Journal of Multimedia Information System VOL. 8, NO. 2, June 2021 (pp. 101-110): ISSN 2383-7632 (Online)
http://doi.org/10.33851/JMIS.2021.8.2.101

Efficient Multi-scalable Network for Single Image Super Resolution

Honnang Alao[1], Jin-Sung Kim[1], Tae Sung Kim[1], Kyujoong Lee[1*]

Abstract

In computer vision, single-image super resolution has been an area of research for a significant period. Traditional techniques involve interpolation-based methods such as Nearest-neighbor, Bilinear, and Bicubic for image restoration. Although implementations of convolutional neural networks have provided outstanding results in recent years, efficiency and single model multi-scalability have been its challenges. Furthermore, previous works haven't placed enough emphasis on real-number scalability. Interpolation-based techniques, however, have no limit in terms of scalability as they are able to upscale images to any desired size. In this paper, we propose a convolutional neural network possessing the advantages of the interpolation-based techniques, which is also efficient, deeming it suitable in practical implementations. It consists of convolutional layers applied on the low-resolution space, post-up-sampling along the end hidden layers, and additional layers on high-resolution space. Up-sampling is applied on a multiple channeled feature map via bicubic interpolation using a single model. Experiments on architectural structure, layer reduction, and real-number scale training are executed with results proving efficient amongst multi-scale learning (including scale multi-path-learning) based models.

Key Words: Single-image Super Resolution, post-up-sampling, multi-scalable network.

I. INTRODUCTION

Single image super resolution (SISR) is a classical computer vision task to reconstruct a high-resolution (HR) image from a low-resolution image. SISR is used for various applications such as surveillance imaging [1], medical imaging [2] and, HDTV in recent years. In order to improve accuracy in the restored image, there have been a lot of efforts, such as SRCNN [3], VDSR [4], and EDSR [5] which use deep learning that is a breakthrough in image restoration.

Solutions to more accurate image restoration had previously been thought to require deeper networks, with techniques such as deep residual learning [6] and batch normalization [7], which unfortunately lead to a huge sum of parameters. Several other state-of-the-art methods emphasize the importance of architectural structure for better performance, but still require heavy computation. On the other hand, studies for practical operation and high efficiency aim to reduce computation whilst maintaining accuracy. However, studies on multi-scalability have not been enough. Multi-scalability refers to multiple scale image restoration by using a single model, which is essential in practical applications.

Methods implementing pre-up-sampling techniques like SRCNN [3], VDSR [4], DRCN [8] involve an interpolation-based up-sampling method (bicubic interpolation). Networks up-sampling with bicubic interpolation can be trained to restore images by multiple scale factors via a single model. VDSR [4] showed better performance by using a single model trained on multiple scales compared to the performance by using different models for each scale. Scale augmentation can also be considered as data augmentation, thus yielding generalization of the model. Techniques which are also multi-scaled and involve multi-path learning ex. MDSR [5] prove the existence of shared parameters across different restoring scale factors.

1.1. Research contributions

Studies show that pre-up-sampling techniques induce a significant number of operations compared to post-up-sampling SR frameworks. In this paper, we introduce an efficient multi-scalable convolutional neural network constituting post-up-sampling but interpolation-based upscale technique. Similar to BTSRN [9], the proposed network consists of convolutional layers applied on the low-resolution input image and its feature maps, an up-sampling layer, and more convolutional layers applied on the up-scaled feature maps. Unlike BTSRN [9], in the upscale layer, the multiple-channeled feature map of the previous layer is up-scaled by bicubic interpolation inducing multi-scalability. Additionally, unlike other previous works of which the training is performed using only scale factors of 2, 3, and 4, the proposed network is

Manuscript received June 10, 2021; Revised June 23, 2021; Accepted June 24, 2021. (ID No. JMIS-21M-06-021)
Corresponding Author (*): Kyujoong Lee, Sunmoon University, Asan, Rep. of Korea, ***-****-**** kyujoonglee@sunmoon.ac.kr
[1]Department of Computer & Electronic Engineering, Sunmoon University, Asan, Korea, {honnang7, jinsungk, ts7kim}@sunmoon.ac.kr

trained by using real-number scale factors in the range of 1.5 to 4.0.

II. RELATED WORKS

The implementation of convolutional neural networks to execute computer vision tasks such as image classification and image generation has been very successful. We can solve classification tasks, for example, identifying diseases in plants [10], and even group images based on its pixel contents for effective image retrieval from large databases, just as implemented in [11]. Image generation tasks like document binarization [12] are more advanced, and adversarial networks can be implemented. Super resolution falls into the category of image generation, as the output is also an image, but with a higher resolution.

Image restoration can be achieved in several ways, but the applied upscaling method is an essential factor. Traditional upscale methods include nearest-neighboring, bilinear and bicubic interpolation, which are interpolation techniques applied on a 2-Dimensional matrix. Amongst them, bicubic interpolation has the best performance, and it has been applied in various software applications for image upscale. Its efficiency lies in the ability to enlarge a given image to any ratio and scale.

