electronics

MDPI

Article

A Generate Adversarial Network with Structural Branch Assistance for Image Inpainting

Jin Wang, Dongli Jia * and Heng Zhang

School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China
* Correspondence: jiadongli@hebeu.edu.cn

Abstract: In digital image inpainting tasks, existing deep-learning-based image inpainting methods have achieved remarkable staged results by introducing structural prior information into the network. However, the corresponding relationship between texture and structure is not fully considered, and the inconsistency between texture and structure appears in the results of the current method. In this paper, we propose a dual-branch network with structural branch assistance, which decouples the inpainting of low-frequency and high-frequency information utilizing parallel branches. The feature fusion (FF) module is introduced to integrate the feature information from the two branches, which effectively ensures the consistency of structure and texture in the image. The feature attention (FA) module is introduced to extract long-distance feature information, which enhances the consistency between the local features of the image and the overall image and gives the image a more detailed texture. Experiments on the Paris StreetView and CelebA-HQ datasets prove the effectiveness and superiority of our method.

Keywords: structural prior information; structure auxiliary branch; generate adversarial network; image inpainting

Citation: Wang, J.; Jia, D.; Zhang, H. A Generate Adversarial Network with Structural Branch Assistance for Image Inpainting. *Electronics* **2023**, *12*, 2108. https://doi.org/10.3390/electronics12092108

Academic Editors: Jyh-Cheng Chen and Donghyeon Cho

Received: 6 March 2023
Revised: 25 April 2023
Accepted: 29 April 2023
Published: 5 May 2023

1. Introduction

Digital image inpainting technology uses a computer to automatically inpaint the content of an image defect area using the content of a known area in the image and, at the same time, ensures the consistency of the overall structure of the image. Using this technology to inpaint defective images can not only avoid the influence of human subjective ideas about the content, but also allows the results to adequately meet the perceived needs of human vision. The rise of deep learning technology has brought this area to the attention of contemporary researchers in recent times. Digital inpainting technology has become an important aspect of research in the field of digital image processing and computer vision.

Early traditional methods [1–5] mainly use mathematical knowledge and physical knowledge to inpaint images. In the case of simple digital image inpainting tasks, such as small image defect areas, simple textures, and structures, good results can be achieved with these methods. However, when dealing with complex tasks, such as large-area defects, traditional methods cannot perceive and understand the high-level semantics of images, and the results often have inconsistent structures and lack reasonable and clear semantics, resulting in unsatisfactory effects.

In contrast to traditional methods, convolutional neural networks [6,7] and generative adversarial networks [8] are applied to digital image inpainting tasks via deep-learning-based methods [9–30]. Using a convolution operation to extract the high-level semantic information in the image while inpainting the textural details of the defect area, it can be ensured that the image has a consistent structure and reasonable semantics. This makes up for the defects in traditional methods, achieves results that are more satisfactory for the needs of human visual perception, and improves the quality of the image significantly. As the research deepens, it has been found that ordinary convolution cannot distinguish

the defect area of the image from the known area. All the pixels in the current window are regarded as effective pixels as the convolution operation is performed, which leads to a series of problems, such as edge response. The problems caused by the indiscriminate operation of ordinary convolution are more serious in the task of inpainting irregular defective digital images. As a result, two updated versions of common convolutions are proposed [12,13]. The two updated versions of the convolution operation are only aimed at effective pixels and achieved good performance in the task of inpainting irregular defective digital images. In addition, to solve the problem wherein the traditional convolution methods cannot extract features from distant regions, researchers have introduced the attention mechanism [14–18] into the network. Using the attention mechanism, feature blocks can be borrowed from known regions to fill in defective regions.

In addition to the above methods, researchers have combined structural information and proposed some two-stage methods [19–23]. These methods inpaint the structural information of the defect area in the first stage and then use the structural information to guide the synthesis of pixels in the defect area in the second stage. The successive emergence of these studies has proved that the structural information of semantic segmentation maps [19], edge maps [20,21], foreground contours [22], and smooth images [23] play an important role in guiding the generation of better images. However, most of these algorithms use a two-stage network architecture, first predicting structural information, and then inpainting the defect area. Obtaining reasonable structural information from already-defective images is a very challenging task in itself, and, thus, there is a problem in that the results will deteriorate due to structural prediction errors. To solve the problems of the above methods, researchers have improved the performance by simultaneously reconstructing structure and texture features in a single-generation network [24–26]. Recently, some researchers tried to introduce a transformer network into the digital image inpainting task [27,28], and the results better meet the needs of human visual perception.

In this paper, we exploit structural information, such as image edge maps, to propose a dual-branch network with independent structural branches to split image inpainting into two simultaneous subtasks. That is, the structure branch focuses on the structural information of the defect area, and the other texture branch focuses on the synthesis of image texture. In this way, the two parallel and independent branches decouple the inpainting of low-frequency and high-frequency information in the area to be inpainted. Correspondingly, we introduce a feature fusion (FF) module with an information selection function to integrate the structure feature map and texture feature map, so that the feature information sets of the two branches complement each other and enhance the consistency of the image structure and texture. At the same time, we also introduce a feature attention (FA) module to extract features from known regions far away from the defected region, giving the image more detailed textures. Furthermore, we introduce two Markov discriminators to evaluate the performance of the generator and force the generator to produce more realistic images.

