ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

Research on Image Super-resolution Reconstruction Technology Based on Three-Layer V-network

Shuiping Ni[1, a], Pengkun Li[1, b, *]

[1] College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China

[a]nishuiping@hpu.edu.cn, [b]lipk2021@163.com

Abstract

In order to solve the problem of insufficient extraction of low-level information in U-Net network, a three-layer V-network is designed in this paper, and the feature extraction and recovery are performed by three V-networks with different depths to ensure that the features at all levels of the image can be extracted. The dual-channel attention module in the model flexibly controls the channel attention coefficients of the dual branches by the second-order channel attention mechanism, and uses the pixel attention module to adjust the coefficients of different pixels on the same channel to make the recovered images more discriminative. The experimental results show that the three-layer V-network model performs better on the "Urban100" test set than other test sets, and extracts the low-level edge information better than other test sets.

Keywords

U-Net network; Channel attention mechanism; Super-resolution reconstruction.

1. INTRODUCTION

With the increasing requirements for image quality in various industries, how to obtain high-quality images has become a research hotspot for researchers at home and abroad. The (single-image super-resolution reconstruction) technique can recover corresponding high-resolution images from low-resolution images[1], which has been widely used in medical, aviation, security and other fields[2]

The traditional interpolation-based[3] and reconstruction-based[4] SISR algorithms are relatively simple and computationally small, but the reconstruction effect is poor and suffers from severe blurring as well as loss of high-frequency information. The SRCNN[5] model proposed by Dong et al. introduces deep learning into the field of SISR, and the image reconstruction effect is significantly better than the traditional methods. However, because it upsamples the input, it not only leads to the reduction of high-frequency information of the image, but also the computational effort increases significantly. To address this problem, Dong et al. then proposed an improved model FSRCNN[6].FSRCNN takes the original image input to the network and upsamples the image output from the network, which improves the model in terms of computation and training speed. The VDSR[7]model proposed by Kim et al. combines the residual structure with convolutional neural networks for the first time and applies it to the image SR domain. Due to the deepening of the network depth, the learning ability of the VDSR model is significantly enhanced compared to SRCNN and the image reconstruction is better. The ESPCN[8] model proposed by Shi et al. uses a new upsampling method, subpixel convolution. The model upsamples the output feature maps at the end, preserving more texture regions in the low-resolution space. The DRCN[9] model proposed by Kim et al. applies recursive neural

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

network structure to super-resolution reconstruction for the first time. The gradient disappearance is effectively avoided by using recursive supervision. The SRDenseNet[10] proposed by Tong et al. introduces a dense block structure and achieves good results. DenseNet[11] feeds the features of each layer to all subsequent layers in the dense block, which enables the network to mitigate the gradient disappearance problem and enhance feature propagation. The SRGAN[12] proposed by LEDIG et al. The LapSRN[13] model proposed by Lai et al. uses stepwise upsampling to predict the residuals one level at a time to make the network propagate faster. The RCAN[14] model proposed by Zhang et al. introduces a channel attention mechanism to make the network propagate faster by adaptively learning interdependencies between channels enables the network to focus on learning important channel features, thus improving the performance of the network.

However, most of the existing reconstruction models cannot extract the multi-level information of the image at the same time. To optimize this problem, this paper proposes a three-layer V-shaped network structure based on U-Net network. The content of our work is as follows:

1) A three-layer V-network structure is proposed, and this network can extract information to all levels of the image simultaneously.

2) The feature extraction module uses a two-branch structure with different convolutional kernel sizes to extract both high and low frequency information from the image.

3) The pixel attention module is introduced after the channel attention module, which adjusts the coefficients of the different pixels on the channel so that the details of the recovered image are more visible.

2. MODEL STRUCTURE

This model reconstructs the high-resolution image by first compressing and then expanding it while keeping the model lightweight, and its structure is shown in Figure 1. The model mainly contains three layers of V-shaped networks for extracting features at different levels of the image, the first layer contains a Dense Down Projection Network (DDPN) and a Dense Up Projection Network (DUPN), the second layer contains two DDPNs and two The third layer contains three DDPNs and three DUPNs, and finally the three feature maps recovered from the three layers are element-summed and upsampled, and then the original low-resolution image that has been upsampled by double triple interpolation is element-summed again and reconstructed by convolution to obtain the final HR image.