Architectural frameworks like SRCNN [3] show an implementation of bicubic interpolation on low-resolution (LR) images as a preprocessing measure to enlarge them. The images are then refined by the convolutional neural network to produce an output with better quality measured in PSNR. SRCNN [3] was a breakthrough in the area of super resolution due to its deep learning application with the implementation of LR-HR non-linear mapping. However, it proposed a shallow network consisting of only 3-layer and also concluded the impossibility of a deeper network. VDSR [4] on the other hand, was able to implement a deep convolutional network with the application of a residual framework improving output image quality. Further studies such as DRCN [8], DRRN [13] & MemNet [14] on frame-work structure were also made for better performance.

Depending on framework structure, upscaling methods have significant effect on the performance, the number of operations, and the number of learning weights (parameters). FSRCNN [15] and ESPCN [16] do not use interpolation-based upscale. Instead, learning based up-sampling methods (transposed convolution [17] and sub-pixel shuffling [16]) were used, in which up-sampling was implemented at the last layer of the network (which is post-up-sampling), indicating implementation of convolution in the LR space only. This breakthrough improved performance and reduced the number of operations (multi-adds) significantly. It efficiently improved image restoration techniques in general, making them more accurate and faster. SRResNet [18], EDSR [5], SRGAN [18], and other works involving post-up-sampling techniques (mostly sub-pixel shuffle and transposed convolution) have been able to produce state-of-the-art performances in super-resolution. However, these methods require large computing operations and large number of parameters which are impractical. Although performances are outstanding in comparison, they might not be worth it in most application environments, and are thus, not efficient.

Fast, accurate, and efficient approaches such as CARN [19], FALSR [20], BTSRN [9] were made to cope with real-time applications. These have shown possibilities to reduce computations and parameters significantly while maintaining moderate performance, making implementation possible in most environments. Nevertheless, frameworks with transposed convolution or sub-pixel shuffling can train only a single model per a single upscale factor. Therefore, separate models have to be trained to restore images to different scales implying the inability of multi-scalability via a single model.

Looking into earlier SR techniques, VDSR [4] did not only show better performance in comparison to its previous works, but it also introduced multi-scalability via a single model. In previous works, networks are trained separately for upscale factors of 2, 3, and 4. However, VDSR [4] introduced a single model capable of training and testing on different scales. This was possible due to its reliance on the interpolation-based up-sampling technique. At any cost, studies show that pre-up-sampling-based frameworks lead to significantly huge computations and do not perform well compared to post-up-sampling-based SR methods.

More recent studies like MDSR [5] claim to have made a breakthrough on single model multi-scalability by introducing scale multi-path learning. In structures of scale multi-path learning models, there are three output ends for the three (×2, ×3, and ×4) upscale factors, as a result, being able to produce SR outputs of different scales. The first few layers of this architecture have shared parameters proving similarities across different scales. On the downside, practical applications do not involve only fixed number (integers) scale factors. In most practical applications, output images eventually have to implement an interpolation-based technique to produce desired output image size. This would require the need to train a single model involving real-number upscale factors which was unfortunately, even not implemented in VDSR [4] during training. Additionally, when upscaling LR images to a certain HR size, the other ends of the network will be useless taking up memory space. In this paper, we introduce an efficient SR technique, able to output images

of any desired size. We utilize the post-up-sampling method for efficient practical operations.

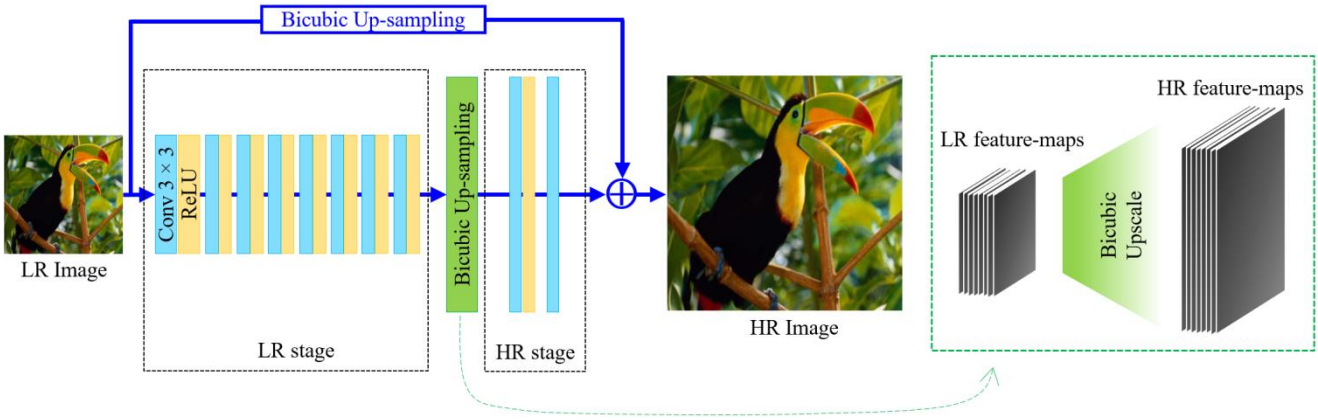

Fig. 1. Architecture of the proposed multi-scale network.

