electronics

MDPI

Article

Multi-Scale Inception Based Super-Resolution Using Deep Learning Approach

Wazir Muhammad and Supavadee Aramvith *

Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand

* Correspondence: supavadee.a@chula.ac.th; Tel.: +66-2-218-6911

Received: 26 June 2019; Accepted: 6 August 2019; Published: 13 August 2019

Abstract: Single image super-resolution (SISR) aims to reconstruct a high-resolution (HR) image from a low-resolution (LR) image. In order to address the SISR problem, recently, deep convolutional neural networks (CNNs) have achieved remarkable progress in terms of accuracy and efficiency. In this paper, an innovative technique, namely a multi-scale inception-based super-resolution (SR) using deep learning approach, or MSISRD, was proposed for fast and accurate reconstruction of SISR. The proposed network employs the deconvolution layer to upsample the LR image to the desired HR image. The proposed method is in contrast to existing approaches that use the interpolation techniques to upscale the LR image. Primarily, interpolation techniques are not designed for this purpose, which results in the creation of undesired noise in the model. Moreover, the existing methods mainly focus on the shallow network or stacking multiple layers in the model with the aim of creating a deeper network architecture. The technique based on the aforementioned design creates the vanishing gradients problem during the training and increases the computational cost of the model. Our proposed method does not use any hand-designed pre-processing steps, such as the bicubic interpolation technique. Furthermore, an asymmetric convolution block is employed to reduce the number of parameters, in addition to the inception block adopted from GoogLeNet, to reconstruct the multiscale information. Experimental results demonstrate that the proposed model exhibits an enhanced performance compared to twelve state-of-the-art methods in terms of the average peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) with a reduced number of parameters for the scale factor of 2×, 4×, and 8×.

Keywords: deep learning; multi-scale information; asymmetric convolution; residual skip connection; inception module

1. Introduction

Super-resolution (SR) is an image, video, and computer vision task that reconstruct the high quality or high-resolution (HR) image with large texture detail information from a single or multiple low quality or low-resolution (LR) image [1,2], under the limited conditional environment and low-cost imaging system. Despite its difficulty and limitations, SR could be applied in real world applications, such as security and surveillance imaging systems [3], face recognition [4], and medical [5] and satellite imaging systems [6].

However, SR is a classical challenging ill-posed problem. To handle the ill-posed problem in SR reconstruction, different algorithms have been proposed by the researchers in the area of image and video recognition. Earlier methods include interpolation and reconstruction-based techniques. Examples of interpolation-based techniques are cubic interpolation [7], nearest neighbor-based interpolation [8], and edge-guided-based interpolation [9]. Usually the performance of these methods is very good, and its implementation is very easy, but still, they generate ringing jagged artifacts and

blurry results in smooth region areas. Furthermore, reconstruction-based methods are very efficient in preserving sharp edges or boundaries and suppressing the jagged ringing artifacts [10]. To reconstruct HR images with complex scenes, prior methods fail to reconstruct the high-frequency information details. Recently, approaches have been used to learn the nonlinear mapping of image space between LR to HR images through millions of image pair co-occurrences, comprising linear regression [11], spare based dictionary learning [12], random forest [13], and neural networks [14,15].

Among them, the neural networks have marked-out significant consideration due to easy, simple elements and excellent performance, but there are still some limitations. First, this type of model is a fully deep neural network and is used to limit the contextual information over all the global image region. Although some methods [16–21] have revised and improved the image restoration quality by stacking side by side convolution layers to display contextual information over a large region but increase the computational cost, as well as memory usage. Second, existing approaches only optimize the network model in the loss function, due to the increase in the blurry edges in the recovered image. Several algorithms [18–20] have concentrated attention on improving the loss function to reconstruct the HR images. However, blurred sharp edges still exist to recover the HR images. In order to handle such issues and to further improve the recent existing methods, we proposed the multi-scale inception-based SR using deep learning approach (MSISRD) to restore the desired high-quality and HR images from observed low quality and LR input images.

In summary, our major contributions through the paper are mainly focused on three aspects:

- We proposed a residual asymmetric convolution block to ease the training complexity, as well as reduce the dimensionality of the intermediate layers.
- We also proposed a multi-scale inception block that can extract the multi-scale feature to restore the HR image.
- Based on the inception block, we designed asymmetric convolution deep model that outperforms the traditional convolutional neural networks (CNNs) model on both effectiveness and efficiency.

The remaining parts of our work are arranged in the following sections. In Section 2, we discuss the literature survey of image SR-related works. Section 3 discusses the proposed network model architecture and training procedure. Experimental evaluation and comparison with existing algorithms are discussed in Section 4. Finally, our work is concluded in Section 5.

2. Related Works

Many single image super-resolution (SISR) methods have been reviewed in the literature to solve the image SR problem. There are three main approaches in SISR. Interpolation-based, sparse coding-based, and deep learning-based methods. We will briefly discuss the first two approaches and focus our discussions on the recent deep CNN-based methods that are related to our work. Earlier algorithms have utilized the interpolation-based techniques [10,22], like bicubic and linear interpolation, adjusted anchored neighborhood regression (A+) [11], super-resolution Forests (RFL) [13] and transformed self-examplars super-resolution (SelfExSR) [23] .These algorithms are easy to implement, and the speed is fast, but they often produce artifacts like pixelization, jagged contours, and blurry results [24]. Hence, it is difficult to reconstruct the detailed, realistic textures in the SR results. Sparse coding-based techniques [12,25] are introduced to alleviate these problems and to improve the performance of previous approaches, but sparsity-based approaches undergo excessive computation to calculate sparse representation of an LR patch from a pre-trained LR dictionary. The neighborhood regression algorithm [11,22] uses the combination of HR image patches to reconstruct an HR image.

Recently, the deep CNN has shown significant improvement for the SR task, thus proving the strong potential for learning a complex non-linear mapping from the LR space domain to the HR space domain. The first concrete architecture, proposed by Dong et al. [14], is the Super-Resolution Convolutional Neural Network (SRCNN) [26], which reported a remarkable progress jump over all previous SR methods. However, there are still some drawbacks. First, the original LR image is upscaled

by bicubic interpolation to the desired size. Second, the reconstruction details information is still unsatisfactory. Third, training convergence is too slow. Z et al. [27] proposed the Deep Networks for Image Super-Resolution with Sparse Prior, named as sparse coding based network (SCN). This approach is simple and achieves notable performance over SRCNN.

Dong et al. [15] improved the SRCNN [26], further named Fast Super-Resolution Convolutional Neural Network (FSRCNN) [15], by introducing a deconvolution layer as the last layer of the model with a stride equal to the size of the scale factor. FSRCNN [15] has a simple network architecture that consists of four convolution layers with one transpose convolution layer and uses the original LR image without bicubic interpolation. FSRCNN [15] has better performance and lower computational cost than SRCNN [26] but has a limited network capacity.

Shi et al. [28] proposed the Efficient Sub Pixel Convolution Neural Network (ESPCN), which uses the same technique introduced by FSRCNN [15], to reduce the model complexity with a sub pixel convolution layer to upscale the information.

Kim et al. [16] proposed Very Deep Super Resolution (VDSR) [16] using the global residual connection to reduce the training complexity, which leads to faster convergence of the model and achieves great performance. The main purpose of VDSR [16] is to predict the residual, rather than the actual, pixel value.

Currently, due to the success of UNet [29] architecture, the work in [30] proposed the idea of the Residual Encoder-Decoder Network (REDNet). REDNet [30] consists of two parts: the encoder network and decoder network. The convolution layer is used at the encoder side, and the deconvolution layer is used at the decoder side.

Kim et al. [17] applied the same convolution layers multiple times and proposed the idea of the Deep Recursive Convolutional Network (DRCN) [17]. The main advantage of this architecture is that the number of model parameters is fixed, even though there are more recursions.

Lai et al. [18] proposed the Laplacian pyramid super-resolution network (LapSRN) [18], which reconstructs multiple images progressively with different scale factors. Deconvolution is proposed in [31–33]. It is observed as pointwise multiplication of each input pixel by a kernel, which could increase the input size if the stride is greater than one. LapSRN [18] uses three types of layers: the convolution layers, leaky rectified linear unit (LReLU) layers, and transpose or deconvolution layers. The training dataset is the same as the SRCNN [26].