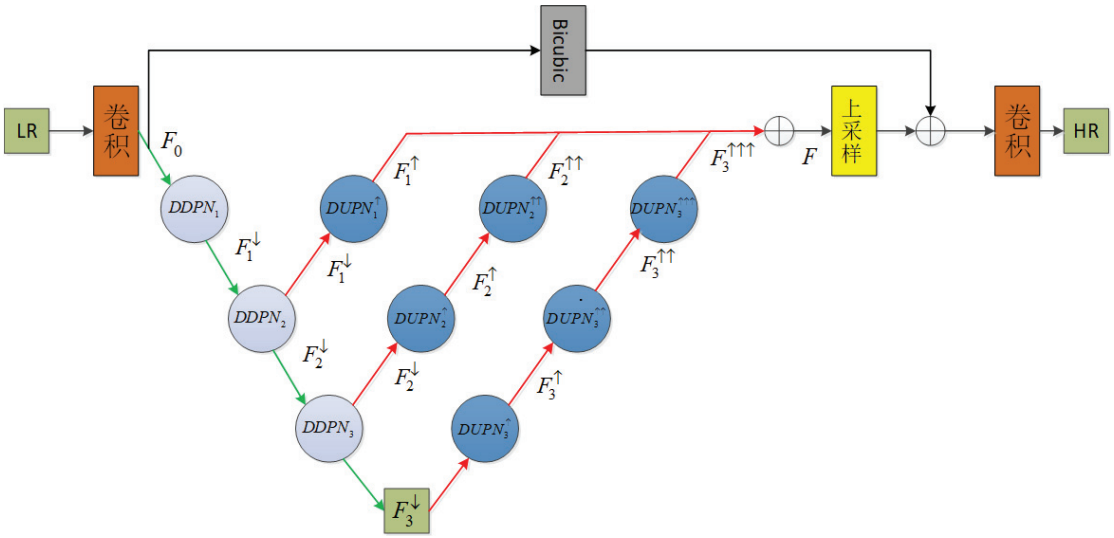

Figure 1. Three-layer V-shaped network structure diagram

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

Firstly, the input low-resolution image F_{LR} is convolved with a 3×3 layer to boost the image features from 3 to 64 dimensions, while the initial features of the image are extracted, and the feature map is denoted as F_0, which can be expressed as:

$$F_0 = \sigma(\omega \times F_{LR} + b) \tag{1}$$

Where, σ is the ReLU activation function, ω denotes the weight, and b is its corresponding bias parameter.

After the initial feature extraction, the compression and expansion phase of the network begins. The feature map size is halved and the number of channels is doubled for each DDPN module that the feature map passes through, and conversely, the feature map size is doubled and the number of channels is halved for each DUPN module that the feature map passes through.

For example, F_0 is compressed by $DDPN_1$ to obtain the feature map $F_1^{\downarrow}$, where the number of channels of $F_1^{\downarrow}$ is 128 and the image size is half of the original one, while $F_1^{\downarrow}$ is expanded by $DUPN_1^{\uparrow}$ of the first layer V network to obtain the final output $F_1^{\uparrow}$ of the first layer V network, where the feature map size is twice of $F_1^{\downarrow}$ and the number of channels is reduced to 64. The specific representation is as follows, where $f_{DDPN}(\bullet)$ represents the compression operation and $f_{DUPN}(\bullet)$ represents the expansion operation.

$$F_1^{\downarrow} = f_{DDPN_1}(F_0) \tag{2}$$

$$F_1^{\uparrow} = f_{DUPN_1^{\uparrow}}(F_1^{\uparrow}) = f_{DUPN_1^{\uparrow}}(f_{DDPN_1}(F_0)) \tag{3}$$

At the same time, $F_1^{\downarrow}$ is compressed again by $DDPN_2$ to obtain the feature map $F_2^{\downarrow}$ with 256 channels, and similar to $F_1^{\downarrow}$, $F_2^{\downarrow}$ is expanded by $DUPN_2^{\uparrow}$ and $DUPN_2^{\uparrow\uparrow}$ of the second layer type network to obtain the final output $F_2^{\uparrow\uparrow}$ of the second layer type network, while $F_2^{\downarrow}$ continues down through the last compression module $DDPN_3$ and reaches the bottom of the network to obtain the feature map $F_3^{\downarrow}$ with 512 channels, which can be expressed by equations (4) to (7):

$$F_2^{\downarrow} = f_{DDPN_2}(f_{DDPN_1}(F_0)) \tag{4}$$

$$F_2^{\uparrow} = f_{DUPN_2^{\uparrow}}(f_{DDPN_2}(f_{DDPN_1}(F_0))) \tag{5}$$

$$F_2^{\uparrow\uparrow} = f_{DUPN_2^{\uparrow\uparrow}}(f_{DUPN_2^{\uparrow}}(f_{DDPN_2}(f_{DDPN_1}(F_0)))) \tag{6}$$

$$F_3^{\downarrow} = f_{DDPN_3}(f_{DDPN_2}(f_{DDPN_1}(F_0))) \tag{7}$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

