Image Super-Resolution Algorithm Based on RRDB Model

HUAN LI
Ningbo Polytechnic, Ningbo, Zhejian 315800, China
e-mail: lihuantougao2021@163.com

This work was supported in part by the Research on the Influence of the Evolution of New Generation Network and Information Technology on Zhejiang Media and its Development Trend under Grant 20NDYD022YB.

ABSTRACT Aiming at the problems of texture distortion and fuzzy details in the existing image super-resolution reconstruction methods, a super-resolution reconstruction network based on multi-channel attention mechanism is proposed. The texture extraction module designs an extremely lightweight multi-channel attention module in the network structure. Combined with one-dimensional convolution, it realizes cross-channel information interaction, focusing on important feature information; The texture restoration module introduces dense remaining blocks to restore some high-frequency texture details, improve the model performance, and generate high-quality reconstructed images. The proposed network can not only effectively improve the visual effect of the image, but the results on the benchmark data set CUFED5 are similar to the classic super-resolution (SRCNN) reconstruction method based on convolutional neural network. The peak signal-to-noise ratio (PSNR) and structure are similar. Degrees (SSIM) increased by 1.76dB and 0.062 respectively. Experimental results show that the designed network can improve the accuracy of texture migration and can effectively improve the quality of the generated image.

INDEX TERMS Image super-resolution reconstruction, texture migration, attention mechanism, one-dimensional convolution, dense residual block.

I. INTRODUCTION

Super-resolution reconstruction of a single image is a technique for recovering high-resolution images from low-resolution images. At present, high-resolution images are widely used in the fields of remote sensing mapping, medical images, video surveillance, and image generation [1]–[3]. Due to current technological development limitations and cost considerations, the use of software processing methods to obtain higher resolution images has become a research hotspot in the field of image processing.

For the traditional interpolation [4] and reconstruction [5] methods, there are usually problems such as poor reconstruction effect and blurred edges. With the development of technology, people began to focus on deep learning technology. Yang et al. [6] introduced deep learning into the field of image reconstruction for the first time, and proposed a convolutional neural network method (super-resolution”convolutional”neur-ural”neet) to achieve reconstruction by constructing 3 layers of SROCKN, volume. Jun [7] proposed a sub-pixel convolution method (efficient, sub-pixel, convolutional, neural network, ESPCN), which does not require preprocessing of low-resolution images.

It is directly used as the input of the network for feature extraction, and the feature map is arranged in the last layer to realize the up-sampling operation, which reduces the destruction of low-resolution image context information and makes the feature information as much as possible. For convolutional networks, the deeper the network, the better its processing capabilities. In image processing, the deep network can also more fully extract the feature information in the image, so that the processing effect is improved, but in the application, it is found that, Increasing the number of network layers will cause gradient dispersion problems. Songsheng et al. [8] combined the residual network [9] and proposed a deep convolutional network method (very deep-network for subper-resolution, VDSR) to solve the problem of the cumulative problem. Yonggong et al. [10] proposed a method of residual dense connection network (residual dense network, RDN), through the mutual connection and fusion of multiple residual dense blocks, it can extract feature information more effectively and improve the quality.
of reconstruction. Xin et al. [11] constructed a method of cascaded channel segmentation network (channel, splitting, network, CSN), and distributed the feature information in the sub-network to reduce the learning burden of the deep network and improve the training effect. Currently, in the field of image reconstruction, learning-based methods have become the focus of research.

Unlike traditional SISR, RefSR extracts the texture of the Ref image to compensate for the missing details in the LR image, so that the generated HR image has more detailed and realistic texture. For example, in 2018 Rui et al. [12] proposed a neural texture transfer model (Super-Resolution by Neural Texture Transfer, SRNTT). SRNTT performs local texture matching in the feature space, and then transfers the matched texture to the final output through the depth model. In 2020, Wei et al. [13] proposed a Texture Transformer Network for Image Super-Resolution (TTSR). TTSR encourages joint feature learning through LR images and Ref images, and discovers deep feature correspondences through the attention mechanism to deliver accurate texture features [14]. However, in the process of restoring the texture of these models, there will be problems such as face distortion and unrealistic texture restoration.

In order to solve the above problems, inspired by the idea of ECA (Efficient Channel Attention) attention mechanism in the literature [15], this paper proposes an image Super-Resolution network by multi-Channel Attention based on the multi-channel attention mechanism, SRCA). Compared with most current RefSR methods, SRCA can recover image details better [15].

Combine the multi-channel attention mechanism with the texture search module, realize local cross-channel information interaction through one-dimensional convolution, and assign different weights to each feature channel of the input image to focus on extracting more important feature information to facilitate Feature reuse. The texture recovery module introduces dense residual blocks to improve the structure of the model, removes the batch normalization layer in the dense residual blocks, and uses residual scaling to restore some high-frequency details to produce high-quality reconstructed images.