We conduct experiments and evaluate using the Paris StreetView and CelebA-HQ datasets. Both the qualitative results display and quantitative numerical comparisons show that our method outperforms existing methods.

The main innovations and contributions of this paper are as follows:

We propose a novel network assisted by structural branches, where two branches in the network focus on image structure and synthesizing image texture, respectively. In this parallel and independent manner, the inpainting of low-frequency and high-frequency information is decoupled.

We introduce a feature fusion (FF) module to integrate the structural information and texture information from the two branches and perform information selection. We allow these sets of information to complement each other so that the image structure and texture are more consistent. At the same time, we also introduce a feature attention (FA) module to extract features from distant regions to generate more detailed textures.

Through experiments on different types of datasets and comparisons with multiple benchmark methods, our method shows strong and advanced performance in both qualitative and quantitative aspects.

2. Related Work

Traditional methods were mainly used in early digital image inpainting tasks. According to previous research, these are mainly divided into two categories: one is the method based on partial differential equations [1–3], and the other is the method based on sample texture. The method based on partial differential equations establishes a geometric model according to the correlation of pixels in the image to inpaint the image defect, and the method based on sample texture fills the defect area by borrowing the texture image block of the known area. The method based on partial differential equations can achieve better results when the proportion of the image defect area is small, and the texture around the defect area is relatively simple. If the proportion of the image defect area is relatively large or the texture around the defect area is relatively complex, the effect will also be poor, and the result will be very blurred. Compared with methods based on partial differential equations, methods based on sample textures can achieve better results in large-area defect image inpainting tasks and can even inpaint texture details in defect areas to a certain extent. However, the images still lack reasonable semantics and cannot meet the needs of human visual perception.

Deep-learning-based methods [9–30] currently occupy a dominant position in the field of image inpainting due to their powerful data-fitting capabilities. This kind of method can effectively extract the high-level semantic information in the image, and the results can meet the perceived needs of human vision to a great extent under the premise of ensuring the content is reasonable. By introducing structural information, some two-stage methods have been proposed. Song et al. [19] proposed the SPG-Net network for some methods that do not make full use of semantic segmentation information to constrain the image structure. Nazeri et al. [20] proposed the EdgeConnect network by explicitly introducing the prior information on the edge structure. Similarly, Shun et al. [21] also explicitly introduced the structural information of the edge map in the E2I network. Xiong et al. [22] did not use a semantic segmentation map and edge map as structural prior information but explicitly introduced foreground contour information into the network. Ren et al. [23] pointed out that smooth images have better global structures, thus explicitly introducing smooth images into the network as structural prior information. These two-stage methods generate images with a more reasonable structure by using structural prior information, but the network in the latter stage is easily affected by the network in the previous stage. If the reasoning structure is unreasonable, the effect will become unsatisfactory. In view of the problems existing in the above two-stage methods, some researchers have begun to try to reconstruct structural features and texture features at the same time. Li et al. [24] proposed a progressive network for edge structure, the PRVS network, which can gradually inpaint the edge structure and related edge features through the designed VSR layer. Liu et al. [25] proposed an inter-encoder-decoder network based on the joint inpainting of structure and texture. Through multi-scale filling of structural features and texture features, the network can inpaint the structure and texture of images at the feature level. Jie et al. [26] designed a structure-embedding layer that was used to gradually embed the structural feature information into the decoding features of the decoder as prior information. Recently, some researchers have tried to use a transformer network to generate multiple results with reasonable content for each defective image. Wan et al. [27] introduced a transformer network to inpaint the overall structure of the damaged image and then used a convolutional network to further refine the local texture, thereby generating diverse images with fine textures. Liu et al. introduced a PUT network [28] based on a transformer network, which further generated high-quality diverse images by reducing the information loss in the transformer network.

3. Method

As shown in Figure 1, the main network of this method is a generative adversarial network [8], and the generator consists of two convolutional autoencoder networks to form a dual-branch network. Among them, the structure branch focuses on the structural information of the defect area, and the texture branch focuses on the synthesis of image texture. The quality of the image is evaluated by a Markov structure discriminator and a Markov texture discriminator. In this section, we will elaborate on the generator of the network, the discriminator, and the loss function of the network training in Sections 3.1–3.3, respectively.