Then $F_3^{\downarrow}$ is successively expanded by $DUPN_3^{\uparrow}$, $DUPN_3^{\uparrow\uparrow}$, and $DUPN_3^{\uparrow\uparrow\uparrow}$ to obtain the output $F_3^{\uparrow\uparrow\uparrow}$ of the third layer V-network, and finally $F_1^{\uparrow}$, $F_2^{\uparrow\uparrow}$, and $F_3^{\uparrow\uparrow\uparrow}$ are summed to obtain the output F of the three-layer V-network. as follows:

$$F_3^{\uparrow} = f_{DUPN_3^{\uparrow}}(f_{DDPN_3}(f_{DDPN_2}(f_{DDPN_1}(F_0)))) \tag{8}$$

$$F_3^{\uparrow\uparrow} = f_{DUPN_3^{\uparrow\uparrow}}(f_{DUPN_3^{\uparrow}}(f_{DDPN_3}(f_{DDPN_2}(f_{DDPN_1}(F_0))))) \tag{9}$$

$$F_3^{\uparrow\uparrow\uparrow} = f_{DUPN_3^{\uparrow\uparrow\uparrow}}(f_{DUPN_3^{\uparrow\uparrow}}(f_{DUPN_3^{\uparrow}}(f_{DDPN_3}(f_{DDPN_2}(f_{DDPN_1}(F_0)))))) \tag{10}$$

$$F = F_1^{\uparrow} + F_2^{\uparrow\uparrow} + F_3^{\uparrow\uparrow\uparrow} \tag{11}$$

2.1. DDPN and DUPN

From the above introduction, it has been known that DDPN and DUPN represent the compression and expansion modules of the three-layer V-network, respectively. Their internal structures are almost the same, and the only difference is that the convolutional layer is used in DDPN to compress the image size and expand the channels, and the deconvolutional layer is used in DUPN to recover the image size and compress the channels. The specific structures are shown in Figure 2 and Figure 3.

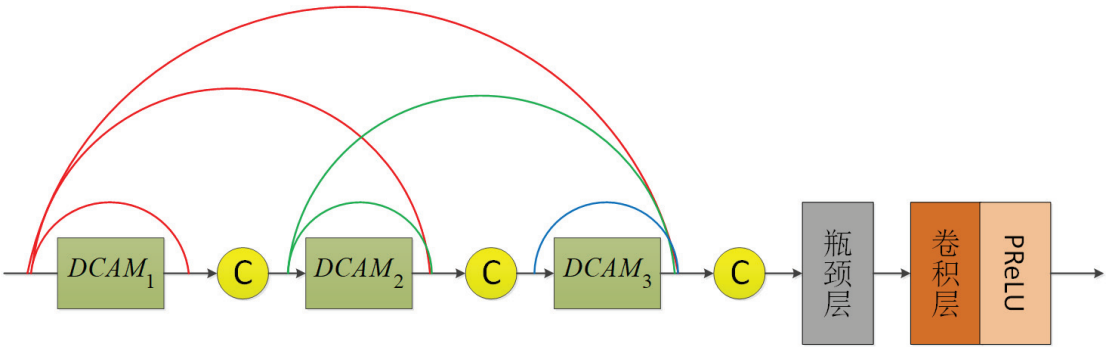

Figure 2 DDPN internal structure diagram

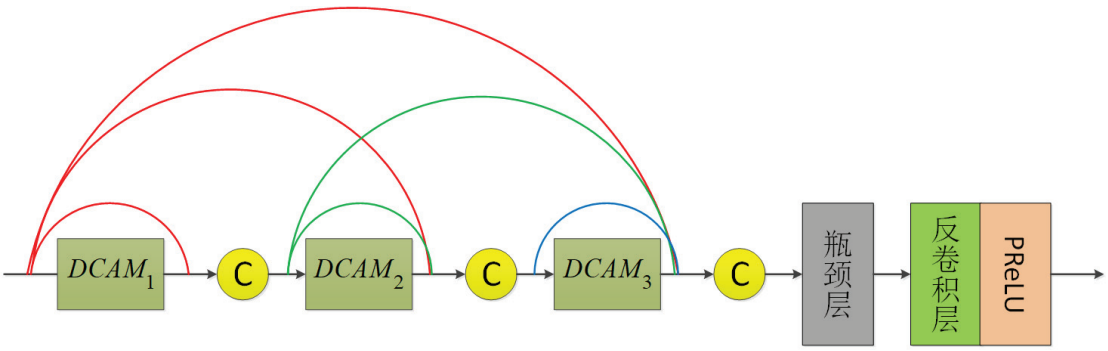

Figure 3. DUPN internal structure diagram

In Figure 2, the model extracts the features of the image step by step through three Dual Channel Attention Modules (DCAM) $DCAM_1$, $DCAM_2$, and $DCAM_3$, and the three DCAMs are

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

densely connected to each other, which effectively mitigates the gradient disappearance and enhances the utilization of shallow features. A "Concat" connection is added after each DCAM module in order to stitch all the outputs of the module, and then a bottleneck layer is used before the convolution layer to downscale the feature map to the same dimension as the input. The final output is obtained by compressing the feature map size to half of the original size and doubling the number of channels through the convolution layer and the "PReLU activation function". In Figure 3, the feature extraction process is the same as that in Figure 2, but after the bottleneck layer, the feature map is upsampled by the deconvolution layer and the "PReLU activation function", and the feature map is expanded to twice the original size and the number of channels is compressed to half the original size to get the final output.