II. METHODS IN THIS PAPER

A. NETWORK ARCHITECTURE

The network structure of this paper is shown in Figure 1. The algorithm is mainly realized by cascading the nonlinear feature mapping module and the up-sampling reconstruction module [16]. \( I_{LR}, I_{SR} \) Represent the input and final output of the model respectively, and use a convolutional layer with a convolution kernel size of \( 3 \times 3 \) to extract the shallow features \( F_0 \) of the input LR image:

\[
F_0 = \text{Conv}_{3, 3}(I_{LR})
\]

The purpose of using \( 3 \times 3 \) convolution is to use fewer parameters to build a lightweight network. At the same time, in super-resolution tasks, especially in the first layer, it is not appropriate to use a core with a larger receptive field. Since each pixel in the down-sampled image corresponds to a small area in the original image, the large receptive field may introduce irrelevant information during the training process. The shallow feature \( F_0 \) completes the information extraction through the nonlinear feature mapping module, which is composed of 20 adaptive residual attention information extraction modules (ARB). After passing through the nth adaptive residual attention information extraction module, the feature \( X_n \) is extracted from the output feature \( X_{n-1} \) of the previous block, and the formula is shown.
in equation (2).

\[ X_n = g_{ABC}^n(X_{n-1}) \] (2)

In formula (2), \( g_{ABC}^n \) represents the nth adaptive residual attention information extraction module. After obtaining the features from the last residual attention information extraction module \( X_L \), after going through a multi-scale upsampling module, as shown in equation (3), \( F_{tail} \) represents a multi-scale upsampling module used to upsample \( X_L \) to the target size \( X_{tail} \):

\[ X_{tail} = F_{tail}(X_L) \] (3)

This module integrates the multi-frequency information generated by the nonlinear mapping module, and uses sub-pixel convolution to up-sample the image. Sub-pixel convolution (sub-pixel) was proposed by ESPCN [16] as a way of upsampling. The purpose is to use sub-pixel convolution for a low-resolution image to rearrange the pixels between the channels of the feature map. Corresponding high-resolution images, in order to make mapping learning easier, the original low-frequency information is introduced to maintain low-frequency accuracy, and the \( F_{tail} \) bilinear interpolation module with the same components \( F_{tail} \), get global information \( X_{skip} \):

\[ X_{skip} = F_{skip}(I_{LR}) \] (4)

Finally, the target image \( I_{SR} \) by \( X_{skip} \) and \( X_{tail} \) add up to get:

\[ I_{SR} = X_{tail} + X_{skip} \] (5)

**B. Texture Extraction Module**

The quality of the features extracted by the texture extraction module has a crucial impact on the generalization ability of the model. This model adds ECA attention mechanism to the VGG19 network [17] to improve feature extraction and increase expressiveness.

The structure of the multi-channel attention mechanism is shown in Figure 2. Given an input \( x \in \mathbb{R}^{W \times H \times C} \), first use GAP (Global Average Pooling) to perform dimensionality reduction operations, then perform fast one-dimensional convolution with \( k = 5 \) to generate channel weights, and finally multiply the output result with the input \( x \). Among them, \( k \) is the convolution kernel of 1D convolution, which represents the local cross-channel interaction coverage, and \( s \) is the Sigmoid function used to generate channel weights. The multi-channel attention mechanism can use the correlation between channels to obtain the weight value of the feature map to adaptively adjust the channel features, and process the high and low frequency information in the feature map to better extract the effective texture.

In the VGG19 network, relu1_1, relu2_1, and relu3_1 are used as multi-scale texture encoders. In order to speed up the matching process, this paper only performs matching on the relu3_1 layer and projects the corresponding relationship to relu2_1 and relu1_1, which can reduce the amount of calculation while ensuring the accuracy of texture migration. \( Q \) (query), \( K \) (key) and \( V \) (value) represent the three basic elements of the attention mechanism inside the texture migration network. \( K \) and \( Q \) represent only the relu3_1 layer features of Ref images, and \( V \) represents the extraction of Ref images Features on the three layers of relu1_1, relu2_1, and relu3_1.

**C. Texture Search Module**

The texture search module determines the texture correlation between the input image and the reference image by comparing the relu3_1 layer features between \( K \) and \( Q \). First, as shown in equation (6), take the output of \( K \) and \( Q \) as input, and calculate their similarity by normalizing the
Among them $Q_i$, $K_j$ Indicates that Q and K are expanded into blocks, the number of blocks are respectively $H_{LR} \times W_{LR}$, $H_{Ref} \times W_{Ref}$, Take the i-th and j-th blocks among them, $S_{i,j}$, Means $Q_i$ versus $K_j$ The similarity between. As shown in formula (7), through $S_{i,j}$, To calculate $Q_i$ versus $K_j$ Index of the most relevant position $h_i$ And value $r_i$. In this way, the transferred HR texture features are obtained from the reference image.

$$h_i = \arg \max_j S_{i,j}, \quad r_i = \max_j S_{i,j}$$ (7)
Among them: SRout represents the output image of this model, and R represents the output image of the texture restoration module.

**F. LOSS FUNCTION**

The loss function can play a role in measuring the performance of the model. In order to preserve the spatial structure of the LR image, improve the visual quality of the deep image and make full use of the rich texture of the Ref image, this paper uses three loss functions: reconstruction loss, counter loss and perceptual loss. Reconstruction loss is used in most SR methods. Combating loss and perceptual loss can improve the visual quality of the generated image.

The reconstruction loss is usually measured by Mean Square Error (MSE) to improve PSNR. In this paper, L1 norm is used. Compared with L2 norm, L1 norm can make the Square Error (MSE) to improve PSNR. In this paper, L1 norm is used. Compared with L2 norm, L1 norm can make the weights sparse, convenient for feature extraction, more sensitive performance, and fast convergence. The reconstruction loss can be expressed by equation (10). Where (C, H, W) is the size of the HR image, \( I^{HR} \) Represents the HR image, \( I^{SR} \) Represents the generated SR image.

\[
L_{rec} = \frac{1}{CHW} \left\| I^{HR} - I^{SR} \right\|_1
\]  

(10)

Combating loss can significantly improve the clarity and visual quality of the generated image. This article uses WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty). WGAN-GP proposes gradient penalty to solve the problem of training gradient vanishing gradient explosion and has a faster convergence speed than WGAN (Wasserstein Generative Adversarial Network) and generates higher quality samples [18]. Confrontation loss can be expressed by equation (11) and equation (12):

\[
L_D = E_{x \sim P_g} [D(\hat{x})] - E_{x \sim P_r} [D(x)]
+ \lambda \left( \| \nabla \bar{D}(\hat{x}) \|_2 - 1 \right)^2
\]  

