Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5579090, 13 pages
https://doi.org/10.1155/2021/5579090

Research Article

Cascading and Residual Connected Network for Single Image Superresolution

Kai Huang[1], **Wenhao Wang**[1], **Cheng Pang**[1], **Rushi Lan**[1,2], **Ji Li**[1,3] and **Xiaonan Luo**[2,3]

[1]*Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*National Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*Guilin Huigu Institute of Artificial Intelligence Industrial Technology, Guilin 541004, China*

Correspondence should be addressed to Cheng Pang; pangcheng3@guet.edu.cn and Rushi Lan; rslan2016@163.com

Received 7 January 2021; Revised 25 July 2021; Accepted 3 August 2021; Published 21 October 2021

Academic Editor: Xiaojie Wang

Convolution neural networks facilitate the significant process of single image super-resolution (SISR). However, most of the existing CNN-based models suffer from numerous parameters and excessively deeper structures. Moreover, these models relying on in-depth features commonly ignore the hints of low-level features, resulting in poor performance. This paper demonstrates an intriguing network for SISR with cascading and residual connections (CASR), which alleviates these problems by extracting features in a small net called head module via the strategies based on the depthwise separable convolution and deformable convolution. Moreover, we also include a cascading residual block (CAS-Block) for the upsampling process, which benefits the gradient propagation and feature learning while easing the model training. Extensive experiments conducted on four benchmark datasets demonstrate that the proposed method is superior to the latest SISR methods in terms of quantitative indicators and realistic visual effects.

1. Introduction

Superresolution (SR) image reconstruction is widely used in various applications, such as military surveillance, medical diagnostics [1, 2], remote sensing [3], and video streaming [4, 5]. Single image superresolution (SISR) is aimed at reconstructing a high-resolution (HR) image from its counterpart low-resolution (LR) input, which is an essential and classic task in computer vision. Recently, high-resolution (HR) demand images have boosted. However, physical constraints limit the conduction of high-resolution pictures. A series of successful works brought attention to the research community.

The task of recovering HR images I^{SR} from its counterpart (LR) version I^{LR} is ill-posed. Researchers have made many efforts to this task and invented numerous algorithms, including interpolation-based, reconstruction-based, and learning-based methods [1], respectively.

The traditional SISR algorithms, for instance, bicubic interpolation [6], are high-speed while suffering from poor accuracy. It is easy to fail in practice. To limit possible solving space, researchers present more advanced methods, reconstruction-based algorithms [7, 8], by introducing available prior knowledge. These algorithms may restore clear details (i.e., texture details), but extensive experiments show that they degrade sharply when the scale factors increase; subsequently, the algorithms with learnable parameters [9] are proposed to analyze relationships between the I^{LR} image and their counterpart I^{HR} image by training concrete instances [10, 11]. Although such learning-based methods perform very well, the time-consuming optimization problems they involve are very tricky.

In recent years, CNNs have been introduced to facilitate the progress of the SISR field because of their excellent feature representation ability. Dong et al. [12] were the first to propose a three-stage convolutional network to solve the

SISR problem, which has become a milestone in this field. Since then, the research community has set out to design more complex networks to improve performance. EDSR, a very large network with residual blocks, was presented by Lim et al. [13] and achieved satisfactory performance in both PNSR and SSIM [14]. However, these state-of-the-art methods still have some limitations:

(1) The state-of-the-art (SotA) models [13] mainly improve the performance by considerably growing the depth and width of the proposed methods. Therefore, massive parameters and increasing resource-consuming problems are inevitable

(2) Many progressive models do not fully take advantage of the hierarchical information from the primary LR images, which are essential for improving visual performances

To address these shortcomings, we present a model named CASR, exploring two separate strategies to functionally extract features for precise SISR. Figure 1 shows the ×4 SR results of our proposed model on dataset DIV2K [15]. First, we propose a small but functional depthwise separable convolution network named head module aimed at more systematic feature extraction.

Second, we present another cascaded residual network (CAS-Block) for better feature and gradient propagation. Our proposed method combines features from excessive layers at both the regional and global levels with such architecture. Moreover, a stacking broader local residual connection is applied to exploit the feature of the I^{LR} and let the vast low-level mappings be transmitted. This schema unites nonlocal actions to capture remote spatial features from former inputs.

As the crucial integrant of the presented method, the CAS-Block includes six subtrunks, each of which consists of two convolutional layers and a nonlinear activation pReLU. Because using the activation function in bottlenecks does affect the performance, we take advantage of channels before the pReLU layer to construct the inverse residual block, resulting in performance improvement.

The three main contributions of this article are summarized as follows.

(1) We propose a head module applied with a series of depthwise separable convolution operations for feature extraction. In addition, we replace all existing conventional convolution operations with deformable convolution layers in the module. At the same time, in order to effectively retain the features, we extend the low-dimensional representation to high-dimensional before passing the activation function. This maintains a balance between a large number of parameters and excellent performance

(2) In order to effectually raise feature fusion and gradient propagation, we introduce a cascaded block called CAS-Block. This mechanism allows our network to combine features from diverse layers. Furthermore, such a structure is also used to construct the network and promote its functional expression

(3) We utilize the L_1 with the addition of total variance loss $\mathscr{L}_{TV}$ instead of the traditional sole L_1 loss function, which significantly improves the quality of the reconstruction image I^{SR}. Meanwhile, to obtain better optimization weights, we explored various parameter settings

2. Related Works

2.1. SISR Using Deep CNNs. In the field of superresolution, compared to conventional image restoration methods, CNN-based models have a stronger feature expression ability and have achieved great success. Dong et al. [12] first proposed an algorithm named SRCNN, which is an end-to-end algorithm based on CNN. It consists of three convolutional layers, and its performance is impressive compared to traditional methods (i.e., sparse coding [7] and bicubic interpolation [6]). Later, the research community designed more intricate CNN architectures and developed more profound networks. For example, in order to grow the depth of the network, VDSR [16] introduced residual learning, and the verification experiment proved that this strategy heightens the SR image qualities and promotes convergence. DRCN [17] uses deep recursion to construct a neural network and uses the same convolution kernel 16 times in the reference network, effectually dropping the number of parameters. He et al. [18], inspired by the ordinary differential equation (ODE), propose an intriguing network named OISR, which provides a new understanding of network designs. It is worth noting that most of these latest methods use interpolated images for input, which will not only cause the details to be too smooth but also boost additional computational cost and time consumption.

2.2. Skip Connection. ResNet [19] was the first to adopt the concept of skip connection, and then, the idea was extended to various computer vision tasks, such as image restoration [20] and semantic segmentation [21]. Since it is difficult for ordinary SR networks to construct extremely deep networks, employing various skip connections avoids the gradient vanishing trap and boosts performance. The strategy is roughly divided into two categories, namely, global or local residual connections and dense connections.