The residual neural network (ResNet) [34], proposed by He et al., solves the vanishing/exploding gradient problem in a very deep neural network during the training. ResNet [34] uses many numbers of layers, like 34, 50, 101, 152, and also 1202. The most popular version is the ResNet50 contains 50 CNN layers and one fully-connected layer at the end of the network. In [19], the authors proposed the SRResNet [19] architecture with 16 residual blocks. Each block is made up of two convolution layers, followed by a batch normalization (BN) layer [35] and parametric rectifying linear unit (PReLU) activation function. It does not use any pre-processing nor residual learning. Transposed convolution is used to upscale the LR image. BN [35] is used to stabilize the training procedure.

Ren et al. [36] proposed Context-wise Network Fusion (CNF), in which each model of the SRCNN [26] is constructed with a different number of layers and, finally, each SRCNN [26] model output is passed through a single convolution layer and fused with the sum-pooling layer.

The Deep CNN with Skip Connection and Network in Network, abbreviated as DCSCN network architecture [37], proposed a shallower model than VDSR [16], introducing the skip connections at different stages and directly using the LR image as an input. The DCSCN [37] model consists of different modules, such as feature extraction and reconstruction network, which provide better SR performance.

Han et al. [38] considered the DRCN [17] and the Deep Recursive Residual Network (DRRN) [21] as the Recurrent Neural Networks (RNNs) employing recurrent states and proposed Dual-State Recurrent Network (DSRN) [38], which uses dual recurrent states.

In [39], super-resolution network for multiple degradations (SRMD) proposed a concatenated LR image and its degradation mappings. The network architecture is the same as in [14,40,41]. First, a size of 3×3 convolution filter is cascaded and followed by a sequence of convolution, rectified linear unit (ReLU) [42], and BN [35] layers. The authors also introduce the SR network for multiple degradations noise-free degradation model (SRMDNF).

Mei at el. [41], inspired by image SR via SRResNet [19] and LapSRN [18], proposed a new concept—the Super-Resolution Squeeze and Excitation (SrSE) Network (SrSENet) Network [41] for SISR. Utilizing SrSEBlock with deep residual networks in this approach can provide better feature extraction due to the channels correlations model between feature mappings from LR image.

In [43], Chu et al. introduced the idea of a multi-objective oriented algorithm, known as Multi-Objective Reinforced Evolution in Mobile Neural Architecture Search (MOREMNAS) by good virtue from both evaluation algorithm (EA) and reinforced learning (RL) methods. Authors also introduced a different version of models, like MOREMNAS-A, -B, -C, and the dominates version, MOREMNAS-D [43].

Many modern SR networks, such as FSRCNN [15], LapSRN [18], SrSENet [41], and DCSCN [37], achieved better results by using deconvolution as the upsampling module. However, the computation complexity of forward and back propagation of deconvolution [44] is still a major concern. They promise low computational complexity and better perceptual quality, but there possibly exists plenty of room for improvement in SR performance.

3. Proposed Method

In this section, we describe the design procedure of our proposed MSISRD method in detail. Initially, input LR image passes through three stacked CNN layers, followed by ReLU [42] using skip connection. This process produces a summed output that contains detailed feature information. As such, the number of parameters is thus reduced. Afterword, the information is fed to the deconvolution layer for upsampling purposes. The upsampled LR information is sent through two asymmetric residual blocks to reduce the training complexity and reconstruct the middle-level feature information. The inception block is used in the multi-scale reconstruction stage-II to reconstruct the final HR image, as shown in Figure 1.

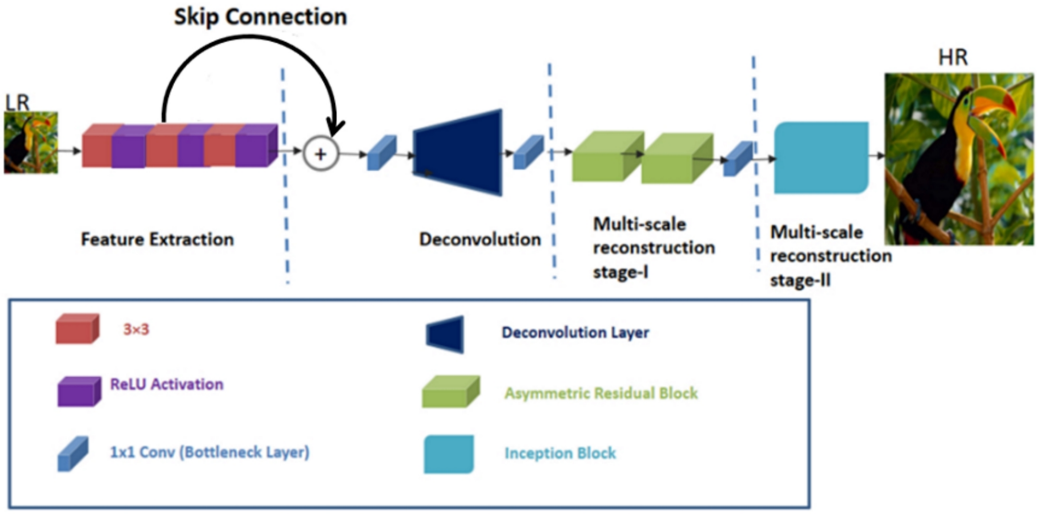

Figure 1. The complete network architecture of our proposed super-resolution (SR) method. Our network consists of feature extraction, deconvolution, multi-scale reconstruction stage-I, and multi-scale reconstruction stage-II. LR = low-resolution; HR = high-resolution; ReLU = rectified linear unit; Conv = convolution.

3.1. Feature Extraction

Inspired by VDSR [16], we proposed three trainable convolution layers of 3×3 kernel size with 64 filters, followed by ReLU [42] activation function. ReLU [42] directly extracts the feature information from the original LR image as Y. Mathematically convolution layer can be represented as,

$$F_l(Y) = W_l * G_{l-1}(G), \tag{1}$$

where l is the lth convolution layer, W_l represents the number of filters of the lth layer, and G_{l-1} denotes the previous layer output feature map. F_l is an output feature map and '*' represents the convolution operation. ReLU [42] activation response can be calculated as general activation function as,

$$ReLU(Y) = max(0, x), \tag{2}$$

where x is the input of activation on the lth layer and Y is an ReLU [42] activation output of the feature maps. The final out put of the convolution layer can be defined as,

$$G_l(Y) = ReLU(W_l * G_{l-1}(Y) + b_l), \tag{3}$$

where G_l represents the final output of the feature map of the lth layer, and b_l, W_l denotes a bias and weight of the convolution filter of the lth layer, respectively. Inspired by ResNet [34], we applied the first layer feature map output that is added in the third layer, using skip connection with identity mapping.

3.2. Deconvolution

In order to recover the SR images, the basic concept is to upscale the original LR image using interpolation techniques to get the HR image. The implementation of such an approach is very easy and fast. Actually, interpolation techniques were not designed for upscaling the original LR to recover the HR image. Additionally, the said approaches even damage the important LR information. Furthermore, it takes more computational time in pre-processing without any obvious advantages. Shi et al. [28] proposed the idea of a sub-pixel convolution layer to recover the HR image directly, but this approach does not completely utilize the related information from the LR domain to HR. LapSRN [18] introduced the concept of multiple transposed convolution layer in a progressive way with different upscale and obtained relatively faster and more accurate information from LR to HR image.

Based on the common architecture of CNN SR, the deconvolution layer is used to upsample the previous feature results with a number of convolution kernels. The quality of the LR image is improved by increasing the kernel size of the deconvolution layer, but a larger kernel size also increases the computational complexity. In our proposed approach, we apply two 1×1 operation of convolution before and after the deconvolution layer. The first 1×1 kernel operation performs the function of dimension reduction to change the 64 feature maps into 4 feature maps for the upsampling purpose, and the last convolution kernel is used to recover the feature information back to the 64 number of channels. The upsampling layer serves as the bridge between two 1×1 convolution layers, which uses the different kernel size for different scale factor like 14×14, 16×16, and 18×18 for enlargement factor of $2\times$, $4\times$, and $8\times$, respectively.