(11)

\[
L_G = - E_{x \sim P_g} [D(\hat{x})]
\]  

(12)

Perceptual loss has been proven to significantly improve visual quality. It calculates the loss by comparing the convolution output of the original image with the convolution output of the generated image. The perceptual loss in this paper can be expressed by equation (13):

\[
L_{per} = \frac{1}{CHW} \left\| \phi^1_v \left( \hat{I}^{SR} \right) - \phi^1_v \left( I^{HR} \right) \right\|^2 + \frac{1}{CHW} \left\| \phi^{ATE} \left( \hat{I}^{SR} \right) - V_{by} \right\|^2
\]  

(13)

Among them: the first part \( \phi^1_v \) Represents the feature map of the i-th layer in the VGG19 network; the second part represents the feature map of the j-th layer of the texture extraction module; \( V_{by} \) It represents the HR texture feature of the j-th layer extracted and transferred for V. Using this perception function can make the network more effectively transfer the texture features of the Ref image.

Table 1 shows the convolution kernel parameter settings of the network structure of this algorithm, where “ARB” represents the adaptive residual attention information extraction module, the number is 20, divided into two branches, one of which is the “RB” residual branch, The other is the “CoordA” coordinate attention branch; “kernel_size” represents the size of the convolution kernel; “input_channel” and “output_channel” represent the number of input channels and output channels of each convolution kernel, respectively. In addition, in this algorithm, up-sampling uses the “pixelshuffle” sub-pixel convolution method to enlarge the input image, and the loss function uses the L1 regularization method to constrain the algorithm.

| Layer_name | Kernel_size | Input_channels | Output_channels |
|------------|-------------|----------------|-----------------|
| Conv1      | (3, 3)      | 3              | 64              |
| [ARB]*20   | (1, 1)      | 3              | 32              |
| (RB)       | (3, 3)      | 32             | 32              |
|            | (3, 3)      | 32             | 32              |
|            | (1, 1)      | 64             | 32              |
|            | (1, 1)      | 32             | 16              |
|            | (1, 1)      | 16             | 32              |
|            | (1, 1)      | 64             | 64              |
| Upconv1    | (3, 3)      | 64             | 256             |
| Upconv2    | (3, 3)      | 64             | 256             |
| Conv_out   | (3, 3)      | 64             | 3               |

**G. COORDINATE ATTENTION**

Coordinate attention is pooled by one-dimensional global average pooling in the horizontal and vertical directions of the feature map, generating feature maps with spatial orientation in both directions, and then performing different convolution operations to generate features that focus on location information Figure. The existing super-resolution network models based on the attention mechanism usually use channel attention and spatial attention networks. Although these two networks can significantly improve the performance of the model, the position information is not taken into consideration. It is very important for generating spatial selective attention feature maps. Inspired by the literature [19], the coordinate attention network is improved: after the image passes through the batch normalization layer, its color distribution information is normalized, destroying the original contrast information, so the network removes the batch normalization information. One layer.

As shown in Figure 5, X represents the input feature whose shape size is (nf, 64, 64) after 1 × 1 convolution, “nf” represents the number of channels, and 64 represents...
The \( \sigma \) in equations (17) and (18) represents the sigmoid function. The output \( g^h_c \) and \( g^w_c \) are used as attention weights, and finally \( Y \) output from the coordinate attention module is expressed as:

\[
Y_c(i, j) = X_c(i, j) \cdot g^h_c(i) \cdot g^w_c(j)
\]  

(19)

This paper uses the heat maps of the attention maps of the “baboon” images in the Set14 data set under different network layers to show the effect of coordinate attention on the high-frequency details of the image, as shown in Figure 3, “block1”, “block9”, “Block15” and “block20” respectively represent the output images of the 1, 9, 15, and 20 ARB modules. When outputting features, the sigmoid function is used to constrain the feature value to \([0, 1]\). The brighter part of the image indicates attention. Where the force is more concerned. It can be seen from the figure that in the low-level network, attention pays more attention to the overall outline of the image, and in the deep layer of the network, attention is more inclined to the high-frequency details of the image, which shows that the coordinate attention is more sensitive to the location information of the high-frequency details. Have stronger attention.

H. ADAPTATION RESIDUAL ATTENTION INFORMATION EXTRACTION MODULE

The self-correcting convolution proposed in [15] splits the feature channel and uses two branches to process the input features with different functions. The self-correcting branch obtains the attention feature map through the attention module, and the convolution branch performs conventional feature extraction. Integrating the features of the two as the final output, the self-correction branch can only focus on the position information of the attention map, reducing additional learning parameters. Since low-resolution images contain a large amount of low-frequency information and a small amount of high-frequency information, low-frequency information is often easier to learn. At the same time, in order to better enable the model to learn high-frequency information, the skip link in the residual block is used to output the low-frequency information. The addition of features allows the model to directly learn more complex high-frequency information, and at the same time relieves the training pressure. Therefore, this paper is inspired by self-correcting convolution [20] and proposes an adaptive residual attention information extraction module, which is the core of the nonlinear mapping module. Different from the design of a single information flow, the adaptive residual attention information extraction module has the following advantages.

1. The coordinate attention branch can directly learn complex high-frequency information, and at the same time relieves the training pressure.
2. It can be seen from the figure that in the low-level network, attention pays more attention to the overall outline of the image, and in the deep layer of the network, attention is more inclined to the high-frequency details of the image, which shows that the coordinate attention is more sensitive to the location information of the high-frequency details. Have stronger attention.