2.2.1. Global or Local Residual Connections. In image restoration tasks, LR images are closely related to HR ones. Obtaining the residual feature maps among the image's pixels can learn the absent high-frequency detailed information. VDSR is the first residual model for superresolution. Extensive experiments have proved that residual learning can enhance reconstruction performance and promote convergence speed. Therefore, this method has been widely used in various computer vision tasks [22].

2.2.2. Dense Connections. A dense connection allows the current layer to connect with all former layers, and the architecture provides more intriguing effects on restoring high-

(a) (b)

Figure 1: ×4 superresolution results of our proposed SR model on dataset DIV2K.

resolution patterns. DenseNet [23] first presents the dense connection in the SR field, starting from the features, achieving better results with fewer parameters through the extreme use of the features.

Traditional neural networks are basically unidirectional propagations, and the signals received in the later layers are very weak. To solve this problem, MemNet [20] stacks memory blocks and adds dense connections among each block, which is called the long-term memory model. Such architecture reduces the weight of the entire network, facilitates convergence, and deepens the network.

RDN [24] uses a similar architecture, but MemNet does not take all the intermediate feature information, while RDN applies global residual learning to use all of them.

Jiang et al. [25] proposed a hierarchical dense network (HDRN) in 2019, which can effectively establish realistic mapping relationships between the LR and HR image, promoting information interaction and representation.

Different from the above models, CARN also uses a cascade mechanism at the local and global levels to integrate features from multiple layers, which can reflect input representations at different levels for receiving more information [26]. Haris et al. [27] proposed D-DBPN, which connects the features of the up- and downsampling stages and improves the SR result.

2.3. Depthwise Separable Convolution. The cross-channel correlation and spatial correlation of the convolutional layer can be decoupled, and they can be mapped separately to achieve better results. Some lightweight networks, such as MobileNet [28], apply depthwise separable convolution, which is a combination of depthwise (DW) and pointwise (PW) to extract feature maps. Compared with the conventional convolution operation, the number of parameters and operation cost are relatively small. In Figure 2, we use the separable convolution operation in the depth direction in the head module.

2.4. Multiscale Learning. So as to utilize computing resources more efficiently and extract more features under the same amount of calculation, Szegedy et al. [29] present the inception module. There are two main contributions of the inception structure: one is to use 1×1 convolution to perform dimensionality reduction; the other is to simultaneously perform convolution and reaggregation on multiple sizes. Inspired by [29] and [30], MSRB [31] was proposed. Multiscale feature fusion and local residual learning can be applied to adaptively detect images of different scales with different sizes of convolution kernel features. The results show that performing different kernel operations can provide better extraction capabilities. However, this method cannot expand more receptive fields and cannot generate more detailed structural information.

2.5. Deformable Convolution. Conventional convolution kernels are usually of fixed size (for example, $3 \times 3, 5 \times 5$, and 7×7.). The biggest problem with this convolution kernel is that it has poor adaptability to unknown changes and weak generalization ability. In order to solve the object space deformation problem, deformable convolution [32] is proposed to heighten the transformation modeling ability of CNN. Deformable convolution is based on traditional convolution, adding the direction vector of the adjustment convolution kernel to make the shape of the convolution kernel closer to the feature.

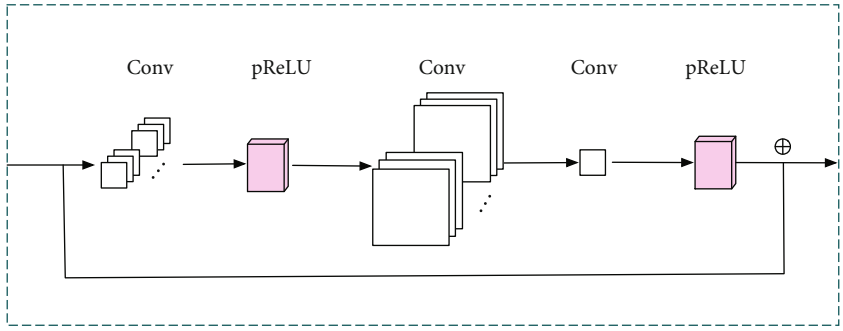


Figure 2: The architecture of head module is abundantly applied with depthwise separable convolution and deformable convolution operations. ⊕ stands for the element-wise additional operation.

2.6. Real-World Image Superresolution. In real-world image restoration scenarios, lacking corresponding high-quality references usually conduct poor experimental results. We additionally introduce the naturalness image quality evaluator (NIQE) [33] and Perceptual Index (PI) [22] to perform the evaluation. In fact, these indicators can sensitively reflect content sharpness, detail contrast, and texture diversity. These evaluation indexes have a high consistency with the subjective quality and can effectively reflect the visual quality of images without reference. In particular, the smaller values of NIQE/PI indicate better perceptual quality and clearer content. We intend to apply it to the DRIVE [34] dataset to estimate the restoration capability of the proposed method.

3. Proposed Approach

3.1. Network Architectures. SISR's algorithm, such as ESPCN [35] and FSRCNN [36], does not take full advantage of low-level feature information. With a deeper structure, there are more parameters. As shown in Figure 3, the proposed CASR consists of three components: (1) head module, (2) cascading block, and (3) upsample module. All we want is the balance between the performance and the cost.

To better explore the mentioned issues, we adopt two different strategies: (1) original feature extraction and (2) cascading connection structure.

3.1.1. Original Feature Extraction. We depict I^{LR} and I^{SR} as the input and output of our models, respectively. Figure 2 illustrates how the head module extracts the original information from LR images:

$$F_{\mathrm{ext}} = H_{\mathrm{ext}}\left(I^{\mathrm{LR}}\right), \tag{1}$$

where $H_{\mathrm{ext}}(\cdot)$ means a series of convolution operations. In the head module, we first replace the conventional one with a depthwise convolution layer for reducing parameters. Through an activation layer, the feature maps are sent to another specific convolution layer, deformable convolution. As we discussed in Section 2, deformable convolution adds an offset to each convolution sampling point, thus achieving free deformation of the sampling grid. Then, after passing through a specific convolutional layer with 1×1 kernel and another pReLU activation function, F_{ext} is sent to the next stage for a higher-level abstraction.

