Superresolution Reconstruction of Remote Sensing Image Based on Generative Adversarial Network

Qiaoliang Zhou

School of Information Engineering, JinZhou University, JinZhou 434000, China

Correspondence should be addressed to Qiaoliang Zhou; zhouqiaoliang@cumt.edu.cn

Received 1 November 2021; Revised 5 December 2021; Accepted 29 December 2021; Published 10 March 2022

Academic Editor: Mu-Yen Chen

Copyright © 2022 Qiaoliang Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To recreate high-resolution, more detailed remote sensing images from existing low-resolution photos, this technique is known as remote sensing image superresolution reconstruction, and it has numerous uses. As an important research hotspot of neural networks, generative adversarial network (GAN) has made outstanding progress for image superresolution reconstruction. It solves the computational complexity and low reconstructed image quality of standard superresolution reconstruction algorithms. This research offers a superresolution reconstruction strategy with a self-attention generative adversarial network to improve the quality of reconstructed superresolution remote sensing images. The self-attention strategy as well as residual module is utilized to build a generator in this model that transforms low-resolution remote sensing images into superresolution ones. It aims to determine the discrepancy between a reconstructed picture and a true picture by using a deep convolutional network as a discriminator. For the purpose of enhancing the accuracy, content loss is used. This is done to obtain accurate detail reconstruction. According to the findings of the experiments, this approach is capable of regenerating higher-quality images.

1. Introduction

People’s expectations for image quality have risen in tandem with scientific and technological advancements and the widening of practical application domains. Image resolution is the key basis for measuring image accuracy and clarity, and the level of image resolution is positively correlated with image clarity. As a result, meeting the public’s desire for high-quality photographs is now a top priority for image processing researchers.

In remote sensing imaging, people can use the acquired remote sensing data for military target identification, environmental monitoring, disaster research, and timely understanding and tracking of changes on the ground. A high-resolution remote sensing image can provide more detailed texture information and higher resolution. The richer feature information contained in the image provides a favorable basis for the various research and application of remote sensing images. However, in the remote sensing imaging process, the distance between the target and the imaging system is relatively long, and it is also affected by some other factors. The quality of the obtained remote sensing image is often low, which makes the local area in the image blurred, part of the information is lost, and the target recognition is low, which cannot meet people’s data requirements [1–5].

Improving the hardware conditions of imaging equipment is the most direct and effective way to obtain superresolution images. This method is mainly divided into two kinds: improving the manufacturing process of the sensor to reduce the pixel size. However, as the pixel size continues to decrease, the amount of light it obtains becomes smaller, and shot noise will eventually be introduced. This leads to reduced image quality. The second is to increase the number of photosensitive elements on the sensor to increase the number of pixels. This approach will lead to a significant increase in economic costs and limit the development and application of high-resolution image technology through hardware methods.

Image superresolution reconstruction is a software technique that reconstructs a high-resolution image from a series of lower-resolution photos. As a result of this method’s ability to overcome the limits of current hardware technology, it...
is inexpensive and easy to implement and can be used with any current picture system. Economic, practical, and extremely viable are all attributes of superresolution reconstruction technology. This technology is not just useful for image processing in remote sensing. Medical imaging and video surveillance are two examples of where it might be used. It offers major theoretical significance and vast application potential to use software technologies to improve the resolution of remote sensing images [6–10].

Scholars in the United States and elsewhere have used cutting-edge artificial intelligence algorithms in the field of image superresolution reconstruction, thanks to the rapid growth of AI, machine learning, computer vision, and other technologies. A new area of machine learning called deep learning enables computers to build more complex notions from simpler ones. In deep learning, it is possible to improve the quality of the rebuilt image while also reducing the computation time. There are numerous academic benefits and potential applications for using the network model of deep learning to rebuild satellite picture resolution [11–15].

The following are the paper’s innovations: (1) Using a self-attention generating adversarial network, a technique for recovering the texture features from a rebuilt superresolution remote sensing image is presented to address this challenge. (2) In order to enhance the quality of superresolution image reconstruction, optimize content and perceptual loss. (3) Many experiments have proven that the proposed method is reliable and correct, and it performs better on common datasets.

