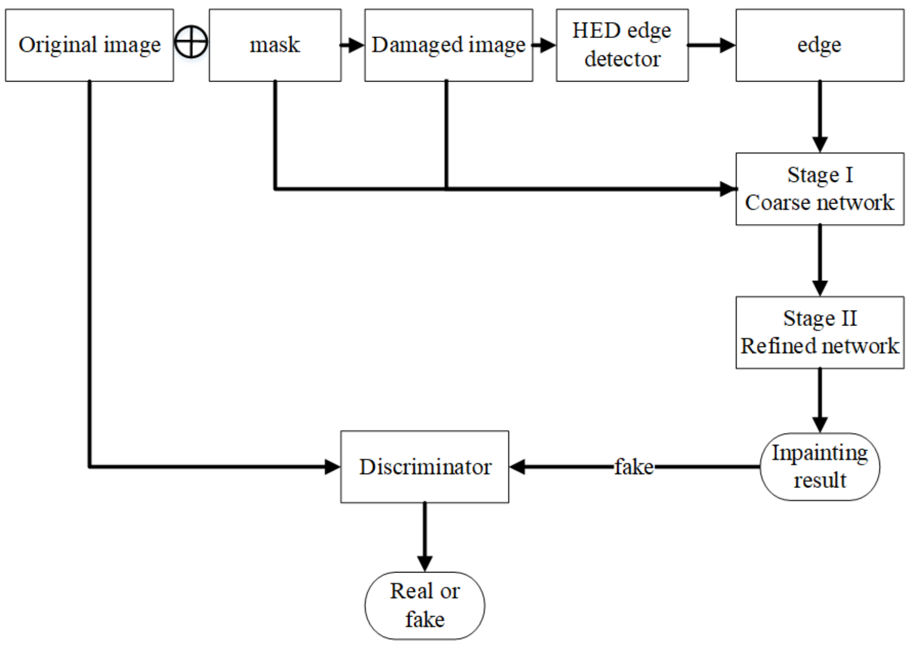

Figure 4: Schematic diagram of the damaged image inpainting process

As an input, the damaged image can be represented as: x_{Di}. First, the edge feature is extracted through the HED edge detector as shown in Formula (7):

$$x_{edge} = H\left(x_{Di}\right) \tag{7}$$

where $H()$ means that the HED edge detector extracts edge information x_{edge} from the damaged image x_{Di}. Then the original damaged image x_{Di}, mask x_{mask} and extracted edge information x_{edge}

are input into the generation network for image inpainting as shown in Formula (8):

$$x_{in} = x_{Di} + x_{mask} + x_{edge} \tag{8}$$

In the generation network part, the input image x_{in} is first coarsely generated in stage I to inpaint the damaged area over a large range. Then, the stage II refinement generation network is used to inpaint the texture and edge details of the damaged area. The inpainting process can be expressed as Formula (9).

$$x_{out} = G_{refine}\left[G_{coarse}\left(x_{in}\right)\right] \tag{9}$$

As shown in Formula (9), the input image x_{in} is first coarsely generated using $G_{coarse}()$ and then refined using $G_{refine}()$ to obtain the inpainting result x_{out}.

In the discriminator part, the image x_{out} inpainted by the generator is taken as the input, and the discriminator needs to judge whether the image is consistent with the image in the training sets. If the discriminator output is real, it indicates that x_{out} is consistent with the image in the original complete training sets; otherwise, it is false. The generator mainly generates images that the discriminator judges as real. The discriminator mainly determines whether the generated image is true. The generator and the discriminator play games with each other to improve the inpainting effect of the generator.

3.3 HED edge detector introduction process

In this paper, the HED edge detector is introduced in the preprocessing part to extract the edge structure information of the damaged image as the prior knowledge of the subsequent image inpainting. Edge detection is an algorithm used to extract the edge information of an image. It marks the places with obvious differences between adjacent pixel values of the image, and then connects them to form the edge information of the image. At present, there are many edge detection algorithms, among which Xie, S. [22] proposed HED as an end-to-end learning method, which can be easily nested in other network models as a whole.