3.3. Multi-Scale Reconstruction Stage-I

As the depth of network increases, the flow of information becomes weak at the final layers [33]. This leads to the vanishing/exploding gradient issue during the training [45] . The ResNet proposed by He et al. [34] intends to solve this problem and widely uses the idea of skip connection in [19,20] to construct a very deeper model for image SR. The residual network blocks [16,19,34,46] are shown to improve training accuracy on the SR work. In Figure 2, we show the residual network block of original ResNet [34], SRResNet [19], and our proposed ResNet block. In the original ResNet block [34], their architecture consists of a direct path and skip connection for propagating the information through

the residual block. Resultantly, the summed up information finally passes through the ReLU [42] activation layer. In the SRResNet block [19], the ReLU [42] activation function has been removed to provide the clean path from one block to the next one. Our proposed block removes two BN [35] layers to reduce the memory usage of the Graphics Processing Unit (GPU) and minimize the computational complexity. Compared to the original ResNet block [34] and SRResNet block [19], which use the standard convolution operation, our proposed block uses the idea of asymmetric convolution operation, which reduces the size of the model, as well as increases the training efficiency of the model.

For multi-scale reconstruction stage-I, we applied eight asymmetric convolution trainable layers, which are interleaved followed by *ReLU* [42] nonlinearity. The asymmetric convolution (AConv) is to factorize a standard two-dimensional convolution kernel into two one-dimension convolution kernels. In other words, a 3×1 convolution, followed by a 1×3 convolution, is substituted for a 3×3 convolution [47,48]. This mechanism can be expressed as,

$$\sum_{i=-M}^{M} \sum_{j=-N}^{N} W(i,j) I(x-i, y-j) = \sum_{i=-M}^{M} w_x(i) [\sum_{i=-N}^{N} W_y(j) I(x-i, y-j)], \tag{4}$$

where I is a 2D image, W is a 2D kernel like 3×3, W_x is a 1D kernel along x-dimension as 1×3, and W_y is a 1D kernel along y-dimension as 3×1.

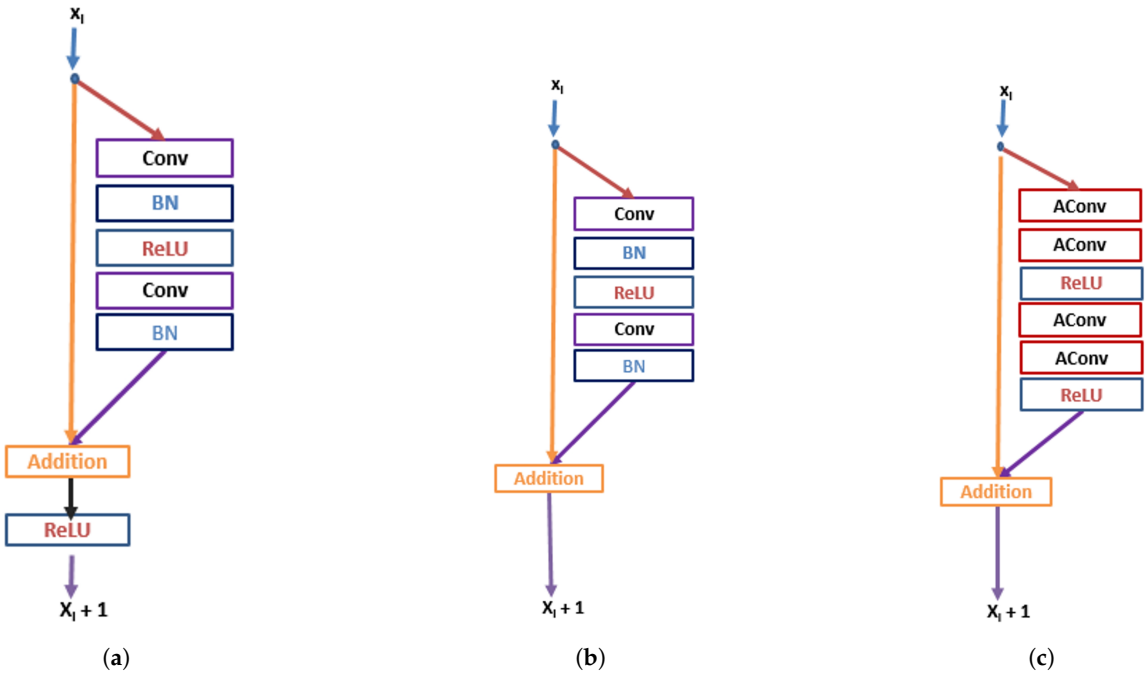

Figure 2. The comparison design of residual connection blocks [49] with our proposed residual block. (**a**) Original residual neural network (ResNet) [34]. (**b**) SRResNet [19]. (**c**) Proposed. BN = batch normalization; AConv = asymmetric convolution.

The relationship between standard convolution kernel size and asymmetric convolution kernel size in terms of a number of parameters is shown in Table 1. For example, we took a single layer of 3×3, where the number of filters is 10 and image patch size is 28×28, and the calculated number of parameters is 900. Similarly, after applying asymmetric convolution operation on the 3×3 layer and splitting the same into 3×1 and 1×3, with the same number of filters and image patch size, the calculated number of parameters is 600. Results clearly show that asymmetric convolution type kernel has a lesser number of parameters compared to standard convolution kernel size. This approach

is considered to be one of the most suitable options due to the fact that it reduces the size of a deeper model, increases the computational efficiency during the training, and avoids the overfitting problems.

Table 1. Comparison of standard and asymmetric convolution in terms of kernel size and number of parameters.

Kernel Size	No: of Layers	No: of Filters	Image Patch Size	No: of Parameters
3×3	1	10	28×28	900
3×1 and 1×3	2	10	28×28	600
5×5	1	10	28×28	2500
5×1 and 1×5	2	10	28×28	1000
7×7	1	10	28×28	4900
7×1 and 1×7	2	10	28×28	1400
9×9	1	10	28×28	8100
9×1 and 1×9	2	10	28×28	1800
11×11	1	10	28×28	12,100
11×1 and 1×11	2	10	28×28	2200

In our proposed architecture, we used four CNN layers of size 3×1 and 1×3 asymmetric convolution operation, with each layer taking the previous input feature and generating 16 channels of the new features. In order to facilitate the flow of training, we used the skip connection after every two convolution layers and added the input to the next block as output. In order to decrease the number of parameters we used, we used a 1×1 bottleneck CNN layer [50] after the final asymmetric residual block.

3.4. Multi-Scale Reconstruction Stage-II

At the final stage, we used a multi-scale block adopted from GoogLeNet [51] to select the appropriate kernel size. The size of the kernel plays a very important role in the model design, as well as the training procedure, because it is a very close relation to extracting the more useful information. The smaller size of the kernel is better for capturing the information locally, and the larger size of the kernel is more preferable for information distributed globally. The inception network [52] uses this idea and includes many convolutions with a different size kernels. Furthermore, the second and third version of inception architecture uses the idea of asymmetric convolution. For example, $n \times n$ shape of the kernel can translate into a combination of two $1 \times n$ and $n \times 1$ convolutions, which is the most efficient convolution kernel, rather than the standard convolution kernel. For example, a convolution with kernel size is 3×3 is equivalent to a 1×3 followed by 3×1 , which was found to be 33% of the low computational cost in the standard convolution [52].

Figure 3 shows the comparison between traditional convolution operation with asymmetric convolution operation. In Figure 3a, plain architecture with many layers is stacked in a single path, used by SRCNN [26] and FSRCNN [15]. These types of architecture design are very simple, but a deeper model increases the size of the model and consumes more memory.

In Figure 3b, a conventional inception block is used to extract the multi-scale feature information. This block allows the extraction of the multi-scale feature information more efficiently. However, the problem with this type of block is that it has a higher number of parameters, and so does the higher computational complexity of the model. We proposed the multi-scale asymmetric convolution block, as shown in Figure 3c, to solve the problem of training complexity. Our proposed inception block can reduce the computational time and can extract the multi-scale feature information to reconstruct the SR image.

