

# Interpolation based Single-path Sub-pixel Convolution for Super-Resolution Multi-Scale Networks

Honnang Alao<sup>1</sup>, Jin-Sung Kim<sup>1</sup>, Tae Sung Kim<sup>1</sup>, Juhyen Oh<sup>1</sup>, and Kyujoong Lee<sup>1\*</sup>

## Abstract

Deep learning convolutional neural networks (CNN) have successfully been applied to image super-resolution (SR). Despite their great performances, SR techniques tend to focus on a certain upscale factor when training a particular model. Algorithms for single model multi-scale networks can easily be constructed if images are upscaled prior to input, but sub-pixel convolution upsampling works differently for each scale factor. Recent SR methods employ multi-scale and multi-path learning as a solution. However, this causes unshared parameters and unbalanced parameter distribution across various scale factors. We present a multi-scale single-path upsample module as a solution by exploiting the advantages of sub-pixel convolution and interpolation algorithms. The proposed model employs sub-pixel convolution for the highest scale factor among the learning upscale factors, and then utilize 1-dimension interpolation, compressing the learned features on the channel axis to match the desired output image size. Experiments are performed for the single-path upsample module, and compared to the multi-path upsample module. Based on the experimental results, the proposed algorithm reduces the upsample module's parameters by 24% and presents slightly to better performance compared to the previous algorithm.

**Key Words:** Super-resolution, multi-scalable network, multi-scale single-path Super-resolution.

## I. INTRODUCTION

Image super-resolution (SR) is an important task in computer vision to increase or recover the size of a low-resolution (LR) image, generating a high-resolution (HR) output. This is usually referred to as single image super-resolution (SISR). SISR is an ill-posed problem, as there are various solutions for any LR image. Applications on SISR in recent years can be found in surveillance imaging [1], medical imaging [2], High-definition television, and more.

One of the traditional image upscaling methods involve the use of interpolation algorithms to increase image sizes. Thus, surrounding pixel data are utilized for generating the required additional pixel values.

Image super-resolution using Deep Convolutional Networks [3], also known as SRCNN was proposed by Dong et al to well tackle this problem and is the pioneer of deep learning-based SR study.

SRCNN [3] was performed in the HR space only by upscaling LR images before input via bicubic interpolation, but a Fast Super-Resolution Convolutional Neural Networks [4] (FSRCNN) was also proposed with transposed convolution to learn the upsampling process

with the LR image as input. Additionally, an Efficient Sub-pixel Convolutional Neural Network (ESPCN) [5], which was proposed by Wenzhe Shi, efficiently generates HR images directly from the LR space and has been used by various algorithms as the standard upsampling module.

More complex and advance SISR algorithms successfully improve the performance in terms of PSNR (peak-to-signal-ratio). Namely, Enhanced Deep Residual Networks for Single Image Super-Resolution [6], and Residual Channel Attention Networks [7], usually known as EDSR and RCAN respectively. They prove the impact of network architecture on recovering image details for better performance. Attention mechanisms utilizing the channel and spatial attention are also key factors on SR performance in recent years. However, these algorithms do not only require heavy computation and huge amounts of parameters but are also able to upsample images by only a single scale factor with a single network.

More efficient lightweight SR networks like the Fast, Accurate, and Lightweight Super-Resolution with Neural Architecture Search [8], known as FALSr, use advance and more complex algorithms aiming at maintaining moderate performance while reducing the computational burden. Nevertheless, they aren't able to upsample LR images by various scale factors with a single model.

**Manuscript received September 10, 2021; Revised October 04, 2021; Accepted October 20, 2021. (ID No. JMIS-21M-09-031)**

Corresponding Author (\*): Kyujoong Lee, Sunmoon University, Asan, Rep. of Korea, 010-5219-0716, kyujoonglee@sunmoon.ac.kr

<sup>1</sup>Department of Computer & Electronic Engineering, Sunmoon University, Asan, Korea, {honnang7, jinsungk, ts7kim, ohjuhyeon03}@sunmoon.ac.kr

### 1.1. Research contributions

Multi-scale Deep Super-Resolution [6], Cascading Residual Network [9], and Multi-path Residual Network [10] referred to as MDSR, CARN, and MPRNet respectively, are multi-path and multi-scale SR algorithms, which can output HR images of various sizes via a single model. They require separate pathways depending on the selected upscale factor and have outstanding results. However, each pathway has to be trained for a specific upscale factor, leaving out the rest, which can be considered a waste of parameters. In this paper, we propose a single-path upscale algorithm, utilizing all parameters of the model for every upscale factor. This reduces the network overall parameters while maintaining its performance.