Our paper is organized as follows: Section 1 is an introduction. In Section 2, the related work is introduced. In Section 3, the proposed method is explained in detail. In Section 4, extensive experiments are conducted to verify the effectiveness of the proposed method. Section 5 is a conclusion.

2. Related Work

The interpolation-based superresolution method used different interpolation functions to fit the known pixel values of the surrounding neighborhood and then calculated the pixel for the position which is needed to be interpolated. Literature [16] used the geometric duality of images with different resolutions to estimate the pixels to be interpolated by calculating the local covariance for low-resolution images. Literature [17] proposed an image interpolation algorithm combining soft decision estimation and piecewise autoregressive reconstruction model. Each interpolation process could estimate a group of pixels at the same time, which effectively improved the edge and texture structure for interpolated images. Literature [18] proposed an adaptive interpolation algorithm, which first estimated high-resolution image gradient from low-resolution picture, and then used it as a constraint, which could reconstruct the high-resolution picture. Literature [19] introduced new model parameters on the basis of moving least square interpolation to recover more image details.

The reconstruction-based superresolution method analyzed factors such as downsampling, blur, and noise that cause image degradation, established a physical model for image degradation, and estimated the original high-resolution picture by optimizing the objective function. Literature [20] proposed a convex set projection method to achieve superresolution reconstruction and used the intersection of convex sets generated by the solution space under different constraints to obtain high-resolution images. Literature [21] proposed an iterative back-projection method, which substitutes the initially estimated high-resolution picture to the degradation model to generate the degraded image. That iteratively updated the reconstructed image by optimizing the error between it and the actual low-resolution image. Literature [22] utilized the maximum posterior probability estimation method to estimate high-resolution pictures when posterior probability is the largest. Literature [23] proposed a regional spatial adaptive full variational reconstruction model and added the processing process of spatial information filtering and information weight clustering. While restoring the edge details of the image, the interference of noise is suppressed. Literature [24] proposed a superresolution reconstruction algorithm with adaptive regularization parameters, which used the particle swarm optimization algorithm to solve appropriate regularization parameters for different image regions, which improved the quality of image reconstruction. Literature [25] estimated the high-resolution target picture with maximizing the likelihood, and the reconstructed image had obvious edge and texture features.

In the learning-based superresolution method, a large number of training samples were used to understand the mapping relationship between high-resolution and low-resolution images. The high-resolution image that corresponds to the low-resolution image could then be obtained by using this mapping relationship. Literature [26] suggested an instance learning-based strategy for superresolution reconstruction building a Markov network to learn the mapping from low-resolution picture block sizes to high-resolution image block sizes. After that, you could perform superresolution reconstruction using the network model you learned before. Literature [27] built a Markov random field using the maximum posterior probability and then used the belief propagation technique to maximize the estimate of model parameters. This enhanced the rebuilt image’s edge sharpness and texture details. In the literature, a neighborhood embedding reconstruction algorithm based on local linear embedding was proposed by literature [28]. The approach leveraged the local linear embedding of manifold learning to rebuild high-resolution images when the associated low-resolution image blocks had similar local manifold structures. Literature [29] had shown that using a binary tree complex wavelet transform to extract the image’s feature information allows for superresolution reconstruction while still preserving the image’s rich features. Literature [30] showed that only the final layer uses subpixel convolution for image upsampling, which minimized the computational complexity of the network model. Reconstruction networks with 20 convolutional layers were proposed in literature [31], and residual learning and gradient clipping were used to ensure the efficiency and stability of the networks’ convergence. Deep networks had fewer training parameters...
because literature [32] employed a recursive structure to share weight parameter values. Literature [33] employed an adversarial network to generate a superresolution image reconstruction and recovered more image texture information. A single network model was presented in literature [34] for achieving arbitrary scale superresolution reconstruction. The network’s upsampling module accepted the super-resolution reconstruction multiple as an input parameter, and the weight of the convolutional layer is dynamically predicted, and upsampling in various proportions was performed on that anticipated weight. Literature [35] introduced the superresolution reconstruction network with a feedback mechanism and used the feedback’s in-depth data for guidance in expressing the network appropriately, improving picture reconstruction.