In equation (16), \( \delta \) represents the merging operation, and \( \delta \) represents the \( H \)-sigmoid nonlinear activation function. Then input \( f \) divided into two independent tensors along the spacial dimension \( f^h \) and \( f^w \). After two \( 1 \times 1 \) convolutional layers, \( f^h \) and \( f^w \). Substituting a tensor with the same number of channels as the input \( X \), we get:

\[
g^h_c = \sigma(\text{conv}1 \times 1(f^h))
\]  

(17)

\[
g^w_c = \sigma(\text{conv}1 \times 1(f^w))
\]  

(18)

The height and width of the image, respectively. Binary adaptive average pooling in the horizontal and vertical directions (AdaptiveAvgPool2d), By adaptively extracting the average value of the features from the feature space to smooth and eliminate the noise in the LR image, reduce the dimensionality, and retain important information, using two spatial pooling kernels \( (H, 1) \) and \( (1, W) \) Encode each channel along the abscissa and ordinate respectively, the height of the input \( X \) is \( h \), and the output \( Y \) of the c-th channel is expressed as:

\[
Y^h_c = \frac{1}{W} \sum_{0 \leq i < W} X_c(h, i)
\]  

(14)

Similarly, the width of the input \( X \) is \( w \), and the output \( Y \) of the c-th channel is expressed as:

\[
Y^w_c = \frac{1}{H} \sum_{0 \leq i < H} X_c(j, w)
\]  

(15)

After the adaptive pooling layer, a pair of feature maps that are sensitive to location information are obtained \( Y^h \), \( Y^w \). Both have the same number of channels as the input \( X \), and the shapes and sizes are \( \text{nf}, 64, 1 \) and \( \text{nf}, 1, 64 \), and then \( Y^h \), \( Y^w \) Perform a merge operation and obtain \( f \) through a \( 1 \times 1 \) convolution operation:

\[
f = \delta(\text{conv}1 \times 1(\text{cat}(Y^h, Y^w)))
\]  

(16)

FIGURE 5. Coordinate attention.
module. The input feature $X_{n-1}$ will pass through a $1 \times 1$ convolutional layer $f^1$ before passing through two branches, $f^2$, its main function is to separate channels, achieve dimensionality reduction and reduce the amount of calculation, the expression is:

$$H_1 = f^1(X_{n-1}) \quad (20)$$

$$H_2 = f^2(X_{n-1}) \quad (21)$$

$H_1$ and $H_2$ in equations (20) and (21) respectively represent the output of the two branches after $1 \times 1$ convolution. After $H_1$ and $H_2$ go through the residual module and coordinate attention respectively, the output features are fused, and finally go through a $1 \times 1$ convolutional layer $f^3$. At the same time, the method of skip link is adopted to alleviate the problem of gradient disappearance, and the final $X_n$, as shown in formula (22):

$$X_n = f^3([RB(H_1), CA(H_2)]) + X_{n-1} \quad (22)$$

RB() in equation (22) represents the output of the residual module, CA() represents the output of coordinate attention, and [] represents the joining of two output features.

**I. LOSS FUNCTION**

This paper uses the L1 loss function to optimize the network. Compared with the MSE loss, the L1 loss penalizes the relative error less, and the effect on the image texture detail reconstruction will be better. Given training set $\{(I^{LR}_j, I^{HR}_j)\}_{j=1}^M$ among them $I^{LR}_j$, $I^{HR}_j$ Represents low-resolution images and high-resolution images respectively, M represents the size of the batch training set, and the parameter set is $\theta$. The loss function expression of the algorithm in this paper is shown in equation (23), and the L1 loss function is used for constraints:

$$||L(\theta)||_1 = \frac{1}{M} \sum_{j=1}^{M} |f_{ARASR}(I^{LR}_j) - I^{HR}_j| \quad (23)$$

In formula (23), $f_{ARASR}$ represents the network model of this paper.

**III. EXPERIMENT**

**A. DATA SAT**

In order to test the feasibility of this model, this paper trains and tests the model on the recently proposed RefSR data set CUFED5 [24]–[26]. The training set contains 11,842 pairs of pictures, and each pair is composed of an input image and a reference image. The test set contains 126 groups of pictures, each group is composed of an HR image and four reference images. In order to fully train the network, this article uses three methods to enhance the training data: 1) Rotate the image by $90^\circ$, $180^\circ$ and $270^\circ$; 2) Flip the image horizontally and vertically; 3) Process the LR image to $40 \times 40$ pixels, the Ref image is processed into $160 \times 160$ pixels.

In order to evaluate the generalization ability of SRCA on the CUFED5 data set, this paper conducted model tests on the CUFED5 [27], Sun80 [28], Urban100 [29], and Manga109 [18] data sets. The Sun80 data set contains 80 natural images, and each image contains multiple reference images to pair with it; the Urban100 data set contains 100 architectural images without reference. Due to the high similarity of the architectural images, the LR image is set As a reference image for texture search and transfer; Manga109 contains 109 comic images without reference images, so HR images are randomly selected as reference images in this dataset.