The rest of the paper is organized in the following order. Section II gives reference to existing SR multi-scale learning algorithms. Section III shows an analysis of the problems in sub-pixel convolution for upsampling in SR and proposes a solution. Experimental results are shown in section IV, which leads to conclusions given in section V.

## II. RELATED WORKS

In recent years, deep learning has been utilized for various computer vision tasks such as facial expression recognition [11], segmentation, etc. Dong et al were the first to use the deep learning convolutional neural network in SISR. The algorithm is known as SRCNN [3].

The SRCNN [3] network requires an upscaled image input to construct the desired HR image output. Bicubic interpolation is used to perform the input image preprocessing task not only in SRCNN [3], but also in models from algorithms including Very Deep Super-Resolution Networks [12] (VDSR), Deeply-Recursive Convolutional Network [13], and others. Therefore, the networks process images in the HR space only, which increases computation significantly and makes it impossible to analyze images in the LR space. Algorithms like the Efficient Multi-scale Super-Resolution [14], and, Balanced Two-Stage Super-Resolution [15] operate on images in both the LR and HR space for more accurate results.

### 2.1. Learning-based upsampling

Transpose convolution, also known as deconvolution, was proposed in FSRCNN [4] aiming at generating the HR output image in the last layer for efficiency and acceleration. This significantly reduced the computational burden and did not require bicubic input preprocessing. Improvement in performance with even fewer parameters compared to SRCNN [3] was also realized, concluding that operation on the LR space is essential in SISR.

ESPCN [5] proposed by Wenzhe Shi, introduced sub-pixel convolution for image upsampling, which also operates on images in the LR space.

Utilizing the sub-pixel convolution upsampling module, huge and complex models like EDSR [6], and RCAN [7] offer outstanding performance. Algorithms that generate images visually pleasing to the human eye such as Super-Resolution using a Generative Adversarial Network [16], also use sub-pixel convolution to upsample images to the desired size. A similar technique known as sub-pixel mapping was implemented for text detection from video frames [17].

Unlike complex and huge computational models, FALSr [8] and CARN mobile [9] are lightweight models for efficient real-time implementation with a good performance-to-efficiency trade-off. They also utilize a learning-based upsampling technique.

Although learning-based upsampling methods are efficient and effective in terms of performance and efficiency, the limitation of these methods is that multi-scale learning is not possible. As first proven in VDSR [12], networks utilizing the interpolation-based upsampling method can train on images with various scales as the input image is upscaled before implementation. VDSR [12] proved better performance when trained with various scales. Compared to the methods training separate models for separate upscale factors, the advantage of VDSR [12] is that it trains a single model for multiple upscale factors, which saves parameters considerably.

### 2.2. Multi-scale multi-path learning

Multi-scale multi-path learning or scale-specific multi-path learning is the process of learning for various scale factors with separate paths. This algorithm utilizes a single model efficiently for various scale factors. It is widely used in various methods such as MDSR [6], CARN [9], and MPRNet [10].

### 2.3. EDSR and MDSR architecture

MDSR [6], which is an extension of the EDSR [6] model claim to realize a breakthrough in multi-scale training with sub-pixel convolution. They implement multi-path learning for the separate scale factor. FSRCNN [4] shows that after the whole model was trained for a certain scale factor, the whole model didn't need training for the other scale factors. The performance obtained by training only the transpose convolutional layer for the other scale factors is the same as training the whole model for the other scale factors. This proves that shared parameters across various scale factors are present, which MDSR [6] use to their advantage.

MDSR [6] used multi-path learning for various scale factors. Multi-path learning in MDSR [6] consists of two

