BIBNet: An Efficient Super Resolution with Bottleneck-In-Bottleneck

Simyung Chang*, Keuntek Lee†, Shobhit Jain‡, Cheul-hee Hahm¶
Visual Display Division
Samsung Electronics
Suwon, Korea
Email: *sm5.chang@samsung.com, †keuntek.lee@samsung.com, ‡shob.jain@samsung.com,
¶chhahm@samsung.com

***Abstract*—Deep Neural Networks have enabled remarkable progress in the field of single image super resolution (SR). However, these models are often large and complex to be applied for real-world applications with limited resources as in mobile and embedded systems. We investigate whether the typical low latency models as MobileNet can be expected of comparable efficiency at SR tasks with recently reported SR performance in the literature. To this end, a moderate and effective architecture, Bottleneck-In-Bottleneck (BIB), is introduced in this paper. The BIB uses multiple expansion factors of the residual blocks in the form of a bottleneck, reducing computation complexity while utilizing advantageous factors of large feature dimensions. We also propose BIBNet with multiple BIB blocks, which can easily adjust its size and computational cost to create a variety of efficient and high-performance models. Extensive experiments show that, with fewer parameters and computations, BIBNet achieves highly competitive performance compared to other conventional SR methods with more complex architectures.**

***Index Terms*—super-resolution, on-device**

I. Introduction

Single image super resolution (SISR) is a computer vision task that aims at reconstructing higher resolution images from lower resolution ones. In recent years, SISR has become an active area of research because of outstanding performance improvements achieved with convolutional neural network (CNN) techniques. These CNN-based methods have shown much superior results over traditional methods [1]–[8].

Super-Resolution Convolutional Neural Network (SRCNN) [9] is the first pioneering work that demonstrates the exceptional superiority of deep-learning-based methods to solve ill-posed problems of SR. After SRCNN, there has been a significantly active studies of applying deep learning to SR related tasks. Models including VDSR [10], EDSR [11], RCAN [12], LapSRN [13], DBPN [14] and MemNet [15] have consistently obtained state-of-the-art results, pushing boundaries of the research literature. However, the models are rarely useful for real world applications because of costly memory requirements or high computational loads. For real world scenarios, it is extremely important to build lightweight deep learning models while maintaining accurate results. Though methods such as DRCN [16], DRRN [17] are introduced to reduce the number of parameters tremendously, but their methods come with the cost of increased number of operations at inference time.

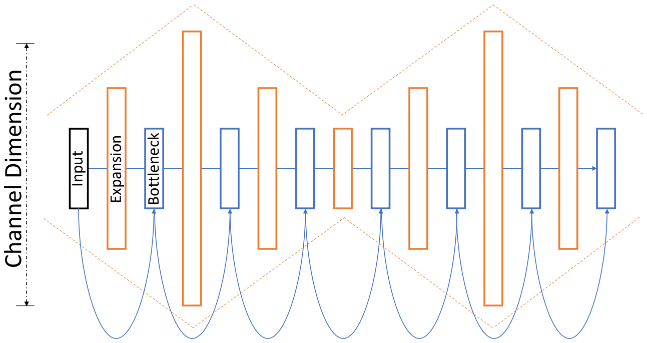


Fig. 1: **The concept of Bottleneck-In-Bottleneck (BIB)** Expansion layers of each residual block consist of a bottleneck. BIB can handle high dimensional features effectively while reducing computational cost.

There have been attempts to address this problem by proposing lightweight models for efficient SR execution such as FSRCNN [18], CARN [19], or by finding efficient structure through NASNet [20]. The methods have been able to solve the problem to an extent, but the literature still needs improvements in handling higher resolution images, such as 4K, on devices with limited resources.

Since SR is an ill-posed inverse many-to-one mapping problem, problems of SR domain should be managed distinctively. Model compression approach for SR tasks should therefore differ from typical lightweight model architectures (e.g. MobileNet [21]) used in classification or object detection problems although their compression performance is still effective. In this paper, we aim to efficiently apply these conventional compressed models specifically to the SR problem. Particularly, we propose a new lightweight architecture, Bottleneck-in-Bottleneck(BIB), that utilizes some of the compression techniques from previously reported methods along with our unique contributions.

A BIB module is composed of a bottleneck structure that uses only 3x3 convolution operation instead of 1x1 and an outer bottleneck, using variable expansion factor of inner bottleneck. Fig 1 illustrates this concept. With BIB, we can take advantage of using large feature maps while reducing

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

computation. In our experiments, despite our simple structure and much fewer operations, our model outperforms not only the typical lightweight models but also more complex models found with Network Architecture Search (NAS). Additionally, with BIB, we are able to control the depth and width of the network and obtain networks of varying sizes depending on the computational constraints.