III. PROPOSED METHOD

The CNN deep residual network learns to a non-linear mapping between the ground-truth (high-resolution images) and its low-resolution counterpart.

The identity image is the upscaled LR image via bicubic interpolation, and the network learns its residual for the reconstructed SR result. Therefore, the dataset consists of the LR image, its bicubic upscaled image, and the HR ground-truth image. The residual image, *r*, is given by:

$$r = y - B(x), \tag{1}$$

where B represents the bicubic operation, x is the LR image, and y is the ground truth. The loss function is defined as the mean squared error (MSE) of the residual and the predicted output of the LR image input:

$$L(\theta) = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| F(x_{ij}) - r_{ij} \right\|^2, \tag{2}$$

where n represents the number of training samples, m is the number of image data per a single training sample. F refers to the operation of the network performed on the x input data to produce the predicted output.

The architecture of the proposed method is a two-staged residual network. As expressed in Figure 1, Convolutional layers and ReLU activation layers are for feature extraction, bicubic interpolation is then used for up-sampling, and additional layers are applied for SR image reconstruction. Inspired by VDSR [4], we up-sample the identity image via bicubic interpolation, and add it to the residual output making it a residual network. In the convolutional neural network (CNN), the kernel size and number of filters are 3 × 3 and 64 respectively. ReLU is used as its activation function. In the LR stage, the residual network is deployed with 8 blocks while 2 more blocks are deployed at the HR stage after up-sampling. This network consists of 10 residual blocks in total.

3.1. Real-number multi-scalability

We perform up-sampling with bicubic interpolation on all channels of the extracted feature map on the last layer of the LR stage as shown in Figure 1. Interpolation-based (bicubic) pre-up-sampling SR methods [21] have always been used on the LR image, which is only 1(grayscale) channeled. Up-sampling in this case is applied on channels of extracted feature (64 channels). The up-sampling layer is located not at the very end but a few layers before the last. Therefore, this can be called a post-up-sampling network.

Restored images are able to possess any possible size and ratio utilizing interpolation-based up-sampling techniques due to their nature of referencing surrounding pixels for the upscaled image reconstruction. Super resolution in previous works has not been able to emphasize real-number upscale factors. In practical application, upscale factors to enlarge images are not always fixed. For example, upscaling an HD+ display (1024 × 768) size to fit a 4K UHD (3840 × 2160) display while maintaining the same aspect ratio is impossible with a fixed upscale factor of 2. Its upscale factor is 2.4. The image will have to upscale by a factor of 2, thereby depending on the remaining 0.4 to be upscaled by an interpolation-based technique. Conclusively, interpolation-based up-sampling techniques are essential in almost all applications. Correspondingly, we learn a mapping between LR and HR image datasets, not only with fixed scale factors but also with real-number upscale factors within the range of 1.5 to 4.0. Compared to previous works training on only 3 different upscale factors (×2, ×3, and ×4), we train with 11 different scales for better accuracy in all circumstances. The performance also improves as the complexity in real-number training scale factors increases. We train our model on the limited number of upscaling factors, but

inference implementation can be done to output any size and ratio.

Compared to a pre-up-sampling network like VDSR [4], computation is reduced and performance is better. In comparison with the VDSR [4] model, the Number of parameters, multi-adds, and other properties is as shown in Table 1. More computation is executed in bicubic interpolation compared to bilinear and nearest-neighboring. Figure 1 shows that 64 channeled feature-maps are being up-scaled via bicubic interpolation. Hence, we also calculate and add the number of operations executed by the upscaling layer to the multi-adds column in Table 1. We use the bicubic polynomial equation to calculate the multi-add operations of the up-sampling layer, just as done in [18]. According to the bicubic polynomial equation, to fill up every missing pixel after spreading pixels apart (for upscaling), 9 multi-adds operation has to be executed per missing pixel.

Table 1. Properties in comparison with VDSR [4]. Multi-adds are calculated assuming the resolution of the HR image is 1280 × 720 and upscale factor is ×4.

Properties	Proposed	VDSR [4]
Training scales	1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4	2, 3, 4
Parameters	296K	665K
Multi-adds	49.9G	612.6G
Number of layers	10	20

3.2. LR and HR stages

The architecture consists of three main factors - *feature extraction*, *up-sampling,* and *image reconstruction*. Residual blocks in the LR stage learn a set of 64 channeled feature map to be up-sampled via bicubic interpolation as shown in Figure 1. It plays the most important role in this framework, hence consists of 8 layers. Feature-maps in the up-sampling layers are the results of the analyzed LR image, creating the best format to up-sample via bicubic interpolation. After feature map up-sampling, the HR stage has 2 residual layers for HR image reconstruction.

IV. EXPERIMENTS AND RESULTS

The performance of the proposed method is evaluated, and it is compared with the performance of the VDSR [4], and other multi-scale networks. For all experiments excluding benchmarking, we utilize the 291 images in [4].

