electronics

Article

A Dual CNN for Image Super-Resolution

Jiagang Song [1,†], Jingyu Xiao [2,*,†], Chunwei Tian [3,4,5], Yuxuan Hu [2], Lei You [6] and Shichao Zhang [2]

[1] School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, China; songjg@stu.gxnu.edu.cn
[2] School of Computer Science, Central South University, Changsha 410083, China; hyxx_08@163.com (Y.H.); zhangsc@csu.edu.cn (S.Z.)
[3] Research & Development Institute, Northwestern Polytechnical University, Shenzhen 518057, China; chunweitian@nwpu.edu.cn
[4] School of Software, Northwestern Polytechnical University, Xi'an 710129, China
[5] Yangtze River Delta Research Institute, Northwestern Polytechnical University, Taicang 215400, China
[6] School of Biomedical Informatics, University of Texas Houston Science Center at Houston, Houston, TX 77030, USA; Lei.You@uth.tmc.edu
* Correspondence: jyxiao@csu.edu.cn
† These authors contributed equally to this work.

Abstract: High-quality images have an important effect on high-level tasks. However, due to human factors and camera hardware, digital devices collect low-resolution images. Deep networks can effectively restore these damaged images via their strong learning abilities. However, most of these networks depended on deeper architectures to enhance clarities of predicted images, where single features cannot deal well with complex screens. In this paper, we propose a dual super-resolution CNN (DSRCNN) to obtain high-quality images. DSRCNN relies on two sub-networks to extract complementary low-frequency features to enhance the learning ability of the SR network. To prevent a long-term dependency problem, a combination of convolutions and residual learning operation is embedded into dual sub-networks. To prevent information loss of an original image, an enhanced block is used to gather original information and obtained high-frequency information of a deeper layer via sub-pixel convolutions. To obtain more high-frequency features, a feature learning block is used to learn more details of high-frequency information. The proposed method is very suitable for complex scenes for image resolution. Experimental results show that the proposed DSRCNN is superior to other popular in SR networks. For instance, our DSRCNN has obtained improvement of 0.08 dB than that of MemNet on Set5 for ×3.

Keywords: dual networks; enhanced CNN; fine learning block; image super-resolution

check for updates

Citation: Song, J.; Xiao, J.; Tian, C.; Hu, Y.; You, L.; Zhang, S. A Dual CNN for Image Super-Resolution. *Electronics* **2022**, *11*, 757. https://doi.org/10.3390/electronics11050757

Academic Editor: Soon Ki Jung

Received: 22 January 2022
Accepted: 22 February 2022
Published: 1 March 2022

1. Introduction

Due to effects of human factor and camera hardware, captured images often are not clear. To overcome these challenges, single super-resolution (SISR) techniques are presented [1]. For instance, priori knowledge is used to guide the SR model [2] Zha et al. [3] embedded a sparse idea into dictionary learning to repair high-quality images. To obtain richer information, Zhang et al. [4] combined non-local and local priors to achieve a non-local mean SR model with steering kernel regression. Zhang et al. [5] used a Monte Carlo-based Markov chain to train an SR model for improving visual effects. There are other popular SR methods, i.e., random forest [6], gradient profile [2], and regression [7]. Although these methods can repair low-resolution images well, they are faced with two challenges as follows:

(1) They referred to complex optimization methods to mine more detailed information for promoting super-resolutions of repaired images.
(2) They reply on manually chosen parameters to promote visual effects of predicted images.

To handle these problems, convolutional neural networks (CNNs) with strong self-learning abilities composed of common components are developed [8,9]. For instance, Dong et al. [10] designed a shallow architecture via pixel mapping operations to automatically obtaining clearer images rather than manual setting parameters. To pursue more perfect restoration effects, residual learning techniques and concatenation operations are applied in image restoration [11]. Tai et al. [12] used two different residual operations to fuse obtained local features to improve the learning ability of CNN in SISR. Kim et al. [13] used residual learning operations to gather hierarchical features for the final layer in order to enhance the robustness of obtained features in SISR. Although these methods have obtained remarkable results in SISR, they upsampled given low-resolution images as inputs of CNNs, which can increase computational cost [14]. To solve this problem, an upsampling operation set in the final layer of CNN is developed [15]. For instance, Dong et al. [14] set a deconvolution operation as the final layer to reduce the complexity of the whole SR network. To improve the SR performance, Zhang et al. [16] enlarged the depth of SR network and repeatedly used concatenation operations to facilitate obtained features in SISR. Although these methods perform well in SR, they may depend on deeper architectures to extract more accurate features in promoting SR performance, which may have higher requirements on hardware devices. In addition, obtained features from single architecture may not fully deal with complex screens.

In this paper, we propose a dual super-resolution CNN (DSRCNN) via three blocks (i.e., two sub-network enhanced block (TSEB), enhanced block (EB), and feature learning block (FLB) to obtain high-quality images. TSEB used two sub-networks to extract complementary low-frequency features to enhance the learning ability of SR networks. To prevent a long-term dependency problem, a combination of convolutions and residual learning operation is embedded into dual sub-networks. To prevent information loss of an original image, an enhanced block is used to gather original information and obtain high-frequency information of deeper layers via sub-pixel convolutions. To obtain more high-frequency features, a feature learning block is used to learn more details of high-frequency information.

The main contributions of the proposed DSRCNN are as follows:

(1) DSRCNN uses two sub-networks to extract complementary features to enhance the learning ability of an SR model, which is very suitable to complex screens. The multiple combinations of residual learning operation, convolutional layer, and ReLU are embedded into two sub-networks to enhance the memory abilities of shallow layers to deep layers and extract more accurate information as well as a large amount of information of dual networks in SISR.

(2) Combining low-frequency and high-frequency features to train a robust SR model.

The remainder of this paper is as follows: Section 2 gives the related work; Section 3 describes the proposed method; Section 4 shows experiments; and Section 5 presents the conclusions.

2. Related Work

2.1. Deep CNNs for Image Super-Resolution

Big data and strong hardware devices, i.e., a graphic processing unit (GPU), contribute the success of CNNs in image applications, i.e., image super-resolution [17]. These SR methods can be summarized as two kinds: upsampling low-resolution image-based CNNs and upsampling obtained low-frequency feature-based CNNs. The first method requires that input and output images have the same sizes. That is, they used upsampling operations to amplify low-resolution images as inputs of CNN to predict super-resolution images. Inspired by that, a deep network with a sparse coding algorithm was used to improve the SR execution speed and performance [18]. Kim et al. [19] combined a deeper architecture and residual learning operation to extract more accurate structure information in SR. Alternatively, Mao et al. [20] utilized skip connections to construct a symmetrical architecture to enhance the learning ability of designed CNN in SR. Although these methods were very effective in image super-resolution, they are still faced with big complexity. To address this issue, the second method is developed. The second method directly puts the given low-resolution images as inputs of CNNs and uses upsampling operations at deep layers to amplify obtained low-frequency features to obtain high-frequency features for constructing high-quality images. Motived by that, scholars conducted a lot of SR methods [21]. For instance, Tian et al. [22], respectively, enhanced low-frequency features and high-frequency features to achieve a robust SR model. Ahn et al. [22] combined residual blocks and smaller kernels to make a trade-off between SR performance and execution speed. In addition, Tian et al. [15] found local key features to obtain richer features in a horizontal and vertical way for promoting visual effects. Chen et al. [23] conducted an efficient network via multi-scale ideas. Geng et al. [24] used the combination of Shearlet and residual network to extract more accurate information in SISR. Nathan et al. [25] used the combination of attention ideas and multi-scale to improve the performance of SISR. According to mentioned illustrations, we can see that upsampling obtained low-frequency feature-based CNNs are popular in SISR. Thus, we use this idea in this paper.