3. Method

Using a self-attention generative adversarial network, this chapter provides a paradigm for superresolution reconstruction (SAGAN). The discriminator employs a deep convolutional neural network and adversarial loss to optimize the training of the superresolution model, while the generator uses a deep residual network that combines an attention mechanism with a residual network module. The content loss function of the model is based on the Charbonnier loss function. While this is going on, the feature value before activation for the VGG network is applied to generate perceptual loss, which can obtain precise texture detail reconstruction.

3.1. Overall Framework. To estimate the superresolution image from a low-resolution image, the superresolution reconstruction problem uses a low-resolution image. This image is as close as possible to the original superresolution photograph. A generator network $G$ must be trained to solve this problem, and the parameters of the network can be represented by the symbol $\theta$ for the network. Low-resolution images are fed into the generator network, and the output is rebuilt superresolution images, namely,

$$\hat{I}_S = G_\theta(I_L),$$

where $\hat{I}_S$ is the reconstructed superresolution image, $I_L$ represents the low-resolution image, and $\theta$ should satisfy

$$\theta = \arg \max_{\theta} \sum L(\hat{I}_S, I_S),$$

where $L$ is loss and $I_S$ is the true superresolution image.

Superresolution picture reconstruction requires training a generator network and a discriminator network to tackle the task effectively. Generators and discriminators make up the system’s overarching framework. To begin, the generator tries to make the low-resolution picture look like a true superresolution picture, while the discriminator looks for differences between the two. Figure 1 depicts the overall model’s block diagram.

Ultimately, the goal of this research is to develop a generator network that can produce superresolution images that are as close to actual ones as feasible. The self-attention layer as well as residual self-attention is introduced and embedded to accomplish this. It is capable of superior superresolution picture reconstruction by utilizing global feature information. Our method replaces the standard BatchNorm layer for the generator with an instance normalization layer. Additionally, the resemblance between generated picture and the original picture can be assessed further thanks to the use of content loss optimization techniques. The perception loss is also optimized, and perception loss is calculated using features before activation of the VGG19 network.

3.2. Self-Attention Mechanism Layer. Deep learning researchers originally advocated using the attention mechanism for natural language processing. It is capable of removing regions from the global area that are not as important, which is crucial for jobs like natural language processing. Internal correlation in the data distributions can be captured better with this technique, and the need for external information is reduced. To focus on a specific location, the human eye quickly analyzes the entire image. The primary objective of the self-attention model is also to choose from a big amount of information that is more critical for the current aim.

It is more efficient to simulate the multilevel dependencies between picture regions when using the self-attention mechanism module in deep learning. The self-attention mechanism is critical in models that must account for global interdependencies. The majority of GAN-based picture generating models now use convolutional layers to create their convolutions. However, because convolution only processes local information, modeling global image dependencies with just the convolution layer is computationally inefficient. With the self-attention mechanism, the image’s long-range and multilevel dependencies can be dealt with more effectively, and the generation for near and far details in the picture may be coordinated. To better reconstruct superresolution picture’s texture details, consider adding the self-attention layer to the generator’s residual module as well as using global information. Figure 2 depicts the hypothesized self-attention mechanism’s structural diagram.

As a result of applying two convolutional layers on top of the residual block’s second feature map, we can get the two feature spaces $a(x)$ and $b(x)$, respectively. Pixel features are extracted using $a(x)$, and global features are extracted using $b(x)$. The attention map is created by taking $a(x)$ and $b(x)$ and translating them into the following:

$$a_{ji} = \frac{\exp(\beta_{ij})}{\sum_{j} \exp(\beta_{ij})},$$

$$\beta_{ij} = a(x_i) b(x_j),$$

where $a_{ji}$ represents the model’s attention for the $i$th position while processing the $j$th area; then, the output is

$$e_i = \sum_{j=1}^{N} a_{ji} c(x_i).$$
Add the output from the attention layer back to the input feature map after multiplying it by the scale parameter. As a result, here is what we have:

\[ y_i = \gamma e_i + x_i, \]  

(6)

where \( y_i \) is the final output and the initial value of \( y \) is set as zero. The extracted feature map will be fed into the next attention mechanism network, and the feature extraction and learning process will be repeated. Adding a self-attention layer to the generator’s residual module improves image reconstruction by making greater use of global feature data. This aids in the recovery of high-resolution photos’ texture information.