4.1. Training

For training, we crop images in the dataset to make the LR sub-images (image patches) the size of 20 × 20. The network is trained with 11 different scales increasing by 0.25 from 1.5 to 4.0, which means that the sizes of HR sub-images are 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, and 80, respectively. Because sub-image sizes cannot be represented in float types, the LR sub-image size is carefully chosen to put the receptive-field concept into account. The LR sub-image size and the 0.25 step size of scale are determined in order to make the corresponding HR sub-image sizes integer figures. Data augmentations included: flip, rotation, and downsizing, with dataset increasing due to scale number complexity. Cropped HR image patch was downscaled via bicubic interpolation to create x (LR) dataset. LR and HR sub-images in each batch have to be of the same size, hence each batch represented an upscale factor, and training iteration was performed on a randomly assigned batch.

We train the models with adaptive momentum optimizer [22] and 128 mini-batch sizes. Training is done over 275,000 iterations with an initial learning rate of 10^{-3} exponentially decaying to 10^{-5}. Iterations are roughly the same (a little over 275,000) for all performed experiments. Maintaining iteration number, more epochs (to repeat iterations of the same set of data) are needed for training on the deeper networks due to the reduced amount of dataset. Deeper networks have larger receptive fields, hence require larger sub-image sizes. Xavier normal [23] is used to initialize weights before training. All implementations are executed utilizing the PyTorch [24] deep learning tool and training lasts for roughly 5 hours on RTX 2080.

4.2. Datasets

After image cropping, the size of the HR sub-images ranges from 30 up to 80. In VDSR [4], the dataset was created downsizing by all required scales, upscaling via bicubic and cropping. All cropped sub-images are combined forming a larger dataset. In the proposed method, however, cropping images for a single network is a tricky task, because the number of all datasets have to be equal across all scales for equality during training. Data augmentation techniques with the priority being – original image, left-right flipping, rotation by 90°, 180°, 270°, and downsizing are used to solve this problem. First, we crop the dataset images to sub-images for the largest HR sub-image size needed (80 × 80). The amount of the cropped HR sub-images is then used as the limit value to stop creating more sub-images (by cropping) when reached by other HR sub-image sizes. Hence, not all augmentation techniques will be utilized as scale factor decreases.

Set 5 [26], Set14 [27], BSD 100 [28] and Urban 100 [30] datasets are used for testing and comparing results with previous works. The datasets, especially Set 5 [26] with different scale factors are used to evaluate the performance

Table 2. Results based on LR-HR stage residual blocks. The Set 5 dataset is used for testing. SSIM is calculated with the aid of [25].

Proposed PSNR (dB) and SSIM results				VDSR [4]
LR-HR Scale	17 - 3 PSNR / SSIM	18 - 2 PSNR / SSIM	19 - 1 PSNR / SSIM	0 - 20 PSNR / SSIM
× 2	37.5733 / 0.9588	37.5659 / 0.9588	37.0928 / 0.9513	37.53 / 0.9587
× 3	33.9024 / 0.9232	33.8976 / 0.9229	33.3120 / 0.9072	33.66 / 0.9213
× 4	31.5304 / 0.8865	31.5334 / 0.8859	30.8687 / 0.8622	31.35 / 0.8838
Average	34.34 / 0.9228	34.33 / 0.9225	33.76 / 0.9069	34.18 / 0.9213

of different structures and strategies of the proposed method.

4.3. LR and HR stages

It was mentioned in Section 3 that the LR stage and HR stage represent feature extraction and image reconstruction respectively. However, to reduce the computational complexity of the network, experiments are made to reduce the number of residual blocks on the HR stage while increasing those of the LR stage. This experiment is also done in comparison with VDSR [4].

For a fair comparison, training is done with 20 layers and with the upscaling factor of 2, 3, and 4. The number of the LR and HR stages residual blocks lead to difference in performance, showing the importance of the up-sample layer's location. To maintain the number of residual blocks in the network, the increased feature extracting layers (layers in LR stage) means reducing HR image reconstructing layers. As shown in Table 2, when LR-HR blocks are 17 - 3 and 18 - 2, the difference in performance measured in PSNR is less than 0.01 which is negligible.

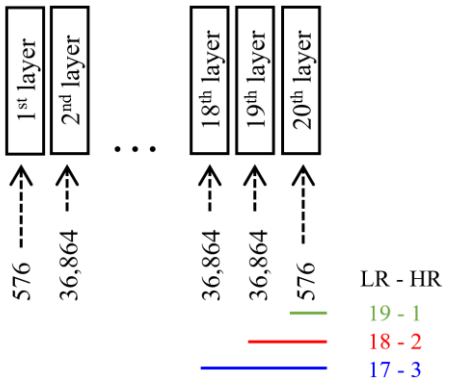

Fig. 2. The number of parameters per layer. The colored lines' lengths cover the number of layers and parameters in the HR stage. Green shows the 19-1, red is the 18-2, and blue is the 17-3 LR-HR stages' layers. Each hidden layer has 36,864 parameters.