In summary, our contributions are as follows:
(1) We propose a novel lightweight architecture, Bottleneck-In-Bottleneck(BIB).
(2) We introduce BIBNet, a simple and efficient super resolution model. Since BIBNet has simple sequential architecture, total parameter size and computation cost of model can be adjusted easily.
(3) The experimental results show that our BIBNet generates high-quality output with much fewer parameters and operations than other complicated SR models.

II. Related Work

Deep Learning Based Super-Resolution. Recently, CNN-based SISR techniques have achieved great success and have become an active research area. SRCNN [9] proposes a deep learning-based approach for SISR, which was also later adapted for face hallucination [22], depth map super-resolution [23]. SRCNN is a lightweight model consisting of three layers: patch extraction/feature representation, non-linear mapping, and high-resolution image reconstruction. Their work has produced state-of-the-art results by demonstrating the first deep learning model for super-resolution. SCN [24] builds upon the idea of SRCNN and exploits the domain knowledge of conventional sparse-coding-based method to replace the non-linear mapping layer with a set of sparse coding sub-networks. VDSR [10] significantly improves the SR performance over the SRCNN model, using the power of deeper models by increasing the network depth from 3 to 20 convolutional layers. The network is trained with the *residual* image,i.e., the difference in ground truth high-resolution images and bicubic upsampled low-resolution images, to enable stably converging of the model. DRCN [16] proposes a memory-efficient model that uses stacked recursive layers, widening their receptive fields without increasing the model capacity. DRRN [17] extends the idea of DRCN to the utilization of recursive blocks with multiple residual units. The method uses global residual learning similar to VDSR, along with the local residual learning approach as used in ResNet [25]. The local residual learning paths facilitate back-propagation and thus prevent vanishing/exploding gradients.

All models mentioned above use bicubic interpolations to upsample the low-resolution images before feeding them into the networks, which increases the computational cost drastically (quadratic to the order of scale factor) and also accompanies with large memory requirements.

As a later work, LapSRN [13] uses a Laplacian pyramid structure to reconstruct progressively high-resolution residual images at multiple pyramid levels. Unlike other models previously mentioned, the method directly extracts features from low-resolution images, thus reducing the computational load. To reduce the number of network parameters, LapSRN uses recursive layers similar to DRCN/DRRN. One major shortcoming of these recursion based approaches is their heavy computational requirements. Though they are effective in reducing the number of parameters, they require noticeably deep networks to compensate for the loss in performance. For example, DRCN uses up to 16 recursions, whereas DRRN uses 52 recursions. This increases the number of operations, which is a crucial factor to be considered for models running on edge devices.

A different set of models aim at achieving real-time speed for SISR. The FSRCNN [18] method formulate an hourglass-shaped CNN structure with transposed convolution layers at the end, which operates directly at the low-resolution images for fast-image SR. The FSRCNN is a lightweight and shallow model with the number of parameters of the order of 12k. The The ESPCN [26] model uses a sub-pixel convolution for upsampling and attach it to end of the network. By this method, it reduces the number of operations and also increases the receptive fileds of each layer.

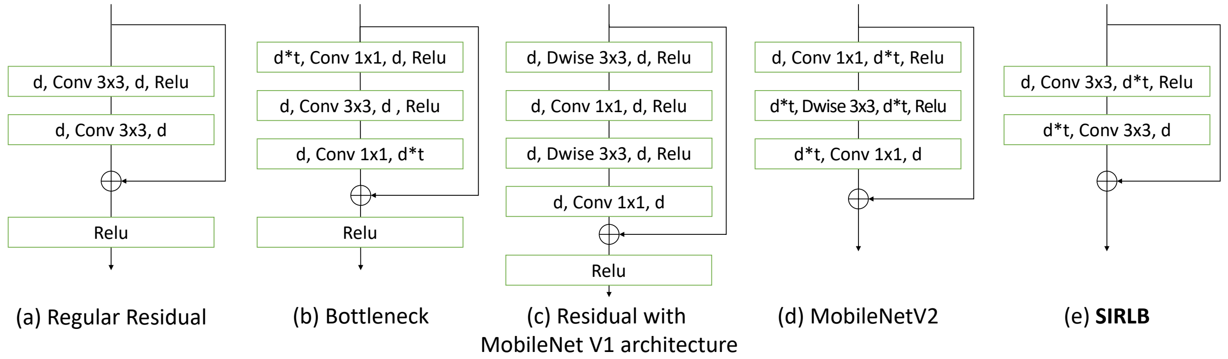
Built upon residual blocks, CARN [19] uses multiple cascading modules to incorporate features from multiple layers. Similar to DRRN, they add multiple cascading connections between intermediary layers on both local and global levels for efficient flows of information and gradients. However, it still requires a lot of memory due to multiple cascading connections.

Lightweight CNNs Architecture. Deep neural networks in computer vision tasks have been successful with Convolutional Neural Network(CNN) architecture. CNN models have allowed superhuman accuracy on image recognition tasks. However, the needs of high performance convolutional neural networks on embedded devices are increasing recently.