This paper uses 800 high-quality RGB training images in the public data set DIV2K [30] as the training set, Set5 [17], Set14 [18], BSD100 [19], Urban100 [20] four data sets as the
TABLE 2. PSNR / SSIM comparison between different algorithms on four different datasets.

| Method | Algorithm | CUFED5 | Sun80 | Urban100 | Manga109 |
|--------|-----------|--------|-------|----------|----------|
|        | PSNR/dB   | SSIM   | PSNR/dB | SSIM | PSNR/dB | SSIM | PSNR/dB | SSIM |
| SISR   | SRCNN[4]  | 24.21  | 0.545  | 27.14  | 0.581  | 23.29  | 0.538  | 26    | 0.65  |
|        | MDSR[19]  | 24.81  | 0.577  | 27.4   | 0.592  | 24.39  | 0.583  | 27.81 | 0.691 |
|        | RDN[20]   | 24.83  | 0.569  | 28.51  | 0.606  | 24.26  | 0.568  | 28.12 | 0.694 |
|        | RCAN[21]  | 24.94  | 0.569  | 28.74  | 0.61   | 24.3   | 0.568  | 28.26 | 0.695 |
|        | SRGAN[22] | 23.28  | 0.502  | 25.64  | 0.525  | 22.95  | 0.529  | 24    | 0.602 |
|        | ENet[23]  | 23.12  | 0.495  | 25.12  | 0.502  | 22.51  | 0.511  | 24.13 | 0.602 |
|        | ESRGAN[24] | 20.78  | 0.433  | 23.06  | 0.451  | 19.79  | 0.42   | 22.41 | 0.597 |
|        | RSRGAN[25] | 21.19  | 0.435  | 24.48  | 0.467  | 20.35  | 0.424  | 23.92 | 0.603 |
| RefSR  | CrossNet[26] | 24.36  | 0.564  | 27.4   | 0.593  | 23.99  | 0.564  | 22.24 | 0.541 |
|        | SRNTT-rec[10] | 25.12** | 0.584 | 27.42  | 0.593  | 24.38  | 0.583  | 27.83 | 0.685 |
|        | SRNTT[10]  | 24.49  | 0.564  | 26.47  | 0.556  | 23.97  | 0.574  | 26.42 | 0.662 |
|        | TTSR-rec[11] | 25.97  | 0.804** | 30.02* | 0.814* | 25.87** | 0.784** | 30.09** | 0.907** |
|        | TTSR[11]   | 24.41  | 0.765  | 28.59  | 0.774  | 24.62  | 0.747  | 28.70 | 0.886 |
|        | SRCA-rec   | 25.53  | 0.807* | 29.93** | 0.813** | 25.93* | 0.786* | 30.25* | 0.909* |
|        | SRCA      | 25.87  | 0.771  | 28.75  | 0.777  | 25.04  | 0.757  | 29.33 | 0.891 |

In this section, quantitative and qualitative evaluations of SRCA are carried out. The quantitative evaluation is shown in Table 2. Through comparison, it can be concluded that SRCA has the best performance on the Urban100 and Manga109 datasets. On CUFED5 and Sun80, SRCA can achieve performance comparable to the latest models.
model in Figure 9(h) in this paper has a high degree of realism in the line details, and the edge contours are clearer and natural.

The pictures restored by SRCA also have higher visual quality, and can be more accurate in Ref image texture search and transfer. When the image similarity between Ref image and LR image is not high, the SRCA model can still search for the corresponding texture and transfer to the SR image, so that the restored image is more vivid, as shown in Figure 10 and Figure 11. In Fig. 10, the reconstructed window lines in Fig. 10(d), Fig. 10(f) and Fig. 10(g) are blurred, and the reconstructed image in Fig. 10(e) is visually improved but the enlarged window glass edge not clear. The model reconstructed in this paper is shown in Figure 10(h). The edges of the window glass are clear and enjoyable. In terms of face recovery, as shown in Figure 11.

In this paper, the model training performance is compared with TTSR, and the experimental results are shown in Figure 12. Figure 8 shows the comparison of the PSNR
FIGURE 11. Reconstructed result comparison on Manga109 by different super-resolution models (×4).

FIGURE 12. SRCA and TTSR training comparison chart and SSIM curves on the ×4 CUFED5 verification set during the 200 rounds of training for the two networks. It can be seen that both networks show a growth trend, but the overall growth of SRCA is above TTSR, and that of SRCA The average PNSR and average SSIM are approximately 0.12dB and 0.0035 higher than TTSR. This proves that SRCA has better performance under the same number of training.

C. OBJECTIVE EVALUATION RESULTS AND ANALYSIS

1) PERFORMANCE EVALUATION OF IMPROVED MODULES IN THE ALGORITHM

In order to verify the effectiveness of the adaptive residual attention information extraction module in the algorithm of this paper and the feasibility of improving the coordinate attention network to compare other attention, in order to ensure the fairness of the experiment, the EDSR [12] algorithm is used as the benchmark model. The number of modules is 16. “Baseline” means the EDSR benchmark model, which is composed of 16 residual modules; “Baseline+CA” means the residual module is cascaded with the channel attention module; “Baseline+CoordA” means the residual module is cascaded with the coordinate attention module, “ARASR” means that the residual module and the coordinate attention module are connected in parallel, which is also the network structure used in this article. In order to train fairness, the experiment has trained 5 × 105 iteration rounds. Table 3 shows the PSNR values of different module combinations on the Set5 data set at 4 times magnification. From the table, it can be found that the adaptive residual attention information extraction module “ARASR” proposed in this paper

| method            | method        | PSNR  |
|-------------------|---------------|-------|
| Baseline          | 1,517K        | 25.98 |
| Baseline+CA       | 1,526K        | 26.02 |
| Baseline+CoordA   | 1,544K        | 26.03 |
| ARASR (Ours)      | 922K          | 26.09 |

FIGURE 13. Effect of the number of ARB modules on model performance.
has obvious parameters compared with other algorithms. Reduced, the PSNR value also has a certain increase, which proves the effectiveness of the adaptive residual attention information extraction module.