19 - 1 LR-HR blocks display reduction in performance due to excessive reduction in the number of parameters as shown in Figure 2. 19-1 implies 19 LR blocks and just 1 layer on the HR space, which is the last layer of the network (the first and last layers of SR networks usually possess the least number of parameters). The difference in parameter numbers in the HR stages between 17-3, 18-2, and 19-1 are 74,304, 37,440, and 576 (each hidden layer has 36,864 parameters) respectively. 576 parameters are too small for image reconstruction regardless of having more in the LR stage, therefore 37,440 parameters in the HR stage (18-2 LR-HR blocks) were concluded as the best for a trade-off between performance and efficiency of the network. Compared to VDSR [4], this (20 layers 18-2) network reduces the computation significantly. The training process can be observed in Figure 3, and there is little to no difference in performance between 17-3 and 18-2.

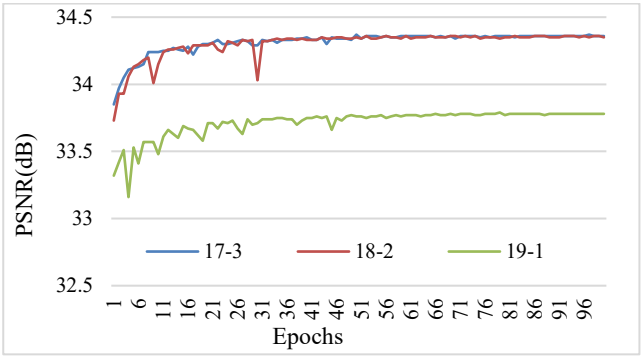

Fig. 3. Training comparison between 17-3, 18-2, 19-1 LR-HR residual blocks. Results are the average PSNR of ×2, ×3, and ×4 upscale factors, on the Set 5 dataset

4.4. Reduction of residual blocks

For more efficient results, we reduce the number of residual blocks to 10 layers. The previous experiment emphasizes the need for 2 layers in the HR stage, which denotes that reduction should be done in the LR stage layers if required. Reduction from 18-2 (20 layers) to 8-2 (10 layers) LR-HR stages are experimented and results were similar as shown in Table 3. The difference of the overall average on the testing datasets is approximately 0.1 (in PSNR), which is trivial. Although the VDSR [4] model is successfully trained on the 291 [29], [30] images dataset, its receptive field was not large (due to pre-up-sampling) compared to the proposed method, hence sub-images had to be cropped to a large size, making 291 [29],

[30] images dataset insufficient on the proposed 20 layers (18-2) network. The 10 layers network had a similar performance while the amount of computation is reduced by 29.8% compared to the 18-2 LR-HR layers, and by 91.9% compared to VDSR [4], when we assume the HR image resolution is 720P and the upscaling factor is ×4.

Table 3. Results based on LR-HR stage residual blocks. To calculate the multi-adds, we assume that the HR image is 720P and the upscaling factor ×4.

Dataset	LR-HR / Scale	18 - 2 PSNR/SSIM	8 - 2 PSNR/SSIM	VDSR [4] PSNR/SSIM
Set 5	× 2	37.57/0.9588	37.52/0.9585	37.53/0.9587
	× 3	33.90/0.9229	33.85/0.9226	33.66/0.9213
	× 4	31.53/0.8859	31.48/0.8854	31.35/0.8838
Set 14	× 2	33.01/0.9139	33.01/0.9137	33.03/0.9124
	× 3	29.87/0.8367	29.86/0.8364	29.77/0.8314
	× 4	28.26/0.7787	28.26/0.7782	28.01/0.7674
B100	× 2	31.91/0.8959	31.89/0.8954	31.90/0.8960
	× 3	28.85/0.7985	28.84/0.7981	28.82/0.7976
	× 4	28.34/0.7266	27.33/0.7262	27.29/0.7251
Urban 100	× 2	30.77/0.9146	30.73/0.9138	30.76/0.9140
	× 3	27.14/0.8289	27.14/0.8286	27.14/0.8279
	× 4	25.25/0.7554	25.26/0.7553	25.18/0.7524
Average		30.53/0.8514	30.43/0.8510	30.37/0.8490
Parameters		665K	296K	665K
Multi-adds		71.1G	49.9G	612.6G

4.5. Real-number multiscale training

All results in Tables 2 and 3 are based on experiments done by training with only 2, 3, and 4 upscale factors for fair comparison and accurate evaluation. In reality, however, image restoration to up-sample images to any size should be possible. Table 4 shows results based on training the 10 layers network by more complex upscaling factors. The model was additionally trained by upscaling factors in the range of 1.5 to 4.0 with a step of 0.5 (1.5, 2, 2.5, 3, 3.5, 4), and also with a step of 0.25 (1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4). The difference is between 3, 6, and 11 upscaling factor values. Greater numbers of upscaling factor values give more complexity to the model. The results of VDSR [4] with real-number upscaling factors are obtained by using the official model in [31]. VDSR [4] results have poor performance especially on 1.5 and 1.75 scales, possibly due to its nature of pre-upscaling.

Results are tested with Set 5 [26] and with the 11 scaling factors. The results in Table 4 lead to a conclusion that scale complexity improves performance. More scale augmentation creates more batches for training which is essential. Theoretically, reconstructed (SR) images would have more quality on real-number upscale compared to models that are trained with only 2, 3, and 4 upscale factors.