2.2. Fusion of Multiple CNNs for Image Restoration

Some SR methods used a single network architecture to extract representative information to construct high-quality images. However, obtained features may be affected by different screens, which are not beneficial to complex screens. To address this issue, fusion multiple CNNs are employed in image restoration [26]. Tian et al. [27] utilized two different sub-networks to extract different features to enlarge differences of the CNN for promoting denoising effects. Pan et al. [28] used dual CNN to extract a different structure and detailed information in image restoration. Tian et al. [29] fused a signal processing idea, a sparse method into dual CNNs to remove the noise. Xin et al. [30] proposed a dual recursive network with a wavelet idea to predict high-quality images. Inspired by that, we also choose the fusion of multiple CNNs in SISR.

2.3. Peak Signal-to-Noise Ratio (PSNR)

Given a clean image I and noise image K with size MXN, MSE is defined as [27]:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{1}$$

Then, PSNR(dB) is defined as [27]:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{2}$$

2.4. Structural SIMilarity (SSIM)

The SSIM formula is based on three comparative measures between samples X and Y: luminance, contrast, and structure [27].

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{3}$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{5}$$

Usually, take $c_3 = c_2/2$. $\mu_x$is the mean of x. μ_y is the mean of y. σ_x^2 is the variance of x. σ_y^2 is the variance of y. σ_x^2 is the covariance of x and y. $c_1 = (k_1L)^2, c_2 = (k_2L)^2$

are two constants, avoiding division by zero. L is the range of pixel values, $2^B - 1$. $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ default value [27]:

$$\text{SSIM}(x,y) = \left[l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}\right] \tag{6}$$

Setting $\alpha, \beta, \gamma$to 1, you can obtain [27]:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{7}$$

In each calculation, a window of N×N is taken from the picture, and then the window is continuously sliding for calculation, and finally the average value is taken as the global SSIM [27].

3. The Proposed Method

The proposed DSRCNN is given in Figure 1. DSRCNN contains TSEB, EB, and FLB. TSEB depended on two sub-networks to extract complementary low-frequency features to improve the SR performance of DSRCNN. Specifically, the simultaneous use of convolutions and residual learning operation are used to enhance the effects of hierarchical features to prevent a long-term dependency problem. To prevent information loss of the given low-resolution image, EB employs a residual operation and sub-pixel convolutions to gather obtained different high-frequency features. FLB used several stacked convolutions to refine high-frequency features for obtaining more accurate high-frequency features in SISR. More information can be shown as follows.

3.1. Network Architecture

The proposed 23-layer DSRCNN consists of TSEB, EB, and FLB. The 17-layer TSEB utilizes dual CNNs to obtain complementary low-frequency information to promote learning ability of an SR model. In addition, using residual learning technique to fuse local hierarchical features in the TSEB can maintain memory ability of shallow layers for SISR. Then, EB fuses features of two different paths via a residual operation and sub-pixel convolutions to prevent information loss of given low-resolution image. Finally, FLB is used to refine high-frequency information to better represent predicted high-quality images. To clearer express the process above, the following symbols are defined. Let I_{LR} and I_{SR} be defined as the given LR image and predicted SR image. f_{TSEB} , f_{EB} and f_{FLB} are functions of TSEB, EB, and FLB, respectively. The execution process of the DSRCNN can be expressed as follows:

$$\begin{aligned} I_{SR} &= f_{FLB}(f_{EB}(f_{TSEB}(I_{LR}))) \\ &= f_{DSRCNN}(I_{LR}) \end{aligned} \tag{8}$$

where f_{DSRCNN} stands for the function of DSRCNN. In addition, DSRCNN relies on the following loss function to find optimal parameters.

3.2. Loss Function

The mean squared error (MSE) [31] is used to test the difference between a real high-quality image and predicted SR image for finding optimal parameters. The MSE value is computed via a training pair of $\{I_{LR}^k, I_{HR}^k\}_{(k=1)}^N$, where I_{LR^k} and I_{HR^k} express the k-th given low-resolution image and high-resolution image, respectively. In addition, N is the total of training samples. In addition, we minimize the loss function to train DSRCNN as follows:

$$l(p) = \frac{1}{2N} \sum_{(k=1)}^{N} \|f_{DSRCNN}(I_{LR}^k - I_{HR}^k)\|^2 \tag{9}$$

where l stands for loss function, and p is used to represent the parameter set of training a DSRCNN.