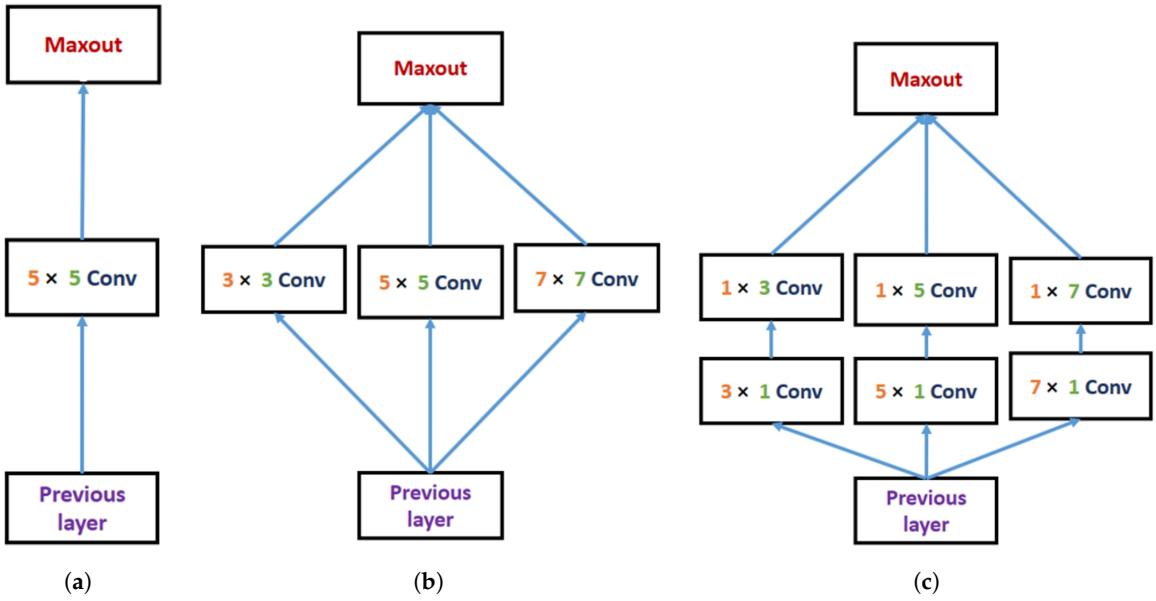

Figure 3. Proposed multi-scale inception block of two successive convolutions [53]. (**a**) Single Scale Convolution (**b**), Multi-Scale Convolution, and (**c**) proposed.

In Figure 4, we introduced a new module inspired by the idea of a naive version inception module and inception module with dimension reduction [51]. In the naive version inception module, the convolution operation is performed on a previous layer output, with three different sizes of the filters having order 1×1, 3×3, and 5×5. In order to achieve dimension reduction, the max-pooling operation is also employed. The output of these layers is concatenated and sent to the next inception module, as shown in Figure 4a. The major problem with the naive version inception module is a larger number of kernel size. Even the modest number of kernel size can be more expensive on the top of the convolutional layer. This problem becomes serious after the fusion of max-pooling layer output with the output of convolutional layers from one stage to another stage. With a view of making it computationally efficient and reducing the number of input channels, the authors have revised the naive version inception module with dimension reductions by adding an extra 1×1 convolution layer before the 3×3 and 5×5 convolution layers, as well as after max-pooling layer, as shown in Figure 4b. Followed by the aforementioned successful model, we proposed an asymmetric inception block to learn the multi-scale information for reconstructing the HR image, as shown in Figure 4c.

In the suggested asymmetric inception block, the standard convolution layers are replaced with asymmetric convolution layers. For multi-scale reconstruction purpose, we used five towers with four different sizes of the asymmetric convolution filter. These filters are followed by ReLU having 16 features of various asymmetric convolutional filter size. In the first branch/tower 1, we split the two filters having layers of 3×3 and 5×5 into four asymmetric convolution filters of the order 3×1, 1×3, 5×1, and 1×5 to reduce the number of parameters. Similarly, in tower 2 and tower 3, we applied the same size of the asymmetric convolution filter operation. In tower 4 and tower 5, we divided the larger filter size of 7×7 and 9×9 into an asymmetric convolution filter of size 7×1, 1×7, 9×1, and 1×9. Finally, we concatenated values of all the towers followed by ReLU activation nonlinearity. With the aim of improving the compactness, achieving the computational efficiency, and experiencing better performance, we used a 1×1 bottleneck CNN layer [50]. Remarkably, the 1×1 bottleneck CNN layer [50] not only reduced the dimensions of the previous layers for higher computational efficiency but also added more nonlinearity information to enhance the representation of the reconstructed LR image. The 1×1 bottleneck CNN layer [50] has less computational cost as compared to 3×3 CNN layer. As a result, our proposed block is relatively lighter, more efficient, and computationally effective in comparison to the other deep learning-based reconstruction blocks.

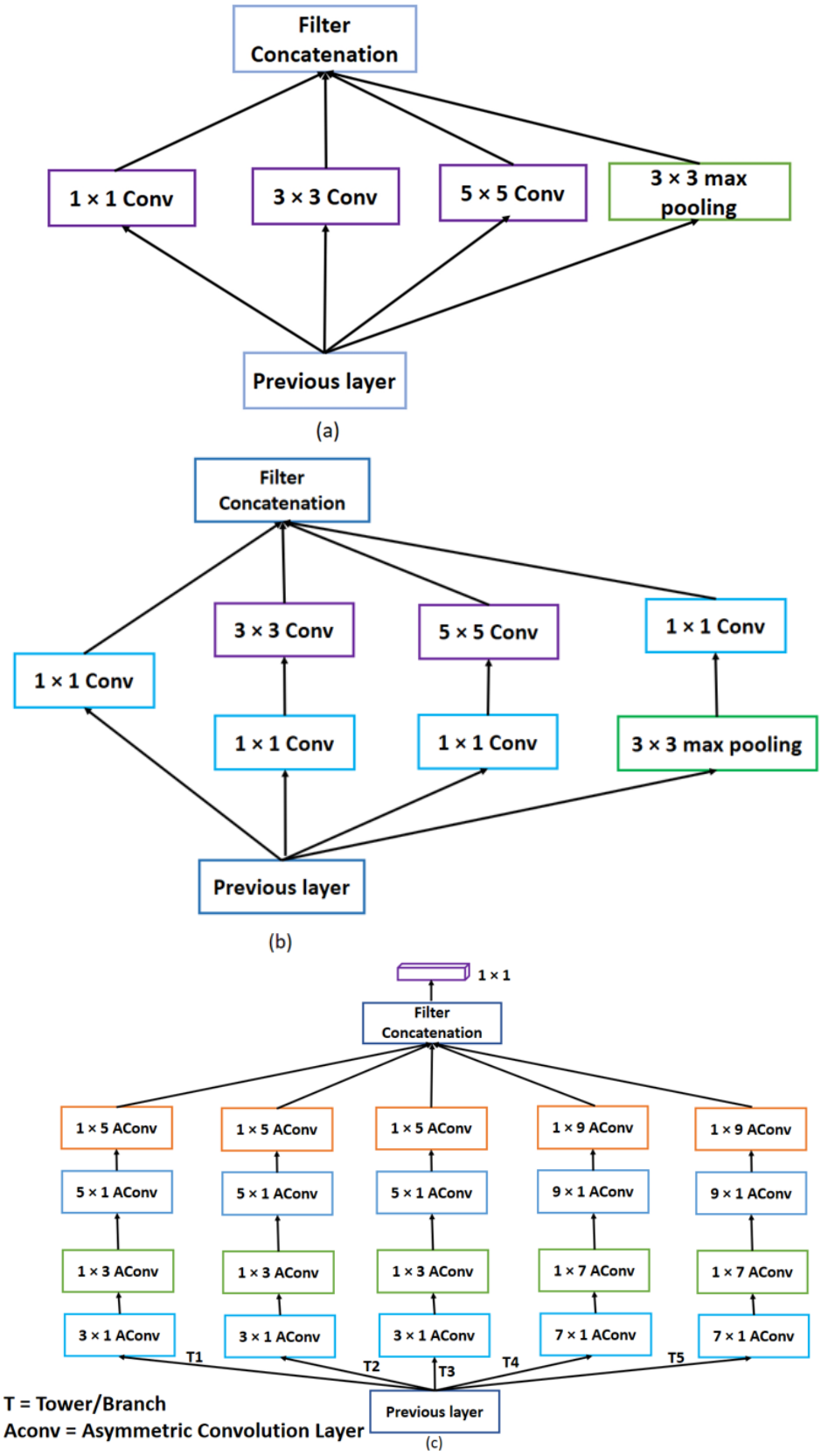

Figure 4. Inception block with different towers [54] used for multi-scale reconstruction stage-II. (**a**) Inception module, naive version; (**b**) inception module with dimension reductions [55]; and (**c**) our proposed module.