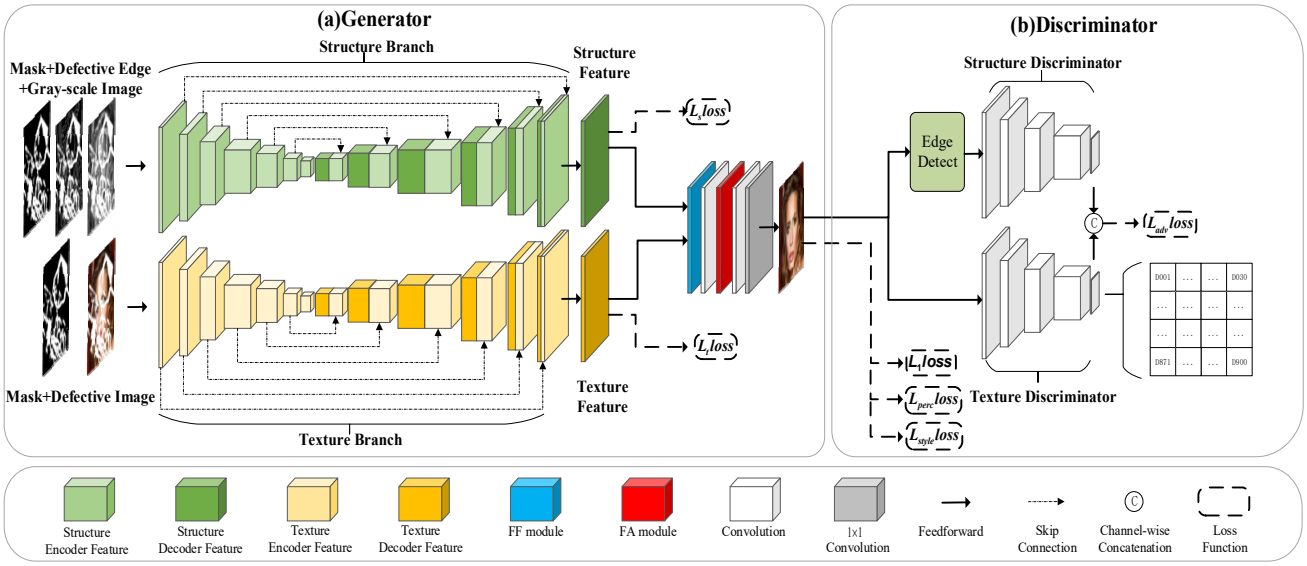


Figure 1. Overall architecture of the proposed network.

3.1. Generator

The generator body consists of two U-net networks [31] as shown in Figure 1a. In the encoding stage, the encoder uses convolution to encode the defect image and its corresponding edge map step by step and then projects them into the latent space after step-by-step compression. The structure branch mainly focuses on structural features, and the texture branch mainly focuses on textural features. In the decoder, to ensure the quality of the generated image, some low-level image features such as texture need to be generated after several decodings. However, because the network layer is too deep, the low-level image features extracted by the shallow layer are likely to have been lost. This is because in the process of network propagation, as the network becomes deeper and deeper, the receptive field of the corresponding feature map will gradually become larger, resulting in less detailed information being retained. By adding skip connections, the convolutional features in the encoding process are copied and passed to the corresponding transposed convolutional layer in the decoding process, which supplements low-level image features such as a texture for the decoder.

The dual-branch network splits image inpainting into two simultaneous subtasks, which decouple the inpainting of low-frequency and high-frequency information in a parallel and independent manner. The structure branch focuses on restoring the structural information of the defect area and can explicitly utilize the image edge map as a structural prior to ensure that the inpainted image has a more reasonable structure. The texture branch focuses on the synthesis of image textures, using skip connections to ensure that the generated images have texture details.

In the two branches of the generator, we use partial convolutional layers instead of ordinary convolutional layers to improve the performance of the network in irregular defect inpainting tasks. In addition, the decoded feature maps output by the two branches

are input to the feature fusion (FF) module with information selection and integration functions for feature fusion and then input to the feature attention module (FA) to obtain the final results.

3.1.1. Feature Fusion (FF) Module

This module is designed to integrate the edge map and texture feature map decoded by the dual-branch network so that the feature information decoded by two parallel and independent branches can be integrated. This module is shown in Figure 2.

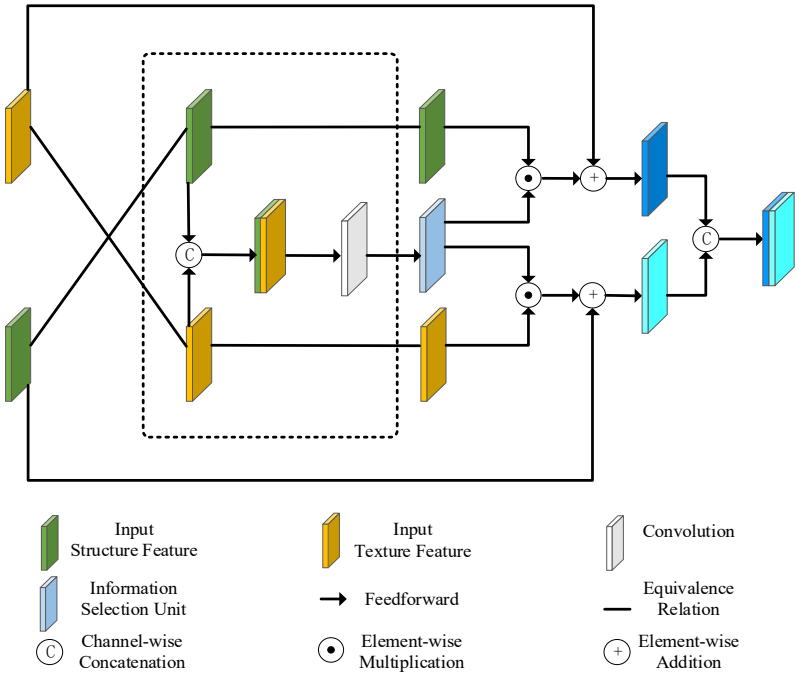


Figure 2. Illustration of Feature Fusion (FF) Module.