2) THE INFLUENCE OF THE NUMBER OF ARB MODULES IN THE ALGORITHM ON THE PERFORMANCE OF THE MODEL

In order to explore the impact of model network depth on model performance, 16, 18, 20, and 22 ARB modules were selected for the experiment. Figure 13 shows the PSNR values of models with different ARB modules on the Set14 data set. It can be seen that when the network is not deep enough, the PSNR value is low. The reason may be that the network does not extract sufficiently deep features, which makes the model unable to learn the high-frequency details of the image. After the network is deepened to a certain level, continuing to deepen will not increase the PSNR value. When the number of ARB modules is 20, the PSNR value is the highest, and the model’s ability to extract features tends to be saturated. Therefore, the number of ARB modules is selected as 20 in this paper.

3) ALGORITHM OVERALL PERFORMANCE EVALUATION

In order to evaluate the performance of the algorithm in this paper, the results obtained by the algorithm in this paper are compared with SRCNN [5], FSRCNN [6], VDSR [8], DRCN [9], LapSRN [21], DRRN [10], CARN [22], MemNet [11], SRMDNF [23] and EDSR-baseline [19] and other algorithms are compared with the experimental results, and the PSNR, SSIM values and parameter sizes under three magnifications of ×2, ×3, and ×4 are compared.

Table 4 shows the model parameters and objective quality evaluation index values under different magnifications. It can be seen that the PSNR and SSIM values of the algorithm ARASR in this paper have a certain increase compared with other algorithms under the three magnifications, especially when the magnification is 2, the PSNR values of the four data sets of the benchmark model EDSR are increased respectively. After reading 0.03dB, 0.14dB, 0.03dB, 0.26dB, we can see that the model in this paper performs better in the Urban100 dataset of complex scenes. At the same time, the parameters are reduced by 818K and 596K respectively compared with CARN and the benchmark model EDSR. In order to explore the influence of the width of the network on the performance of the model, this paper sets up two sets of experiments. The number of modules in the ARB network of the ARASR-s algorithm is 20, and the number of input channels is 40; the number of modules in the ARB network of the algorithm in this paper (ARASR) is 20, the number of input channels is 64. It can be seen from Table 3 that the width of the network has a huge impact on the amount of model parameters. When the magnification is 4, ARASR-s reduces the amount of parameters by 60% compared with ARASR, and the model performance also declines.

Figure 14 shows the corresponding relationship between the model parameters of each algorithm on the Set14 data set and the PSNR value at 2 times magnification. The text model has a good trade-off between performance and the amount of parameters.