4.6. Comparisons with State-of-the-Art methods

All the previous experiments are done by training using the addition of T91 [29] and BSD200 [30] image datasets just as performed in VDSR [4]. We use the addition of T91 [29], BSDS 200 [30] and General 100 [32] to train the model for benchmarking results on Table 5 and Figure 4.

Two main key elements in this paper are real-number multi-scalability and efficiency in practical implementation. Models such as CARN [19], FALSR [20], BTSR [9], and OISR [33] are state-of-the-art methods excelling in efficiency while maintaining impressive levels of performance. Nevertheless, they do not have the ability to restore images with multiple upscaling factors, thus needing multiple trained models for implementation on several upscale factors, which we can arguably be referred to as inefficient. Therefore, the comparison was done with state-of-the-art methods that can perform multi-scale learning using a single model.

Table 4. Real-number multi-scale training comparison. The red shows the best while the blue indicates the second-best performance.

Dataset	Trained scales / Testing scale	**2**, **3**, **4** PSNR/SSIM	1.5, **2**, 2.5, **3**, 2.5, **4** PSNR/SSIM	1.5, 1.75, **2**, 2.25, 2.5, 2.75, **3**, 3.25, 3.5, 3.75, **4** PSNR/SSIM	VDSR [4, 28] (**2**, **3**, **4**) PSNR/SSIM
Set 5	1.5	40.40/0.9766	40.62/0.9771	40.59/0.9770	33.54/0.9503
	1.75	38.81/0.9677	38.81/0.9677	38.82/0.9677	35.91/0.9572
	2	37.52/0.9585	37.49/0.9584	37.49/0.9585	37.53/0.9587
	2.25	36.40/0.9496	36.34/0.9495	36.37/0.9495	35.12/0.9416
	2.5	35.39/0.9404	35.41/0.9406	35.45/0.9406	34.34/0.9325
	2.75	34.66/0.9326	34.63/0.9327	34.68/0.9328	33.54/0.9265
	3	33.85/0.9226	33.83/0.9226	33.86/0.9228	33.66/0.9213
	3.25	33.18/0.9137	33.13/0.9137	33.20/0.9138	32.47/0.9080
	3.5	32.64/0.9042	32.66/0.9045	32.69/0.9046	32.18/0.9006
	3.75	32.00/0.8957	31.99/0.8955	31.98/0.8953	31.57/0.8914
	4	31.48/0.8854	31.51/0.8852	31.50/0.8853	31.34/0.8838

Journal of Multimedia Information System VOL. 8, NO. 2, June 2021 (pp. 101-110): ISSN 2383-7632 (Online)
http://doi.org/10.33851/JMIS.2021.8.2.101

Table 5. Comparison with light weight state-of-the-art models. We assume the HR image is 720P. The red shows the best while the blue indicates the second-best performance.

Scale	Model	Params	Multi-Adds	Set 5 PSNR/SSIM	Set 14 PSNR/SSIM	B 100 PSNR/SSIM	Urban 100 PSNR/SSIM	Real-number Upscale
2	VDSR [4]	665K	612.6G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	No
	LapSRN [34]	813K	29.9G	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100	No
	MPRNet [31]	538K	-	38.08/0.9608	33.79/0.9196	32.25/0.9004	32.52/0.9317	No
	EMSR (ours)	296K	94.5G	37.57/0.9588	33.16/0.9144	31.95/0.8959	30.99/0.9162	Yes
3	VDSR [4]	665K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	No
	MPRNet [31]	538K	-	34.57/0.9285	30.42/0.8441	29.17/0.8073	28.42/0.8578	No
	EMSR (ours)	296K	61.4G	33.92/0.9235	29.91/0.8370	28.89/0.7980	27.31/0.8324	Yes
4	VDSR [4]	665K	612.6G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	No
	LapSRN [34]	813K	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560	No
	MPRNet [31]	538K	31.3G	32.38/0.8969	28.69/0.7841	27.63/0.7385	26.31/0.7921	No
	EMSR (ours)	296K	49.9G	31.59/0.8863	28.33/0.7797	27.36/0.7267	25.38/0.7590	Yes