Here, we denote the feature maps decoded by the structure branch and the texture branch by F_s and F_t, respectively. First of all, we first construct an information integration unit I_{U}, and the obtained process of I_{U} is shown in Formula (1):

$$I_{\mathrm{U}} = \sigma_{\mathrm{s}}(\mathrm{Conv}(\mathrm{C}(F_s \cdot F_t))), \tag{1}$$

where $\mathrm{C}(F_s \cdot F_t)$ represents the concatenation of F_s and F_t in the channel dimension, $\mathrm{Conv}(\cdot)$ represents the convolution operation with a convolution kernel size of 3, and $\sigma_{\mathrm{s}}(\cdot)$ represents the Sigmoid activation function.

Using I_{U}, we fuse F_s and F_t to obtain the texture-based and structure-assisted feature F_{ts}, and the process of obtaining F_{ts} is shown in Formula (2):

$$F_{ts} = \alpha_1(I_{\mathrm{U}} \odot F_s) \oplus F_t, \tag{2}$$

where α_1 is a trainable parameter whose initial value is 0. $\odot$ and $\oplus$ represent element-wise multiplication and element-wise addition operations, respectively.

Correspondingly, the structure-based and texture-assisted feature F_{st} is obtained as shown in Formula (3):

$$F_{st} = \alpha_2(I_{\mathrm{U}} \odot F_t) \oplus F_s, \tag{3}$$

where α_2 is a trainable parameter, whose initial value is 0.

Finally, we obtain the fusion feature F_F by concatenating F_{ts} and F_{st} in the channel dimension. The process of obtaining F_F is shown in Formula (4):

$$F_F = \mathrm{C}(F_{ts} \cdot F_{st}), \tag{4}$$

where $\mathrm{C}(F_s \cdot F_t)$ represents the concatenation of F_{ts} and F_{st} in the channel dimension.

3.1.2. Feature Attention Module (FA)

Traditional convolution methods cannot extract features from distant regions. We have introduced an attention mechanism into the network, and the attention mechanism can be used to borrow feature blocks from known regions to fill in missing regions, effectively solving this problem. By introducing an attention mechanism, we designed this module to effectively enhance the correlation between image parts, thereby generating more detailed textures. This module is shown in Figure 3.

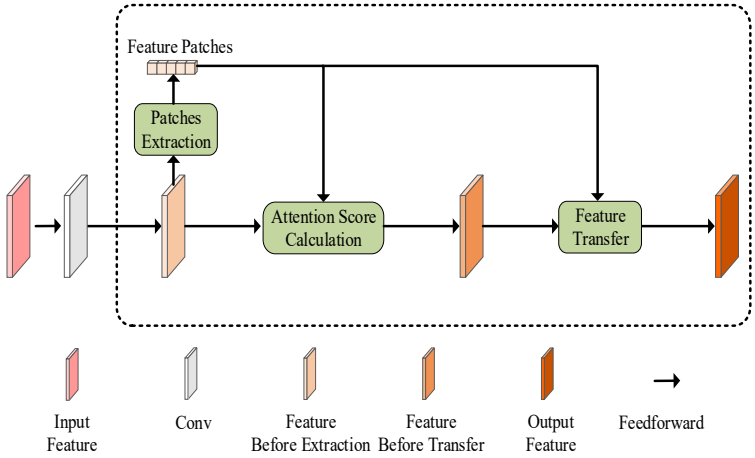


Figure 3. Illustration of Feature Attention Module (FA).

This module mainly performs two steps of attention score calculation and feature transfer on the feature map. To compute the attention score, here, we denote an unspecified input feature map by F. We need to first divide F into small block feature maps of 3×3 size and calculate the cosine similarity between the small block feature maps. The cosine similarity calculation process is shown in Formula (5):

$$CS^{i,k} = \left\langle \frac{f_i}{\|f_i\|}, \frac{f_k}{\|f_k\|} \right\rangle, \tag{5}$$

where f_i and f_k are the i-th and k-th small block feature maps, respectively.

Then we use the softmax function to obtain the final attention score of each small block feature map. This calculation process is shown in Formula (6):

$$AS^{i,k} = \frac{\exp\left(\beta CS^{i,k}\right)}{\sum_{m=1}^{N} \exp\left(\beta CS^{i,m}\right)}, \tag{6}$$

where β is a trainable parameter.

Finally, the feature transfer is completed according to the attention score, and the transfer feature map is obtained. This process is shown in Formula (7):

$$f_i' = \sum_{k=1}^{N} f_k \cdot AS^{i,k}, \tag{7}$$

where f_i' is the i-th small block feature map of the transfer feature map.

3.2. Discriminator

The GLCIC network [10] uses local and global double discriminators. The local discriminator acts on a local area of the image, and the global discriminator acts on the entire image. The two discriminators work at the same time to ensure the consistency of local and global information and improve the quality of the image. Later, to generate better image texture details, a Markov discriminator [32] is introduced into the digital image inpainting task; one of the discriminators used in the PGGAN network [11] is a Markov discriminator. Inspired by the above network, for the dual-branch structure of the generator network, two Markov discriminators are used to evaluate the performance of the generator and to force it to generate more realistic images. The network discriminator is shown in Figure 1b. One of the two Markov discriminators of our network is used to evaluate the image structure information to guide the structure reconstruction and is called the structure discriminator. The other evaluates the whole image and focuses on evaluating the detailed texture, which is called the texture discriminator.