4) SUBJECTIVE EFFECT DISPLAY AND ANALYSIS

Figure 15 shows the comparison of the model generation effect diagram when the magnification is 4, you can see that the zebra stripe image obtained by the bicubic interpolation is blurred as a whole, and the reconstruction effect diagram of SRCNN and other algorithms can clearly see the phenomenon of local blur, and this article The algorithm restores high-frequency details to the greatest extent for this irregular zebra stripe image and greatly reduces the ringing effect.
| method          | scale | Params | Set5 PSNR/SSIM          | Set14 PSNR/SSIM          | BSD100 PSNR/SSIM            | Urban100 PSNR/SSIM           |
|-----------------|-------|--------|-------------------------|-------------------------|----------------------------|----------------------------|
| Bicubic         | -     | 32.54/0.8099 | 29.12/0.8488 | 28.44/0.8231 | 25.76/0.8403 |                        |
| SRCNN           | 57K   | 35.54/0.8342 | 31.33/0.8867 | 30.24/0.8679 | 28.38/0.8946 |                        |
| FSRCNN          | 13K   | 35.54/0.8342 | 31.33/0.8867 | 30.41/0.872 | 28.76/0.902 |                        |
| VDSR            | 665K  | 35.54/0.8342 | 31.91/0.8924 | 30.78/0.8742 | 29.63/0.9133 |                        |
| DR CN           | 1,774K | 35.54/0.8342 | 31.91/0.8924 | 30.73/0.8742 | 29.63/0.9133 |                        |
| LapSRN          | ×2    | 813K   | 35.54/0.8342 | 31.91/0.8924 | 30.68/0.8752 | 29.29/0.9103 |                        |
| DRRN            |       | 297K   | 36.62/0.8391 | 32.11/0.8936 | 30.93/0.8773 | 30.11/0.9188 |                        |
| MemNet          |       | 678K   | 36.66/0.8397 | 32.16/0.8942 | 30.96/0.8787 | 30.19/0.9195 |                        |
| EDSR-baseline   |       | 1,518K | 36.87/0.8404 | 32.45/0.8975 | 31.04/0.8794 | 30.86/0.9272 |                        |
| SRMDNF          |       | 1,552K | 36.87/0.8404 | 32.22/0.8959 | 30.93/0.8785 | 30.21/0.8204 |                        |
| CARN            |       | 1,592K | 36.87/0.8404 | 32.42/0.8966 | 30.97/0.8778 | 30.80/0.9256 |                        |
| ARASR-S         |       | 365K   | 36.87/0.8404 | 32.37/0.8971 | 31.07/0.8971 | 30.80/0.9266 |                        |
| ARASR           |       | 922K   | 36.87/0.8404 | 32.59/0.8989 | 31.07/0.888 | 31.12/0.9297 |                        |
| Bicubic         | ×3    | -      | 30.39/0.8682 | 27.55/0.7742 | 27.21/0.7385 | 24.46/0.7349 |                        |
| SRCNN           |       | 57K    | 32.75/0.9090 | 29.30/0.8215 | 28.41/0.7863 | 26.24/0.7989 |                        |
| FSRCNN          |       | 13K    | 33.18/0.9140 | 29.37/0.8240 | 28.53/0.7910 | 26.43/0.8080 |                        |
| VDSR            |       | 665K   | 33.66/0.9213 | 29.77/0.8314 | 28.82/0.7976 | 27.14/0.8279 |                        |
| DR CN           |       | 1,774K | 33.82/0.9226 | 29.76/0.8311 | 28.80/0.7963 | 27.15/0.8276 |                        |
| LapSRN          |       | 813K   | 33.81/0.9220 | 29.79/0.8325 | 28.82/0.7980 | 27.07/0.8275 |                        |
| DRRN            |       | 297K   | 34.03/0.9244 | 29.96/0.8349 | 28.96/0.8004 | 27.53/0.8378 |                        |
| MemNet          |       | 678K   | 34.09/0.9248 | 30.00/0.8350 | 28.96/0.8001 | 27.56/0.8376 |                        |
| EDSR-baseline   |       | 1,518K | 34.37/0.9270 | 30.28/0.8417 | 29.09/0.8052 | 28.15/0.8527 |                        |
| SRMDNF          |       | 1,552K | 34.12/0.9254 | 30.04/0.8382 | 28.97/0.8025 | 27.57/0.8398 |                        |
| CARN            |       | 1,592K | 34.29/0.9255 | 30.29/0.8407 | 29.06/0.8034 | 28.06/0.8493 |                        |
| ARASR-S         |       | 365K   | 34.24/0.9262 | 30.27/0.8411 | 29.03/0.8038 | 27.98/0.8488 |                        |
| ARASR           |       | 922K   | 34.37/0.9270 | 30.39/0.8433 | 29.11/0.8056 | 28.29/0.8552 |                        |
| Bicubic         | ×4    | -      | 27.30/0.6940 | 26.00/0.7027 | 25.96/0.6675 | 23.14/0.6577 |                        |
| SRCNN           |       | 57K    | 29.36/0.7428 | 27.50/0.7513 | 26.90/0.7101 | 24.52/0.7221 |                        |
| FSRCNN          |       | 13K    | 29.60/0.746 | 27.61/0.7550 | 26.98/0.7150 | 24.62/0.7280 |                        |
| VDSR            |       | 665K   | 30.23/0.7638 | 28.01/0.7674 | 27.29/0.7251 | 25.18/0.7524 |                        |
| DR CN           |       | 1,774K | 30.41/0.7654 | 28.02/0.7670 | 27.23/0.7233 | 25.14/0.7510 |                        |
| LapSRN          |       | 813K   | 30.42/0.7652 | 28.09/0.7700 | 27.32/0.7275 | 25.21/0.7562 |                        |
| DRRN            |       | 297K   | 30.56/0.7688 | 28.21/0.7720 | 27.38/0.7284 | 25.44/0.7638 |                        |
| MemNet          |       | 678K   | 30.62/0.7693 | 28.26/0.7723 | 27.40/0.7281 | 25.50/0.7630 |                        |
| EDSR-baseline   |       | 1,518K | 30.97/0.7738 | 28.58/0.7813 | 27.57/0.7357 | 26.04/0.7849 |                        |
| SRMDNF          |       | 1,552K | 30.84/0.7725 | 28.35/0.7787 | 27.49/0.7337 | 25.68/0.7731 |                        |
| CARN            |       | 1,592K | 31.01/0.7737 | 28.60/0.7806 | 27.58/0.7349 | 26.07/0.7837 |                        |
| ARASR-S         |       | 365K   | 30.94/0.7732 | 28.50/0.7790 | 27.51/0.7342 | 25.93/0.7807 |                        |
| ARASR           |       | 922K   | 31.08/0.7751 | 28.61/0.7821 | 27.58/0.7367 | 26.15/0.7876 |                        |
Figure 16 and Figure 17 show the reconstruction renderings of the two images “img_012” and “img_046” in the Urban100 dataset at a magnification of 4. From “img_046”, it can be seen that the algorithm in this paper can better restore the edges and textures of the image. The details are closer to the real image than other algorithms. From “img_012”, it can be seen that other algorithms will produce incorrect reconstruction of the texture direction in the image to a certain extent, and at the same time will cause the checkerboard effect. The algorithm in this paper can reconstruct the correct texture details more closely to the real image.

IV. IN CONCLUSION

In this paper, an adaptive residual attention network is proposed, which can achieve lightweight and accurate single image super-resolution. The experimental results show that the proposed method can reduce the number of parameters and achieve the best objective evaluation index compared
with other algorithms; in terms of visual quality, the proposed method can better deal with fuzzy artifacts and reconstruct texture details. At the same time, it can be seen that the network width has a great impact on the number of parameters and performance of the model.

REFERENCES

[1] W. Heming and S. Wenzhong, “Super-resolution reconstruction of iris image based on dual-path residual network,” J. Shanghai Electr. Power Univ., vol. 37, no. 3, pp. 289–294, 2021.

[2] Y. Tao, S. J. Conway, J.-P. Müller, A. R. D. Putri, N. Thomas, and G. Cremonese, “Single image super-resolution restoration of TGO CaSIS colour images: Demonstration with perseverance rover landing site and Mars science targets,” Remote Sens., vol. 13, no. 9, p. 1777, May 2021.

[3] W. Yanfan, Z. Junlin, L. Jianli, N. Feiyang, and Z. Sheng, “Application of low-dose multi-slice spiral CT high-resolution reconstruction in the screening of new coronavirus pneumonia,” Chin. J. Med. Phys., vol. 38, no. 4, pp. 456–460, 2021.

[4] Z. Tibo, L. Lin, Y. Kai, Z. Yu, and L. Jinlong, “Image reconstruction algorithm based on improved super-resolution generative confrontation network,” Prog. Laser Optoelectron., vol. 58, no. 8, pp. 93–103, 2021.