3.3. Instance Normalization. In many image classification tasks, batch normalization is proved to be efficient. Nevertheless, this normalizing strategy reduces the performance. Instance normalization was first proposed in image style conversion. The instance normalization layer allows instance-specific contrast information to be removed from content images. This simplifies the generation process and can greatly improve image quality.

Instance normalization itself is a very simple algorithm, especially suitable for scenes where the batch size is small and each pixel is considered separately, because it calculates the normalized statistics without mixing the data between batches and channels. For this kind of application scenario, it is generally considered to use instance normalization. In image applications, the value on each channel is relatively large, so a more appropriate normalized statistic can be obtained. When doing picture style conversion, this is employed to greatly improve the end result. SAGAN’s picture super-resolution reconstruction model considers using an instance normalizing layer in place of the generator’s batch normalization layer in order to increase speed. One way to generate a single picture instance is by doing instance normalization on a single image. Using this example regularization formula, we may express as

\[ y_{tcmn} = \frac{x_{tcmn} - \mu_{tc}}{\sqrt{\sigma_{tc}^2 + \nu}}, \]  

(7)

\[ \mu_{tc} = \frac{\sum_{h=1}^{H} \sum_{w=1}^{W} x_{tchw}}{HW}, \]  

(8)

\[ \sigma_{tc}^2 = \frac{\sum_{h=1}^{H} \sum_{w=1}^{W} (x_{tchw} - \mu_{tc})^2}{HW}, \]  

(9)

in which \( H \) and \( W \) denote the image’s height and width and \( x_{tcmn} \) denotes the \( tcmn \)th image in the batch, where \( m \) and \( n \) are the image’s spatial dimensions, \( c \) is the input feature channel, and \( t \) is the batch’s index.
3.4. Loss Function. SAGAN uses adversarial loss in the loss function to reconstruct images with crisp textures at high resolution. The discriminator network is deceived as much as possible by the superresolution image rebuilt by the generator network. The adversarial loss is

\[ L(G, D) = E_{I_L \sim I_L}[\log D_\theta(I_L)] + E_{E_{IS} \sim E_{IS}}[\log (1 - D_\theta(G_\theta(I_L)))], \]  

(10)

where \( D_\theta(I_S) \) represents the reconstructed superresolution picture, \( I_L \) is the true superresolution picture, and \( I_L \) represents the low-resolution image.

GANs have traditionally used JS divergence as a way to gauge how far real-world data differs from the generated data’s probability distribution. There are no valid gradients for training if you use JS divergence to estimate the probability distribution. Because the distance between the generated sample and the real sample is discontinuous in high-dimensional space, the created sample cannot be compared to the genuine sample. GAN cannot practice because there are not any junction points. The distance between two distributions is calculated using the Wasserstein distance rather than the JS divergence in this article. The lowest gap between the real sample distribution and the produced sample distribution is used to calculate the Wasserstein distance. In theory, the Wasserstein distance is differentiable almost everywhere. Therefore, it can quickly and effectively guide the training of the GAN model. In the adversarial training process, the adversarial loss based on the Wasserstein distance is

\[ L_w(G, D) = E_{I_L \sim I_L}D_\theta(I_L) - E_{E_{IS} \sim E_{IS}}D_\theta(G_\theta(I_L)). \]  

(11)

When all samples are considered, the generator loss and discriminator loss are both defined as

\[ L_G = -\sum D_\theta(G_\theta(I_L)), \]  

(12)

\[ L_D = \sum (D_\theta(I_S) - D_\theta(G_\theta(I_L))). \]  

(13)

To determine how comparable reconstructed superresolution picture and target superresolution picture are, a loss must be built between the two images during the experimental training procedure. As long as the difference between the reconstructed and target superresolution images is being measured with MSE, model training will have a much smaller impact, because the reconstructed image will be excessively smooth and lack realism if only pixel-level reconstruction is used. In most image superresolution reconstruction techniques, the difference between reconstructed and target images is calculated using perceptual loss at the moment. Contrary to popular belief, perceptual loss has been extensively studied and exploited in image superresolution reconstruction methods.