In the aspect of implementation, first, HED is used to extract the edge of the original incomplete image, and the extracted edge information is input to the proposed model together with three RGB channels and a mask channel to inpaint the damaged area of the image. An example of the edge detection effect of the HED edge detector on damaged images is shown in Figure 5. Figure 3 shows that the HED edge detector is located in front of the image inpainting network in the network. In this way, the edge detector can be used to extract the edge of the input damaged or affected image as prior knowledge for subsequent image inpainting. Using the edge information of images to guide network inpainting can reduce artifacts or other visual problems in the inpainting process.

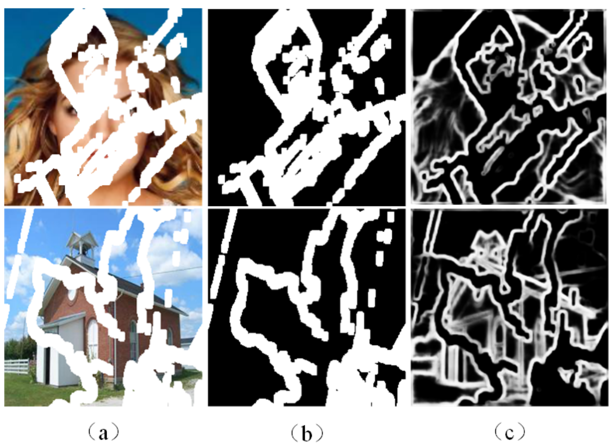

Figure 5: Example of the edge extraction effect of the HED edge detector on an image. (a) is the original incomplete image, (b) is the mask added to the real image, and (c) is the extracted edge information.

It can be seen from Figure 5 that HED mainly marks and connects areas with obvious changes in the image to form an edge structure, and this edge information is used as a supplementary for subsequent inpainting.

3.4 Dense blocks structure and interlayer indirect connection

This section mainly focuses on the composition and structure of dense blocks, the structure of dense layers, the transition layer between dense blocks and the interlayer indirect connection. The dense connection mode adopts a special feature transmission mode, which fuses the features of the previous layer with each subsequent layer and limits the number of output channels of each dense layer to the parameter K to achieve feature reuse while reducing the number of parameters. To improve the inpainting effect of the network and reduce the risk of gradient disappearance in the training process of the gated convolutional network, this paper improves the original 17 layer architecture of the gated convolution coarse inpainting network and replaces the previous direct connections between network layers with dense connections and indirect connections. To keep consistent with the number of layers of the original gated convolutional network, in this paper, only three dense blocks are introduced, and simple adjustments are made to other network layers. The positions of dense blocks are shown in the green blocks in Figure 3. The use of dense blocks can make it easier to modify the number of layers of the network. The dense block structure is shown in Figure 6.

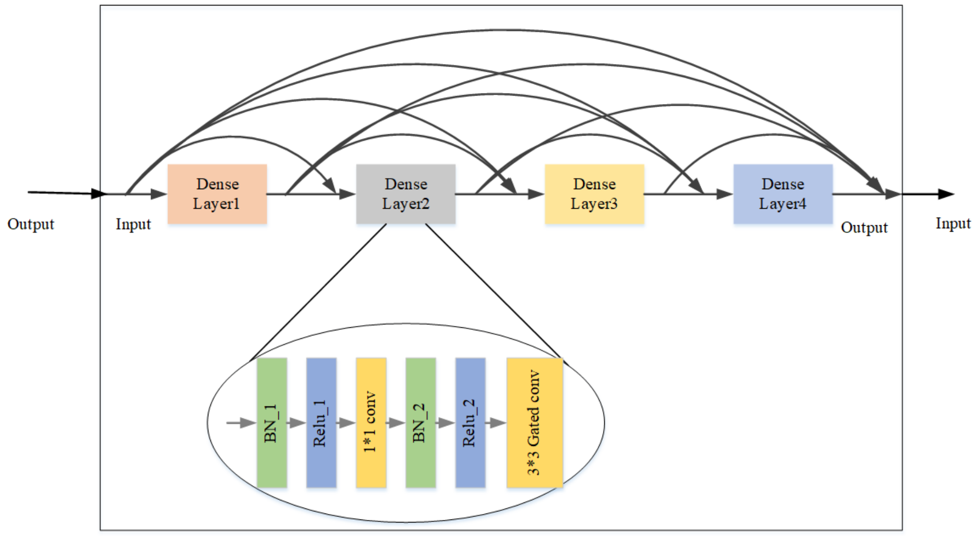

Figure 6: Structure of dense block

The structure of the dense block has the following characteristics:

1) Each dense block consists of several dense layers and each dense layer consists of some special layers.

2) Each dense layer integrates several special network layers such as a normalization layer, activation layer and convolution layer with convolution kernels of different sizes.

In Figure 6, the dense block is composed of several dense layers, and dense connections (or special full connections) are adopted among different dense layers; that is, the input of the previous layer is not only used for the input of the next layer but also for the input of all subsequent layers. This special connection mode enables the image features to be effectively utilized. Each dense layer consists of an ordinary convolution with a convolution kernel of size 1×1 and a gated convolution with a convolution kernel of size 3×3. The convolution of a 1×1 convolution kernel can reduce the number of parameters of the model, and the core of dense blocks uses gated convolution of a 3×3 convolution kernel to extract features. Before the convolution operation of each convolution layer, batch normalization and activation are carried out, and then 1×1 convolution and 3×3 convolution are carried out. The 1×1 convolution layer can be called the bottleneck layer in this special structure.

Each dense block reads the input of the previous layer and carries out corresponding operations.

Meanwhile, due to the existence of the parameter K, the output of each dense layer will be limited to K, which can greatly reduce the number of parameters of the network model. After the dense block structure, subsequent upsampling and refinement operations are carried out. Because the size of the feature information extracted from the dense blocks of different layers is different, the feature of the upper level cannot be reused directly in the dense connection of the next layer due to the difference in the size information, thus affecting the extraction of the dense block feature of the next layer. To solve this problem, as shown in Figure 7, a transition layer is inserted between adjacent dense blocks to change the number of channels for transmitting information from the previous layer to keep the size of the feature information of the preceding and following dense blocks consistent to realize the reuse of features. The transition layer mainly uses a 1×1 convolution kernel to carry out convolution operations to reduce the size of feature graphs. Meanwhile, a 1×1 convolution kernel can also reduce the number of network parameters and network complexity.

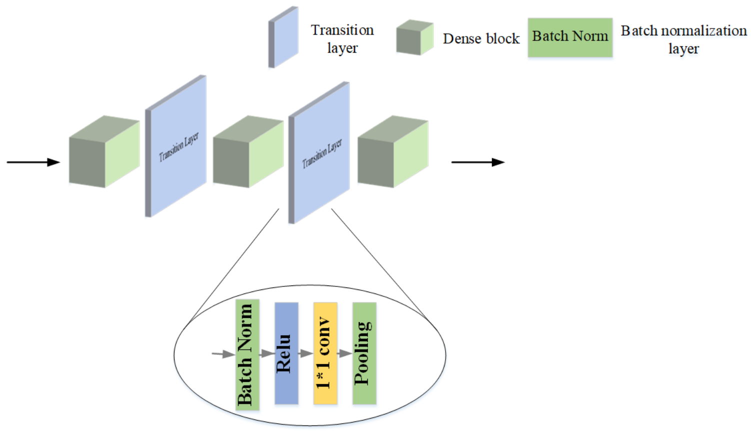

Figure 7: Connections between dense blocks

The direct connection between network layers will make the training unstable and increase the risk of gradient disappearance and overfitting. All convolution layers of the DGCN are arranged and combined in the order of convolution layer-batch normalization layer-activation layer. By means of batch normalization [36], the increasingly biased distribution that may appear in the training process is pulled back to the standardized distribution so that the input value of the activation function falls in the area where the activation function is sensitive to the input, thus making the gradient larger, accelerating the learning convergence speed and avoiding the problem of gradient disappearance. The connection mode between convolution layers is shown in Figure 8.