[5] Z. Yan, L. Xuanming, L. Guorui, L. Shudong, and S. Yemei, “Remote sensing image super-resolution reconstruction algorithm based on multi-path feature fusion,” Remote Sens. Inf., vol. 36, no. 2, pp. 46–53, 2021.

[6] X. Yang, T. Xie, L. Liu, and D. Zhou, “Image super-resolution reconstruction based on improved dirac residual network,” Multidimensional Syst. Signal Process., vol. 32, no. 4, pp. 1065–1082, Oct. 2021.

[7] S. Jun, “Dual-path image super-resolution reconstruction network based on adaptive reconstruction,” Electron. Technol. Softw. Eng., vol. 6, no. 7, pp. 134–135, Oct. 2021.

[8] C. Songsheng and P. Jinshan, “Video super-resolution method based on deep learning feature matching,” Comput. Sci., vol. 48, no. 7, pp. 184–189, 2021.

[9] L. Jiaqi, L. Huanyu, and L. Junbao, “Experiment of super-resolution reconstruction of medical MRI based on deep learning network,” Chin. J. Med. Phys., vol. 38, no. 1, pp. 30–39, 2021.

[10] F. Yonggong and Y. Chunting, “LatSRN image super-resolution reconstruction combined with error feedback,” Electron. Des. Eng., vol. 29, no. 2, pp. 185–189, 2021.

[11] Q. Xin and W. Yan, “An improved SRGAN image super-resolution reconstruction algorithm,” Comput. Times, vol. 1, no. 4, pp. 72–75, Sep. 2021.

[12] S. Rui, Z. Han, C. Zhikang, and Z. Xudong, “Infrared image super-resolution reconstruction algorithm based on channel attention and transfer learning,” Photoelectr. Eng., vol. 48, no. 1, pp. 4–16, 2021.

[13] W. Wei, H. Tao, L. Xinwei, S. Siwan, J. Xiaoming, and L. Junyuan, “Research on super-resolution reconstruction of white blood cell images,” Comput. Sci., vol. 48, no. 4, pp. 164–168, 2021.

[14] L.-H. Fu, X.-W. Sun, Y. Zhao, R.-J. Chen, H. Chen, and R. Zhao, “Video super-resolution reconstruction method based on deep back projection and motion feature fusion,” Multimedia Tools Appl., vol. 80, no. 8, pp. 11423–11441, Mar. 2021.

[15] Z. Dong, Research on Face Super-Resolution Method Preserved by Manifold Structure. Nanjing, China: Nanjing Univ. Posts and Telecommunications, 2020.

[16] X. Jingjing, Research on the Phantom Structure and Detection Depth Method of Very Low-Resolution Face Images. Nanjing, China: Nanjing Univ. Posts and Telecommunications, 2020.

[17] L. Yinghua, Z. Chunna, and J. Murong, “An image super-resolution reconstruction algorithm combining self-attention and gating mechanism,” Comput. Sci. Appl., vol. 10, no. 12, pp. 2323–2330, 2020.

[18] L. Q. Ming, J. R. Sheng, L. Y. Bo, S. H. Bin, Y. J. Zhi, and S. H. Mei, “Infrared image super-resolution reconstruction by using generative adversarial network with an attention mechanism,” Appl. Intell., vol. 51, no. 4, 2020.

[19] F. Weijie, Research on Single Image Super-Resolution Reconstruction Based on Joint Training Regressor. Nanning, China: Guangxi Univ., 2020.

[20] S. Yujian, L. Luying, and Y. Wenyuan, “Based on Laplace super-resolution printed circuit board defect detection method,” Chin. Stereol. Image Anal., vol. 25, no. 3, pp. 295–303, 2020.

[21] J. Lingjia and Z. Yue, “An image super-resolution algorithm based on prediction kernel pixel-by-pixel reconstruction,” Chin. Stereol. Image Anal., vol. 25, no. 3, pp. 304–311, 2020.

[22] P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M. E. Celebi, and J. Yang, “Image synthesis with adversarial networks: A comprehensive survey and case studies,” Inf. Fusion, vol. 72, pp. 126–146, Aug. 2021.

[23] X. Nie, K. Huang, J. Deasy, A. Rimner, and G. Li, “Enhanced super-resolution reconstruction of T1w time-resolved 4DMRI in low-contrast tissue using 2-step hybrid deformable image registration,” J. Appl. Clin. Med. Phys., vol. 21, no. 10, pp. 25–39, Oct. 2020.

[24] L. Renu, Z. Li, M. Chenhui, L. Xuan, and Z. Hao, “Improved super-resolution reconstruction of infrared images based on deep back-projectation networks,” Inf. Technol., vol. 42, no. 9, pp. 873–879, Sep. 2020.

[25] Z. Yongmei, H. Ruimin, M. Jianzhe, and H. Lei, “High spatiotemporal fusion method of remote sensing based on deep learning and super-resolution reconstruction,” Comput. Eng. Sci., vol. 42, no. 9, pp. 1578–1586, 2020.

[26] Z. Chuanchu, W. Lucky, and L. Ming, “Single-frame face image super-resolution algorithm based on WGAN,” Comput. Technol. Develop., vol. 30, no. 9, pp. 29–35, 2020.

[27] S. Yongxiang, J. Bin, H. Yongxu, Y. Guisheng, L. Qingwu, and Z. Zhiliang, “CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 589–597.

[28] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1091–1100.

[29] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, “Crowd counting via multi-column convolutional neural network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 5245–5254.

[30] R. Liang, Y. Zhu, and H. Wang, “Counting crowd flow based on feature points,” Neurocomputing, vol. 153, no. 3, pp. 377–384, Sep. 2014.

[31] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 589–597.