4. Experimental Results

Under the experimental results section, we first explain the construction of the training datasets and model hyperparameters. Next, we compare the quantitative and qualitative performance on five benchmark test datasets. Finally, we compare the complexity of the model in terms of peak signal-to-noise ratio (PSNR) [56] versus a number of parameters.

4.1. Training Datasets

There have been many training datasets available for single image-super resolution, but commonly used datasets are Yang et al.'s [57] image dataset and the Berkeley Segmentation Dataset (BSDS) [58]. To evaluate the proposed method, we selected 91 images from [57] and another 200 images from [58]. As followed by [21], to take the benefit of the full training dataset and avoid the over-fitting problem, we applied the data augmentation technique randomly by flipping all images and then performing the rotation operation to increase the training dataset [59]. All experiments were performed on the HR ground truth image and randomly cropped and flipped the training sample images as an original ground truth image. For data processing, we used MATLAB 2018a and the Keras 2.2.1 framework [60], with TensorFlow as back-end, and LR images were generated by built-in function bicubic. Several loss functions have been used in deep learning techniques. Most deep neural network based SR methods have used the mean squared error (MSE) loss function, so we adopted the same loss function with our proposed model. The end-to-end mapping function required the estimation parameters of the network θ, which consists of a set of weights and biases. This is obtained by minimizing the objective loss between the restored image $F(Y,\theta)$ and the corresponding original HR ground truth image X. The set of HR and high-quality images are X_i and their corresponding LR images Y_i, and m is the number of samples in each batch during the training; we used the MSE as a loss function that can be calculated as:

$$L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left\|F(Y_i;\theta) - X_i\right\|^2. \quad (5)$$

To minimize the objective of the loss function, we used the adaptive momentum estimation (Adam) [61] optimizer, and its initial learning rate set as 0.0003, with 32 mini-batch sizes during the training. The training takes 100 epochs to converge properly, and all experiments were conducted on an NVIDIA Titan Xp GPU, under an Ubuntu 18.04 operating system of 3.5 GHz Intel i7-5960x CPU and 64 GB RAM. For a fast training procedure, we trained our model only on single channel, i.e., Y-channel, so we converted the RGB channel into YCbCr and finally added the enlarged color channel using bicubic interpolation technique.

4.2. Testing Datasets

We evaluated our model's performance on five publicly available benchmark datasets, such as the Set5 [62], Set14 [63], BSDS100 [58], Urban100 [23], and Manga109 [64]. The Set5 [62] dataset consists of five images with various sizes between 228 × 228 and 512 × 512 pixels. Set14 [63] consists of 14 images, and the BSDS100 [58] dataset consists of 100 natural scenes of images. The Urban100 [23] dataset consists of different challenging images with many frequency bands and details of the information available, and the Manga109 [64] dataset consists of many comic images with fine structure. However, for fair comparison purposes, our proposed method used the recently published data, such as that presented by Lai et al. (2017) [18] and Yulun et al. (2018) [65].

4.3. Comparison with Other Existing State-of-the-Art Methods

There are many techniques to validate the effectiveness of the proposed model. In image SR literature, it is common to use two metrics for quality measurement, i.e., PSNR and structural similarity

index (SSIM) [56]. Both quality metrics have measured the difference between upscaled or interpolated LR image and its original high-quality HR image. The higher value of $PSNR$ and $SSIM$ [56] of two images should correlate to a higher degree of similarity between them and shows the better reconstruction quality of the image. The value of $PSNR$ [56] is measured in decibels (dB), and the ranges from 0 to infinity. $SSIM$ means the perfect recovery of the LR image and the ranges from 0 to 1. The main expression of $PSNR$ and $SSIM$ [56] are shown in Equations (6) and (7), respectively,

$$PSNR(r,s) = 10 * log_{10}\left[\frac{(2^k-1)^2}{MSE}\right], \tag{6}$$

where k is the bit depth, and MSE is the mean square error.

$$SSIM(r,s) = \frac{(2\mu_r\mu_s + C_1)(2\sigma_{rs} + C_2)}{(\mu_r{}^2 + \mu_s{}^2 + C_1)(\sigma_r{}^2 + \sigma_s{}^2 + C_2)}, \tag{7}$$

where μ_r and μ_s denote the mean value of r and s. The variance of r and s are denoted by $\sigma_r{}^2$ and $\sigma_s{}^2$. The covariance of r and s represents as σ_{rs}. C_1 and C_2 are the constants to maintain the formula validity and to avoid the denominator being zero.

Quantitative results of PSNR and SSIM were evaluated on the public benchmark of Set5 [62], Set14 [63], BSDS100 [58], Urban100 [23], and Manga109 [64] datasets, with scale factor 2×, 4×, and 8×, as shown in the Table 2.

For qualitative and quantitative comparison, we selected twelve different state-of-the-art algorithms, along with the baseline. PSNR and SSIM [56] are the most popular reference metrics, widely used in the image SR tasks, and they directly apply on the intensity of the image. As can been seen from Table 2, our method achieves, on average, better PSNR and SSIM [56] than all existing methods. Furthermore, overall on five datasets with upscale factor 2x, our MSISRD can improve 1.33 dB, 1.04 dB. 0.95 dB, 0.42 dB, 0.39 dB, 0.49 dB, 0.32 dB, 0.37 dB, 0.37 dB, 0.32 dB, 0.24 dB, and 0.16 dB on average PSNR, in comparison with SRCNN [26], ESPCN [28], FSRCNN [15], VDSR [16], DCSCN [37], LapSRN [18], DRCN [17], SrSENet [41], SRMD [39], REDNet [30], DSRN [38], and CNF [36], respectively.

Table 3 shows the quantitative comparison results for scale 4×, on the Set5 [62] dataset of PSNR/SSIM [56] versus a number of parameters. Our model yields higher performance with fewer numbers of parameters than other SR methods, which proves the best efficiency of our proposed model. Furthermore, the proposed method employs a much fewer number of parameters than REDNet [30], DRCN [17], and SRMD [39]. For instance, our model uses up to 94% less the number of parameters than REDNet [30] , 86% less than DSRN [38], and 70% less than LapSRN [18].

Figure 5 shows the relationship between the number of parameters and PSNR [56]; our proposed model presents a favorable trade-off between the model complexity and the performance of the SR image.

Figure 6–9 show the perceptual quality performance on the Set5 [56], Set14 [63], BSDS100 [58], and Urban100 [23] datasets for scale 4× enlargements image SR. Figures 10–13 present the visual performance of above datasets on scale factor 8×, including one image from the Manga109 [14] dataset. The results of the Bicubic, SRCNN [26], and FSRCNN [15] look blurry and lack high-frequency details. Image SR on scale 8× is a very challenging problem, but our method accurately reconstructs the texture details, suppresses the artifacts, and recovers the details of the LR image with sharp edges. Figure 10 clearly shows that our method accurately reconstructs the fine texture details, such as the eyebrow of a baboon, leading to the pleasing visual perceptual quality of the image.

Table 2. Quantitative evaluation of existing SR algorithms with our proposed approach; reported results is the average value of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [56] using 2×, 4×, and 8× enlargement scale factors; red color with bold value indicates the best value, and the blue color with underline indicates the second best value.