Specifically, suppose Figure 2 shows the internal structure of $DDPN_1$, then its input and output are F_0 and $F_1^{\downarrow}$, respectively. F_0 is extracted by $DCAM_1$ to get the feature map F_{DCAM_1}, at this time F_{DCAM_1} is still a 64-dimensional feature map, because DCAM only extracts features without changing the size and number of channels of the feature map. Then F_{DCAM_1} is stitched with F_0 to get $\{F_0, F_{DCAM_1}\}$ ({,} means stitching operation), the dimension of the stitched feature map is 128, the stitched feature map $\{F_0, F_{DCAM_1}\}$ is fed into $DCAM_2$ to get the feature map F_{DCAM2}, and similarly, the dimension of the feature map does not change, then $\{F_0, F_{DCAM2}\}$ is fed into $DCAM_3$ to get the feature map F_{DCAM3}, the dimension of F_{DCAM3} is 192, Finally, all the inputs of $DCAM_1$, $DCAM_2$ $DCAM_3$, and $\{F_0, \{F_0, F_{DCAM_1}\}, \{F_0, F_{DCAM2}\}\}$, are stitched together to get the feature map, which has a dimension of 384. Since DCAM does not change the number of channels of the feature map, before the compression operation, the feature map should be sent to the bottleneck layer to downscale to 64 dimensions to obtain the feature map F_BL, and then the convolution layer and the " PReLU activation function" to obtain a feature map with half of the image size and 128 channels, which is expressed as follows:

$$F_{DCAM_1} = f_{DCAM_1}(F_0) \tag{12}$$

$$F_{DCAM_2} = f_{DCAM_2}(\{F_0, f_{DCAM_1}(F_0)\}) \tag{13}$$

$$F_{DCAM_3} = f_{DCAM_3}\left(\left\{F_0, f_{DCAM_2}\left(\left\{F_0, f_{DCAM_1}\left(F_0\right)\right\}\right)\right\}\right) \tag{14}$$

$$F_{BL} = f_{BL}\left(\left\{F_0, \left\{F_0, f_{DCAM_1}\left(F_0\right)\right\}, \left\{F_0, f_{DCAM_2}\left(\left\{F_0, f_{DCAM_1}\left(F_0\right)\right\}\right)\right\}\right\}\right) \tag{15}$$

$$F_1^{\downarrow} = \varphi(\omega_s \times F_{BL} + b_s) \tag{16}$$

where F_0 is the input of $DDPN_1$, $F_1^{\downarrow}$ is the output of $DDPN_1$, $f_{DCAM_1}(\bullet)$, $f_{DCAM2}(\bullet)$, and $f_{DCAM3}(\bullet)$ are the extracted feature operations of the three DCAM blocks, respectively, $f_{BL}(\bullet)$ denotes the bottleneck layer operation, ω_s and b_s denote the weights and bias parameters of the convolutional layer, respectively, and $\varphi(\bullet)$ denotes the PReLU activation function.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

2.2. Dual Channel Attention Module(DCAM)

As the core module of DDPN and DUPN, DCAM is mainly used to extract the image features, and its structure is shown in Fig. 4. DCAM uses both 3×3 and 5×5 convolutional kernels to enable the network to extract both low and high frequency information of the image. It is worth noting that since the size of the image is not changed during the feature extraction stage, the convolution kernels are convolved in the form of "same", and then the two extracted features are fused together and subjected to global covariance pooling (GCP) GCP is used instead of global covariance pooling to obtain a 1×1×C vector, i.e., to avoid the destruction of image details and to obtain a more differentiated feature representation. The purpose of this is to reduce the computational effort, and then use the "Softmax1" and "Softmax2" functions to obtain two different sets of channel attention coefficients for the feature vector, and the sum of the two sets of coefficients is 1, so it can be adjusted by The ratio of the two sets of coefficients can be adjusted to adjust the attention of the image on the low and high frequency information, thus allowing more flexibility in extracting different features. After obtaining the two sets of channel attention coefficients, they are fed into the pixel attention module to fine-tune the attention coefficients on each channel so that each pixel in the feature map has a more accurate coefficient, and finally the two sets of attention coefficients are applied to the feature maps extracted from the 3×3 and 5×5 convolution kernels and summed up to obtain the final output.