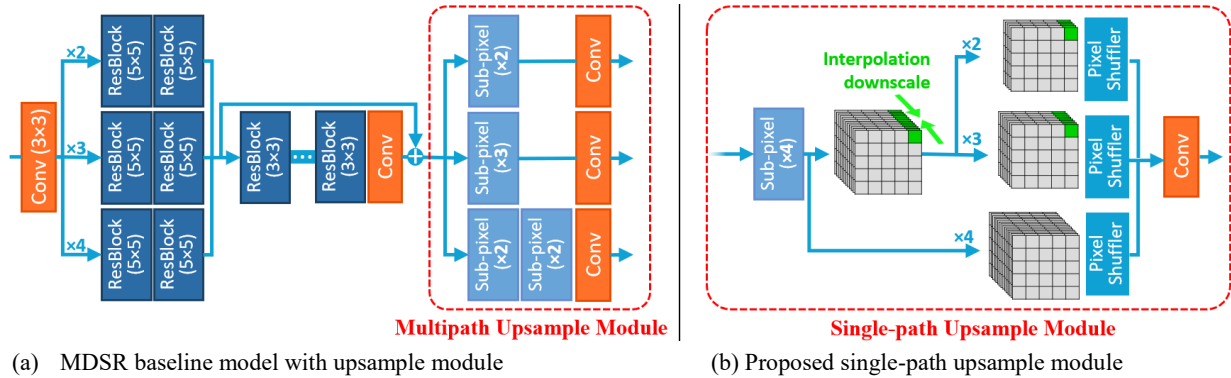


Fig. 1. MDSR model architecture with its sub-pixel convolutional upsample module, and the proposed upsample module represented in the red broken lines. The proposed module uses a linear downscale on the channel axis for each scale.

elements. The first one is the preprocessing module for each scale factor separately, the second one is the sub-pixel upsample module for the separate scale factors. The central part of the network has shared parameters across all scale factors. During training, the central convolutional layers are trained for every scale factor, but each of the multi-path layers is trained for only one of the scale factors.

### III. PROPOSED METHOD

We present a multi-scale single-path module exploiting the strong points of sub-pixel convolution and multi-scale training, and utilizing a single path for training. Our proposed method overcomes the need for multi-path learning and uses all the parameters for all upscale factors.

#### 3.1. Sub-pixel convolution for upsampling in SR

As first proposed in ESPCN [5], sub-pixel convolution is achieved by applying convolution to output a feature of  $s^2 \times n$  channels. Then, pixel shuffling is applied by rearranging the pixels to increase the width and height of the feature by  $s$  while reducing the channel dimension to  $n$ . Therefore, channel dimensions are different for all scale factors. Note that  $s$  in  $s^2 \times n$  represents the upscaling factor. As a result, MDSR [6] uses separate sub-pixel convolutional layers for different scale factors.

#### 3.2. Problems in multi-path Upsample Module

As shown in figure 1(a) MDSR [6] upsample module, the sub-pixel convolutional layer is used for each scale factor separately. MDSR [6] network use 64 filters for every layer, therefore, for scale factor  $\times 2$  in sub-pixel convolution,  $2^2 \times 64 \times 64$  filters are needed for the pixel shuffle upscaling process. Scale factor  $\times 3$  path layer needs  $3^2 \times 64 \times 64$  filters, which is reasonable. However, in the  $\times 4$  upscale layer,  $4^2 \times 64 \times 64$  filters cause a very huge number of parameters

and therefore is replaced by doubling the scale factor  $\times 2$  upscale filter. Although it is an intuitive solution, it causes an imbalance in the number of parameters on the  $\times 3$  and  $\times 4$  upscale modules ( $3^2 \times 64 \times 64 > 2^2 \times 64 \times 64 \times 2$ ), which makes the  $\times 3$  upscale module possess the greatest number of parameters.

Another observed problem is that during training, each multi-path branch representing a certain scale factor will be trained one-third of the time compared to the central layers of the network. Moreover, when training for scale  $\times 2$  for example,  $\times 3$  and  $\times 4$  upscale layers are useless for the performance of the  $\times 2$  upscale layer.

#### 3.3. Single-path interpolation-based Upsample Module

We propose a solution as shown in figure 1(b). The green pixels shown in figure 1(b) are treated as the points to be downsampled via 1-dimensional interpolation. Note that the number of the green pixels represents the number of channels. Therefore, downsampling the pixels means reducing the number of channels. Thus, we reduce the channel dimension of the features from  $4^2 \times n$  to  $s^2 \times n$  when the required upscale factor is less than 4.

We firstly utilize the sub-pixel convolution for scale factor  $\times 4$  with  $4^2 \times 64 \times 64$  filters and use a 1-dimensional linear downscale on the channel axis of the feature map depending on the training scale factor. This reduces  $4^2 \times 64$  channeled output to  $2^2 \times 64$ , and  $3^2 \times 64$  channels for  $\times 2$  and  $\times 3$  scale factors while applying no reduction for  $\times 4$  factor upscale. Implementing sub-pixel convolution for scale factor  $\times 4$  allows the model to gather more parameters, and also lets the model perform channel compression for the  $\times 2$ , and  $\times 3$  upscale factors. Thus, it exploits all its parameters for all the needed upscale factors. It can be formulated as:

$$U(F_{LR}) = S_p[D_{it}(W_{sc} * F_{LR})], \quad (1)$$

where the low-resolution feature-map  $F_{LR}$ , upsamples itself by an upsample function  $U$ .  $W_{sc}$ ,  $D_{li}$ , and  $S_p$  represent sub-pixel convolution, linear downscale, and pixel shuffle upscale respectively.