3.1.2. Cascading Connection Structure. Now, we present the CAS-Block. The cascade connection allows information to spread across multiple paths in the network, which greatly enhances feature fusion. It [10] has been widely applied in various computer vision tasks. In Figure 4, the mapping process of our cascade network includes C CAS-Blocks, each with a skip connection:

$$H_{\mathrm{map}_0} = C_0(F_{\mathrm{ext}}), \tag{2}$$

$$H_{\mathrm{map}_i} = C_i\left(H_{\mathrm{map}_0}\right), \tag{3}$$

where H_{map_i} presents the output of the C_ith CAS-Block. Each CAS-Block contains one group convolution layer (with 3×3 or 1×3 kernel), one traditional convolution layer for adjusting the number of channels, and a pReLU layer. We prefer stacking several kernels with smaller sizes (such as 1×3 and 3×3) to directly applying larger kernels (such as 5×5 and 7×7) for enlarging the receptive field of the feature extraction module and decreasing the number of learnable parameters:

$$H_{\mathrm{middle}} = C_{m+1}\left(H_{\mathrm{cas}}\left(\left(\left(H_{\mathrm{map}_m} + H_{\mathrm{map}_{m-1}}\right), \left(H_{\mathrm{map}_m} + H_{\mathrm{map}_{m-2}}\right), \cdot \left(H_{\mathrm{map}_{m-1}} + H_{\mathrm{map}_{m-2}}\right)\right)\right)\right), \tag{4}$$

where $H_{\mathrm{map}_m}, H_{\mathrm{map}_{m-1}}, H_{\mathrm{map}_{m-2}}$ means all the outputs of the middle three CAS-Blocks. H_{cas} denotes the cascading operation:

$$H_{\mathrm{map}} = C_{m+2}(C_1 + H_{\mathrm{middle}}), \tag{5}$$

where $H_{\mathrm{map}}(\cdot)$ demotes our proposed mapping function. Finally, we use a common upsampling module to fuse the hierarchical structural features and amplify the image size:

$$F_{\mathrm{up}} = H_{\mathrm{up}}\left(H_{\mathrm{map}}\right), \tag{6}$$

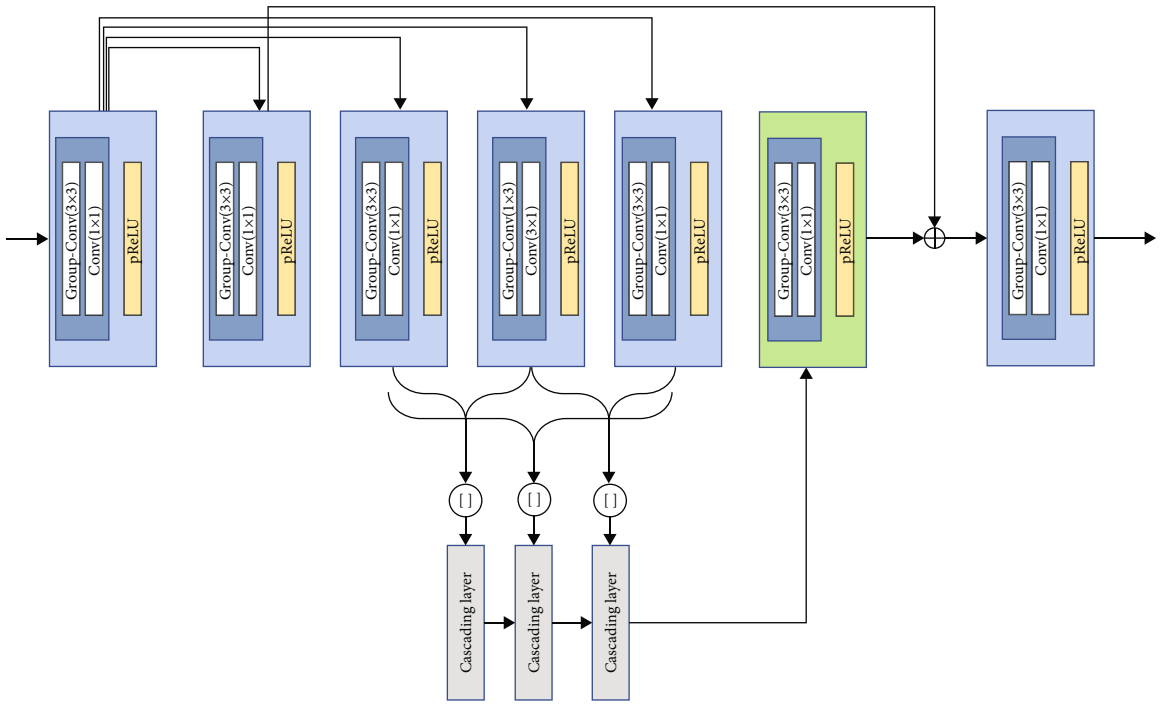


FIGURE 3: Network design for our model. The green cuboid means the head module; the yellow one represents the cascading block, and the last blue rectangle depicts the upsampling process.

where $H_{\text{up}}(\cdot)$ indicates an upscale module. In recent years, many upsampling methods have been proposed, such as [12, 27, 36]. We adopt the postupsampling method, which has been proven effectively outstanding. The process of our model roughly includes three steps. First, taking the low-resolution I^{LR} image as the original input, the feature extraction module obtains the initial features from the low-quality image. Then, these features are delivered to a higher abstraction layer. Finally, we adopt a simple upsampling block, including a convolutional layer, and a pixel-shuffle layer to enlarge the SR image.

3.2. Total Variation Loss. Aly and Dubois bring the total variation (TV) [37] loss to the SR field in order to suppress noise in generated images, and for imposing spatial smoothness, Yuan et al. also select this TV loss:

$$\mathscr{L}_{\text{TV}}\left(\widehat{I}\right) = \frac{1}{hwc}\sum_{i,j,k}\sqrt{\left(I\wedge_{i,j+1,k} - I\wedge_{i,j,k}\right)^2 + \left(I\wedge_{i+1,j,k} - I\wedge_{i,j,k}\right)^2}, \tag{7}$$

where $\widehat{I}$ depicts the reconstructed HR image, h, w, represent the dimensions of the corresponding feature maps, and c symbolizes the number of channels. On the other hand, although mean square error (MSE) is available, previous work [38] proved that it is not a good choice. Thus, the second loss function is defined as follows:

$$\mathscr{L} = \mathscr{L}_1 + \lambda\mathscr{L}_{\text{TV}}. \tag{8}$$

We applied these loss functions in the training process of our presented model. From the experiment, we found that adopting the $\mathscr{L}$ loss compared with the simple $\mathscr{L}_1$ loss, the model achieves better performance, and set $\lambda = 1e^{-4}$ works well. As shown in Figure 5, the loss $\mathscr{L}$ enables the network to generate smoother recovery images, and Figure 6 comparatively illustrates that the combined loss function may produce sharper SR results.