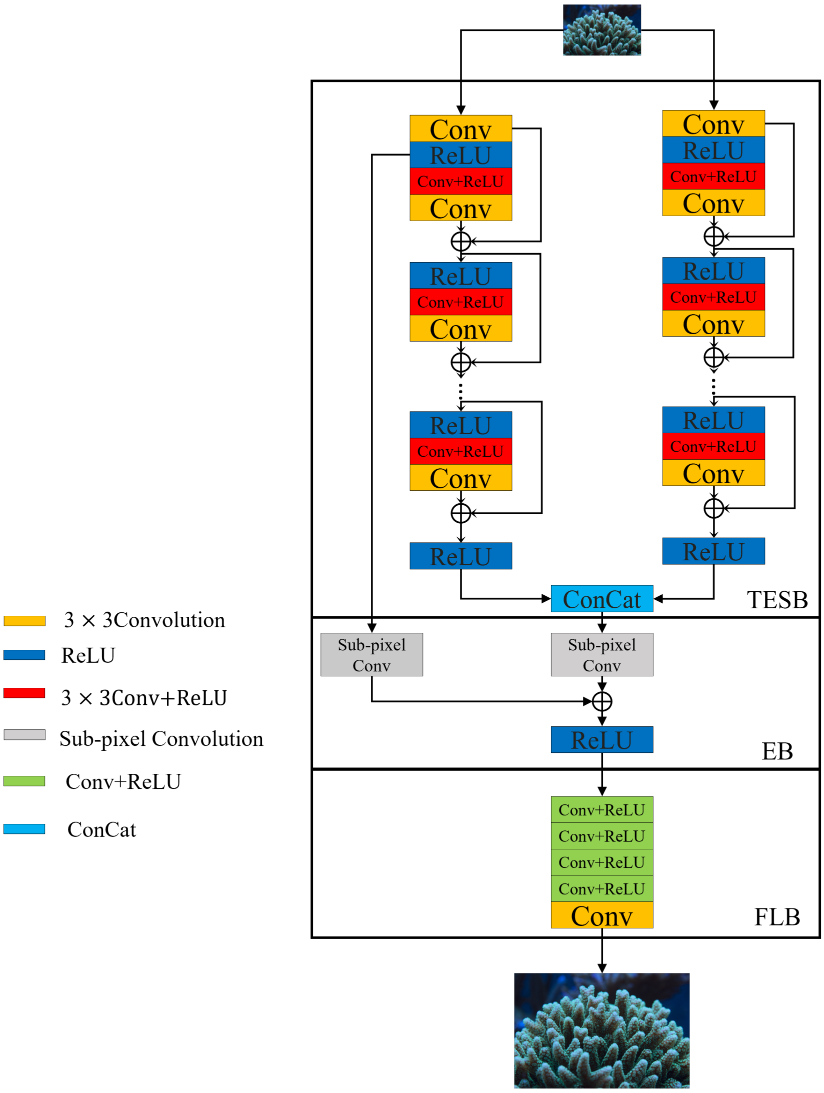


Figure 1. Network architecture of the proposed DSRCNN.

3.3. Two Sub-Network Enhanced Block

According to previous illustrations, it is known that obtained features of single architecture cannot fully deal with complex screens. In this paper, a two sub-network enhanced block is used to overcome this phenomenon. The two sub-network enhanced block consists of two phases. The first phase fuses two sub-networks via a concatenation operation to extract robust low-frequency features in SISR, where a concatenation operation is shown as Figure 2. The second phase is used to gather local hierarchical information to enhance the memory ability of shallow layers for improving the SR effect. Each sub-network from two phases is composed of 17 combinations of convolution and Rectified Linear Unit (ReLU) [32], and a single convolution layer. In addition, input and output channels of the first convolutional layer in each sub-network are 3 and 64. Input and output channels from 2nd to 16th convolutional layers are 64. Two sub-networks are fused via a concatenation operation at the end of the 16th convolutional layer. Thus, the input channel of the 17th convolutional layer is 128. To flexibility operate the convolution layer, the output channel of the 17th convolutional layer is 64. The input and output channels of the 18th convolutional layer are 128 and 64. In addition, kernel sizes of all convolutions are 3 × 3. The mentioned process can be explained via the following Equation (10).

$$\begin{aligned} O_{TSEB} &= f_{TSEB}(I_{LR}) \\ &= (Cat(O_{(TSEB_1)}, O_{(TSEB_2))}) \end{aligned} \tag{10}$$

where O_{TSEB_1} and O_{TSEB_2} express the outputs of two sub-networks. Cat denotes a concatenation operation as well as operation in Figures 1 and 2. O_{TSEB} stands for output of TSEB. More detailed information of two sub-networks can be shown in the second phase of TSEB. The second phase repeatedly uses a combination of residual operation and convolutions to enhance the effect of local hierarchical information for improving SR effects. That is, obtained features of odd convolutional layers from the 1st layer to the 17th layer can be gathered via residual operations to enhance robustness of obtained features for SISR. According to Figure 1 and illustrations above, the second phase in two sub-networks can be shown as follows:

$$O_{(TSEB_k)} = R(O_{L1} + O_{L3} + O_{L5} + O_{L7} + O_{Li} \ldots + O_{17}) \tag{11}$$

where $O_{(TSEB_k)}$ denotes the output of the kth each sub-network, and O_{Li} is the output of the ith convolution layer, where $i = 2, \ldots, 17$. $O_{L1} = R(C(I_{LR}))$ and $O_{Li} = C(O_{(Li-1)})$, where R denotes the function of ReLU. In addition, $O_j = R(C(O_{(j-1)}))$, where $j = 2, 4, 6, 8, \ldots, 16$. In addition, the output TSEB acts an enhanced block as follows.

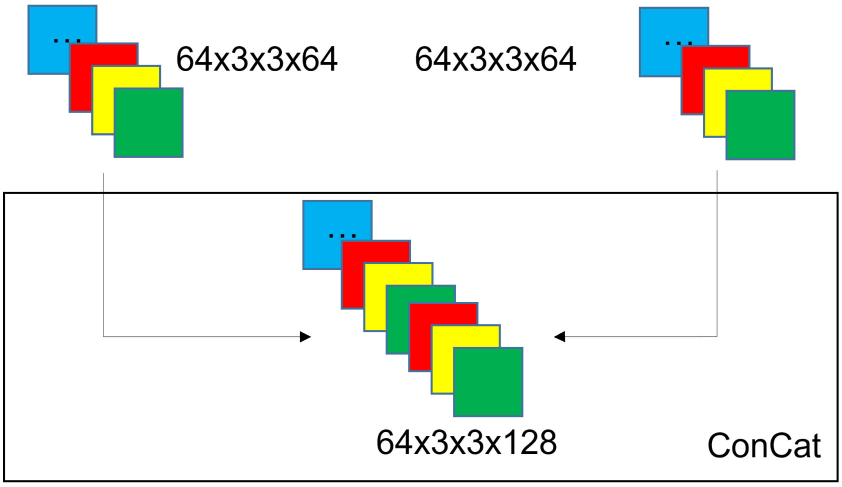


Figure 2. Sketch figure of concatenation operation.

3.4. Enhanced Block

A one-layer enhanced block is used to fuse information of an original image and obtained information from a deeper layer. This is implemented via the following steps. The first step obtains the information of the original image and a deeper layer, respectively. That is, we use sub-pixel operation to respectively amplify obtained features of the TSEB and the first layer of the below sub-network in Figure 1 as follows:

$$O_{(EB_s)} = Subp(R(O_{L1})) \tag{12}$$

$$O_{(EB_d)} = Subp(O_{TSEB}) \tag{13}$$

where $O_{(EB_s)}$ and $O_{(EB_d)}$ are obtained high-frequency features of shallow and deep layers, respectively. $Subp$ denotes sub-pixel convolutional techniques, which is expressed as Subpixel Conv in Figure 1. In addition, the sub-pixel convolution consists of convolution with a kernel of 3 × 3, and the input and output channels in Equation (12) total 64. Input and output channels in Equation (13) are 128 and 64. The second step fuses the obtained features above via a residual operation to obtain more commentary information as follows:

$$\begin{aligned} O_{EB} &= f_{EB}(O_{TSEB}) \\ &= R(O_{(EB_s)} + O_{(EB_d)}) \end{aligned} \tag{14}$$

where O_{EB} is the output of EB. In addition, $+$ denotes a residual operation, which is expressed as $\oplus$ in Figure 1. In addition, O_{EB} acts as a feature learning block.