Method	Factor PSNR/SSIM	Params PSNR/SSIM	Set5 [62] PSNR/SSIM	Set14 [63] PSNR/SSIM	BSDS100 [58] PSNR/SSIM	Urban100 [23]	Manga109 [64]
Bicubic	2×	-	33.69/0.931	30.25/0.870	29.57/0.844	26.89/0.841	30.86/0.936
A+ [11]	2×	-	36.60/0.955	32.32/0.906	31.24/0.887	29.25/0.895	35.37/0.968
RFL [13]	2×	-	36.59/0.954	32.29/0.905	31.18/0.885	29.14/0.891	35.12/0.966
SelfExSR [23]	2×	-	36.60/0.955	32.24/0.904	31.20/0.887	29.55/0.898	35.82/0.969
SRCNN [26]	2×	57k	36.72/0.955	32.51/0.908	31.38/0.889	29.53/0.896	35.76/0.968
ESPCN [28]	2×	20k	37.00/0.955	32.75/0.909	31.51/0.893	29.87/0.906	36.21/0.969
FSRCNN [15]	2×	12k	37.05/0.956	32.66/0.909	31.53/0.892	29.88/0.902	36.67/0.971
SCN [27]	2×	42k	36.58/0.954	32.35/0.905	31.26/0.885	29.52/0.897	35.51/0.967
VDSR [16]	2×	665k	37.53/0.959	33.05/0.913	31.90/0.896	30.77/0.914	37.22/**0.975**
DCSCN [37]	2×	244k	37.62/0.959	33.05/0.912	31.91/0.895	30.77/0.910	37.25/0.974
LapSRN [18]	2×	813k	37.52/0.959	33.08/0.913	31.80/0.895	30.41/0.910	37.27/0.974
DRCN [17]	2×	1774k	37.63/0.959	33.06/0.912	31.85/0.895	30.76/0.914	37.63/0.974
SrSENet [41]	2×	-	37.56/0.958	33.14/0.911	31.84/0.896	30.73/0.917	37.43/0.974
MOREMNAS-D [43]	2×	664k	37.57/0.958	33.25/0.914	31.94/0.896	**31.25**/**0.919**	37.65/**0.975**
SRMD [39]	2×	1482	37.53/0.959	33.12/0.914	31.90/0.896	30.89/0.916	37.24/0.974
REDNet [30]	2×	4131k	37.66/0.959	32.94/0.914	31.99/**0.897**	30.91/0.915	37.45/0.974
DSRN [38]	2×	1200k	37.66/0.959	33.15/0.913	**32.10**/**0.897**	30.97/0.916	37.49/0.973
CNF [36]	2×	337k	37.66/0.959	33.38/0.914	31.91/0.896	31.15/0.914	37.64/0.974
MSISRD (ours)	2×	240k	**37.80**/**0.960**	**33.84**/**0.920**	32.09/0.895	31.10/0.913	**37.70**/**0.975**
Bicubic	4×	-	28.43/0.811	26.01/0.704	25.97/0.670	23.15/0.660	24.93/0.790
A+ [11]	4×	-	30.32/0.860	27.34/0.751	26.83/0.711	24.34/0.721	27.03/0.851
RFL [13]	4×	-	30.17/0.855	27.24/0.747	26.76/0.708	24.20/0.712	26.80/0.841
SelfExSR [23]	4×	-	30.34/0.862	27.41/0.753	26.84/0.713	24.83/0.740	27.83/0.8663
SRCNN [26]	4×	57k	30.49/0.863	27.52/0.753	26.91/0.712	24.53/0.725	27.66/0.859
ESPCN [28]	4×	20k	30.66/0.864	27.71/0.756	26.98/0.712	24.60/0.736	27.70/0.856
FSRCNN [15]	4×	12k	30.72/0.866	27.61/0.755	26.98/0.715	24.62/0.728	27.90/0.861
SCN [27]	4×	42k	30.41/0.863	27.39/0.751	26.88/0.711	24.52/0.726	27.39/0.857
VDSR [16]	4×	665k	31.35/0.883	28.02/0.768	27.29/0.726	25.18/0.754	28.83/0.887
DCSCN [37]	4×	244k	30.86/0.871	27.74/0.770	27.04/0.725	25.20/0.754	28.99/0.888
LapSRN [18]	4×	813k	31.54/0.885	28.19/**0.772**	27.32/0.727	25.21/0.756	29.09/0.890
DRCN [17]	4×	1774k	31.54/0.884	28.03/0.768	27.24/0.725	25.14/0.752	28.98/0.887
SrSENet [41]	4×	-	31.40/0.881	28.10/0.766	27.29/0.720	25.21/**0.762**	29.08/0.888
SRMD [39]	4×	1482	31.59/**0.887**	28.15/**0.772**	27.34 /**0.728**	25.34/0.761	30.49/0.890
REDNet [30]	4×	4131k	31.51/0.886	27.86/0.771	**27.40**/**0.728**	25.35/0.758	28.96/0.887
DSRN [38]	4×	1200	31.40/0.883	28.07/0.770	27.25/0.724	25.08/0.747	30.15/0.890
CNF [36]	4×	337k	31.55/0.885	28.15/0.768	27.32/0.725	25.32/0.753	30.47/0.890
MSISRD (ours)	4×	240k	**31.62**/0.886	**28.51**/0.771	27.33/0.727	**25.42**/0.757	**31.61**/**0.891**
Bicubic	8×	-	24.40/0.658	23.10/0.566	23.67/0.548	20.74/0.516	21.47/0.650
A+ [11]	8×	-	25.53/0.693	23.89/0.595	24.21/0.569	21.37/0.546	22.39/0.681
RFL [13]	8×	-	25.38/0.679	23.79/0.587	24.13/0.563	21.27/0.536	22.28/0.669
SelfExSR [23]	8×	-	25.49/0.703	23.92/0.601	24.19/0.568	21.81/0.577	22.99/0.719
SRCNN [26]	8×	57k	25.33/0.690	23.76/0.591	24.13/0.566	21.29/0.544	22.46/0.695
ESPCN [28]	8×	20k	25.75/0.673	24.21/0.510	24.73/0.527	21.59/0.542	22.83/0.671
FSRCNN [15]	8×	12k	25.60/0.697	24.00/0.599	24.31/0.572	21.45/0.550	22.72/0.692
SCN [27]	8×	42k	25.59/0.706	24.02/0.603	24.30/0.573	21.52/0.560	22.68/0.701
VDSR [16]	8×	665k	25.93/0.724	24.26/0.614	24.49/0.583	21.70/0.571	23.16/0.725
DCSCN [37]	8×	244k	24.96/0.673	23.50/0.576	24.00/0.554	21.75/0.571	23.33/0.731
LapSRN [18]	8×	813k	26.15/**0.738**	24.35/0.620	24.54/**0.586**	21.81/0.581	23.39/0.735
DRCN [17]	8×	1775k	25.93/0.723	24.25/0.614	24.49/0.582	21.71/0.571	23.20/0.724
SrSENet [41]	8×	-	26.10/0.703	**24.38**/0.586	24.59/0.539	21.88/0.571	**23.54**/0.722
MSISRD (ours)	8×	240k	**26.26**/0.737	**24.38**/**0.621**	**24.73**/**0.586**	**22.53**/**0.582**	23.50/**0.738**

Table 3. Quantitative results of computational complexity in terms of number of parameters versus PSNR [56] on Set5 [62] with 4× scale enlargement factor [66].

Models	PSNR/SSIM [56]	Parameters
SRCNN [26]	30.50/0.863	57k
ESPCN [28]	30.66/0.864	20k
FSRCNN [15]	30.72/0.866	12k
SCN [27]	30.41/0.863	42k
VDSR [16]	31.35/0.883	665k
DCSCN [37]	30.86/0.871	244k
LapSRN [50]	31.54/0.885	813k
DRCN [17]	31.54/0.884	1775k
SRMD [39]	31.59/0.887	1482k
REDNet [30]	31.51/0.886	4131k
DSRN [38]	31.40/0.883	1200k
CNF [36]	31.55/0.885	337k
MSISRD (ours)	31.62/0.886	240k

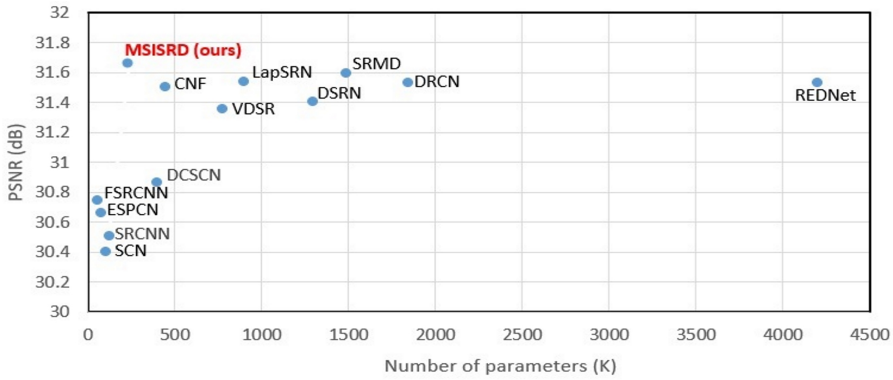

Figure 5. The model complexity comparison in number of parameters versus PSNR [56]. The performance of model complexity are evaluated on Set5 [56] dataset for 4× SR [67].