3.3. Comparison with Other CNN-Based Methods

3.3.1. Comparison with MSRN. Compared with MSRN, our CASR is different as follows. First, the basic module design is distinct. In MSRN, the multiscale residual block (MSRB) incorporates parallel convolution with multiple feature channels. The output of each multiscale residual block is cascaded together through a hierarchical feature fusion to produce the final result, which leads to a lot of calculations. However, our multiscale modules are branch-based, using regional skip connections and cascades extensively, scaling down parameters. Second, it is the difference in the activation function. MSRN utilizes the ReLU function, while we employ PReLU as the activation function. According to the comparison in Figure 7, PReLU optimizes and improves ReLU. Under the premise of almost no increase in the amount of calculation, the PReLU function effectually improves the overfitting problem of the model, accelerates the convergence, and lowers the error. Therefore, our proposed multiscale module owns more effective representation capabilities.

3.3.2. Comparison with MemNet. We summarize the main differences between MemNet [20] and our CASR. The

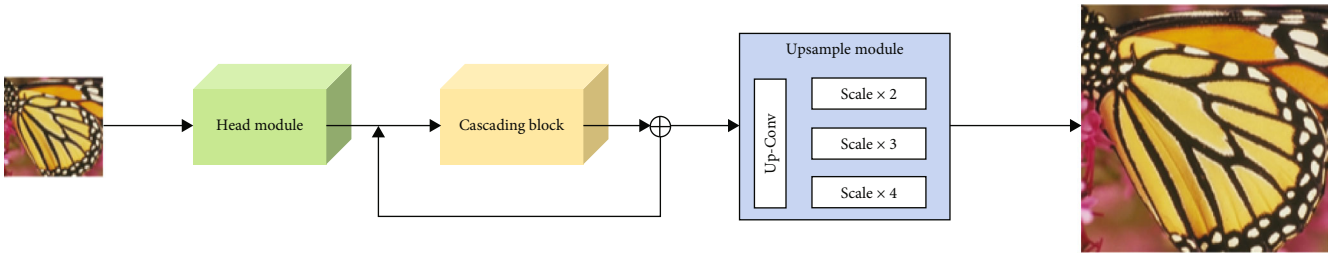


FIGURE 4: The architecture of cascading block (CAS-Block). ⊕ operator denotes the element-wise additive operation and [ ] means cascading operation.

FIGURE 5: ×3 SR's loss function comparison. In the first row, it is the *comic* images in the Set14 dataset. The image processed with $L_1 + L_{TV}$ has precise details in the area around the eyes. The bottom line is the "img020" images in the Urban100 benchmark dataset. This method applies the $L_1 + L_{TV}$ method to reconstruct the clear details, such as the windows.

former employs stacking memory blocks and massive shortcuts, while our method avoids extensive dense connections for lowering the number of parameters. What is more, Lim et al. trained their network with the L_2 loss, but we prefer L_1 loss to L_2 loss function. Besides, MemNet regards the interpolated images as input. Contrastively, our proposed method directly extracts hierarchical features from the original LR images upsampled at the end of the process for computational efficiency and SR performance improvement.

4. Experimental Results

4.1. Training Details. We set depthwise separable convolution operations in head module shown in Figure 2, which were first illustrated in the Inception net in the proposed model, and were able to reduce the size of the network parameters effectively. Figure 4 graphically illustrates the cascading process occurring. The medial layers' outcomes are cascaded into the posterior layers and finally assemble in a convolutional trunk consisting of a depthwise separable convolutional operation with three times the input and output features and then thorough a pReLU activation function.

We prefer $L_1 + \mathscr{L}_{TV}$ to L_2 loss as the loss function, though the latter has been generally applied in computer vision tasks because of its intimate relation with PSNR's calculation. However, the research community recently indicates that L_1 loss provides better accuracy and faster convergence; TV loss ($\mathscr{L}_{TV}$) imposes spatial smoothness on reconstructed images. Specifically, we set training patches with a size of 128×128, and batch size = 16. We employ