3.5. Feature Learning Block

To further learn high-frequency features, a 5-layer feature learning block is presented. It includes four Conv+ReLU and a Conv. Conv+ReLU denotes the combination of convolution and ReLU, where their input and output channels total 64, and a convolutional kernel is 3×3 . In addition, a Conv denotes a convolution, where its input and output channels are 64 and 3, and a convolutional kernel is 3×3. This is used to construct predicted high-quality images. The mentioned descriptions are visualized as Equation (15):

$$\begin{aligned} I_{SR} &= f_{FLB}(O_{EB}) \\ &= C(R(C(R(C(R(C(R(C(O_{EB}))))))))) \end{aligned} \tag{15}$$

where C is convolutional operation as well as Conv in Figure 1. Finally, we give a pseudo-code to show implementations of the proposed method as shown in Algorithm 1:

Algorithm 1 The process of converting an LR image into an SR image.

```
Input: Put an LR image $I_{LR}$ into the DSRCNN model. Enter the scale factor
for Patch is 64 do
    The residual network is used to retain low-level features and fuse high-level features.
    The feature outputs of the two models are merged through ConCat.
    Updated $I_{LR}$ feature (the first model $O_{TSEB_1}$ & the second model $O_{TSEB_2}$ )through a two sub-network
    enhanced block by Equation (10);
    The enhanced block contains two upsampling layers.The first upsampling roughly extracts the spliced
    low-frequency features and converts them into high-frequency features. The second upsampling extracts the
    features of the first layer and stacks them with the first upsampling.
    Updated $O_{EB_s}$ and $O_{EB_d}$ through an enhanced block by Equation (12) and (13);
    After five layers of convolutional layers (feature learning block ), the high-frequency features are further
    learned and extracted by Equation (15).
end for
Output: Obtain a super-resolution image $I_{SR}$ with an inpainting scale as the input scale.
```

4. Experimental Results

4.1. Training Dataset

All DIV2K images are saved in PNG format and DIV2K dataset of RGB images with a large diversity of contents. To conduct fair experiments, the DIV2K dataset [33] is chosen to train a DSRCNN model. DIV2K consists of three scales, i.e., ×2, ×3 and ×4. Each scale includes 800 training images. In addition, test images and validation images are 100 natural images. To enlarge differences of training images, we gather the given training dataset of DIV2K and the validation dataset as a new training dataset under the same scale. In addition, some data augmentation operations, i.e., random horizontal flips and 90° rotation operations, are used to enhance training data. To improve the speed of training DSRCNN, given LR images are cropped as an image patch with 64×64.

4.2. Test Dataset

To fairly and effectively test the SR performance, Set5 [34], Set14 [34], BSD100 (B100) [35] and Urban 100 (B100) [36] are chosen to conduct comparative experiments for ×2, ×3, and ×4. The Set5 and Set14 are captured under the same conditions, which have five and fourteen natural images. B100 and U100 respectively contain 100 natural color images. These datasets can be further introduced as follows:

- The Set5 dataset is a dataset consisting of five images, i.e., baby, bird, butterfly, head, and woman [34].
- The Set14 dataset is a dataset consisting of 14 images, and it is commonly used for testing performance of Image Super-Resolution models [34].
- BSD is a dataset used frequently for image denoising and super-resolution. BSD100 have 100 images, which was conducted by Martin et al. The dataset is composed of a large variety of images ranging from natural images to object-specific such as plants, people, food, etc.
- The Urban100 dataset contains 100 images of urban scenes. It is commonly used as a test set to evaluate the performance of super-resolution models [36].

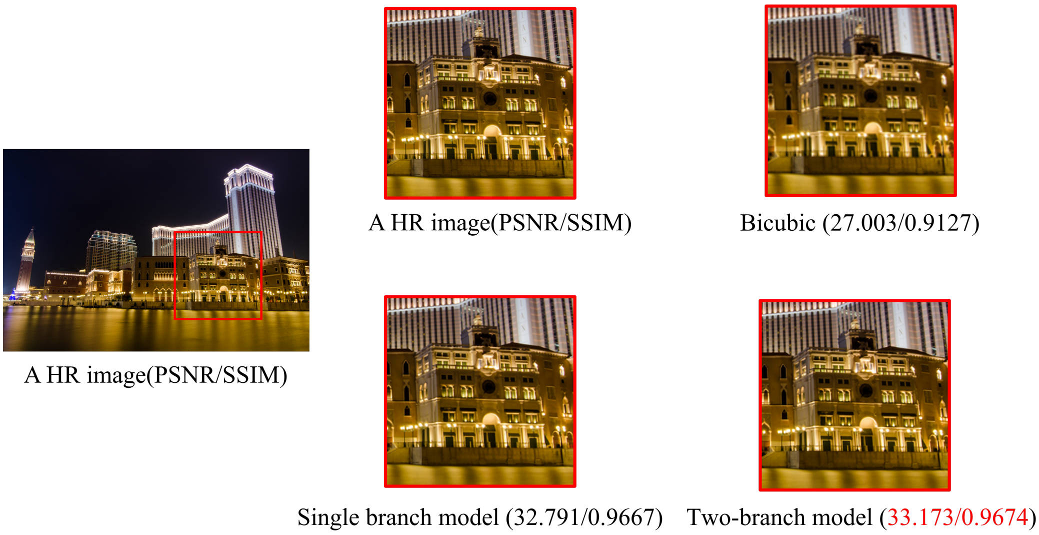
Because most of the SR methods use the Y channel to test the SR performance of their proposed methods, we also choose a Y channel to test the effect of our method on SR. That is, obtained RGB images of the DSRCNN are converted to the Y channel to verify the SR performance.

4.3. Implementation Details

This paper has the following initial parameters. We set a batch size as 64. In addition, the initial learning rate is 1×10^3, Beta_1 is 0.9, and Beta_2 is 0.999. In addition, the training steps are 6×10^5, for which the learning rate will be divided in half every 4×10^5. That is, or $1 \sim 4 \times 10^5$, the learning rate is 0.0001 for training a DSRCNN model. For (4×10^5)+1 to 6×10^5, the learning rate is 0.00005. Epsilon size is 1×10^8. Additionally, other initial parameters can refer to Ref. [8]. In addition, we use an Adam optimizer [37] to update parameters. Codes of LSRCNN are programmed via Python of 0.41 and Python of 2.7. In addition, it runs on Ubuntu of 16.04, CPU of Inter Xeon 8163 and two NVIDIA Tesla P100. The Nvidia CUDA is 9.0 and CuDNN is 7.6.4.