Table 1. Parameters comparison between multi-path and single-path upscale modules. Note, these are parameters of the sub-pixel convolutional layers only.

pscaling factor	Multi-path upscale module par.	Single-path upscale module par.
×2	147.5K	589.8K
×3	331.8K	
×4	294.9K	
Total	774.1K	589.8K

With this solution, all parameters can be used across various scales without waste, which reduces the need for excessive parameters. As expressed in Table 1, the multi-path upscale branches require separate parameters for various scale factors. Consequently, compared to the single-path module, there are fewer parameters for each scale, and an imbalance in parameter numbers between scale factors ×3 and ×4 is observed. The single-path upscale module, on the other hand, uses all its parameters for all upscaling factors and is reduced compared to the total parameters of the multi-path upscale module. The parameters are reduced by 24% in the single-path module. More reduction can be identified when we also consider the last convolutional layers of the network shown in figure 1. This reduction is the same not only for MDSR [6] model but also for CARN [9] and MPRNet [10] as their multi-scale upsampling algorithms are the same as MDSR [6]. However, the lightweight mobile model from the CARN [9] paper called ‘CARN-M’ utilized group convolution in the upsample module, which makes it different from the MDSR [6] upsample algorithm. Nevertheless, parameters also reduce by 24% with the single-path algorithm.

## IV. EXPERIMENTS

For a fair comparison, we train the MDSR [6] baseline model and train for the proposed algorithm by modifying the upsample module only. The same experiment is also performed for CARN-M [9] because its upsample module is composed of group convolutions with a group of 4. Although cubic interpolation is less memory efficient according to [18], we perform experiments on cubic downscale to compare their results with the multi-scale and linear downscale. Therefore, experiments are performed for multi-path, linear single-path, and cubic single-path using the MDSR [6] and CARN-M [9] models.

### 4.1. Datasets and training details

We employ the Div2K [19] RGB data images for training. Image data are cropped into 48×48 patches before training, and data augmentations include; flip and rotation to 90°, 180°, and 270°. The Set5 [20], Set14 [21], B100 [22], and Urban [23] datasets are used for evaluation and comparison.

We use a mini-batch of 16 and the L1 (Mean Absolute Error) loss also known as the MAE loss. Adaptive momentum optimizer [24] with a learning rate of  $10^{-4}$  and halved at every  $2 \times 10^5$  iteration updates. Xavier normal [25] is used as the weight initializer. The models are trained for  $6 \times 10^5$  iterations. Although we trained the MDSR models almost exactly as presented in the EDSR-MDSR [6], paper, we didn’t implement the geometric self-ensemble procedure which was expressed as ‘MDSR+’. We utilize the same settings to train the CARN-M [9] model. For implementation, we utilize the PyTorch deep learning tool, with GPU RTX 2080.

### 4.2. Upsample module performance comparison

As shown in Table 2, we compare results between multi-path, and linear single-path for MDSR [6]. We utilize the

Table 2. Performance comparison with the MDSR [6] model. The red color indicates the best performance while the blue color shows the second-best performance in terms of PSNR and SSIM [26]. We also measure performance with MS-SSIM [27] and UQI [27]