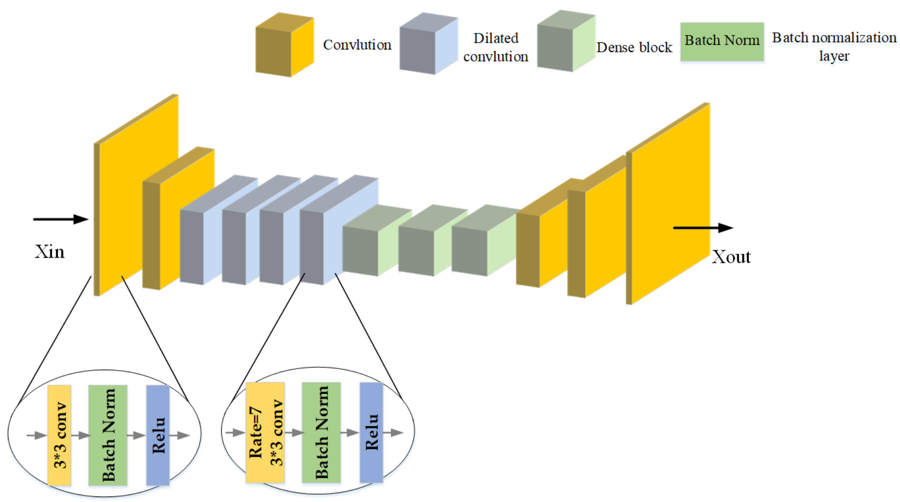

Figure 8: Interlayer indirect connection

4 Experiments

The DGCN model proposed in this paper is trained and tested on different datasets, and the implementation details are described in Section 4.1. In Section 4.2, the proposed model was compared with several classical methods, and PSNR and SSIM [35] were selected as objective evaluation indicators for the integrity of the inpainting effect. In terms of qualitative analysis, the visual effect is selected as the subjective evaluation index. The visual effect is often more direct than the objective evaluation index. The visual effect is displayed in Section 4.3, and ablation contrast is also analyzed in this section.

4.1 Implementation Details

The dense gated convolution network model proposed in this paper has been trained and tested on CelebA facial datasets and Places2 datasets. Different datasets have been trained and tested separately. During image preprocessing, different masks provided by partial convolution [17] are selected and processed by adding noise, rotating and flipping. These masks are added to different datasets to simulate real damaged images. The images in the datasets are also enhanced by rotating and flipping, which improves the generalization ability of the network model. For different datasets, 150000 of them are randomly selected for training, and 1000 of the remaining images are randomly selected for testing. And K with different parameters is trained.

4.2 Objective Evaluation

To facilitate the comparison with the original gated convolution model, three dense blocks are introduced into the modified network model. The number of dense layers is 1, 4, and 1. So the number of coarse generation network layers is 17, which is equal to the number of coarse generation network layers of the original gated convolution. Table 1 shows the comparison results of PSNR and SSIM of the same batch of damaged images inpainted by DGCN and several typical image inpainting models including Contextual Attention (CA) [31], Globally and Locally Consistent Image Completion (GLCIC) [32], EdgeConnect (EC) [33] and original gated convolution (GC) [18].

Table 1: Comparison of PSNR and SSIM objective indices of the same batch of images inpainted by the proposed model and other typical image restoration models

Methods	PSNR	SSIM
CA	20.03	0.793
GLCIC	23.49	0.865
GC	26.90	0.883
EC	27.95	0.920
DGCN(Ours)	**30.79**	**0.927**

It can be seen from Table 1 that compared with several other several classical models, the model proposed in this paper has significantly improved the image inpainting performance according to the objective indicesindexes of PSNR and SSIM. The objective indicesindexes of PSNR and SSIM of this model have increased by 10.16% and 1.09%, respectively, compared with the original GC model.