4.4. Ablation Study

The proposed uses of TSEB, EB, and FLB to implement a robust SR model. In addition, TSEB is composed of two sub-networks and an enhanced technique. The enhanced technique uses local residual learning operation to enhance effects of local hierarchical layers to promote SR performance. These are verified in Table 1. DSRCNN is obtained higher PSNR and SSIM values than that of DSRCNN with residual learning operations, which shows the effectiveness of local residual learning operations. DSRCNN outperforms DSRCNN without RLO, one sub-network in Table 1, which shows the effectiveness of two sub-networks and RLO. It is known that enlarging the depth of a network is very useful to extract complementary information [38,39]. It is known that increasing the width of network can improve the performance of image tasks, according to GoogLeNet [40]. Although wider networks perform well in image applications, they will increase the complexity. In addition, two sub-networks can effectively address this question in image restoration. Thus, taking into account performance and complexity, we also choose two sub-networks to design network architecture in this paper. According to mentioned illustrations, we design two sub-networks for SISR [41–43]. In addition, it is proved that DSRCNN without RLO, EB_S outperforms improvement of 0.511dB compared to that of DSRCNN without RLO, one sub-network and EB_S in PSNR in Table 1, which shows the effectiveness of dual networks for SR. DSRCNN without RLO, one sub-network exceeds DSRCNN without RLO, one sub-network and EB_S in both PSNR and SSIM on U100 in Table 1, which tests the effectiveness of EB. Additionally, DSRCNN without RLO, one sub-network, and EB_S have obtained improvement of 0.16 dB in PSNR and 0.001 in SSIM than that of DSRCNN without RLO, one sub-network EB_S and RO in both PSNR and SSIM on U100 in Table 1, which tests the effectiveness of FLB. Specifically, EB_S and RO denote EB without enhancement from the shallow layer and FLB without four Conv+ReLU. In addition, several visual figures from an HR image, Bicubic, single branch model, and a two-branch model are conducted to test excellent performance of two-branch architecture, which can show the complementary of two branches as Figure 3 on page 9. In addition, these visual figures are obtained via amplifying one area of predicted high super-resolution images as observation areas. According to mentioned illustrations, we can see the effectiveness of key techniques for SISR.


A HR image(PSNR/SSIM)

A HR image(PSNR/SSIM)

Bicubic (27.003/0.9127)

Single branch model (32.791/0.9667)

Two-branch model (33.173/0.9674)

Figure 3. Visual figures of different methods.

Table 1. PSNR and SSIM of different methods on U100 for ×2.

Methods		U100
		PSNR/SSIM
Scale	DSRCNN	31.833/0.9252
	DSRCNN without RLO, one sub-network	31.676/0.9237
	DSRCNN without RLO, one sub-network and EB_S	31.220/0.9181
	DSRCNN without RLO, one sub-network EB_S and RO	31.060/0.9171
	DSRCNN without RLO, EB_S	31.649/0.9238
	DSRCNN without RLO	31.701/0.9241

4.5. Experiment Results

To fairly evaluate the SR performance of DSRCNN, quantitative and qualitative analysis are used to conduct experiments. The quantitative analysis includes PSNR [44] and SSIM [44] of popular methods, i.e., Bicubic, A+ [7], jointly optimized regressors (JOR) [45], RFL [6], self-exemplars super-resolution (SelfEx) [36], CSCN [18], RED [19], a denoising convolutional neural network (DnCNN) [46], trainable nonlinear reaction diffusion (TNRD) [47], fast dilated residual SR convolutional network (FDSR) [48], SRCNN [10], fast SR CNN (FSRCNN) [14], very deep SR network (VDSR) [19], deeply-recursive convolutional network (DRCN) [13], context wise network fusion (CNF) [49], Laplacian SR network (LapSRN) [50], deep persistent memory network (MemNet) [11], CARN-M [22], wavelet domain residual network (WaveResNet) [51], convolutional principal component (CPCA) [52], new architecture of deep recursive convolution networks for SR (NDRCN) [53], LESRCNN [8], LESRCNN-S [8], and DSRCNN on four public datasets, i.e., Set5, Set14, B100, and U100. In terms of quantitative analysis, our proposed DSRCNN has obtained the best SR results in most circumstances as shown in Tables 2–5. For example, our method has obtained gain PSNR of 0.08 dB than that of LESRCNN on Set 5 for ×2 in Table 2 In addition, DSRCNN has achieved gain PSNR of 0.16 dB and SSIM of 0.0035 than that of CARN-M on Set14 for ×3 in Table 3. In addition, our method has obtained an excellent SR performance on B100 in Table 4 and on U100 in Table 5, respectively. Our method is very competitive in complexity in Table 6. In terms of qualitative analysis, we choose Bicubic, SelfEx, SRCNN, and CARN-M as comparative methods to test the visual effects of DSRCNN. Amplify a chosen area from predicted high-resolution images from these methods as an observation area, where observation area is clearer and corresponding SR methods are better in SISR. As shown in Figures 4–6, we can see that the observation areas of our method are clearer than other SR methods. This shows that our method is more effective in SISR. According to quantitative analysis and qualitative analysis, our method is robust in different screens.

According to descriptions, we can see that the proposed method can reply with two sub-networks, the combination of residual learning, convolutional layer, and ReLU to obtain excellent SR performance. However, it has slower execution speed in SR than that of the single network with the same parameters. Thus, how to develop an efficient and robust SR network is very important for us in our work in the future.

Table 2. PSNR and SSIM of different techniques with scale factors of ×2, ×3 and ×4 on Set5.

Dataset	**Model**	**×2**	**×3**	**×4**
		PSNR/SSIM	**PSNR/SSIM**	**PSNR/SSIM**
Set5	Bicubic	33.66/0.9299	30.39/0.8682	28.42/0.8104
	A+ [7]	36.54/0.9544	32.58/0.9088	30.28/0.8603
	JOR [45]	36.58/0.9543	32.55/0.9067	30.19/0.8563
	RFL [6]	36.54/0.9537	32.43/0.9057	30.14/0.8548
	SelfEx [36]	36.49/0.9537	32.58/0.9093	30.31/0.8619
	CSCN [18]	36.93/0.9552	33.10/0.9144	30.86/0.8732
	RED [19]	37.56/0.9595	33.70/0.9222	31.33/0.8847
	DnCNN [46]	37.58/0.9590	33.75/0.9222	31.40/0.8845
	TNRD [47]	36.86/0.9556	33.18/0.9152	30.85/0.8732
	FDSR [48]	37.40/0.9513	33.68/0.9096	31.28/0.8658
	SRCNN [10]	36.66/0.9542	32.75/0.9090	30.48/0.8628
	FSRCNN [14]	37.00/0.9558	33.16/0.9140	30.71/0.8657
	RCN [54]	37.17/0.9583	33.45/0.9175	31.11/0.8736
	VDSR [19]	37.53/0.9587	33.66/0.9213	31.35/0.8838
	DRCN [13]	37.63/0.9588	33.82/0.9226	31.53/0.8854
	CNF [49]	37.66/0.9590	33.74/0.9226	31.55/0.8856
	LapSRN [50]	37.52/0.9590	-	31.54/0.8850
	MemNet [11]	37.78/0.9597	34.09/0.9248	31.74/0.8893
	CARN-M [22]	37.53/0.9583	33.99/0.9236	31.92/0.8903
	WaveResNet [51]	37.57/0.9586	33.86/0.9228	31.52/0.8864
	CPCA [52]	34.99/0.9469	31.09/0.8975	28.67/0.8434
	NDRCN [53]	37.73/0.9596	33.90/0.9235	31.50/0.8859
	LESRCNN [8]	37.65/0.9586	33.93/0.9231	31.88/0.8903
	LESRCNN-S [8]	37.57/0.9582	34.05/0.9238	31.88/0.8907
	DSRCNN(Ours)	37.73/0.9588	34.17/0.9247	31.89/0.8909

Table 3. PSNR and SSIM of different techniques with scale factors of ×2, ×3, and ×4 on Set14.