Dataset	scale	Multi-path		Linear single-path		Cubic single-path	
		PSNR / SSIM	MSSSIM / UQI	PSNR / SSIM	MSSSIM / UQI	PSNR / SSIM	MSSSIM / UQI
Set5	×2	<b>37.86</b> / 0.9594	0.9953 / 0.9990	<b>37.87</b> / <b>0.9595</b>	0.9953 / 0.9990	37.85 / <b>0.9595</b>	0.9953 / 0.9990
	×3	<b>34.31</b> / 0.9253	0.9908 / 0.9980	<b>34.31</b> / <b>0.9256</b>	0.9908 / 0.9980	34.29 / <b>0.9256</b>	0.9908 / 0.9980
	×4	<b>32.03</b> / 0.8921	0.9826 / 0.9968	<b>32.05</b> / <b>0.8929</b>	0.9828 / 0.9968	<b>32.03</b> / <b>0.8924</b>	0.9827 / 0.9968
Set14	×2	33.49 / 0.9165	0.9861 / 0.9977	<b>33.54</b> / <b>0.9167</b>	0.9861 / 0.9977	<b>33.53</b> / <b>0.9170</b>	0.9863 / 0.9977
	×3	30.24 / 0.8396	0.9738 / 0.9954	<b>30.30</b> / <b>0.8408</b>	0.9740 / 0.9955	<b>30.29</b> / <b>0.8408</b>	0.9739 / 0.9954
	×4	28.49 / 0.7788	0.9546 / 0.9931	<b>28.51</b> / <b>0.7796</b>	0.9548 / 0.9932	<b>28.54</b> / <b>0.7802</b>	0.9550 / 0.9932
BSDS100	×2	32.10 / 0.8981	0.9823 / 0.9971	<b>32.14</b> / <b>0.8985</b>	0.9824 / 0.9971	<b>32.13</b> / <b>0.8985</b>	0.9824 / 0.9971
	×3	29.04 / 0.8027	0.9666 / 0.9943	<b>29.07</b> / <b>0.8033</b>	0.9668 / 0.9944	<b>29.07</b> / <b>0.8035</b>	0.9668 / 0.9944
	×4	27.53 / 0.7334	0.9433 / 0.9922	<b>27.56</b> / <b>0.7343</b>	0.9436 / 0.9922	<b>27.55</b> / <b>0.7344</b>	0.9436 / 0.9922
Urban100	×2	31.77 / 0.9246	0.9873 / 0.9966	<b>31.90</b> / <b>0.9260</b>	0.9877 / 0.9967	<b>31.89</b> / <b>0.9261</b>	0.9877 / 0.9967
	×3	27.91 / 0.8467	0.9749 / 0.9925	<b>28.00</b> / <b>0.8488</b>	0.9753 / 0.9926	<b>27.99</b> / <b>0.8491</b>	0.9754 / 0.9926
	×4	25.86 / 0.7778	0.9542 / 0.9887	<b>25.94</b> / <b>0.7809</b>	0.9550 / 0.9887	<b>25.93</b> / <b>0.7810</b>	0.9550 / 0.9887

Table 3. Performance comparison with the CARN-M[9] model. The red color shows the best performance and the blue color shows the second-best performance in terms of PSNR and SSIM [23] comparison. We also measure performance with MS-SSIM [27] and UQI [27]

Dataset	scale	Multi-path		Linear single-path		Cubic single-path	
		PNSR / SSIM	MSSSIM / UQI	PNSR / SSIM	MSSSIM / UQI	PNSR / SSIM	MSSSIM / UQI
Set5	×2	37.61 / <b>0.9587</b>	0.9952 / 0.9990	<b>37.62 / 0.9586</b>	0.9951 / 0.9990	<b>37.64 / 0.9586</b>	0.9952 / 0.9990
	×3	<b>33.94 / 0.9228</b>	0.9903 / 0.9979	<b>33.92 / 0.9227</b>	0.9903 / 0.9979	33.91 / 0.9226	0.9903 / 0.9979
	×4	<b>31.69 / 0.8870</b>	0.9815 / 0.9966	<b>31.70 / 0.8873</b>	0.9817 / 0.9966	31.63 / 0.8867	0.9815 / 0.9965
Set14	×2	<b>33.24 / 0.9142</b>	0.9855 / 0.9976	<b>33.26 / 0.9140</b>	0.9855 / 0.9976	33.23 / <b>0.9142</b>	0.9855 / 0.9976
	×3	30.04 / <b>0.8356</b>	0.9729 / 0.9952	<b>30.05 / 0.8355</b>	0.9729 / 0.9952	<b>30.08 / 0.8364</b>	0.9731 / 0.9953
	×4	<b>28.30 / 0.7736</b>	0.9531 / 0.9930	<b>28.31 / 0.7738</b>	0.9531 / 0.9930	<b>28.30 / 0.7739</b>	0.9532 / 0.9929
BSDS100	×2	<b>31.93 / 0.8960</b>	0.9818 / 0.9970	<b>31.94 / 0.8961</b>	0.9818 / 0.9970	<b>31.93 / 0.8960</b>	0.9818 / 0.9970
	×3	28.87 / <b>0.7988</b>	0.9657 / 0.9942	<b>28.88 / 0.7988</b>	0.9657 / 0.9942	<b>28.88 / 0.7987</b>	0.9657 / 0.9942
	×4	27.35 / 0.7278	0.9416 / 0.9920	<b>27.36 / 0.7280</b>	0.9416 / 0.9920	<b>27.37 / 0.7281</b>	0.9417 / 0.9919
Urban100	×2	<b>31.09 / 0.9170</b>	0.9856 / 0.9962	<b>31.09 / 0.9171</b>	0.9856 / 0.9962	<b>31.15 / 0.9176</b>	0.9857 / 0.9962
	×3	<b>27.35 / 0.8331</b>	0.9719 / 0.9918	<b>27.34 / 0.8331</b>	0.9719 / 0.9918	<b>27.39 / 0.8339</b>	0.9721 / 0.9918
	×4	<b>25.38 / 0.7606</b>	0.9488 / 0.9876	<b>25.38 / 0.7607</b>	0.9487 / 0.9876	<b>25.41 / 0.7613</b>	0.9491 / 0.9876