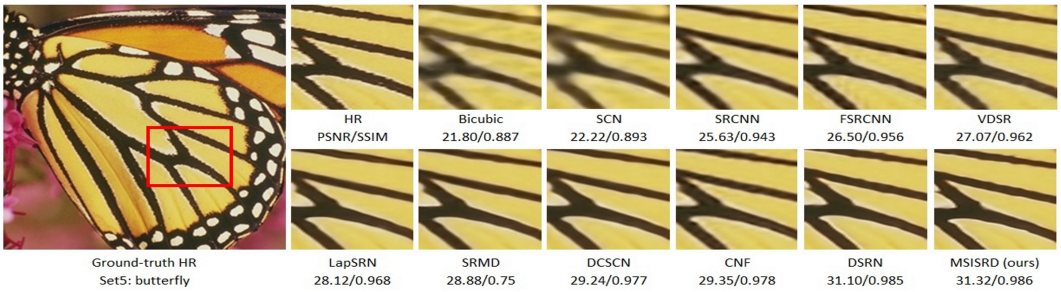

Figure 6. Visual performance of image "butterfly" of Set5 [56] dataset with 4× scale factor enlargements [68].

Figure 7. Visual performance of image "ppt3" from Set14 [63] dataset with 4× scale factor enlargements [68].

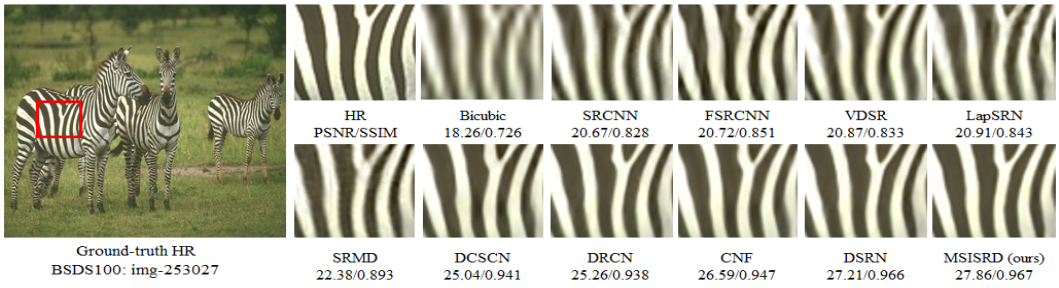

Figure 8. Visual performance of image "253027" from Berkeley Segmentation Dataset (BSDS)100 [58] dataset with 4× scale factor enlargements [69].

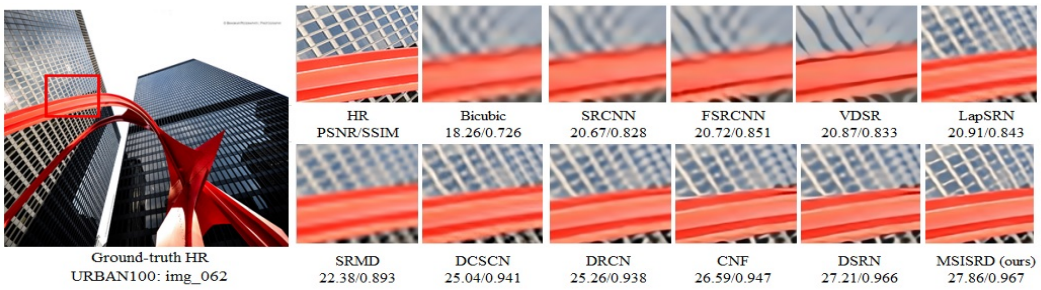

Figure 9. Visual performance of the image "img-062" from URBAN100 [23] dataset with 4× scale factor enlargements [70].

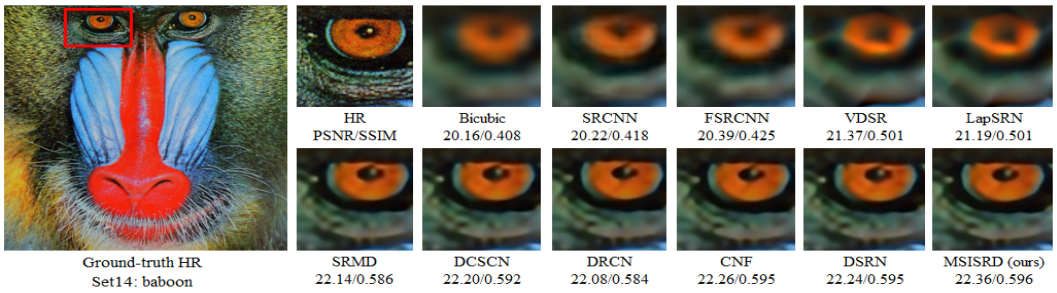

Figure 10. Visual performance of the image "baboon" from Set14 [63] dataset with 8× scale factor enlargements [71].

Figure 11. Visual performance of the image "302008" from BSDS100 [58] dataset with 8× scale factor enlargements [71].

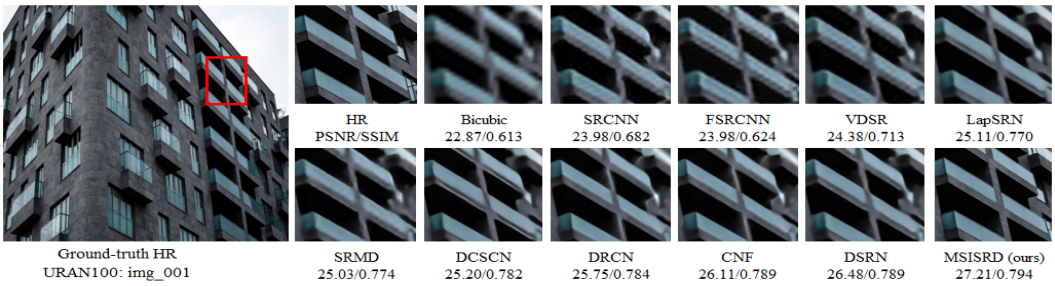

Figure 12. Visual performance of the "img-001" from URBAN100 [23] dataset with 8× scale factor enlargements [71].

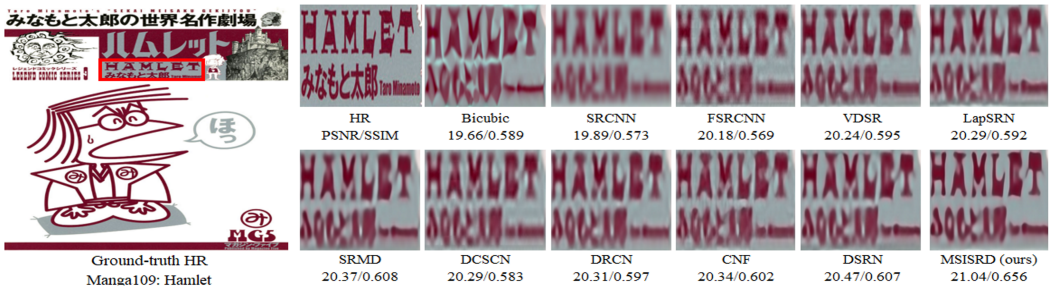

Figure 13. Visual performance of the image "Hamlet" from Manga109 [64] dataset with 8× scale factor enlargements [72].

5. Conclusions

In this paper, we proposed a multi-scale inception-based SR using deep learning approach. Our model uses the locally residual asymmetric convolution block and inception-based asymmetric convolution block architecture to directly extract the short and long feature information. For upscaling purposes, we used the learned transposed convolution layer in the latent feature space. In the reconstruction part, an asymmetric convolution type kernel is applied for better reconstruction of vertical and horizontal edges. Furthermore, we used an inception module to obtain better feature reconstructions with less computational complexity. To our knowledge, this is the first network in which asymmetric convolution kernel has been used in whole architecture. The results show, both qualitatively and quantitatively, a large upscaling factor of 2×, 4×, and 8× enlargements, along with a number of parameters. The proposed method achieves high competitive performance on five benchmark datasets. In the future, we will stack more residual and inception blocks to further improve the quality of SISR.

Author Contributions: Conceptualization, investigation, software, writing—original draft preparation, W.M.; supervision, writing—review and editing, S.A.

Funding: The research grant have been funded by "The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship" and "The 90th Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)".