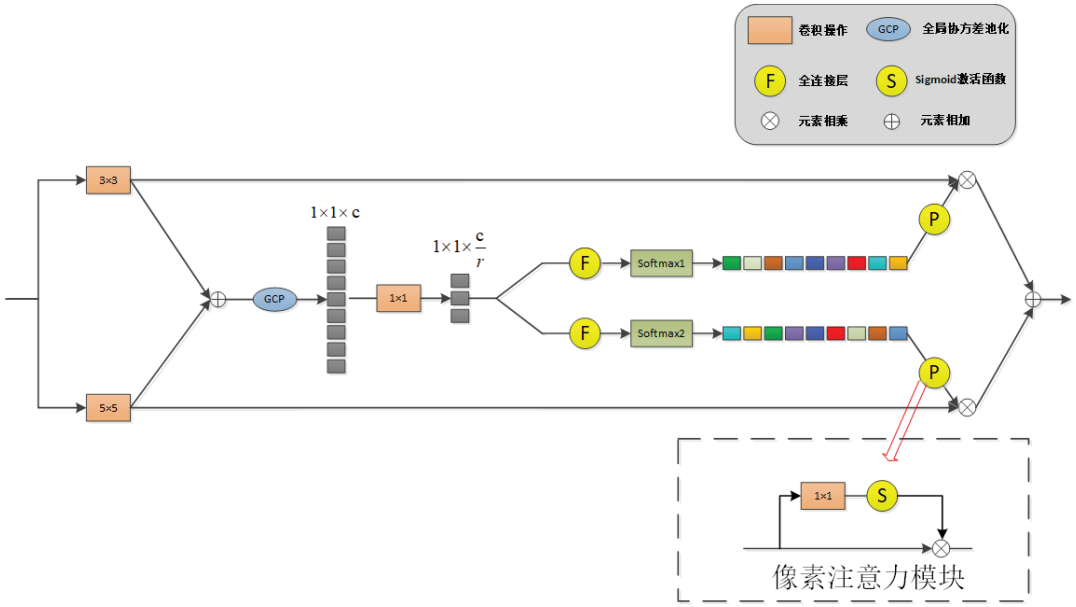

Figure 4. DCAM network structure diagram

Specifically, assuming that Figure 4 shows the internal structure of $DCAM_1$ in $DDPN_1$, the input of Figure 4 is F_0 and the output is F_{DCAM_1}, F_0 is first passed through 3×3 and 5×5 convolution kernels to obtain two feature maps $F_0^{3\times3}$ and $F_0^{5\times5}$ respectively, and the two feature maps are element-summed to obtain F_0^{+}, which can be expressed by the following equation:

$$F_0^{3\times3} = H_{3\times3}(F_0) \tag{17}$$

$$F_0^{5\times5} = H_{5\times5}(F_0) \tag{18}$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

$$F_0^+ = H_{3\times3}(F_0) + H_{5\times5}(F_0) \tag{19}$$

where $H_{3\times3}(\bullet)$ and $H_{5\times5}(\bullet)$ denote the convolutional operations with convolutional kernels of size 3×3 and 5×5, respectively. Immediately after F_0^+ is pooled by global covariance to obtain the feature vector t , t is compressed by a 1×1 convolution kernel to obtain $t^{'}$, and then $t^{'}$ goes through the fully connected layer and Softmax1 and Softmax2 to obtain different two channel attention coefficients ω_1 and ω_2 , respectively, and $\omega_1 + \omega_2 = 1$.

$$t = f_{GCP}(H_{3\times3}(F_0) + H_{5\times5}(F_0)) \tag{20}$$

$$t^{'} = H_{1\times1}\left(f_{GCP}\left(H_{3\times3}\left(F_0\right) + H_{5\times5}\left(F_0\right)\right)\right) \tag{21}$$

$$\omega_1 = \text{Softmax}\,1\left(f_F\left(H_{1\times1}\left(f_{GCP}\left(H_{3\times3}\left(F_0\right) + H_{5\times5}\left(F_0\right)\right)\right)\right)\right) \tag{22}$$

$$\omega_2 = \text{Softmax}\,2\left(f_F\left(H_{1\times1}\left(f_{GCP}\left(H_{3\times3}\left(F_0\right) + H_{5\times5}\left(F_0\right)\right)\right)\right)\right) \tag{23}$$

where $H_{1\times1}(\bullet)$ denotes the convolution kernel size of 1 × 1 convolution operation, $f_{GCP}(\bullet)$ denotes the global covariance operation, $f_F(\bullet)$ denotes the fully connected operation, and $\text{Softmax}(\bullet)$ denotes the Softmax activation function. Then ω_1 and ω_2 are fed into the pixel attention module to obtain pixel attention coefficients $\omega_1^{'}$ and $\omega_2^{'}$. Then, $\omega_1^{'}$ and $\omega_2^{'}$ are applied to $F_0^{3\times3}$ and $F_0^{5\times5}$ to obtain the outputs of the two branches, $F_{DCAM_1}^{(1)}$ and $F_{DCAM_1}^{(2)}$, and finally, $F_{DCAM_1}^{(1)}$ and $F_{DCAM_1}^{(2)}$ are element-summed to obtain the final output, F_{DCAM_1} which can be expressed as follows, representing the sigmoid activation function.