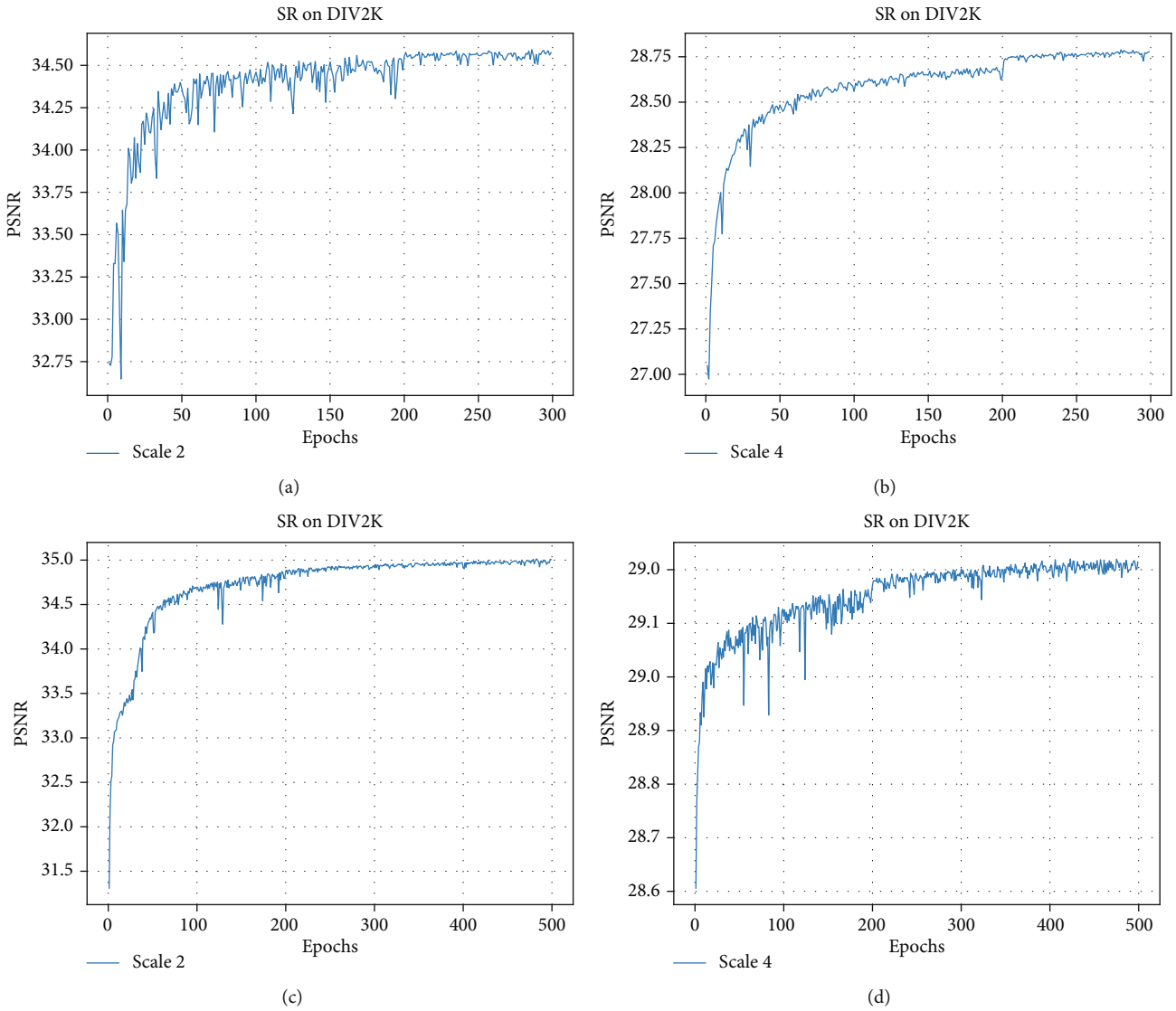


Figure 6: PSNR (dB) of training process on scales ×2 and ×4 with L_1 and $L_1 + \mathscr{L}_{TV}$ loss, respectively.

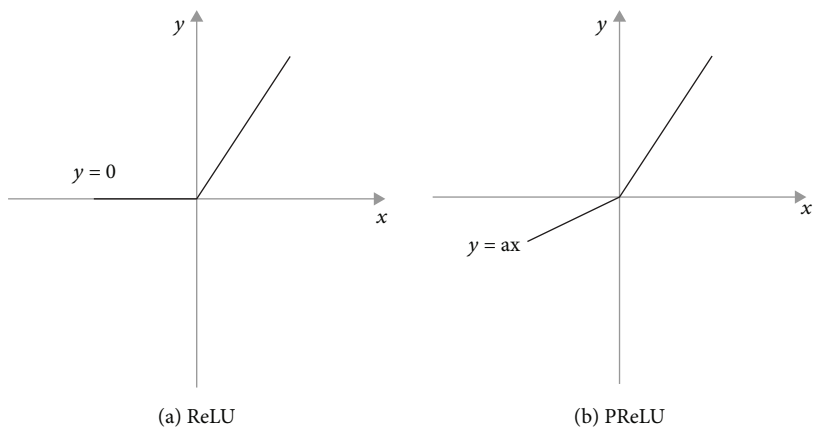


Figure 7: Comparison: ReLU versus PReLU. PReLU optimizes and improves ReLU. Under the premise of almost no increase in the amount of calculation, the overfitting problem of the model is effectively improved. The convergence is faster, and the error is lower.

Table 1: Public benchmark test results (PSNR (dB)/SSIM). Bold illustrates the best result, and the underline indicates the second-best ones.

Method	Scale	Params	Multi-adds	Set5 [40]	Set14 [41]	B100 [42]	Urban100 [43]
SRCNN [12]	×2	57K	52.7G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
VDSR [16]	×2	665K	612G	37.52/0.9586	33.01/0.9123	31.89/0.8960	30.76/0.9140
LapSRN [44]	×2	813K	29.9G	37.51/0.9590	33.07/0.9129	31.89/0.8959	30.40/0.9100
DRCN [17]	×2	1774K	17974G	37.43/0.9587	33.23/0.9137	31.29/0.8911	30.22/0.9023
DRRN [8]	×2	297K	6796.9G	37.74/0.9523	32.79/0.9121	31.67/0.8900	30.14/0.9112
MemNet [20]	×2	677K	2662.4G	37.83/0.9623	33.15/0.9192	31.99/0.9000	30.51/0.9177
RDN [24]	×2	22.12M	5096.2G	38.30/0.9616	**34.10/0.9218**	**32.40/0.9022**	**33.09/0.9368**
HDRN [25]	×2	—	—	37.75/0.9590	33.49/0.9150	32.03/0.8980	31.87/0.9250
OISR [18]	×2	41.91M	9656.5G	37.98/0.9604	33.58/0.9172	32.18/0.8996	32.09/0.9281
IDN [45]	×2	590K	174.1G	38.10/0.9601	33.91/0.9194	32.31/0.9012	32.92/0.9269
CASR (ours)	×2	501K	113G	**38.49/0.9607**	33.97/0.9204	32.33/0.9017	32.90/0.9244
SRCNN [12]	×3	57K	52.7G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7963
VDSR [16]	×3	665K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
LapSRN [44]	×3	813K	29.9G	33.82/0.9207	29.89/0.8304	28.82/0.7950	27.07/0.8298
DRCN [17]	×3	1.77M	17974G	33.84/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
DRRN [8]	×3	297K	6796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
MemNet [20]	×3	677K	2662.4G	34.09/0.9590	30.07/0.8429	29.89/0.8110	28.41/0.8610
RDN [24]	×3	22.31M	2281.2G	**34.78/0.9300**	**30.67/0.8482**	**29.33/0.8105**	**29.00/0.8683**
HDRN [25]	×3	—	—	34.24/0.9240	30.23/0.8400	28.96/0.8040	27.93/0.8490
OISR [18]	×3	44.86M	4590.1G	34.43/0.9273	30.33/0.8420	29.10/0.8053	28.20/0.8534
IDN [45]	×3	590K	105.6G	34.46/0.9282	30.52/0.8462	29.25/0.8093	28.80/0.8653
CASR (ours)	×3	597K	52G	34.49/0.9297	30.57/0.8450	29.29/0.8099	28.90/0.8704
SRCNN [12]	×4	57K	52.7G	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221
VDSR [16]	×4	665K	612.6G	31.35/0.8830	28.02/0.7680	27.29/0.7226	25.18/0.7540
LapSRN [44]	×4	813K	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560
DRCN [17]	×4	1774K	17974G	31.53/0.8846	28.02/0.7670	27.23/0.7233	25.14/0.7510
DRRN [8]	×4	297K	6796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
RDN [24]	×4	22.27M	1309.2G	32.61/0.9003	28.92/0.7893	**27.80/0.7434**	**26.82/0.8069**
HDRN [25]	×4	—	—	32.23/0.8960	28.58/0.7810	27.53/0.7370	26.09/0.7870
OISR [18]	×4	44.27M	2962.5G	32.21/0.8950	28.63/0.7822	27.58/0.7364	26.14/0.7874
MemNet [20]	×4	677K	2662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7286	25.50/0.7630
IDN [45]	×4	590K	81.8G	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033
CASR (ours)	×4	597K	51G	**32.67/0.9006**	**28.96/0.7899**	27.77/0.7428	26.67/0.8057

16000 iterations of back-propagation per epoch; we adopt the ADAM [39] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$ for optimization. We set 850 training epochs and the learning rate of all layers to 1×10^{-4} initially, which will be reduced to half for every 50 epochs. All experiments are run under the PyTorch framework and deployed on NVIDIA RTX 2080Ti GPU.