In DGCN, there is an important parameter K, which is related to the size of the network parameters and will inevitably affect the inpainting effect of the network. The smaller K is, the fewer the output characteristics of the dense layer, the fewer the number of output characteristic graphs of each dense block, and the fewer the network parameters. However, if K is too small, excessive compression will result in the loss of original image features and unreasonable use. Therefore, to select suitable and more effective network parameters, we choose different sizes of K for training and compare the impact of different K values on the network effect. The corresponding effects of different K on damaged image inpainting are shown in Table 2.

Table 2: Comparison of objective indices of different K for image inpainting effects

K	PSNR	SSIM
12	19.39	0.743
24	25.45	0.866
36	30.79	0.927
48	31.22	0.933

It can be seen from Table 2 that with the increase in K, the inpainting effect has a certain improvement in the objective indices of PSNR and SSIM. The reason is that with the increase in K, the number of features obtained in the process of feature information extraction increases, which has a certain role in promoting the subsequent image inpainting. However, with the increase in K, the increase in network parameters is also very large.

The number of network architecture parameters is an important standard to measure network performance, which is directly related to network computation. Table 3 shows the comparison of the number of generator parameters between the original gated convolution and DGCN under different parameters.

It can be seen from the comparison in Table 3 that with the increase in the parameter K, the number of parameters of the network model also increase. Compared with the inpainting effect of the image when K=36, for inpainting the same damaged image, the DGCN can improve 16.95% in terms of parameters. The DGCN model proposed in this paper achieves a better inpainting effect with fewer parameters.

Table 3: Comparison of generator parameters between GC and DGCN corresponding to different K values

Methods		Parameters
GC		16171254
DGCN(Ours)	K=12	10836007
	K=24	11834587
	K=36	13428693
	K=48	15628143
	K=60	18415996

By combining Table 2 and Table 3, it can be seen that the image inpainting effect will also improve with increasing K. When K is small (12 or 24), the number of DGCN parameters is low, but the image inpainting effect is poor. When K increases to 48, the number of DGCN parameters is slightly different from that of the original gated convolutional network. From K=36 to K=48, there was no significant increase in the objective indices of image inpainting. At the same time, when K=60, the number of DGCN parameters exceeds that of the original model, which is not considered in this paper.

Based on the above data, compared with other typical image inpainting models, the model proposed in this paper achieves a better inpainting effect in terms of the objective indices of PSNR and SSIM. Compared with the original gated convolutional network, the DGCN can reduce the number of network parameters and improve the inpainting effect of the network. However, when the value of K is increased, the number of parameters in the network will greatly increase, and the cost of training requirements will also increase. Specific conditions should be considered regarding how to select the relationship between the number of parameters in the network and the inpainting effect. For example, when K is selected as 36 in this paper, the number of parameters in the model and the inpainting effect are more compromised.

4.3 Subjective Evaluation

Visual effect is the subjective standard to evaluate the quality of image inpainting. It is more representative of the effect of image inpainting to evaluate the quality of image inpainting with people's subjective consciousness.

The comparison of inpainting effects between DGCN proposed in this paper and several other typical networks is shown in Figure 9.

Figure 9: Comparison of damaged image inpainting results of different algorithms, where (a) represents the damaged image, (b) represents the added mask, (c) represents the inpainting results of the original gated convolution, (d) represents the inpainting results of the EdgeConnect algorithm, (e) represents the inpainting results of the dense gated convolution network, and (f) represents the original images.

The comparison of inpainting effects between the DGCN proposed in this paper and several other typical networks is shown in Figure 9. Figure 9(a) represents the damaged image or the image that needs to be inpainted. Figure 9(b) is the mask that needs to be added to the original image and is also named the damaged area. Figure 9(c) is the inpainting result of the original gated convolution, and compared with other networks, the clarity and inpainting effect are very clear. However, since edge detectors are not introduced to extract edge information as prior knowledge, blur or artifacts will appear at the edge of the damaged area. Figure 9(d) shows the inpainting results of the EdgeConnect algorithm. Compared with Figure 9(c), the integrity of the inpainting is improved by considering the information of the edge. Figure 9(e) shows the inpainting results of the DGCN, which possesses the advantages of both and has a certain visual improvement. Figure 9(f) is the original complete image. Compared with Figure 9(c) and Figure 9(d), it can be seen that the inpainting effect of the image after introducing the HED edge detector is improved at the edge of the damaged area and does not produce blur and artifacts as heavy as that in Figure 9(c). However, the inpainting effect and clarity of the detail generated in Figure 9(d) is not as good as that in Figure 9(c). Compared with Figure 9(d) and Figure 9(e), clarity continues the advantages of gated convolution. Compared with Figure 9(c) and Figure 9(e), it can be seen that the DGCN proposed on the basis of the original gated convolution network improves to some extent in elevating details.