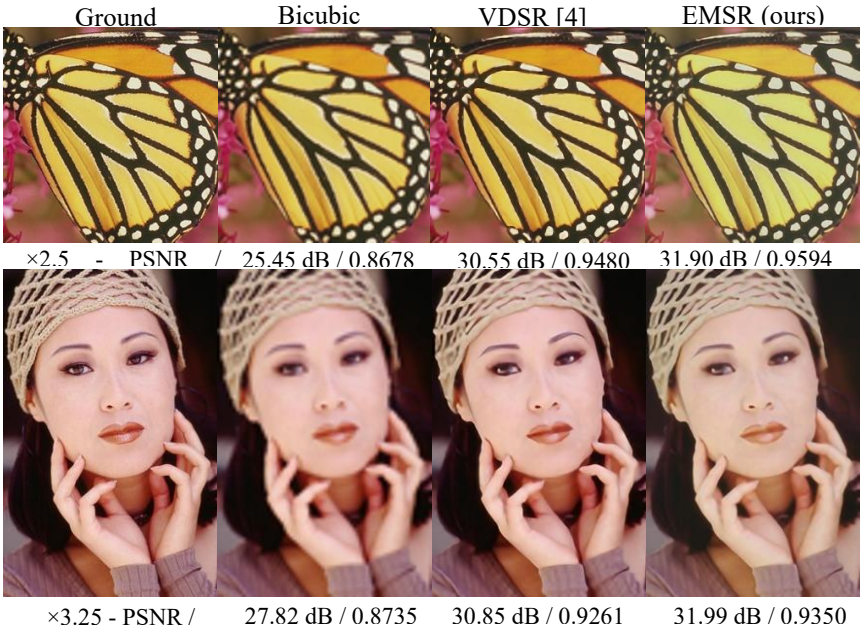

Fig. 4. Trade-off between performance vs number of parameters and number of operations computed, assuming that the HR resolution is 720P. Results with upscale factor ×4 tested with Set5 [26], Set14 [27], BSD100 [29] and Urban100 [28] datasets. Note, the *x* axis is a log scale.

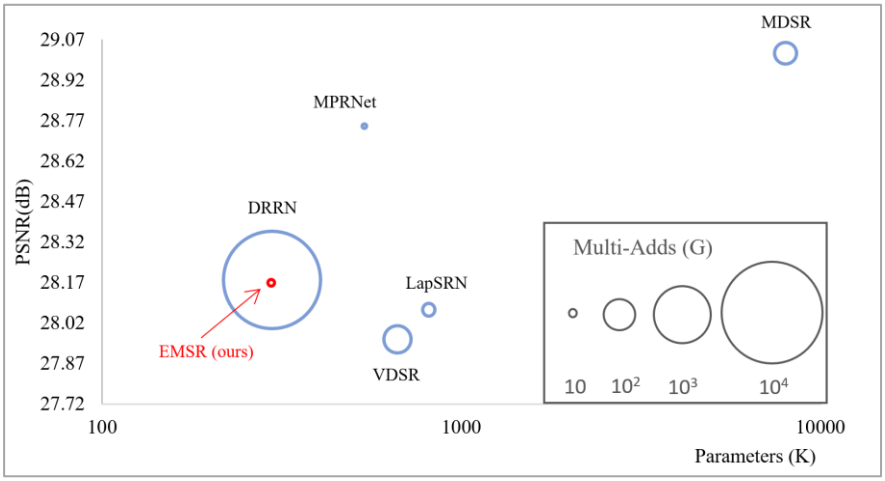

Fig. 5. Visual comparison between the ground-truth, bicubic and restored image, with the butterfly and woman images of the Set 5 [26] image dataset. Scale factors ×2.5 and ×3.25 are used.

The comparison is performed with VDSR [4], LapSRN [33], MDSR [5], DRRN [13] and MPRNet [35]. Results in Figure 4 are based on scale factor ×4 to show superiority in post-up-sampling techniques. The results prove efficiency in parameters and operation numbers while maintaining good performance.

The proposed model is also able to perform real-number upscaling. The results in table 5 leaves out MDSR [5] and DRRN [13] due to their bulkiness in computation and parameters. MPRNet [35] provides good results with less computation, but Table 5 show that more parameters are used compared to the proposed model.

Figure 5 shows visual comparison with VDSR [4]. Major differences cannot be seen visually, but the results in PSNR indicate improvement in performance.

V. CONCLUSION

The proposed (EMSR) model is able to find a break-through in interpolation-based post-up-sampling with a more realistic and efficient outcome.

Using a single model for vast complexity with a small number of parameters and less computation pushes its ability to the limit without waste. Our model can generate images to any possible size and ratio well within the range of trained upscale while maintaining a very reasonable amount of quality. Further works can be done on the architectural structure for even better performance whilst maintaining its efficiency.

Acknowledgement

This work was supported by the R&D Program of MOTIE/KEIT (No. 20010582, Development of deep learning based low power HW IP design technology for image processing of CMOS image sensors).

REFERENCES

[1] Wilman WW Zou and Pong C Yuen, "Very low-resolution face recognition problem," *IEEE Transactions on image processing*, vol. 21, no. 1, pp. 327-340, 2011.

[2] Wenzhe Shi, Jose Caballero, Christian Ledig, Xiahai Zhuang, Wenjia Bai, Kanwal Bhatia, Antonio M Simoes Monteiro de Marvao, Tim Dawes, Declan O'Regan, and Daniel Rueckert, "Cardiac image super-resolution with global correspondence using multi-atlas patchmatch," in *Proceeding of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 9-16, 2013.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, "Image superresolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI)*, vol. 38, no. 2, pp, 295-307, 2015.

[4] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1646-1654, 2016.

[5] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 136-144, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv*:1502. 03167, 2015.

[8] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1637-1645, 2016.

[9] Y. Fan, H. Shi, J. Yu, D. Liu, W. Han, H. Yu, Z. Wang, X. Wang, T. S. Huang, "Balanced two-stage residual networks for image super-resolution" in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 161-168, 2017.