PSNR, structural similarity index (SSIM) [26], multi-scale structural similarity index (MSSSIM) [27], and universal quality index (UQI) [27] to show the achieved result. We only utilize the PSNR and SSIM to show the best and the second-best results. The linear single-path module performs better than the multi-path upsample module. It has a very similar performance to the cubic single-path results.

Shown in Table 3 are the results for the same experiments performed on the CARN-M [9] upsample module. The linear single-path and cubic single-path modules performed better than the multi-path module. However, the cubic single-path module performed slightly better than the linear single-path modules. Thus, the results show the effect the single-path modules have on sub-pixel convolution composed of group convolution.

Figure 2 (a) and (b) show the visual representation implemented with the MDSR [6] and CARN-M [9] model respectively. Due to its computational complexity, cubic single-path was expected to outperform other algorithms, but linear single-path algorithm performs better in terms of PSNR and SSIM in the case of MDSR [6]. It might be that when downscaling the features' channels for pixel shuffling, utilizing cubic interpolation generates unnecessary values. This is because it implements a polynomial curve to fit four reference points. Linear interpolation utilizing just two reference points is enough. However, the output features generated from group convolution are generated from separate input features. That is the reason cubic interpolation works better on CARN-M [9] by using four reference points to downscale the features in the channel axis.

Although it increases the parameters, performance improvement can also be realized if we increase the number of channels of the feature to more than  $4^2 \times n$  before channel compression via interpolation downscale.

## V. CONCLUSION

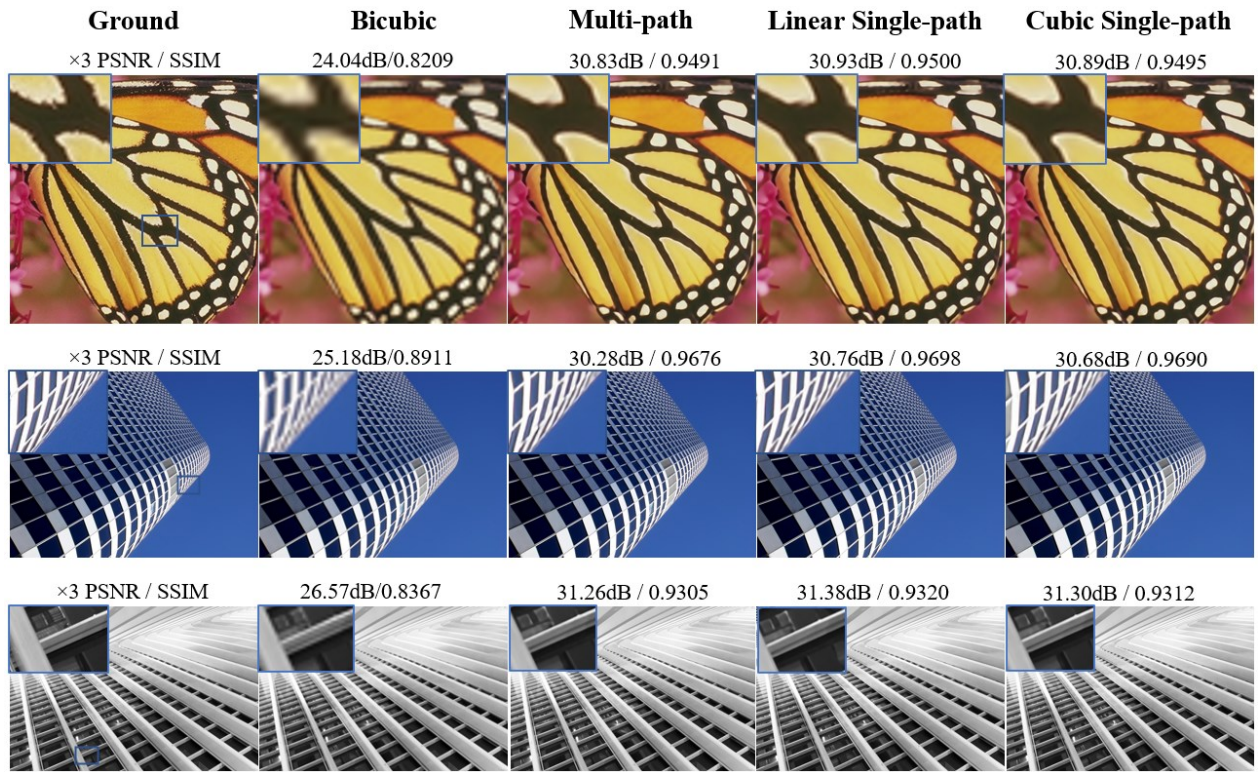
Through multi-scale multi-path SR analysis, we can identify the unshared and unbalanced parameter problems, formulate a solution by utilizing the advantages of interpolation algorithms, and exploiting sub-pixel convolution to its limit.