Due to the sparsity of edge image structure information, in addition to the edge image obtained by passing the original image or the inpainted image through a residual block, the grayscale image is also input to the structure discriminator as an addition [20]. According to the literature [32], we use five convolution layers to form the structure discriminator. The size of the convolution kernel of the five convolutional layers is four and the filling is one. The stride of the first three layers is two, and the stride of the last two layers is one. We apply the Leaky ReLU activation function to the first four layers; the slope is set to 0.2, and the Sigmoid activation function is used in the last layer to make the predicted value fall in the range of 0–1. The input of the texture discriminator is the whole original image or the whole inpainted image, which also consists of five convolutional layers, and the convolution setting is the same as that of the structure discriminator. The final output of the Markov discriminator is not a scalar, and it maps the input image to a matrix D of size $N \times N$. D_i corresponds to the discrimination output of the discriminator for a small block of the input image, and its value represents the probability that the input image block is a true sample block. In our network, the size of N is set to 30. Finally, the output block matrices of the two discriminators are concatenated in the channel dimension, and the generative adversarial loss is calculated from this. Furthermore, we apply spectral normalization [33] to the discriminator network, which effectively constrains the Lipschitz constant of the discriminator network to one by shrinking the respective maximum singular values of the weight matrices, further stabilizing the training of the network.

3.3. Loss Function

According to literature [15,20,25,26], the loss function of the network consists of five parts, namely reconstruction loss, confrontation loss, perception loss, style loss, and branch loss. Here, we use G_{our} to denote a two-branch generator and D_{our} denote two Markov discriminators. I_w denotes the undamaged image, E_w denotes the edge map of the undamaged image, and G_w denotes the grayscale image corresponding to the undamaged image. M represents a binary mask. $I_d = I_w \odot M$ represents the input image of the defect, $E_d = E_w \odot M$ represents the input edge image of the defect, and $G_d = G_w \odot M$ represents the input grayscale image of the defect. The inpainted image I_r and the inpainted edge map E_r are obtained from the output of G_{our}, and this process is represented by (8):

$$I_r, E_r = G_{our}(I_d, E_d, G_d, M). \tag{8}$$

3.3.1. Reconstruction Loss

The reconstruction loss is mainly used to measure the difference between the generated image and the real image. The reconstruction loss can be mainly divided into L_1 reconstruction loss and L_2 reconstruction loss. According to actual needs, the reconstruction loss can be used for the entire image, or it can be used alone for a certain area or jointly

weighted for different areas. Here, we choose L_1 reconstruction loss and apply it to the whole image, and its calculation process is shown in Equation (9):

$$L_1 = \mathbb{E}[\|I_r - I_d\|_1]. \tag{9}$$

3.3.2. Generative Adversarial Loss

The generative adversarial loss is mainly used to ensure the visual authenticity of the image, and its calculation process is shown in Formula (10):

$$L_{adv} = \min_{G_{our}} \max_{D_{our}} \mathbb{E}_{I_w \sim P_w, E_w \sim P_{we}}[\log D_{our}(I_w, E_w)] + \mathbb{E}_{I_r \sim P_r, E_r \sim P_{re}}[\log(1 - D_{our}(I_r, E_r))], \tag{10}$$

where P_w represents the distribution of the undamaged image, P_{we} represents the distribution of the edge map of the undamaged image, P_r represents the fitted distribution of the inpainted image, and P_{re} represents the fitted distribution of the inpainted edge map.

3.3.3. Perceptual Loss

Perceptual loss [34] is mainly used to measure the difference between generated images and real image features, which can force the network to capture high-level semantics and simulate the human perception of image vision. The calculation process is shown in Formula (11):

$$L_{prec} = \mathbb{E}\left[\sum_i \|\phi_i(I_r) - \phi_i(I_w)\|_1\right], \tag{11}$$

where ϕ_i represents the feature map of the i-th layer extracted by the VGG-16 network pre-trained on the ImageNet dataset.

3.3.4. Style Loss

The style loss is mainly used to eliminate the checkerboard artifacts caused by the transposed convolution to improve the quality of the image. Its calculation process is shown in Formula (12):

$$L_{style} = \mathbb{E}\left[\sum_i \|G_j^{\phi}(I_r) - G_j^{\phi}(I_w)\|_1\right], \tag{12}$$

where G_j^{ϕ} represents the $C_j \times C_j$ Gram matrix constructed from the selected feature maps, which are the same as those used in the perceptual loss.

3.3.5. Branch Loss

To allow the structure branch and the texture branch to generate more reasonable structures and finer textures, respectively, we introduce branch losses in the two branches as shown in Formula (13):

$$L_{branch} = L_s + L_t = \mathrm{B}(E_w, P_s(F_s)) + L_1(I_w, P_t(F_t)), \tag{13}$$

where L_s represents the branch loss added to the structural branch. L_t represents the branch loss added to the texture branch. B represents BECloss and P_s represents the mapping function that maps F_s to the edge map. P_t represents a mapping function that maps F_t to a three-channel color image.