[10] S. Mukherjee, P. Kumar, R. Saini, P. P. Roy, D. P. Dogra, B. G. Kim, "Plant disease identification using deep neural networks," *Journal of Multimedia Information System*, vol. 4, no. 4, pp. 233-238, 2017.

[11] S. Gupta, P. P. Roy, D. P. Dogra, B.-G. Kim, "Retrieval of colour and texture images using local directional peak valley binary pattern," *Pattern Analysis and Applications*, vol. 23, no. 4, pp. 1569-1585, 2020.

[12] A. K. Bhunia, A. K. Bhunia, A. Sain, P. P. Roy, "Improving document binarization via adversarial noise-texture augmentation," in *Proceeding of 2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2721-2725, 2019.

[13] Y. Tai, J. Yang, X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the Conference on Computer Vision and Pattern*

Recognition (CVPR), pp. 2790-2798, 2017.

[14] Y. Tai, J. Yang, X. Liu, C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 4539-4547, 2017.

[15] C. Dong, C. C. Loy, X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 392-407, 2016.

[16] W. Shi, J. Caballero, F. Husz´ar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, "Real-time single image and video super-resolution using an efficient sub-Fast, Accurate, and Lightweight Super-Resolution with CARN," *arXiv:1803.08664v5*, 2018.

[17] H. Gao, H. Yuan, Z. Wang, and S. Ji, "Pixel transposed convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI)*, vol. 42, no. 5, pp. 1-10, 2019.

[18] C. Ledig, L. Theis, F. Husz´ar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang et al., "Photorealistic single image super-resolution using a generative adversarial network*,"* in *Proceeding of IEEE Computer Vision and Pattern Recognition*, pp. 105-114, 2017.

[19] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1-17, 2018.

[20] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, Jixiang Li, and Qingyuan Li, "Fast, accurate and lightweight superresolution with neural architecture search," *arXiv preprint arXiv:1901.07261*, 2019.

[21] Lukáš Miňo, Imrich Szabó, and Csaba Török, "Bicubic splines and biquartic polynomials," *Open Comput. Sci*., vol. 6, pp. 1–7, 2016.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR, abs/1412.6980*, pp. 1-15, 2014.

[23] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision (ECCV)*, pp. 184–199, 2014.

[24] Pytorch deep learning framework, *pytorch.org*.

[25] HolmesShuan. EDSR-ssim. github: *https://github.com/HolmesShuan/EDSR-ssim*, 2018.

[26] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel, "Low-complexity single- image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 1-10, 2012.

[27] Roman Zeyde, Michael Elad, and Matan Protter "On single image scale-up using sparse-representations," in *Proceeding of International conference on curves and surfaces,* pp. 711–730, 2010.

[28] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, ZhangyangWang, Shiyu Chang, and Thomas Huang, "Robust video super-resolution with learned temporal dynamics," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2507–2515, 2017.

[29] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image superresolution via sparse representation," *IEEE Transactions on Image Processing (TIP)*, vol. 19, no. 11, pp. 2861–2873, 2010.

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

[31] Huangzehao. Implementation of "Accurate Image Super-Resolution Using Very Deep Convolutional Networks". *Github https://github.com/huangzehao/caffe-vdsr*, 2017.

[32] C. Dong, C. C. Loy, and X. Tang, "Accelerating the superresolution convolutional neural network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1-17, 2016.

[33] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng, "Ode-inspired network design for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1732–1741, 2019.

[34] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and MingHsuan Yang "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 624–632, 2017.

[35] Armin Mehri, Parichehr B. Ardakani, and Angel D. Sappa, "Multi-Path Residual Network for Lightweight Image Super Resolution," in *Proceeding of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2704-2713*,* 2021.

Authors

Honnang Alao received B. S. degree in Electronic Engineering, at Sunmoon University, in 2020.
Currently, he is pursuing Master's degree in Computer and Electronic Engineering, at Sunmoon University, since 2021.

Fields of interest are Deep learning, image processing, Multimedia.

Jin-Sung Kim received B.S., M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 1996, 1998, and 2009, respectively. From 1998 to 2004 and from 2009 to 2010, he was with the PDP Development Group, Samsung SDI Ltd. as a Manager. From 2010 to 2011, he was a Post-Doctoral Researcher with Seoul National University. From 2011 until now, he is a professor in the Electrical Engineering department at Sunmoon University.

Fields of interest are pattern recognition, video compression and image enhancement and driving systems for flat panel displays.

Tae Sung Kim joined the research Institution for new media Communications in Seoul University from 2017 to 2018. He was a senior researcher in Samsung S.LSI from 2018 to 2021 and is currently an assistant professor in the Electronic Engineering department of Sunmoon University, since 2021.

Kyujoong Lee received Bachelor's degree in Electronic Engineering, at Seoul National University, in 2002 and master's degree in Electronic Engineering, at University of Southern California, in 2008. In 2013 he received his Doctorate's degree in Electronic Engin-eering, at Seoul National University, and was a senior researcher in Samsung S.LSI from 2013 to 2017. He joined Sunmoon University since 2017, and is currently an associate professor of the Electronic Engineering department.

Fields of interest are Deep learning, image processing, image co mpression Multimedia, SOC design.