To verify the effect of edge information on image inpainting, an ablation experiment with or without the guidance of edge information is conducted in this part. The inpainting results are shown in Figure 10. Figure 10(a) is the original damaged image. Figure 10(b) shows the inpainting effect without the edge detector, that is, without edge information input as prior knowledge. Figure 10(c) shows the image inpainting results of the DGCN with the edge detector proposed in this paper. In the network model without edge guidance, in some texture details, the inpainting effect is not as good as that with the edge detector, which is specifically reflected in the vivid content of the image and the perfect consistency of the structure. For example, compared with Figure 10(b), the inpainting effect in 10(c) is more natural, and there are fewer artifacts on the damaged edges. The results of the ablation comparison experiment can confirm the feasibility and necessity of adding an edge detector to the architecture of the image inpainting model.

Compared with other typical image inpainting models, the DGCN image inpainting model proposed in this paper can achieve better inpainting results in the objective indicators of PSNR and SSIM. It can reduce the number of network parameters and improve the inpainting effect of the network model. The dense connection mode is used in the image inpainting model, and the special connection mode can strengthen the transmission of feature information between network layers of the model and maximize the use of feature information, thus improving the inpainting effect. In terms of subjective vision, the

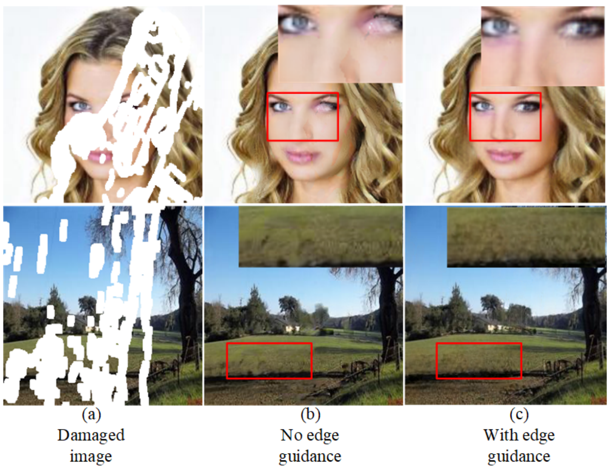

Figure 10: Comparison of image inpainting results with/without edge guidance

DGCN uses edge information as prior knowledge for image inpainting, which can reduce the blur at the edge of damaged images and improve the semantic accuracy and integrity of inpainting. The use of an indirect method between convolution layers can effectively avoid the absolute dependence of the later layers on the previous layers and improve the inpainting effect. DGCN can also be used to assist in the inpainting of cultural relics and calligraphy. To improve the visual effect of cultural relics and calligraphy, images can be collected, DGCN can be used to inpaint them, and the results of inpainting can be used to assist in actual inpainting.

5 Conclusions

In this paper, a novel generative image inpainting model with dense gated convolutional network (DGCN) is proposed. This model integrates edge detector, dense connections and indirect connections between layers. The information extracted by the edge detector can be better used by the generative network as prior knowledge. DGCN can optimize the model training process while reducing network parameters and directly connect all layers to enhance the use of feature information while ensuring the maximum transmission of feature information. Indirect connections between layers mainly rely on batch normalization layers to reduce the instability and gradient disappearance of network training. After training and testing on different datasets, DGCN has a better inpainting effect than other typical network architectures, and at the same time, it has also been optimized in terms of visual effects such as semantic and texture consistency. Other image inpainting models can be further optimized with the help of the DGCN model architecture to obtain a better training effect, and the parameters in the model can also be easily modified for better use.