3.3.6. Total Loss

The total loss is shown in Formula (14):

$$L_{total} = \lambda_1 L_1 + \lambda_2 L_{adv} + \lambda_3 L_{prec} + \lambda_4 L_{style} + \lambda_5 L_{branch}, \tag{14}$$

where λ_1, λ_2, λ_3, λ_4, and λ_5 represent the hyperparameters used to balance different loss functions. Here, according to literature [20,26], we set $\lambda_1 = 10$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$, and $\lambda_4 = 250$. According to literature [15,25], we set $\lambda_5 = 1$.

4. Experiment

4.1. Experimental Setup

We conducted experiments on the Paris StreetView [35] dataset and the CelebA-HQ [36] dataset, and conducted subjective evaluation and objective evaluation of the results using existing evaluation indicators. In addition, we conducted three ablation experiments to demonstrate the effectiveness of our designed network and modules.

Our experiments used the Paris StreetView dataset and the CelebA-HQ dataset as well as the irregular mask dataset proposed in [12]. The irregular mask dataset includes masks with six defect ratios, and the defect ratios are (0.01, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5], (0.5, 0.6]. We use torchvision.transforms.Resize to adjust the image and irregular mask to 256 × 256 pixels. The parameter interpolation in torchvision.transforms.Resize is set to InterpolationMode.BILINEAR. InterpolationMode.BILINEAR refers to the bilinear interpolation method. Furthermore, we generated edge maps using a Canny edge detector whose sensitivity is controlled by the standard deviation of the Gaussian smoothing filter. In our network, we set it to two based on previous experience [20].

We implemented the network using PyTorch, a deep learning framework. The experiments were performed on an NVIDIA 3090 GPU with a batch size of 12. The commonly used Adam optimizer was used to optimize the network, and β_1 and β_2 of the Adam optimizer was set to 0.9 and 0.999, respectively. The learning rate of the generator was set to 2×10^{-4}, and the learning rate of the discriminator was one-tenth of that of the generator, which was set to 2×10^{-5}.

4.2. Benchmark Methods

We selected three benchmark methods, EC network [20], RFR network [29], and CTSDG network [15], for qualitative and quantitative comparison with the methods presented in this paper.

EC network explicitly introduced edge structure prior information. The network proved the effectiveness of structural prior information in inpainting tasks by explicitly introducing edge images, which provided ideas for future research.

Many existing image inpainting methods use generative adversarial networks as the main body of the network to generate more realistic inpainted images. However, the RFR network did not use a generative adversarial network but an autoencoder as the main body of the network.

CTSDG network is a two-stream architecture. In the decoding stage, the texture decoder borrows structural features from the structural encoder, while the structural decoder extracts texture features from the texture encoder.

The main difference between the network proposed in this article and the network proposed in the literature [15]:

1. Network structure; In literature [15], they proposed a novel two-stream network for image inpainting. Structure-constrained texture synthesis and texture-guided structure reconstruction are achieved in a coupled manner through this two-stream generator. Our generator is a dual-branch network composed of two independent U-net networks that decouple the inpainting of low-frequency and high-frequency information in a parallel and independent manner.
2. Attention mechanism; In literature [15], they proposed the Contextual Feature Aggregation module, which utilizes multi-scale feature aggregation to refine the generated images. By introducing the idea of learning and adding learnable parameters to our feature attention (FA) module, the performance of this module is further improved, so that the inpainted image has a more detailed inpainting texture.

3. Parameter setting; The batch size used in literature [15] is six. The batch size used in our experiments is 12.
4. Training process; The network proposed in literature [15] needs to adjust the learning rate to train again after the first training. In contrast, our network training procedure is relatively simple. Our network only needs to be trained once after it is configured.

4.3. Qualitative Comparison

Figure 4 presents the qualitative comparison results of our method with three benchmark methods. The first three columns of Figure 4 show the results on the Paris StreetView dataset, and the last three columns show the results on the CelebA-HQ dataset. It can be seen from Figure 4b that there are some incorrect structures in the repaired image of the EC network [20], and there are obvious artifacts; additionally, the consistency between the structure and the texture is not well guaranteed. It can be seen from Figure 4c that there are some artifacts in the results from the RFR network [29], and the phenomenon of blurred boundaries is more obvious in the results from the CelebA-HQ dataset. It can be seen from Figure 4d that the CTSDG network [15] is very competitive in inpainting images. However, in some details, it is not as good as our network restoration (such as the windowsill in the second row and the nose, ear, and eyes in the face results of the last three rows). Overall, our network can generate more visually realistic images with clear boundaries and consistent structure and texture.

Figure 4. Qualitative comparison of our method with three benchmark methods: (**a**) Input images; (**b**) EC [20]; (**c**) RFR [29]; (**d**) CTSDG [15]; (**e**) ours; (**f**) ground-true images.