We can conclude that the linear single-path technique is a more practical solution compared to the multi-path algorithm because it reduces and exploits all its parameters for all scale factors. It also shows similar performance with less computation compared to the cubic single-path algorithm. The proposed technique can be applied to existing multi-scale multi-path SR models, such as MDSR [6] CARN [9], and MPRNet [10] even if they utilize group convolution.

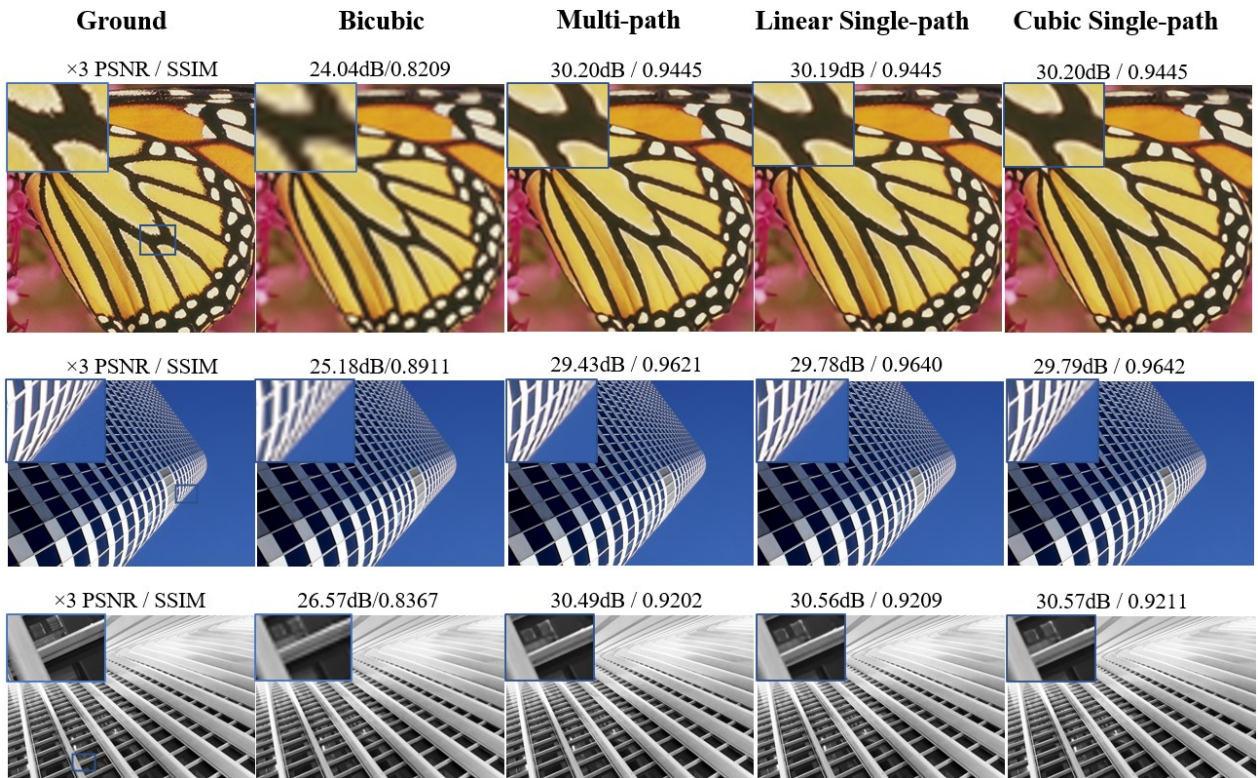
Inspired by the results achieved from CARN-M [9], further research can be done by analyzing the group convolution in sub-pixel convolution to improve efficiency by reducing parameters and computation while maintaining good performance.