Acknowledgments: This research is funded by the Graduate School of Chulalongkorn University scholarship from "The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship" and "The 90th Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)". The first author would like to thanks "Department of Electrical Engineering", Chulalongkorn University, Bangkok, Thailand.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Freeman, W.T.; Jones, T.R.; Pasztor, E.C. Example-based super-resolution. *IEEE Comput. Graph. Appl.* **2002**, *22*, 56–65. [CrossRef]
2. Glasner, D.; Bagon, S.; Irani, M. Super-resolution from a single image. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision 2009, Kyoto, Japan, 29 September–2 October 2009.
3. Zhang, L.; Zhang, H.; Shen, H.; Li, P. A super-resolution reconstruction algorithm for surveillance images. *Signal Process.* **2010**, *90*, 848–859. [CrossRef]
4. Gunturk, B.K.; Batur, A.U.; Altunbasak, Y.; Hayes, M.H.; Mersereau, R.M. Eigenface-domain super-resolution for face recognition. *IEEE Trans. Image Process.* **2003**, *12*, 597–606. [CrossRef] [PubMed]
5. Peled, S.; Yeshurun, Y. Superresolution in MRI: Application to human white matter fiber tract visualization by diffusion tensor imaging. *Magn. Reson. Med. Off. J. Int. Soc. Magn. Reson. Med.* **2001**, *45*, 29–35. [CrossRef]
6. Thornton, M.W.; Atkinson, P.M.; Holland, D. Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping. *Int. J. Remote Sens.* **2006**, *27*, 473–491. [CrossRef]

7. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [CrossRef]
8. Wang, Y.; Wan, W.; Wang, R.; Zhou, X. An improved interpolation algorithm using nearest neighbor from VTK. In Proceedings of the 2010 International Conference on Audio, Language and Image Processing, Shanghai, China, 23–25 November 2010.
9. Zhang, L.; Wu, X. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Process.* **2006**, *15*, 2226–2238. [CrossRef]
10. Tai, Y.-W.; Liu, S.; Brown, M. S.; Lin, S. Super resolution using edge prior and single image detail synthesis. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
11. Timofte, R.; De Smet, V.; 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; Springer: Berlin/Heidelberg, Germany, 2014.
12. Yang, J.; Wright, J.; Huang, T. S.; Ma, Y. Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **2010**, *19*, 2861–2873. [CrossRef]
13. 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.
14. Du, X.; Qu, X.; He, Y.; Guo, D. Single image super-resolution based on multi-scale competitive convolutional neural network. *Sensors* **2018**, *18*, 789. [CrossRef]
15. 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, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016.
16. Kim, J.; Kwon Lee, J.; Mu Lee, K. 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, 26 June–1 July 2016.
17. Kim, J.; Kwon Lee, J.; Mu Lee, K. Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
18. Lai, W.-S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep laplacian pyramid networks for fast and accurate superresolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
19. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv* **2017**.
20. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017.
21. 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.
22. Timofte, R.; De Smet, V.; Van Gool, L. Anchored neighborhood regression for fast example-based super-resolution. in Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 8–12 April 2013.
23. 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, 7–12 June 2015.
24. Giachetti, A.; Asuni, N. Real-time artifact-free image upscaling. *IEEE Trans. Image Process.* **2011**, *20*, 2760–2768. [CrossRef] [PubMed]
25. Jianchao, Y.; Wright, J.; Huang, T.; Ma, Y. Image super-resolution as sparse representation of raw image patches. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008.

26. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014.
27. 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.
28. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z.; Magic Pony Technology; Imperial College London. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015.
30. Mao, X.; Shen, C.; Yang, Y.-B. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
31. Wang, Y.; Wang, L.; Wang, H.; Li, P. *End-to-End Image Super-Resolution via Deep and Shallow Convolutional Networks*; IEEE: Piscataway, NJ, USA, 2019; Volume 7, pp. 31959–31970.
32. Zeiler, M.D.; Taylor, G.W.; Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
33. Hui, Z.; Wang, X.; Gao, X. Fast and Accurate Single Image Super-Resolution via Information Distillation Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
35. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
36. 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, Honolulu, HI, USA, 21–26 July 2017.
37. Yamanaka, J.; Kuwashima, S.; Kurita, T. Fast and accurate image super resolution by deep CNN with skip connection and network in network. In *Neural Information Processing*; Springer: Cham, Switzerland, 2017.
38. Han, W.; Chang, S.; Liu, D.; Yu, M.; Witbrock, M.; Huang, T.S. Image super-resolution via dual-state recurrent networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
39. Zhang, K.; Zuo, W.; Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
40. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
41. Mei, K.; Jiang, A.; Li, J.; Ye, J.; Wang, M. An Effective Single-Image Super-Resolution Model Using Squeeze-and-Excitation Networks. In Proceedings of the International Conference on Neural Information Processing, Siem Reap, Cambodia, 13–16 December 2018; Springer: Cham, Switzerland, 2018; pp. 542–553.
42. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.
43. Chu, X.; Zhang, B.; Xu, R.; Ma, H. Multi-objective reinforced evolution in mobile neural architecture search. *arXiv* **2019**, arXiv:1901.01074.
44. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep backprojection networks for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

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. Romera, E.; Alvarez, J.M; Bergasa, L.M.; Arroyo, R. Efficient convnet for real-time semantic segmentation. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017.
48. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
49. Mahapatra, D.; Bozorgtabar, B.; Hewavitharanage, S.; Garnavi, R. Image super resolution using generative adversarial networks and local saliency maps for retinal image analysis. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Quebec City, QC, Canada, 10–14 September 2017; Springer: Cham, Switzerland; pp. 382–390.
50. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
51. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed,S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
52. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
53. Khened, M.; Kollerathu, V.A.; Krishnamurthi, G. Fully convolutional multi-scale residual DenseNets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers. *Med. Image Anal.* **2019**, *51*, 21–45. [CrossRef]
54. Zhang, H.; Hong, X. Recent progresses on object detection: A brief review. *Multimed. Tools Appl.* **2019**, *78*, 1–39. [CrossRef]
55. Krig, S. Feature learning and deep learning architecture survey. In *Computer Vision Metrics*; Springer: Cham, Switzerland, 2016; pp. 375–514.
56. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
57. Yang, C.-Y.; Ma, C.; Yang, M.-H. Single-image super-resolution: A benchmark. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014.
58. 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 of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, BC, Canada, 7–14 July 2001.
59. Available online: https://github.com/MarkPrecursor/ (accessed on 26 November 2018).
60. Chollet, F. Keras: The Python Deep Learning Library. Available online: https://keras.io/ (accessed on 6 August 2019).
61. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
62. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In Proceedings of the British Machine Vision Conference, Surrey, UK, 3–7 September 2012.
63. Zeyde, R.; Elad, M.; Protter, M. On Single Image Scale-Up Using Sparse-Representations. In Proceedings of the International Conference on Curves and Surfaces, Oslo, Norway, 28 June–3 July 2012; pp. 711–730.
64. Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Sketch-based manga retrieval using manga109 dataset. *Multimed. Tools Appl.* **2017**, *76*, 21811–21838 [CrossRef]
65. Zhang, Y.; Li, K.; Li, K. ; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
66. Zhu, L.; Zhan, S.; Zhang, H. Stacked U-shape networks with channel-wise attention for image super-resolution. *Neurocomputing* **2019**, *345*, 58–66. [CrossRef]
67. Shamsolmoali, P.; Zhang, J.; Yang, J. Image super resolution by dilated dense progressive network. *Image Vision Comput.* **2019**, *88*, 9–18. [CrossRef]
68. Shen, M.; Yu, P ; Wang, R.; Yang, J.; Xue, L.; Hu, M. Multipath feedforward network for single image super-resolution. *Multimed. Tools Appl.* **2019**, *78* , 1–20. [CrossRef]

69. Li, J.; Zhou, Y. Image Super-Resolution Based on Dense Convolutional Network. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Guangzhou, China, 23–26 November 2018; Springer: Cham, Switzerland; pp. 134–145.
70. Kim, H.; Choi, M.; Lim, B.; Mu Lee, K. Task-Aware Image Downscaling. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
71. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
72. Luo, X.; Chen, R.; Xie,Y.; Qu,Y.; Li, C. Bi-GANs-ST for perceptual image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.