Dataset	**Model**	**×2**	**×3**	**×4**
		PSNR/SSIM	**PSNR/SSIM**	**PSNR/SSIM**
Set14	Bicubic	30.24/0.8688	27.55/0.7742	26.00/0.7027
	A+ [7]	32.28/0.9056	29.13/0.8188	27.32/0.7491
	JOR [45]	32.38/0.9063	29.19/0.8204	27.27/0.7479
	RFL [6]	32.26/0.9040	29.05/0.8164	27.24/0.7451
	SelfEx [36]	32.22/0.9034	29.16/0.8196	27.40/0.7518
	CSCN [18]	32.56/0.9074	29.41/0.8238	27.64/0.7578
	RED [19]	32.81/0.9135	29.50/0.8334	27.72/0.7698
	DnCNN [46]	33.03/0.9128	29.81/0.8321	28.04/0.7672
	TNRD [47]	32.51/0.9069	29.43/0.8232	27.66/0.7563
	FDSR [48]	33.00/0.9042	29.61/0.8179	27.86/0.7500
	SRCNN [10]	32.42/0.9063	29.28/0.8209	27.49/0.7503
	FSRCNN [14]	32.63/0.9088	29.43/0.8242	27.59/0.7535
	RCN [54]	32.77/0.9109	29.63/0.8269	27.79/0.7594
	VDSR [19]	33.03/0.9124	29.77/0.8314	28.01/0.7674
	DRCN [13]	33.04/0.9118	29.76/0.8311	28.02/0.7670
	CNF [49]	33.38/0.9136	29.90/0.8322	28.15/0.7680
	LapSRN [50]	33.08/0.9130	29.63/0.8269	28.19/0.7720
	MemNet [11]	33.28/0.9142	30.00/0.8350	28.26/0.7723
	CARN-M [22]	33.26/0.9141	30.08/0.8367	28.42/0.7762
	WaveResNet [51]	33.09/0.9129	29.88/0.8331	28.11/0.7699
	CPCA [52]	31.04/0.8951	27.89/0.8038	26.10/0.7296
	NDRCN [53]	33.20/0.9141	29.88/0.8333	28.10/0.7697
	LESRCNN [8]	33.320.9148	30.12/0.8380	28.44/0.7772
	LESRCNN-S [8]	33.30/0.9145	30.16/0.8384	28.43/0.7776
	DSRCNN(Ours)	33.43/0.9157	30.24/0.8402	28.46/0.7796

Table 4. PSNR and SSIM of different techniques with scale factors of ×2, ×3, and ×4 on B100.

Dataset	**Model**	**×2**	**×3**	**×4**
		PSNR/SSIM	**PSNR/SSIM**	**PSNR/SSIM**
B100	Bicubic	29.56/0.8431	27.21/0.7385	25.96/0.6675
	A+ [7]	31.21/0.8863	28.29/0.7835	26.82/0.7087
	JOR [45]	31.22/0.8867	28.27/0.7837	26.79/0.7083
	RFL [6]	31.16/0.8840	28.22/0.7806	26.75/0.7054
	SelfEx [36]	31.18/0.8855	28.29/0.7840	26.84/0.7106
	CSCN [18]	31.40/0.8884	28.50/0.7885	27.03/0.7161
	RED [19]	31.96/0.8972	28.88/0.7993	27.35/0.7276
	DnCNN [46]	31.90/0.8961	28.85/0.7981	27.29/0.7253
	TNRD [47]	31.40/0.8878	28.50/0.7881	27.00/0.7140

Table 4. *Cont.*

Dataset	Model	×2	×3	×4
		PSNR/SSIM	**PSNR/SSIM**	**PSNR/SSIM**
	FDSR [48]	31.87/0.8847	28.82/0.7797	27.31/0.7031
	SRCNN [10]	31.36/0.8879	28.41/0.7863	26.90/0.7101
	FSRCNN [14]	31.53/0.8920	28.53/0.7910	26.98/0.7150
	VDSR [19]	31.90/0.8960	28.82/0.7976	27.29/0.7251
	DRCN [13]	31.85/0.8942	28.80/0.7963	27.23/0.7233
	CNF [49]	31.91/0.8962	28.82/0.7980	27.32/0.7253
	LapSRN [50]	31.80/0.8950	-	27.32/0.7280
	MemNet [11]	32.08/0.8978	28.96/0.8001	27.40/0.7281
	CARN-M [22]	31.92/0.8960	28.91/0.8000	27.44/0.7304
	WaveResNet [51]	32.15/ 0.8995	28.86/0.7987	27.32/0.7266
	NDRCN [53]	32.00/0.8975	28.86/0.7991	27.30/0.7263
	LESRCNN [8]	31.95/0.8964	28.91/0.8005	27.45/0.7313
	LESRCNN-S [8]	31.95/0.8965	28.94/ 0.8012	27.47/0.7321
	DSRCNN(Ours)	32.05/0.8978	29.01/0.802927.50/0.7341	

Table 5. PSNR and SSIM of different techniques with scale factors of ×2, ×3, and ×4 on U100

Dataset	Model	×2	×3	×4
		PSNR/SSIM	**PSNR/SSIM**	**PSNR/SSIM**
U100	Bicubic	26.88/0.8403	24.46/0.7349	23.14/0.6577
	A+ [7]	29.20/0.8938	26.03/0.7973	24.32/0.7183
	JOR [45]	29.25/0.8951	25.97/0.7972	24.29/0.7181
	RFL [6]	29.11/0.8904	25.86/0.7900	24.19/0.7096
	SelfEx [36]	29.54/0.8967	26.44/0.8088	24.79/0.7374
	DnCNN [46]	30.74/0.9139	27.15/0.8276	25.20/0.7521
	TNRD [47]	29.70/0.8994	26.42/0.8076	24.61/0.7291
	FDSR [48]	30.91/0.9088	27.23/0.8190	25.27/0.7417
	SRCNN [10]	29.50/0.8946	26.24/0.7989	24.52/0.7221
	FSRCNN [14]	29.88/0.9020	26.43/0.8080	24.62/0.7280
	VDSR [19]	30.76/0.9140	27.14/0.8279	25.18/0.7524
	DRCN [13]	30.75/0.9133	27.15/0.8276	25.14/0.7510
	LapSRN [50]	30.41/0.9100	-	25.21/0.7560
	MemNet [11]	31.31/0.9195	27.56/0.8376	25.50/0.7630
	CARN-M [22]	31.23/0.9193	27.55/0.8385	25.62/0.7694
	WaveResNet [51]	30.96/0.9169	27.28/0.8334	25.36/0.7614
	CPCA [52]	28.17/0.8990	25.61/0.8123	23.62/0.7257
	NDRCN [53]	31.06/0.9175	27.23/0.8312	25.16/0.7546
	LESRCNN [8]	31.45/0.9206	27.70/0.8415	25.77/0.7732
	LESRCNN-S [8]	31.45/0.9207	27.76/0.8424	25.78/0.7739
	DSRCNN(Ours)	31.83/0.9252	27.99/0.8483	25.94/0.7815