4.2. Datasets. DIV2K [15] is a high-definition dataset containing various image contents. It has 800 training images, 100 verification images, and 100 test images. We employ 800 training images to train the proposed model and randomly select ten validation images as evaluation. In the testing process, we adopt the following benchmark datasets as test datasets: Set5 [40], Set14 [41], B100 [42], and Urban100 [43]. They contain various scenes in real life, such as landscapes, buildings, and people, while the Digital Retinal Images for Vessel Extraction (DRIVE [34]) dataset is a dataset for retinal vessel segmentation. It consists of a total of JPEG 40 color fundus images, including 7 abnormal pathology cases.

4.3. Experimental Analyses

4.3.1. Results on Benchmark Datasets. Our proposed method will be compared with the state-of-the-art SR model on two commonly adopted image quality metrics (i.e., PSNR and SSIM). We analyze our methods with several progressive networks: (1) bicubic, (2) SRCNN [12], (3) VDSR [16], (4) LapSRN [44], (5) DRCN [17], (6) DRRN [8], (7) MemNet [20], (8) RDN [24], (9) HDRN [25], (10) OISR [18], and (11) IDN [45]. As described in the technical literature, these methods were evaluated on four aforementioned datasets. Table 1 lists the performance of all mentioned algorithms.

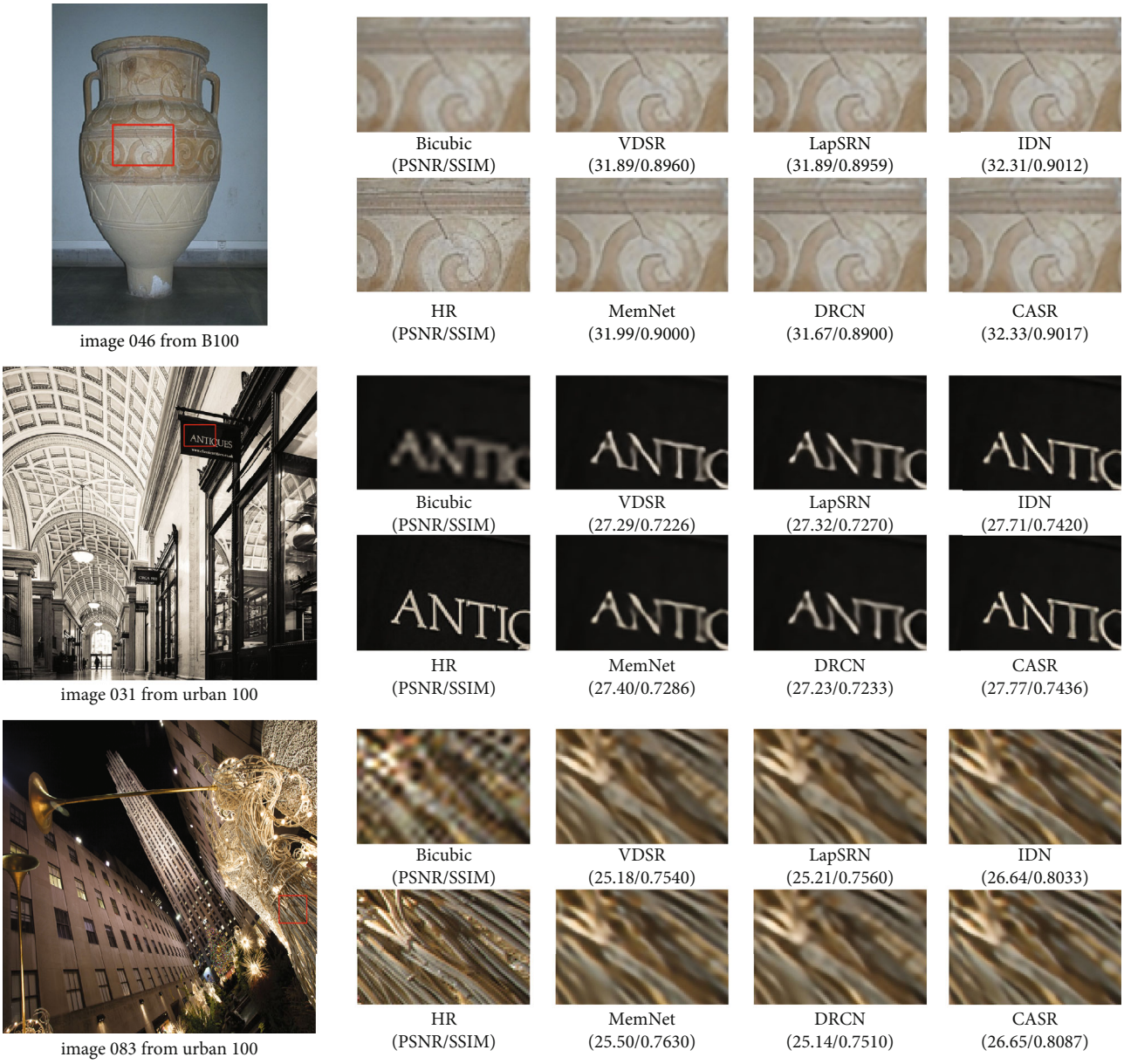


Figure 8: Comparison of reconstructed image details on the benchmark dataset. From top to bottom, the ×2 superresolution experiment on B100, the ×3 and ×4 superresolution experiment on Urban100. Our presented method reconstructs SR images with richer details.

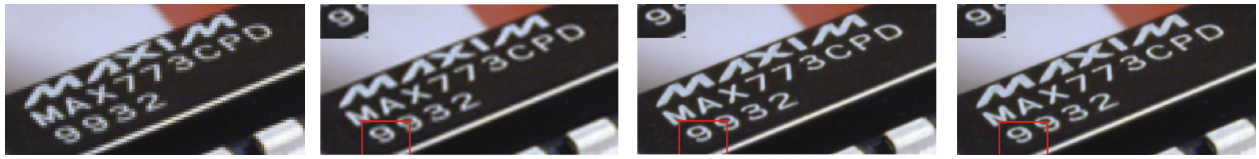
Table 2: Time consumption and experimental performance in dataset Set14 with the scale 4.

Method	Scale	Params	Multi-adds	Runtimes (s)	PSNR/SSIM
VDSR [16]	×4	665K	612.6G	0.4523	28.02/0.7680
RDN [24]	×4	22.27M	1309.2G	0.2114	28.92/0.7893
CASR (ours)	×4	597K	51G	0.1017	28.96/0.7899

Our networks are much better than the comparison model in variant scale factors except for RDN. On some datasets, the performance of CASR is completely close to RDN, while the consumption of RDN is much larger in the meantime. We will particularly discuss it later.