$$\omega_1^{'} = \left(\omega_1 \times \text{sigmoid}\left(H_{1\times1}\left(\omega_1\right)\right)\right) \tag{24}$$

$$\omega_2^{'} = \left(\omega_2 \times \text{sigmoid}\left(H_{1\times1}\left(\omega_2\right)\right)\right) \tag{25}$$

$$F_{DCAM_1}^{(1)} = (\omega_1^{'} \times F_0^{3\times3}) \tag{26}$$

$$F_{DCAM_1}^{(2)} = (\omega_2^{'} \times F_0^{5\times5}) \tag{27}$$

$$F_{DCAM_1} = F_{DCAM_1}^{(1)} + F_{DCAM_1}^{(2)} \tag{28}$$

2.3. Loss function

Since the three-layer V-network model produces feature maps at different scales when compressed and expanded, this paper uses the Multi-scale Structural Sinilarity (MS_SSIM) loss function[15] combined with the L1 loss function[16] as the total loss function, i.e:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

$$L_{total} = \alpha L_{MS\_SSIM} + (1-\alpha) \bullet G L_1 \tag{29}$$

where α uses the same coefficient of 0.16 as in the literature [15], G denotes the Gaussian distribution parameter, L_{MS_SSIM} and L_1 represent the MS_SSIM and the L_1 loss function, respectively, and are calculated as follows:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| H_{TVN}(I_i^{LR}) - I_i^{HR} \right\|_1 \tag{30}$$

$$L_{MS-SSIM} = 1 - MS\_SSIM(H_{TVN}(I_i^{LR}) - I_i^{HR}) \tag{31}$$

where, θ denotes all parameters of the network, $H_{TVN}(I_i^{LR})$ is the network reconstructed image, I_i^{HR} is the original high-resolution image, and MS_SSIM represents the multi-scale structural similarity operation.

3. EXPERIMENTAL SETUP

3.1. Experimental dataset

The dataset for this experiment uses the DIV2K[17] dataset containing 1000 high-definition images, of which 800 are used as the training set and the remaining 200 as the validation set to improve the generalization ability of the model. The test set uses four standard datasets commonly found in the field of super-resolution reconstruction, namely, Set5[18], Set14[19], BSDS100[20], and Urban100[21]. The first three datasets contain rich natural scenes, and the fourth dataset contains various architectural scenes. These datasets include image details of almost all frequency bands, which can well validate the performance of the model. In addition, in order to make full use of the training set and prevent the model from overfitting, this experiment expands the dataset by inverting and rotating the dataset. And the pixels of the images are normalized to between [-1,1] to avoid the gradient dispersion phenomenon during the training process.

3.2. Model parameter settings and experimental environment

The model uses Adam[22] as the optimizer with the initial learning rate set to 10^{-4} , β_1 and β_2 using the default values of the optimizer, i.e., $\beta_1 = 0.9$, $\beta_2 = 0.999$, to avoid the division by 0 error factor $\varepsilon = 10^{-8}$. A total of three DDPN modules and six DUPN modules are used in the model. Pytorch was used as the deep learning framework and Cuda was used for acceleration.

4. RESULTS AND ANALYSIS

4.1. Comparative analysis with the classical model of image super-resolution reconstruction

In this section, the model proposed in this paper (TVN) is compared and analyzed with existing classical models, mainly including SRCNN, FSRCNN, D-DBPN, EDSR and the traditional algorithm Bicubic. PSNR and SSIM are used as the evaluation indexes of the reconstruction effect. Table 1 shows the PSNR and SSIM values of each model at magnifications of ×2, ×3, and ×4, respectively. From the data in the table, it can be seen that TVN performs well in all three magnification factors compared with the rest of the classical models. Among them, there is a

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

more obvious improvement on the Urban100 dataset compared with the other models, which indicates that the present model can recover more edge texture information.

Tabel 1. Data comparison between the model in this chapter and existing classical models on PSNR/SSIM

Method	Scale	Set5	Set14	BSDS100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403
SRCNN	×2	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8948
ESPCN	×2	37.00/0.9559	32.75/0.9098	31.51/0.9065	29.87/0.9065
FSRCNN	×2	37.06/0.9554	32.76/0.9078	31.53/0.8912	29.88/0.9024
VDSR	×2	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140
DRCN	×2	37.63/0.9588	33.06/0.9121	31.85/0.8942	30.76/0.9133
LapSRN	×2	37.52/0.9591	33.08/0.9130	31.80 /0.8950	30.41/0.9101
D-DBPN	×2	38.09/0.9600	33.85/0.9190	32.27/0.9000	32.55/0.9324
EDSR	×2	38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351
RDN	×2	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353
TVN(our)	×2	38.22/0.9614	34.07/0.9215	32.36/0.9018	32.94/0.9359
Bicubic	×3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349
SRCNN	×3	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989
ESPCN	×3	33.02/0.9135	29.49/0.8271	28.50/0.7937	26.41/0.8161
FSRCNN	×3	33.20/0.9149	29.54/0.8277	28.55/0.7945	26.48/0.8175
VDSR	×3	33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290
DRCN	×3	33.82/0.9226	0.2977/0.8314	28.80/0.7963	27.15/0.8277
LapSRN	×3	33.82/0.9227	29.87/0.8320	28.82/0.7980	27.07/0.8280
D-DBPN	×3	-	-	-	-
EDSR	×3	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653
RDN	×3	34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653
TVN(our)	×3	34.73/0.9293	30.62/0.8469	29.30/0.8095	28.89/0.8661
Bicubic	×4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577
SRCNN	×4	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221
ESPCN	×4	30.66/0.8646	27.71/0.7562	26.98/0.7124	24.60/0.7360
FSRCNN	×4	30.73/0.8601	27.71/0.7488	26.98/0.7029	24.62/0.7272
VDSR	×4	31.35/0.8830	28.02/0.7680	27.29/0.0726	25.18/0.7540
DRCN	×4	31.53/0.8854	28.03/0.7673	27.24/0.7233	25.14/0.7511
LapSRN	×4	31.54/88.50	28.19/0.7720	27.32/0.7270	25.21/0.7560
D-DBPN	×4	32.47/0.8980	28.82/0.7860	27.72/0.7400	26.38/0.7946
EDSR	×4	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033
RDN	×4	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028
TVN(our)	×4	32.52/0.8994	28.79/0.7870	27.75/0.7422	26.69/0.8037