When using a pretrained deep network, perceptual loss is often embedded on the activation layer, and associated loss function is derived using the activated feature value. However, as the network depth increases, the number of activated features decreases, making it difficult to keep track of everything. As a result, SAGAN calculates perceptual loss using feature value, and feature value before activation better represents the image’s feature information. It is capable of keeping an eye on the reconstructed image’s texture and comparing it to the original. Prior to applying an activation layer, a trained VGG19 is applied to get the feature value. Perceptual loss is computed by comparing Euclidean distance between superresolution picture feature map generated and image’s initial feature:

\[ L_P = \sum (V(I_S) - V(G_\theta(I_L)))^2, \]  

(14)

where \( V \) is the VGG19 module.

A content loss function must be incorporated into the model to guarantee that the generated superresolution picture and original picture have similar content. SAGAN replaces the usual L2 loss with Charbonnier loss to increase the network’s performance. Charbonnier loss is

\[ L_C = \sum (\rho(I_S - (G_\theta(I_L)))), \]  

(15)

where \( \rho \) is the Charbonnier penalty function. For this reason, it has stronger supervision capabilities than other types of losses and is more robust for Charbonnier’s loss. Adding up all of the above losses, the generator network model has a total loss of

\[ L = L_P + \lambda_1 L_G + \lambda_2 L_C, \]  

(16)

where \( \lambda_1 \) and \( \lambda_2 \) are two weights for loss.

3.5. Generator and Discriminator Structure. The generator’s primary job during training is to produce images with a high level of resemblance to real high-resolution images. In order to discriminate between a reconstructed and true superresolution image, the discriminator must be used. After training the generator and discriminator, the generated network is the needed network for superresolution picture reconstruction. Different convolution kernels, downsampling, self-attention residuals, and picture reconstruction via upsampling all make up the generator network. The generator structure is shown in Figure 3.

An image can be reconstructed using three distinct convolution kernel sizes \((3 \times 3, 5 \times 5, 7 \times 7)\) and then convolved across three different receptive fields \((3 \times 5 \times 5, 5 \times 5, 7 \times 7)\). Next, a \(1 \times 1\) convolutional layer is applied with a PReLU activation function on the convolved feature map and the original picture, and the results are delivered to the Concat layer. Convolutional layers and self-attention layer are employed in a single self-attention residual module. Next comes the instance normalization layer, followed by the convolutional layer. The activation strategy makes use of PReLU, and the jump connection is inserted. An additional jump connection is embedded to the last output layer for the self-attention residual block. It is utilized for image pixel amplification. At the same time,
massive deep network layers are increased to improve image reconstruction in the image upsampling part.

A deep convolutional neural network is utilized in the discriminator network. BatchNorm is still employed in the discriminator network because of its efficacy in image classification tasks, while LReLU is utilized as the activation function. The structure is illustrated in Figure 4.

With the convolutional layer, the feature map is sent to the fully connected layer, where it is classified using the sigmoid activation function. Generator and discriminator minimization leads to better visual quality while generating high-resolution photos.

4. Experiment and Discussion

4.1. Dataset. Three remote sensing picture datasets are used in this article. UC Merced’s land use dataset (UC) is the first one to use [36]. The training picture set has 80 photos chosen at random, whereas the testing image set contains 20 images chosen at random from the training image set. NWPU-RESISC45 (NW) is the second [37]. The dataset consists of 45 different types of scenes, each of which has 700 photos. The dataset’s size is the same as the UC dataset, and all of the images are 256 × 256 pixels. A random sample of 100 photographs from each type of scene is chosen at random and added to the training image set. Then, from the rest of the photos, randomly select 10 images from each scene type to include in the testing set. The dataset is shown in Table 1.

In this work, peak the signal-to-noise ratio (PSNR) and structural similarity (SSIM) are utilized to evaluate the performance. These are two different performance indicators, and the performance of the algorithm can be evaluated from different aspects.

4.2. Comparison with Other Methods. SAGAN proposed in this paper is compared with Bicubic [38], SRCNN [39], ESPCN [30], and MSRNN [40]. To ensure fairness, all methods are all retrained and tested from the same set of remote sensing images. Experimental details are illustrated in Tables 2 and 3.