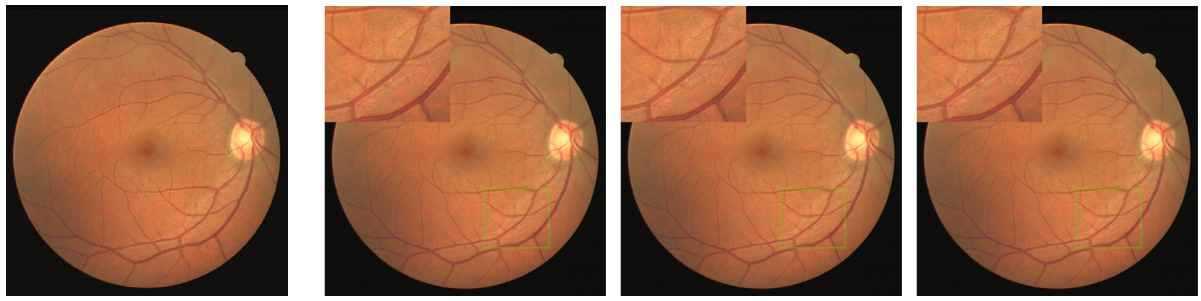
Table 3: Average NIQE and PI values on public dataset Set14 with scale factor ×4.

Metrics	SRCNN [12]	VDSR [16]	IDN [45]	CASR (ours)
NIQE [33]	6.624	5.744	6.472	4.506
PI [22]	5.929	5.121	5.448	3.859



(a) LR Image (b) Bicubic (c) IDN [50] (d) CASR (Ours)

Figure 9: Visual image comparisons with different SR methods on real-world image *chip* with scale ×2.



(a) LR Image (b) Bicubic (c) RDN [18] (d) CASR (Ours)

Figure 10: The qualitative comparisons between different SR methods on the digital retinal image for vessel extraction (DRIVE) dataset with scale ×4.

Compared with other methods, on dataset Set5, our proposed method performs better at all scales. Especially on the ×2 superresolution, the reconstructed image contains very clear texture details.

Our method performs well in image superresolution restoration tasks of various scales on the dataset Set14. Specifically, the best performance of the benchmark SR model is RDN [24] and IDN [45], which reach 30.67/0.8482 and 30.52/0.8462 on PSNR/SSIM metrics, respectively, at the ×3 scale; our model is 0.1 db lower than the former and 0.05 db higher than the latter.

As mentioned earlier, the dataset B100 contains many real-world images. As seen in Figure 8, the vase image recovered by our method has clearer edges, reaching 32.33 db. Figure 8 demonstrates visual comparisons on dataset B100 and Urban100 with scales ×2 and ×4, respectively.

The Urban100 dataset consists of 100 pictures of various buildings, which usually contain clear edges and rich textures. So, according to [24], RDN is expected to perform well on superresolution tasks, reaching 33.09 db on ×2. Our method acts well at ×3 and ×4 superresolution tasks, reaching 28.90 db and 26.67 db, respectively, which is approximately 0.1 db and 0.15 db lower than RDN, while CASR costs much less than the competitors.

4.3.2. Comparison Results on Time Complexity. Besides, we have provided a comparison of the model's efficiency in terms of time complexity on public dataset Set14 (taking ×4 as an instance), as tabulated in Table 2. The table intuitively shows that the CASR model achieves a similar competitive experiment result compared to VDSR [16] and RDN [24], reaching 28.96/0.7899 on PSNR/SSIM metrics, while spending less time (0.1017 s on a single image) and costing the least resource on processing image restoration. We may conclude that our proposed CASR model takes the least time consumption and adopts acceptable parameters compared with VDSR and RDN.

4.3.3. Superresolution on Real-World Images. Table 3 indicates that our proposed CASR method is highly competitive, achieving the lowest average values of NIQE/PI on benchmark dataset Set14 with scale factor ×4. Figure 9 illustrates the visual image restoration comparison with several SR methods on the real-world image chip. Results visually show that our method, compared to others, achieves better restorative performance. It not only achieves competitive PI and NIQE values but also improves more pleasant visual quality in terms of image, edge, texture, color, and feature-rich regions. Besides, as shown in Figure 10, the restorative performance on the larger scale, e.g., ×4, is also acceptable. The vessel in the retinal image is more clear than the competitors, and the edge of the retina is also sharp as we expected. Considering that the whole experiment was designed and conducted on dataset DIV2K, a supervised public dataset with ground truth images, which is acceptable and compromised, we believe that could provide a further research direction, exploring a more realistic oriented image SR process with a better degradation kernel on a real-world image dataset.

4.4. Ablation Study. In order to further explore the details of the experiment, we design 2 ablation experiments: one is to investigate the influence of different dilation factors on deformable convolution, and the other is the experiment of different loss functions' effects.

4.4.1. Study of the Deformable Convolution. Figure 11 illustrates two training processes with variant dilation scales. We examine whether the dilation scale of deformable

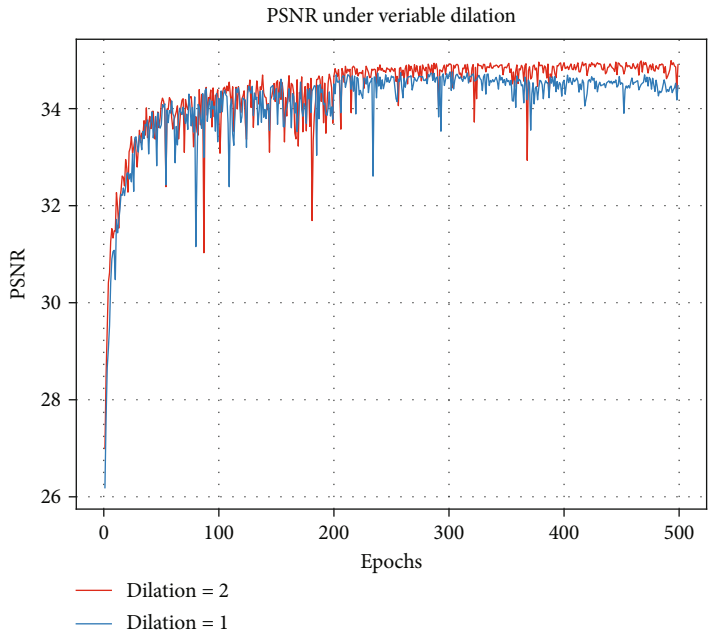


Figure 11: Comparable training results (PSNR) with scale factor ×2 under variable dilations; blue result illustrates the training process without dilation while red one shows deformable convolution with dilation 2.