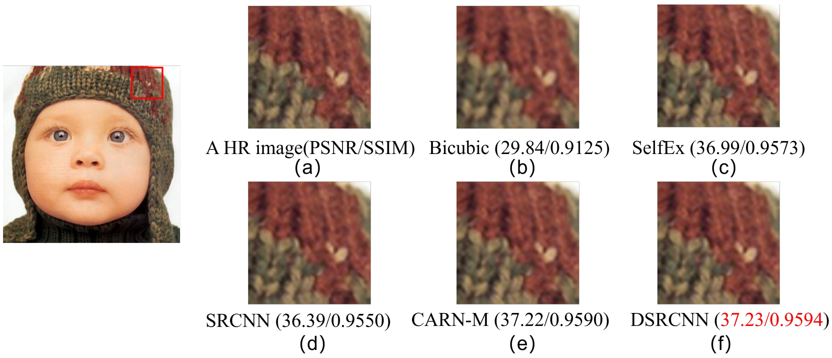


Figure 4. Visual effects of the different SR method on Set5 for ×2 scale: (**a**) A HR image (PSNR/SSIM), (**b**) Bicubic (29.84/0.9125), (**c**) SelfEx (36.99/0.9573), (**d**) SRCNN (36.39/0.9550), (**e**) CARN-M (37.22/0.9590), and (**f**) DSRCNN (37.23/0.9594).

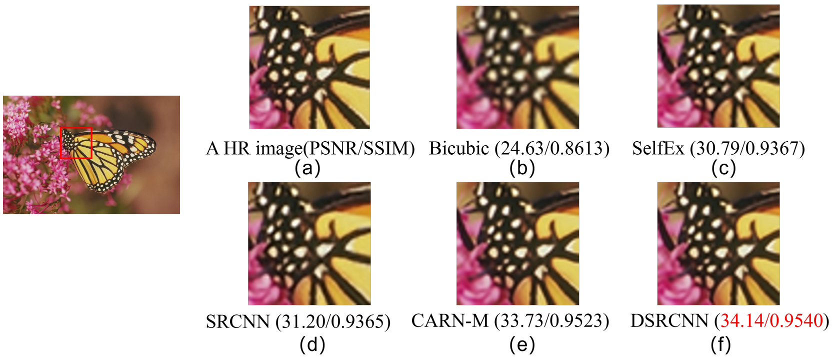


Figure 5. Visual effects of the different SR method on Set14 for ×4 scale: (**a**) A HR image (PSNR/SSIM), (**b**) Bicubic (24.63/0.8613), (**c**) SelfEx (30.79/0.9367), (**d**) SRCNN (31.20/0.9365), (**e**) CARN-M (33.73/0.9523), and (**f**) DSRCNN (34.14/0.9540).

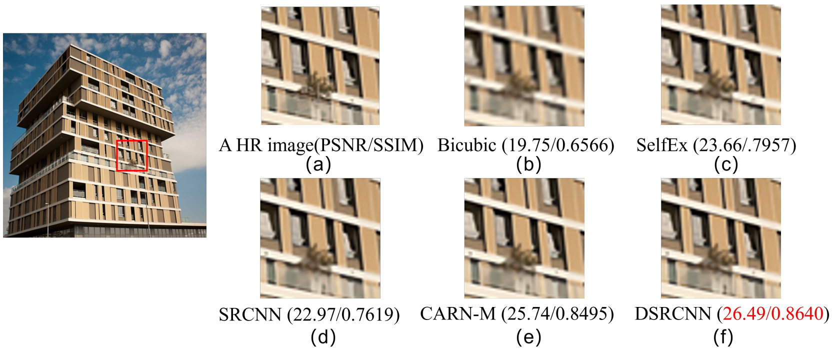


Figure 6. Visual effects of different SR method on U100 for ×4 scale: (**a**) A HR image (PSNR/SSIM), (**b**) Bicubic (19.75/0.6566), (**c**) SelfEx (23.66/.7957), (**d**) SRCNN (22.97/0.7619), (**e**) CARN-M(25.74/0.8495), and (**f**) DSRCNN (26.49/0.8640).

Table 6. Complexity of six networks for SISR.

Methods	Parameters	Flops
VDSR	665K	10.90G
DnCNN	556K	9.11G
DRCN	1,774K	29.07G
MemNet	677K	11.09G
LESRCNN	598K	3.56G
DSRCNN(our)	798 k	4.76G

5. Conclusions

In this paper, we propose a dual super-resolution CNN (DSRCNN) to obtain clear images. DSRCNN uses a two sub-network enhanced block (TSEB) to extract complementary low-frequency features to improve learning ability in SR. Combinations of convolutions and residual learning operation in TSEB are used to facilitate memory abilities of shallow layers, which can prevent a long-term dependency problem. To prevent information loss of an original image, an enhanced block is used to gather original information and obtain high-frequency information from a deeper layer via sub-pixel convolutions. To obtain more high-frequency features, a feature learning block is used to learn more details of high-frequency information. The proposed method can be applied to portable devices. We will use an attention mechanism to obtain more robust SR models in the future.

Author Contributions: Conceptualization, J.S. and J.X.; methodology, J.X.; software, J.S.; validation, C.T., Y.H. and L.Y.; formal analysis, J.S.; investigation, J.X.; resources, J.S.; data curation, J.S.; writing—original draft preparation, J.S.; writing—review and editing, C.T.; visualization, S.Z.; supervision, C.T.; project administration, S.Z.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported in part by the Shenzhen Science and Technology Program under Grant 2021A1515110079, in part by the Fundamental Research Funds for the Central Universities under Grant D5000210966, in part by the Basic Research Plan in Taicang under Grant TC2021JC23, and in part by the in part by the Key Project of NSFC under Grant 61836016.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: There are no conflicts of interest.