There is a 3 dB to 5 dB difference in PSNR between SAGAN and SRCNN/ESPCN for both datasets. The outcomes of image superresolution reconstruction have been greatly improved as a result of significant advancements. SRCNN and ESPCN, on the other hand, are only capable of extracting a few features. Deep networks like SAGAN and MSRN outperform SRCNN and ESPCN in terms of performance. The approach in this research has higher objective indicators PSNR and SSIM on the two datasets for the deep network MSRN when compared to the SAGAN method. Here is a way to demonstrate how much better SAGAN is than the alternatives.

Table 1: The dataset details.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Scenes</th>
<th>Size</th>
<th>Total</th>
<th>Training</th>
<th>Testing</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC</td>
<td>21</td>
<td>256 × 256</td>
<td>2100</td>
<td>1680</td>
<td>420</td>
<td>0.3</td>
</tr>
<tr>
<td>NW</td>
<td>45</td>
<td>256 × 256</td>
<td>31500</td>
<td>4500</td>
<td>450</td>
<td>0.2-30</td>
</tr>
</tbody>
</table>

Table 2: Comparison with other methods on UC.

<table>
<thead>
<tr>
<th>Method</th>
<th>Bicubic</th>
<th>SRCNN</th>
<th>ESPCN</th>
<th>MSRN</th>
<th>SAGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>26.8</td>
<td>29.7</td>
<td>31.5</td>
<td>34.8</td>
<td>35.4</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.64</td>
<td>0.78</td>
<td>0.82</td>
<td>0.93</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Table 3: Comparison with other methods on NW.

<table>
<thead>
<tr>
<th>Method</th>
<th>Bicubic</th>
<th>SRCNN</th>
<th>ESPCN</th>
<th>MSRN</th>
<th>SAGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>27.9</td>
<td>30.8</td>
<td>32.5</td>
<td>34.9</td>
<td>35.6</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.67</td>
<td>0.82</td>
<td>0.85</td>
<td>0.93</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Figure 3: Generator network.

Figure 4: Discriminator network.
4.3. Evaluation on Self-Attention Mechanism Layer. Self-attention mechanisms are used to extract more discriminatory features from the pipeline in this study. We conduct a comparative experiment to examine the model’s performance with and without a self-attention mechanism layer to demonstrate the strategy’s efficacy. Experimental results on two datasets are illustrated in Figure 5. NSA represents having no self-attention layer. SA represents having a self-attention layer.

Obviously, with the introduction of the self-attention layer, both the PSNR and SSIM for the model can be improved. On the UC dataset, the gains of PSNR and SSIM are 1.5% and 0.03. On the NW dataset, the gains of PSNR and SSIM are also 1.5% and 0.03. These data can verify the correctness of using the self-attention mechanism layer.

4.4. Evaluation on Instance Normalization Layer. In this work, an instance normalization layer is embedded in the pipeline to normalize features. To verify the effectiveness of this strategy, a comparative experiment is conducted to compare the performance for the model with and without an instance normalization layer. The experimental results on two datasets are illustrated in Figure 6. NIN represents having no instance normalization layer. IN represents having an instance normalization layer.

Obviously, with the introduction of the instance normalization layer, both the PSNR and SSIM of the model can be improved. On the UC dataset, the gains of PSNR and SSIM are 1.1% and 0.02. On the NW dataset, the gains of PSNR and SSIM are also 0.9% and 0.02. These data can verify the correctness of using the instance normalization layer.
4.5. Evaluation on Loss. In this work, a mixed loss consisting of generator loss, content loss, and perceptual loss is proposed to optimize the network. To prove the effectiveness of this strategy, we first conducted a comparative experiment to compare the training loss of the model with different losses. Experimental results are illustrated in Figure 7. GL is generator loss. CL is content loss. PL is perceptual loss. ML is mixed loss proposed in this work.

As the number of iterations increases, the loss value for the network gradually decreases and finally converges steadily. But it should be noted that, compared with other individual loss functions, the combined loss proposed in this paper has the lowest corresponding loss in any iteration. This can effectively prove the reliability and correctness of the loss in this article.

To further verify the effectiveness of this loss combination strategy in this article, this article also conducts another set of comparative experiments to compare the effects of different loss combinations on model performance. Experimental results on two datasets are illustrated in Table 4.