4.2. Image reconstruction effect comparison and analysis

In this paper, we select the "head" image in Set5 dataset and the "img_046" image in Urban100 dataset to do down-sampling factor of "×2" and "×4" respectively. " and "×4" image reconstruction effect comparison and analysis, as shown in Figures 5 and 6. From Figure 5, we can see that the recovered images of this model have more details, stronger facial texture and clearer eyes than other models. It is also easy to see from Figure 6 that the images recovered by the present model have more obvious edge structure, which is consistent with the results of data analysis in the previous section.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

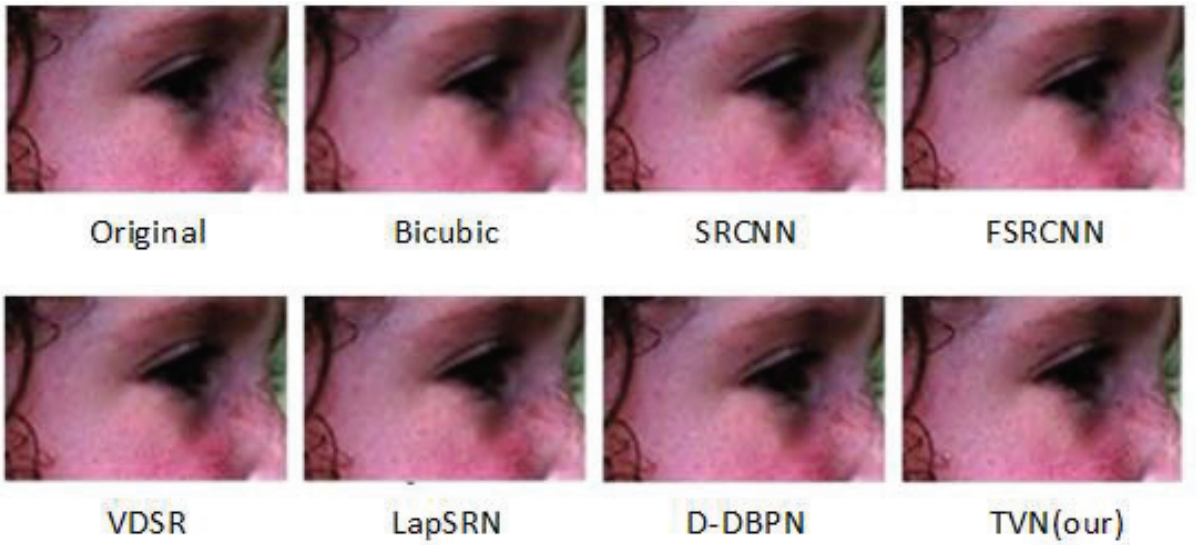

Figure 5. Comparison of reconstruction results of "head" images in Set5 dataset by models (×2)