Table 4: The results of ablation studies on deformable convolutions with distinct dilation factors on dataset Set5 and B100.

Scale	DeformConv	Dilation	Set5 (PSNR)	B100 (PSNR)
2	×	—	37.34	31.83
	✓	1	37.54	30.99
	✓	2	33.97	31.90
4	×	—	31.95	26.76
	✓	1	32.32	26.64
	✓	2	32.53	26.44

Table 5: Effect of the variant loss functions on the Set14 and Urban100 benchmark datasets, with a scale factor of ×2 and ×4, respectively.

Scale	Loss function	Set14	Urban100
2	L_1	33.38	31.62
	$L_1 + L_{TV}$	33.97	31.90
4	L_1	31.95	25.76
	$L_1 + L_{TV}$	28.97	26.65

convolution would affect recovery performance or not. As is shown in Figure 11, with epochs rising, both training results grow as well, while the model with dilation two would achieve better performance but cause a worse fluctuation. We also compare the effect of different scale factors on the experimental performance, as shown in Table 4. It can be learned that with the same scale factor ×2, our proposed method, which replaces the convenient convolution with deformable convolution, would achieve better results on both datasets Set5 and B100. With the dilation factor enlarging, the performances go better. This result mainly occurs since the operation may effectively and dynamically expand the receptive field. Because different input feature maps may correspond to objects with different deformation scales, for some tasks, it is essential to adaptively determine the ratio or receptive field size.

4.4.2. Study of the Loss Function. To examine the effect of the mentioned loss functions, we design an ablation experiment to explore it. Expressed formally, let the first model be "L_1" and the other one be $L_1 + L_{TV}$ (i.e., using the enhanced loss function $L_1 + L_{TV}$). We tried different combinations of scale factor and loss function to examine which one would achieve better performance on dataset DIV2K, as demonstrated in Figure 6 and Table 5. Afterward, the validation process on dataset Set14 and Urban100 proves that the enhanced loss function actually results in a clearer image with more details in Figure 5.

5. Conclusion

This paper presents two novel CNN architectures, namely, head module and CAS-Block, to improve the SISR performance. Compared with the state-of-the-art (SotA) CNN-based algorithms, our presented head module considers low-level feature expression by applying depthwise separable convolution and deformable convolution, which is demonstrated to not only effectively extract the patterns but also reduce the parameter size. At the same time, the CAS-Block employs a global residual connection and abundantly utilizes cascading connections to capture remote spatial features from former inputs. Extensive experiments have

illustrated that our presented model has effectively improved both the quality of the reconstructed images and the processing speed compared with the SotA methods in terms of quantitative indicators and realistic visual effects.

Data Availability

The image data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. U1701267 and 61962014), Guangxi Science and Technology Project (AD18216004 and AD18281079), Guangxi Bagui Scholars Special Project (2019GXNSFFA245014, AA17202024, Ji Li, 2018), Guangxi Key Laboratory of Image and Graphic Intelligent Processing (GIIP202001), and Innovation Project of GUET Graduate Education (No. 2020YCXS053).

References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[2] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020.

[3] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced gan for remote sensing image superresolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5799–5812, 2019.

[4] Z. Wang, P. Yi, K. Jiang et al., "Multi-memory convolutional neural network for video super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2530–2544, 2019.

[5] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multitemporal ultra dense memory network for video super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2503–2516, 2020.

[6] R. Keys, "Cubic convolution interpolation for digital image processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 1, pp. 1153–1160, 1982.

[7] A. Marquina and S. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *Journal of Scientific Computing*, vol. 37, no. 3, pp. 367–382, 2008.

[8] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[9] C. L. P. Chen, L. Liu, L. Chen, Y. Y. Tang, and Y. Zhou, "Weighted couple sparse representation with classified regularization for impulse noise removal," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4014–4026, 2015.

[10] R. Lan, L. Sun, Z. Liu et al., "Cascading and enhanced residual networks for accurate single-image super-resolution," *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 115–125, 2021.

[11] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, and X. Luo, "MADNet: a fast and lightweight network for single-image super resolution," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1443–1453, 2021.

[12] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image superresolution," in *European Conference on Computer Vision (ECCV)*, pp. 184–199, Springer, 2014.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1132–1140, 2017.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[15] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[16] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, 2016.

[17] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[18] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng, "Ode-inspired network design for single image super-resolution," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1732–1741, 2019.

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

[20] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: a persistent memory network for image restoration," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4549–4557, 2017.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 2016no. 6.

[22] X. Wang, K. Yu, S. Wu et al., "ESRGAN: enhanced superresolution generative adversarial networks," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taix'e and S. Roth, Eds., pp. 63–79, Springer International Publishing, 2018.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[24] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, 2018.

[25] K. Jiang, Z. Wang, P. Yi, and J. Jiang, "Hierarchical dense recursive network for image super-resolution," *Pattern Recognition*, vol. 107, p. 107475, 2020.

[26] Y. Li, E. Agustsson, S. Gu, R. Timofte, and L. Gool, *CARN: Convolutional Anchored Regression Network for Fast and Accurate Single Image Super-Resolution*, ECCV Workshops, 2018.

[27] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1664–1673, 2018.

[28] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, http://arxiv.org/abs/1704.04861.

[29] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, pp. 4278–4284, AAAI Press, 2017.

[31] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[32] J. Dai, H. Qi, Y. Xiong et al., "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017.

[33] L. Liu, B. Liu, H. Huang, and A. C. Bovik, "Noreference image quality assessment based on spatial and spectral entropies," *Signal processing: Image communication*, vol. 29, no. 8, pp. 856–863.

[34] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[35] W. Shi, J. Caballero, F. Husz'ar et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, 2016.

[36] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision (ECCV)*, pp. 391–407, Springer, 2016.

[37] H. A. Aly and E. Dubois, "Image up-sampling using total-variation regularization with a new observation model," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647–1659, 2005.

[38] R. Lan, L. Sun, Z. Liu et al., "Cascading and enhanced residual networks for accurate single-image super-resolution," *IEEE Transactions on Cybernetics*, pp. 1–11, 2020.

[39] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *International Conference on Learning Representations*, vol. 12, 2014.

[40] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. A. Morel, "Low-complexity single-image superresolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference*, pp. 135.1–135.10, BMVA Press, 2012.

[41] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," *Proceedings of the 7th International Conference on Curves and Surfaces*, , pp. 711–730, Springer-Verlag, Berlin, Heidelberg, 2010.

[42] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[43] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, 2015.

[44] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 624–632, 2017.

[45] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 723–731, 2018.