Table 4: Evaluation on different losses.

<table>
<thead>
<tr>
<th>Loss</th>
<th>UC PSNR</th>
<th>UC SSIM</th>
<th>NV PSNR</th>
<th>NV SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>33.2</td>
<td>0.89</td>
<td>34.1</td>
<td>0.91</td>
</tr>
<tr>
<td>CL</td>
<td>33.5</td>
<td>0.90</td>
<td>34.3</td>
<td>0.91</td>
</tr>
<tr>
<td>PL</td>
<td>33.7</td>
<td>0.91</td>
<td>34.5</td>
<td>0.92</td>
</tr>
<tr>
<td>GL+CL</td>
<td>34.6</td>
<td>0.92</td>
<td>34.6</td>
<td>0.93</td>
</tr>
<tr>
<td>GL+PL</td>
<td>34.8</td>
<td>0.92</td>
<td>34.8</td>
<td>0.93</td>
</tr>
<tr>
<td>CL+PL</td>
<td>35.1</td>
<td>0.93</td>
<td>35.2</td>
<td>0.94</td>
</tr>
<tr>
<td>ML</td>
<td>35.4</td>
<td>0.94</td>
<td>35.6</td>
<td>0.95</td>
</tr>
</tbody>
</table>

4.5. Evaluation on Loss. In this work, a mixed loss consisting of generator loss, content loss, and perceptual loss is proposed to optimize the network. To prove the effectiveness of this strategy, we first conducted a comparative experiment to compare the training loss of the model with different losses. Experimental results are illustrated in Figure 7. GL is generator loss. CL is content loss. PL is perceptual loss. ML is mixed loss proposed in this work.

As the number of iterations increases, the loss value for the network gradually decreases and finally converges steadily. But it should be noted that, compared with other individual loss functions, the combined loss proposed in this paper has the lowest corresponding loss in any iteration. This can effectively prove the reliability and correctness of the loss in this article.

To further verify the effectiveness of this loss combination strategy in this article, this article also conducts another set of comparative experiments to compare the effects of different loss combinations on model performance. Experimental results on two datasets are illustrated in Table 4.

It can be seen that compared with any single loss, the loss of our combination is higher than their performance. In addition, with the gradual introduction of losses, the performance of the model is gradually increasing. This shows that combining different losses can effectively make network learning more discriminative features. And compared with the loss of any other pairwise combination, the best performance can be obtained by combining three kinds of losses. This also further proves the correctness and reliability of the loss proposed in this article.

5. Conclusion

Superresolution reconstruction technology can use software algorithms to effectively improve the resolution of remote sensing images without being restricted by hardware devices. This method has low cost and high universality. Vigorous development on deep learning technology has made image superresolution reconstruction technology a research hotspot. Among them, the generative adversarial network, as a classic algorithm in deep learning, has developed rapidly in superresolution reconstruction, and many excellent and effective algorithm models have emerged. However, most networks are for general images, and there are still many deficiencies and room for improvement in the reconstruction methods of remote sensing images. This thesis studies the generative countermeasure network, focusing on superresolution reconstruction with a generative adversarial network on remote sensing images. This paper proposes a superresolution reconstruction algorithm with a self-attention strategy. The first step is to add a self-attention layer to the generator network so that global features may be efficiently utilized when reconstructing high-resolution images. Replace the BatchNorm layer with the instance normalization layer based on the deep network structure. Once you have optimized the content loss, compare the reconstructed superresolution image to the original to see how closely they compare in terms of resolution. The final step is to optimize and generate perception loss based on the feature value prior to activation of the VGG19 network. Prior to activation, the features provide a more accurate

Figure 7: The training loss on two datasets.
representation of the image’s feature information. It is now possible to examine the texture consistency in order to confirm that the reconstructed image is accurate. According to results from experiments, the suggested approach improves PSNR and SSIM scores compared to existing algorithms, while also providing greater realism and clarity in the reconstructed image’s textures.

Data Availability
The datasets used are available from the corresponding author on reasonable request.

Conflicts of Interest
The author declares that they have no conflict of interest.

Acknowledgments
The project is sponsored in part by the Science Foundation of JinZhou University, Project No. 2020KY05.

References