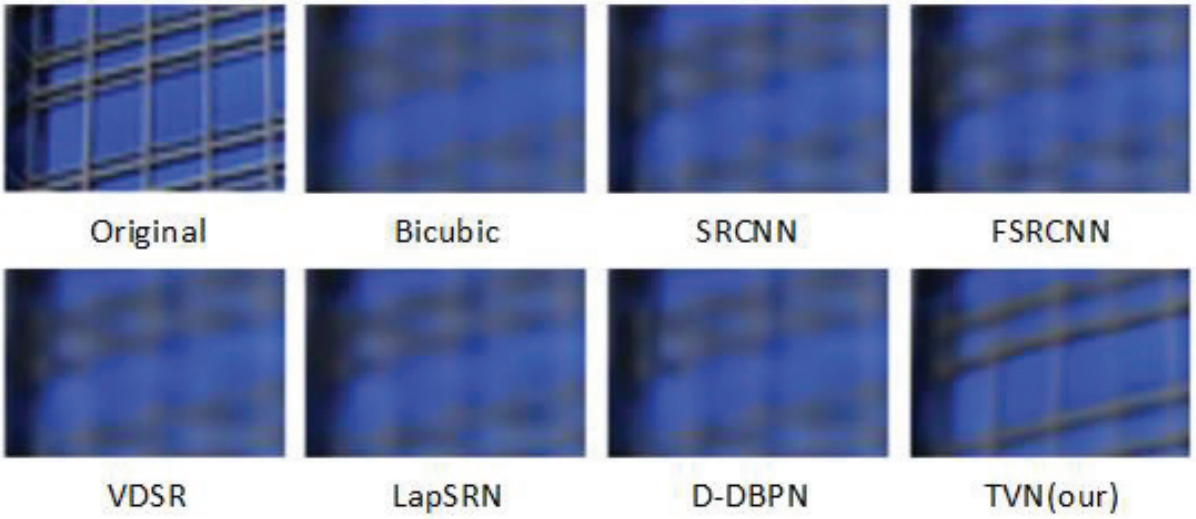

Figure 6. Comparison of the reconstruction effect of "img_046" image in Urban dataset by models (×4)

5. CONCLUSION

This chapter proposes an improved three-layer V-network structure based on the U-Net network. The model contains three V-networks of different depths that can extract features at different levels of the image, and the network is guaranteed to be lightweight by progressive compression and recovery. The network also introduces a dual-channel attention module, which can adaptively adjust the information fusion of the dual channels when extracting features in order to reconstruct more discriminative features. Experimental results show that the present model recovers sharper low-level edge features of images compared with other classical models.

ACKNOWLEDGMENTS

This paper is supported by the National Natural Science Foundation of China (Grant No. 61872126)

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

REFERENCES

[1] Ding L, Ding S, F, Zhang J, et al. Single-image super-resolution reconstruction using VGG energy loss[J]. Journal of Software, 2021.

[2] Zuo L, Zhang P, Jing S X, Zhao Y, Li F. A two-channel residual network for image super-resolution reconstruction[J]. Journal of Xi'an Jiaotong University,2022,56(01):158-164.

[3] Wang Huifeng, Xu Yan, Wei Yiming, et al. Super-resolution reconstruction of images based on parallel convolution and residual networks[J]. Computer Applications, 2022, 42(5):7.

[4] Fu LH, Sun XW, Zhao Y, Li ZG, Huang KY, Wang LY. A fast video super-resolution reconstruction method based on motion feature fusion [J]. Pattern Recognition and Artificial Intelligence,2019,32(11):1022-1031.DOI:10.16451/j.cnki.issn1003-6059.201911007.

[5] Chao D, Chen CL, He K, et al. Learning a Deep Convolutional Network for Image Super-Resolution[C]// ECCV. Springer International Publishing, 2014

[6] Dong C , Loy C C , Tang X . Accelerating the Super-Resolution Convolutional Neural Network[C]// European Conference on Computer Vision. Springer, Cham, 2016.

[7] Kim J, Lee JK, Lee K M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks[J]. IEEE, 2016.

[8] Shi W , Caballero J , F Huszár, et al. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[9] Kim J, Lee J K, Lee K M. Deeply-Recursive Convolutional Network for Image Super-Resolution[J]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[10] Tong T, Li G, Liu X, et al. Image Super-Resolution Using Dense Skip Connections[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2017.

[11] Huang G, Liu Z, Laurens V, et al. Densely Connected Convolutional Networks[J]. IEEE Computer Society, 2016.

[12] Ledig C , Theis L , Huszar F , et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network[J]. IEEE Computer Society, 2016.

[13] Lai W S , Huang J B , Ahuja N , et al. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2017:5835-5843.

[14] Zhang Y , Li K , Li K , et al. Image Super-Resolution Using Very Deep Residual Channel Attention Networks[J]. 2018.

[15] Zhao H, Gallo O, Frosio I, et al. Loss functions for image restoration with neural networks[J]. IEEE Transactions on computational imaging, 2016, 3(1): 47-57.

[16] Xue, Y., Xu, T., Zhang, H. et al. SegAN: Adversarial Network with Multi-scale L1 Loss for Medical Image Segmentation. Neuroinform 16, 383–392 (2018).

[17] Timofte R, Agustsson E, Van Gool L, et al. Ntire 2017 challenge on single image super-resolution: Methods and results[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017: 114-125.

[18] Zhou B, Li C H, Chen W. Region-level channel attention for single image super-resolution combining high frequency loss[J]. Journal of Image and Graphics, 2021, 26(12): 2836-2847.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202401_10(1).0002

[19] Zeyde R, Elad M, Protter M. On single image scale-up using sparse-representations[C]//Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7. Springer Berlin Heidelberg, 2012: 711-730.

[20] Chaowen S U N, Xiao C. Multiscale feature fusion back-projection network for image super-resolution[J]. Acta Auto. Sinica, 2021, 47(7): 1689-1700.

[21] Cai Tiejian,Peng Xiaoyu,Shi Yapeng,Huang Ji. Channel attention and residual cascading for image super-resolution reconstruction[J]. Optical Precision Engineering,2021,29(01):142-151.

[22] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.