4.4. Quantitative Comparison

We mainly use the three metrics of PSNR and SSIM as well as Mean l1 to quantitatively compare the results of the three baseline methods with our method on the Paris StreetView dataset. The proportions of image defects are (0.01, 0.2], (0.2, 0.4], (0.4, 0.6]. From Table 1 (The metrics in Table 1 are the averaged results over all test images), we can see that the performance of various methods gradually deteriorates as the proportion of image defects increases. However, our method outperforms the three baseline methods in all three indicators on the Paris StreetView dataset. The performance of the CTSDG network is very close to our network. Due to the adjustment of the learning rate, the CTSDG network needs to be trained twice. In contrast, our network only needs to be trained once, so the training process of our network is relatively simple. The numerical comparison shows that our method is not only effective but also superior in performance.

Table 1. Quantitative comparison of our method with three benchmark methods.

Metrics	PSNR [1]	PSNR [1]	PSNR [1]	SSIM [1]	SSIM [1]	SSIM [1]	Mean *l*1 [2]	Mean *l*1 [2]	Mean *l*1 [2]
Mask Ratio	0–20%	20–40%	40–60%	0–20%	20–40%	40–60%	0–20%	20–40%	40–60%
EC [20]	33.21436	26.88999	22.36982	0.969226	0.893742	0.741424	0.005998	0.018316	0.039742
RFR [29]	34.17117	26.91602	22.89791	0.974969	0.894790	0.749771	0.005186	0.018146	0.037578
CTSDG [15]	35.93581	28.22912	23.69428	0.980768	0.917800	0.781414	0.004107	0.014532	0.032340
Ours	36.04300	28.30448	23.77611	0.981703	0.918644	0.789554	0.003977	0.014275	0.031878

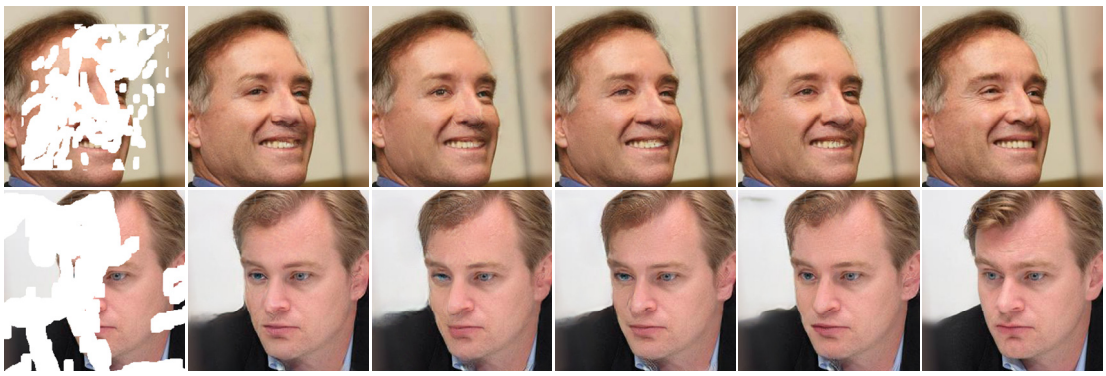
[1] Higher is better; [2] lower is better.

4.5. Ablation Experiment

We conducted ablation experiments on the CelebA-HQ dataset to verify the auxiliary role of the structure branch and the role of the feature fusion (FF) module and feature attention (FA) module.

4.5.1. Auxiliary Role of Structural Branches

To verify the auxiliary role of the structure branch, we constructed a single-branch network, which only uses the texture branch to fill the defect area and inpaint the defective image. Corresponding to keeping only the texture branch, this network also keeps only the texture discriminator. As shown in Figure 5b, the single-branch network is not ideal for nose and face structure inpainting due to the lack of structural information. The results of the quantitative comparison in Table 2 (The metrics in Table 2 are the averaged results over all test images) also demonstrate that structural branches can assist in producing better images.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 5. Qualitative ablation experiment of our method: (**a**) Input; (**b**) single branch; (**c**) w/o FF; (**d**) w/o FA; (**e**) ours; (**f**) ground-true images.

Table 2. Quantitative ablation experiment of our method.

Metrics	PSNR [1]	PSNR [1]	PSNR [1]	SSIM [1]	SSIM [1]	SSIM [1]	Mean *l*1 [2]	Mean *l*1 [2]	Mean *l*1 [2]
Mask Ratio	0–20%	20–40%	40–60%	0–20%	20–40%	40–60%	0–20%	20–40%	40–60%
Single branch	36.04495	28.28100	23.56070	0.988170	0.950049	0.867736	0.003840	0.013230	0.029942
w/o FF	35.94235	28.26042	23.49801	0.986837	0.949126	0.865271	0.003858	0.013239	0.029965
w/o FA	36.38478	28.59062	23.79217	0.988584	0.952620	0.872514	0.003592	0.012827	0.029305
Ours	36.62037	28.71729	23.92498	0.989126	0.953883	0.872923	0.003479	0.012494	0.028673

[1] Higher is better; [2] lower is better.

4.5.2. The Role of the Feature Fusion (FF) Module

The feature fusion (FF) module was introduced to enhance the consistency of the image structure and texture. To reflect the role of this module, we constructed a network without the feature fusion (FF) module. As shown in Figure 5c, this resulted in more blurred edges in the image, especially around the nose and eyes. This phenomenon is more obvious. The quantitative comparison results in Table 2 also demonstrate the effectiveness of this module in improving the quality of images.