### Acknowledgment

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).



(a) MDSR [6] model



(b) CARN-M [9] model

Fig. 2. Visual comparison with scale factor  $\times 3$  between the ground-truth, bicubic, and images restored by multi-path, and single-path algorithms implemented with the MDSR [6] model (a), and (b) CARN-M [9] model (b) respectively. Testing with the butterfly image from the Set 5 [20] dataset and the 2<sup>nd</sup>, 5<sup>th</sup> and, 42<sup>nd</sup> image from the Urban 100 dataset [23].

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, Xiaohai Zhuang, Wenjia Bai, Kanwal Bhatia, Antonio M Simoes Monteiro de Marvaio, Tim Dawes, Declan O'Regan, and Daniel Rueckert, "Cardiac image super-resolution with global correspondence using multi-atlas patch match," 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 super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 2, pp. 295-307, 2015.
- [4] 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.
- [5] W. Shi, J. Caballero, F. Huszar, 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.
- [6] 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.
- [7] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. "Image super-resolution using very deep residual channel attention networks", in *ECCV (7), volume 11211 of Lecture Notes in Computer Science*, pp. 294– 310, 2018.
- [8] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, Jixiang Li, and Qingyuan Li, "Fast, accurate and lightweight super resolution with neural architecture search," *arXiv preprint arXiv:1901.07261*, 2019.
- [9] 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.
- [10] 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.
- [11] JH Kim, BG Kim, PP Roy, and DM Jeong, "Efficient facial expression recognition algorithm based on hierarchical deep neural network structure," in *Proceeding of IEEE access* 7, 41273-41285, 2019.
- [12] 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.
- [13] 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.
- [14] Honnang Alao, Jin-Sung Kim, Tae Sung Kim, and Kyujoong Lee. "Efficient multi-scalable network for single-image super-resolution," in *Journal of Multimedia Information System*, Volume, No. 2, pp. 1-10, 2021.
- [15] 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.
- [16] C. Ledig, L. Theis, F. Huszar, 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.
- [17] A Mittal, P. P. Roy, P. Singh, and B. Raman. "Rotation and script independent text detection from video frames using sub pixel mapping," in *Journal of Visual Communication and Image Representation*, Volume No. 46, pp. 187-19, 2017.
- [18] Young-Hyun Jung, Jong-Ho Yun, and Myung-Ryul Choi. "Modified Cubic Convolution Interpolation for Low Computational Complexity," in *the Korean Information Display Society*, pp. 1259-1261, 2006.
- [19] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. "NTIRE 2017 challenge on single image super-resolution: Methods and results", in *CVPR Workshops*, pp. 1110-1121. IEEE Computer Society, 2017.
- [20] 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.

- [21] 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.
- [22] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Transactions on Image Processing (TIP)*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [23] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, 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.
- [24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR, abs/1412.6980*, pp. 1-15, 2014.
- [25] 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.
- [26] HolmesShuan.EDSR-ssim.github:  
<https://github.com/HolmesShuan/EDSR-ssim>, 2018.
- [27] Sewar python library for image quality metrics:  
<https://sewar.readthedocs.io/en/latest/>

#### Authors



**Honnang Alao** received B. S. degree in Electronic Engineering, at Sunmoon University, in 2020.

Currently pursuing a Master’s degree in Computer and Electronic Engineering, at Sunmoon University, since 2021.

His research interests include 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.

His research interests include 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.



**Juhyen Oh** Currently pursuing his Bachelor’s degree in Theology, and also pursuing an Electronic Engineering degree as his second major, at Sunmoon University, since 2015.

His research interests include deep learning, image processing, and multimedia.



**Kyujoong Lee** received Bachelor’s degree in Electronic Engineering, at Seoul National University, in 2002 and a master’s degree in Electronic Engineering, at the University of Southern California, in 2008. In 2013 he received his Doctorate’s degree in Electronic Engineering, 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.

His research interests include deep learning, image processing, image compression Multimedia, and SOC design.