However, the image inpainting network proposed in this paper still has some limitations, and there are still some challenges in inpainting large-scale damaged areas or damaged areas with more complicated structures or obvious color changes. In addition, DGCN only uses the edge detector to directly extract the edge information of the damaged image, and the damaged area will interfere with the extraction of the correct structural information from the image. Therefore, future work will take the color in the image information into account as part of the prior knowledge to inpaint the damaged image and improve the model to deal with the inpainting of more complex damaged images. At present, as long as the prior knowledge is accurate enough, such as if the structure is more complete, then the generative network can obtain a better inpainting effect on this basis. Therefore, how to better obtain prior knowledge or how to fuse prior knowledge and then apply it to the generative network is also a research direction.

Funding

This research is partially funded by the National Key Research and Development Program of China (2020YFB2103604) and the Ministry of Education Humanities and Social Science Research Project (22YJA630111).

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Efros, A.A.; Leung, T.K. (1999). Texture synthesis by non-parametric sampling, *proc.ieee int.conf.on computer vision*, 1999.

[2] Kwatra, V.; Essa, I.A.; Bobick, A.F.; Kwatra, N. (2005). Texture optimization for example-based synthesis, *ACM Transactions on Graphics (TOG)*, 24(3),795-802,2005.

[3] Ding D.; Ram S.; Rodríguez J.J. (2019). Image inpainting using nonlocal texture matching and nonlinear filtering, *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1705-1719, April 2019.

[4] Erkan, U.; Enginoglu, S.; Dang, N. (2019). An Iterative Image Inpainting Method Based on Similarity of Pixels Values, *IEEE The 6th International Conference on Electrical and Electronics Engineering ICEEE 2019*, 2019.

[5] Wang, H.; Jiang, L.; Liang, R.; Li, X.X. (2017). Exemplar-based image inpainting using structure consistent patch matching, *Neurocomputing*, 269(dec.20), 401-410.

[6] Song, Y.; Yang, C.; Lin, Z.; Liu, X.; Huang, Q.; Li, H. (2017). Contextual-based Image Inpainting: Infer, Match, and Translate, *Proceedings of the European Conference on Computer Vision*, pages 3-19, 2017.

[7] Cheng, Y.; Wan, Y.; Sima, Y.; Zhang, Y.; Hu, S.; Wu, S. (2022). Text Detection of Transformer Based on Deep Learning Algorithm, *Tehnički vjesnik*, 29 (3), 861-866.

[8] Lv, B.; Gao, X.; Feng, S.; Yuan, J. (2022). Deep Learning Detection Algorithm for Surface Defects of Automobile Door Seals, *Tehnički vjesnik*, 29 (5), 1499-1506.

[9] Ma, X.; Li, Z.; Zhang, L. (2022). An Improved ResNet-50 for Garbage Image Classification, *Tehnički vjesnik*, 29 (5), 1552-1559.

[10] Lee, J.M.; Jung, I.H.; Hwang, K. (2022). Classification of beef by using artificial intelligence, *Journal of Logistics, Informatics and Service Science*, Vol. 9, No. 3, pp. 271-283.

[11] Wang, J.; Chen, Y.; Qiu, S.; Cui, Q. (2021). Cuckoo Search Optimized Integrated Framework Based on Feature Clustering and Deep Learning for Daily Stock Price Forecasting, *Economic Computation And Economic Cybernetics Studies And Research*, Vol. 55(3), pg. 55-70, DOI: 10.24818/18423264/55.3.21.04.

[12] Lu, W.; Li, J.; Wang, J.; Wu, S. (2022). A Novel Model for Stock Closing Price Prediction Using CNN-Attention-GRU-Attention, *Economic Computation And Economic Cybernetics Studies And Research*, Vol. 56(3), pg. 251-264, DOI: 10.24818/18423264/56.3.22.16.

[13] Nath, B., Kumbhar, C., Khoa, B.T. (2022). A study on approaches to neural machine translation, *Journal of Logistics, Informatics and Service Science*, Vol. 9, No. 3, pp. 271-283.