4.5.3. The Role of the Feature Attention (FA) Module

The feature attention (FA) module was introduced to give the image finer textures. To reflect the role of this module, we constructed a network without the feature attention (FA) module. As shown in Figure 5d, the rendering of the nose in the resulting image is not ideal, indicating poor image quality. The results of the quantitative comparison in Table 2 also verify the necessity of this module.

5. Conclusions

In this paper, we propose a method for digital image inpainting with structural branch assistance. The method focuses on image structure and synthesizing image texture through a structure branch and texture branch, respectively; this decomposes the image inpainting into two simultaneous subtasks and decouples the inpainting from low-frequency and high-frequency information. The feature fusion (FF) module is introduced to enhance the consistency of the image structure and texture, and the feature attention (FA) module is introduced to give the image a more detailed texture. Experiments show that our method exhibits strong and advanced performance in both qualitative and quantitative aspects.

Author Contributions: Conceptualization, J.W.; methodology, J.W.; software, J.W.; validation, J.W., D.J. and H.Z.; formal analysis, J.W.; investigation, J.W.; resources, J.W.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, D.J.; visualization, J.W.; supervision, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bertalmio, M.; Sapiro, G.; Caselles, V.; Ballester, C. Image inpainting. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 23–28 July 2000; pp. 417–424.
2. Shen, J.; Chan, T.F. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **2002**, *62*, 1019–1043. [CrossRef]

3. Chan, T.F.; Shen, J. Nontexture inpainting by curvature-driven diffusions. *J. Vis. Commun. Image Represent.* **2001**, *12*, 436–449. [CrossRef]
4. Criminisi, A.; Pérez, P.; Toyama, K. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* **2004**, *13*, 1200–1212. [CrossRef] [PubMed]
5. Barnes, C.; Shechtman, E.; Finkelstein, A.; Goldman, D.B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **2009**, *28*, 24. [CrossRef]
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
7. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
8. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *arXiv* **2014**, arXiv:1406.2661.
9. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
10. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph.* **2017**, *36*, 1–14. [CrossRef]
11. Demir, U.; Unal, G. Patch-based image inpainting with generative adversarial networks. *arXiv* **2018**, arXiv:1803.07422.
12. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.C.; Tao, A.; Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 85–100.
13. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Free-form image inpainting with gated convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4471–4480.
14. Zeng, Y.; Fu, J.; Chao, H.; Guo, B. Learning pyramid-context encoder network for high-quality image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1486–1494.
15. Guo, X.; Yang, H.; Huang, D. Image inpainting via conditional texture and structure dual generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 14134–14143.
16. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–20 June 2018; pp. 5505–5514.
17. Sagong, M.C.; Shin, Y.G.; Kim, S.W.; Park, S.; Ko, S.J. Pepsi: Fast image inpainting with parallel decoding network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11360–11368.
18. Shin, Y.G.; Sagong, M.C.; Yeo, Y.J.; Kim, S.W.; Ko, S.J. Pepsi++: Fast and lightweight network for image inpainting. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 252–265. [CrossRef] [PubMed]
19. Song, Y.; Yang, C.; Shen, Y.; Wang, P.; Huang, Q.; Kuo, C.C.J. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv* **2018**, arXiv:1805.03356.
20. Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.Z.; Ebrahimi, M. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv* **2019**, arXiv:1901.00212.
21. Xu, S.; Liu, D.; Xiong, Z. E2I: Generative inpainting from edge to image. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 1308–1322. [CrossRef]
22. Xiong, W.; Yu, J.; Lin, Z.; Yang, J.; Lu, X.; Barnes, C.; Luo, J. Foreground-aware image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5840–5848.
23. Ren, Y.; Yu, X.; Zhang, R.; Li, T.H.; Liu, S.; Li, G. Structureflow: Image inpainting via structure-aware appearance flow. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 181–190.
24. Li, J.; He, F.; Zhang, L.; Du, B.; Tao, D. Progressive reconstruction of visual structure for image inpainting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5962–5971.
25. Liu, H.; Jiang, B.; Song, Y.; Huang, W.; Yang, C. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part II 16. Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 725–741.
26. Yang, J.; Qi, Z.; Shi, Y. Learning to incorporate structure knowledge for image inpainting. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020.
27. Wan, Z.; Zhang, J.; Chen, D.; Liao, J. High-fidelity pluralistic image completion with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4692–4701.
28. Liu, Q.; Tan, Z.; Chen, D.; Chu, Q.; Dai, X.; Chen, Y.; Liu, M.; Yuan, L.; Yu, N. Reduce information loss in transformers for pluralistic image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11347–11357.
29. Li, J.; Wang, N.; Zhang, L.; Du, B.; Tao, D. Recurrent feature reasoning for image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7760–7768.
30. Wang, W.; Zhang, J.; Niu, L.; Ling, H.; Yang, X.; Zhang, L. Parallel multi-resolution fusion network for image inpainting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 14559–14568.

31. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Part III 18. Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
32. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
33. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv* **2018**, arXiv:1802.05957.
34. Johnson, J.; Alahi, A.; Li, F.-F. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part II 14. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 694–711.
35. Doersch, C.; Singh, S.; Gupta, A.; Sivic, J.; Efros, A. What makes paris look like paris? *ACM Trans. Graph.* **2012**, *31*, 1–9. [CrossRef]
36. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.