References

1. Van Ouwerkerk, J. Image super-resolution survey. *Image Vis. Comput.* **2006**, *24*, 1039–1052. [CrossRef]
2. Sun, J.; Xu, Z.; Shum, H.-Y. Image super-resolution using gradient profile prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
3. Zha, Z.; Yuan, X.; Wen, B.; Zhou, J.; Zhang, J.; Zhu, C. A benchmark for sparse coding: When group sparsity meets rank minimization. *IEEE Trans. Image Process.* **2020**, *29*, 5094–5109. [CrossRef]
4. Zhang, K.; Gao, X.; Tao, D.; Li, X. Single image super-resolution with non-local means and steering kernel regression. *IEEE Trans. Image Process.* **2012**, *21*, 4544–4556. [CrossRef]
5. Zhang, H.; Zhang, Y.; Li, H.; Huang, T.S. Generative bayesian image super resolution with natural image prior. *IEEE Trans. Image Process.* **2012**, *21*, 4054–4067. [CrossRef] [PubMed]
6. Schulter, S.; Leistner, C.; Bischof, H. Fast and accurate image upscaling with super-resolution forests. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3791–3799.
7. Timofte, R.; Smet, V.D.; Van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 111–126.
8. Tian, C.; Zhuge, R.; Wu, Z.; Xu, Y.; Zuo, W.; Chen, C.; Lin, C.-W. Lightweight image super-resolution with enhanced cnn. *Knowl.-Based Syst.* **2020**, *205*, 106235. [CrossRef]

9. Lan, R.; Sun, L.; Liu, Z.; Lu, H.; Pang, C.; Luo, X. Madnet: A fast and lightweight network for single-image super resolution. *IEEE Trans. Cybern.* **2020**, *51*, 1443–1453. [CrossRef] [PubMed]
10. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef]
11. Tai, Y.; Yang, J.; Liu, X.; Xu, C. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4539–4547.
12. Tai, Y.; Yang, J.; Liu, X. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3147–3155.
13. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645.
14. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 391–407.
15. Tian, C.; Xu, Y.; Zuo, W.; Lin, C.-W.; Zhang, D. Asymmetric CNN for image superresolution. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**. [CrossRef]
16. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual dense network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2472–2481.
17. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.-W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [CrossRef]
18. Wang, Z.; Liu, D.; Yang, J.; Han, W.; Huang, T. Deep networks for image super-resolution with sparse prior. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 370–378.
19. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26th June 2016; pp. 1646–1654.
20. Mao, X.; Shen, C.; Yang, Y.-B. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2802–2810.
21. Wang, Z.; Chen, J.; Hoi, S.C. Deep learning for image super-resolution: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3365–3387. [CrossRef] [PubMed]
22. Tian, C.; Xu, Y.; Zuo, W.; Zhang, B.; Fei, L.; Lin, C.-W. Coarse-to-fine cnn for image super-resolution. *IEEE Trans. Multimed.* **2020**, *23*, 1489–1502. [CrossRef]
23. Chen, W.; Yao, P.; Gai, S.; Da, F. Multi-scale feature aggregation network for image super-resolution. *Appl. Intell.* **2022**, *52*, 3577–3586. [CrossRef]
24. Geng, T.; Liu, X.-Y.; Wang, X.; Sun, G. Deep shearlet residual learning network for single image super-resolution. *IEEE Trans. Image Process.* **2021**, *30*, 4129–4142. [CrossRef] [PubMed]
25. Nathan, S.; Kansal, P. Leveraging multi scale backbone with multilevel supervision for thermal image super resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, place virtually, 19–25 June 2021; pp. 4332–4338.
26. Ahn, N.; Kang, B.; Sohn, K.-A. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 6 August 2018; pp. 252–268.
27. Tian, C.; Xu, Y.; Zuo, W. Image denoising using deep cnn with batch renormalization. *Neural Netw.* **2020**, *121*, 461–473. [CrossRef]
28. Pan, J.; Liu, S.; Sun, D.; Zhang, J.; Liu, Y.; Ren, J.; Li, Z.; Tang, J.; Lu, H.; Tai, Y.-W.; et al. Learning dual convolutional neural networks for low-level vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, 18–22 June 2018; pp. 3070–3079.
29. Tian, C.; Xu, Y.; Zuo, W.; Du, B.; Lin, C.-W.; Zhang, D. Designing and training of a dual cnn for image denoising. *Knowl.-Based Syst.* **2021**, *226*, 106949. [CrossRef]
30. Xin, J.; Li, J.; Jiang, X.; Wang, N.; Huang, H.; Gao, X. Wavelet-based dual recursive network for image super-resolution. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 707–720. [CrossRef]
31. Douillard, C.; Jézéquel, M.; Berrou, C.; Electronique, D.; Picart, A.; Didier, P.; Glavieux, A. Iterative correction of intersymbol interference: Turbo-equalization. *Eur. Trans. Telecommun.* **1995**, *6*, 507–511. [CrossRef]
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
33. Agustsson, E.; Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 126–135.
34. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. Low-complexity single-image super-resolution based on non-negative neighbor embedding. In Proceedings of the British Machine Vision Conference, Surrey, UK, 3–7 September 2012; pp. 135.1–135.10.
35. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; pp. 416–423.
36. Huang, J.-B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 5197–5206.

37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Shen, Z.; Wang, W.; Lu, X.; Shen, J.; Ling, H.; Xu, T.; Shao, L. Human-aware motion deblurring. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 5572–5581.
39. Zhou, T.; Qi, S.; Wang, W.; Shen, J.; Zhu, S.-C. Cascaded parsing of human-object interaction recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
40. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *CoRR* 2014. Available online: http://arxiv.org/abs/1409.4842 (accessed on 20 January 2022).
41. Wang, Y.; Gong, D.; Yang, J.; Shi, Q.; Hengel, A.v.d.; Xie, D.; Zeng, B. An effective two-branch model-based deep network for single image deraining. *arXiv* **2019**, arXiv:1905.05404.
42. Wang, L.; Li, Y.; Huang, J.; Lazebnik, S. Learning two-branch neural networks for image-text matching tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 394–407. [CrossRef]
43. Faranda, R. A new parameters identification procedure for simplified double layer capacitor two-branch model. *Electr. Power Syst. Res.* **2010**, *80*, 363–371. [CrossRef]
44. Hore, A.; Ziou, D. Image quality metrics: Psnr vs. ssim. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
45. Dai, D.; Timofte, R.; Van Gool, L. Jointly optimized regressors for image super-resolution. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2015; Volume 34, No. 2, pp. 95–104.
46. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
47. Chen, Y.; Pock, T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1256–1272. [CrossRef]
48. Lu, Z.; Yu, Z.; Yali, P.; Shigang, L.; Xiaojun, W.; Gang, L.; Yuan, R. Fast single image super-resolution via dilated residual networks. *IEEE Access* **2018**, *7*, 109729–109738. [CrossRef]
49. Ren, H.; El-Khamy, M.; Lee, J. Image super resolution based on fusing multiple convolution neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 54–61.
50. Lai, W.-S.; Huang, J.-B.; Ahuja, N.; Yang, M.-H. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 624–632.
51. Bae, W.; Yoo, J.; Ye, J.C. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 145–153.
52. Xu, J.; Li, M.; Fan, J.; Zhao, X.; Chang, Z. Self-learning super-resolution using convolutional principal component analysis and random matching. *IEEE Trans. Multimed.* **2018**, *21*, 1108–1121. [CrossRef]
53. Cao, F.; Chen, B. New architecture of deep recursive convolution networks for super-resolution. *Knowl.-Based Syst.* **2019**, *178*, 98–110. [CrossRef]
54. Shi, Y.; Wang, K.; Chen, C.; Xu, L.; Lin, L. Structure-preserving image super-resolution via contextualized multitask learning. *IEEE Trans. Multimed.* **2017**, *19*, 2804–2815. [CrossRef]