[14] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; et al. (2014). Generative Adversarial Nets, *Neural Information Processing Systems*, MIT Press,2014.

[15] Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting, *IEEE*, 10.1109/CVPR.2016.278, 2016.

[16] Wang, Y.; Tao, X.; Qi, X.; Shen, X.; Jia, J. (2018). Image inpainting via generative multi-column convolutional neural networks, , 2018.

[17] Guilin Liu; Fitsum A. Reda; Kevin J. Shih; Ting-Chun Wang; et al (2018). Image inpainting for irregular holes using partial convolutions, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85-100,2018.

[18] Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T. (2018). Free-Form Image Inpainting With Gated Convolution, *2019 IEEE/CVF International Conference on Computer Vision*, pp. 4470-4479, 2018.

[19] He, J.; Zhang, S.; Yang, M.; Shan, Y.; Huang, T. (2019). BDCN: Bi-Directional Cascade Network for Perceptual Edge Detection, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3823-3832, 2019.

[20] Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), 679-698,1986.

[21] Liu, Y.; Cheng, M.M.; Hu, X.; Wang, K.; Bai, X. (2016). Richer convolutional features for edge detection, *IEEE Computer Society*, 2016.

[22] Xie, S.; and Z. Tu. (2015). Holistically-Nested Edge Detection, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1395-1403, 2015.

[23] Poma, X.S.; Riba, E.; Sappa, A.D. (2019). Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection, , 10.48550/arXiv.1909.01955.

[24] Arimoto M.; Hara J.; Watanabe H. (2021). An Image Inpainting Method Considering Edge Connectivity of Defects, *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*, pp. 101-102, 2021.

[25] Yuantao Chen; Runlong Xia; Ke Zou; Kai Yang (2023). E2I: Generative Inpainting From Edge to Image, *Journal of Visual Communication and Image Representation*, Volume 91,2023,103776, ISSN 1047-3203.

[26] Xu, S.; Liu,D.; Xiong Z. (2021). FFTI: Image inpainting algorithm via features fusion and two-steps inpainting, *IEEE Transactions on Circuits and Systems for Video Technology*, 31(),1308-1322,2021.

[27] Haiyan Li; Yingqing Song; Haijiang Li; Zhengyu Wang (2023). Semantic prior-driven fused contextual transformation network for image inpainting, *Journal of Visual Communication and Image Representation*, Volume 91,2023,103777, ISSN 1047-3203.

[28] He, K.; Zhang, X.; Ren, S.; Sun, J. (2016). Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778,2016.

[29] Huang, G.; Liu, Z.; Laurens, V.; Weinberger, K.Q. (2016). Densely connected convolutional networks, *IEEE Computer Society*, 2016.

[30] Pleiss, G.; Chen, D.; Huang, G.; Li, T.; Laurens, V.; Weinberger, K.Q. (2017). Memory-efficient implementation of densenets, , 2017.

[31] Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. (2018). Generative image inpainting with contextual attention, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5505-5514, 2018.

[32] Iizuka, S.; Simo-Serra, E.; Ishikawa, H. (2017). Globally and locally consistent image completion, *ACM Transactions on Graphics (TOG)*, 36(4CD), 107.1-107.14,2017.

[33] Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.Z.; Ebrahimi, M. (2019). Edgeconnect: generative image inpainting with adversarial edge learning, , 2019.

[34] Nair, V.; Hinton, G.E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair, *International Conference on International Conference on Machine Learning*, Omnipress,2010.

[35] Zhou, W.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. (2004). Image quality assessment: from error visibility to structural similarity, *IEEE Trans Image Process*, 13(4),2004.

[36] Ioffe, S.; Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift, *JMLR*, org,2015.

Cite this paper as:

Ma, X.; Deng, Y.; Zhang L.; Li Z. (2023). A Novel Generative Image Inpainting Model with Dense Gated Convolutional Network,*International Journal of Computers Communications & Control*, 18(2), 5088, 2023.

https://doi.org/10.15837/ijccc.2023.2.5088