While ResNet [25] has been successful in image recognition tasks, its deep CNN architecture is not proposed as a lightweight network. However, its bottleneck structure proves its efficiency for its performance. Bottleneck consists of 1×1 pointwise convolutions and parameter-free identity shortcuts (skip connection), having an advantage on computational cost. As ResNet's performance is verified on image classification and object detection tasks, many CNN models use its bottleneck architecture as a baseline.

MobileNet [21] uses depthwise separable convolution, which factorizes a common convolution to build blocks. Depthwise separable convolution consists of a depthwise convolution and a 1×1 pointwise convolution, effectively reducing cost and amount of parameters. ShuffleNet [27] proposes pointwise group convolutions and channel shuffle operations to increase the information changes within multiple groups. Further, these methods improve network efficiency with the reduction of computational cost.

MobileNetV2 [28] proposes an inverted residual block for the mobile-purposed lightweight network. Inverted residual block reduces the dimensionality of the activation space by using 1×1 pointwise convolution and uses shortcuts directly



(a) Regular Residual (b) Bottleneck (c) Residual with MobileNet V1 architecture (d) MobileNetV2 (e) **SIRLB**

Fig. 2: Various structure of residual blocks. SIRLB uses a linear bottleneck consisting of 3x3 convolutions. The four columns of convolution layer represent input dimension, convolution type, output dimension, and activation function, respectively. t indicates expansion factor.

between the bottlenecks.

All these networks [21], [27], [28] focus on designing lightweight CNN architecture for a real-world scenario. However, the main purpose of these networks is not super-resolution since super-resolution is essentially an image reconstruction problem that is inherently different from classification/detection vision tasks. In this paper, we propose a lightweight network specifically for super-resolution, which consists of Bottleneck in Bottleneck(BIB) architecture.

III. BIBNet

Our method aims to create a simple SR model that has fewer parameters and operations with minimal operators so that it can be easily ported and operated on-device, such as a mobile system. In this section, we first describe the *Simple Inverted Residual and Linear Bottleneck* (SIRLB) and *Bottleneck-In-Bottleneck* (BIB) structure. Then we introduce BIBNet, an efficient and straightforward SR model based on BIB.

A. Simple Inverted Residual and Linear Bottleneck

Fig 2 shows various types of residual blocks with some typical lightweight architecture. SIRLB is a component of BIB and consists of a standard 3x3 convolution using a linear bottleneck in an inverted residual. This structure is designed based on the following intuition.

Bottleneck with 3x3 Convolution. A Bottleneck commonly refers to layers with fewer channels than previous layers. 1x1 convolution, called pointwise convolution, is used to change the dimension of the channel to form a bottleneck. Our intuition is that this pointwise convolution may not be appropriate even in the SR domain. The bottleneck structure using 1x1 convolution was mainly verified in fields such as classification and object detection. It handles input from image space and produces abstracted results such as class or object location. In particular, in classification, only the relationship between channels is finally considered through average pooling before the last fully connected layer. However, super-resolution (SR) accepts image space input and output, even with a higher dimension of output than input. It can also be viewed as a generation problem in terms of restoring lost information. With this reason, SIRLB has a bottleneck using only 3x3 convolution. It allows networks to handle high-dimensional features while reducing computations compared to not using bottlenecks.

Inverted Residual. Inverted Residual is proposed in [28]. The bottleneck layer also contains all the necessary information, so they apply the shortcut to the bottleneck. It reduces the amount of element-wise sum for shortcut connection, and increase the memory efficiency by reducing the size of the feature map stored for skip connection. For BIBNet, we can expect additional advantages. Even though the expansion factor is different for each layer, the shortcut size is the same. So the shortcut connection can be performed without an additional 1x1 convolution. And expansion is done at low resolution while keeping small bottleneck in BIBNet. It can greatly reduce the operations at high resolution due to fewer filters of convolution.

Linear Bottleneck. Sandler *et al.* proposed a linear bottleneck in their paper [28]. Assuming the manifold of interest is low-dimensional, a linear bottleneck layer can be inserted into the convolution block to capture it. They show that it is essential to use a linear layer because non-linearity can cause too much information loss. This assumption can be also valid for SR models using bottlenecks, we adapt linear transformation also. The usefulness of removing this non-linearity has also been reported in [29], and we can see this in our session IV experiment.

For input of size $h \times w$, input channel d, output channel d' with kernel size k and expansion factor t, SIRLB block requires total number of multiply adds operation:

$$h \times w \times k^2 \times d \times t(d + d'), \tag{1}$$

which includes two 3x3 convolution.

B. Bottleneck-In-Bottleneck

In general, Deep Neural Networks have the advantage of having large dimension feature maps, and for this purpose,

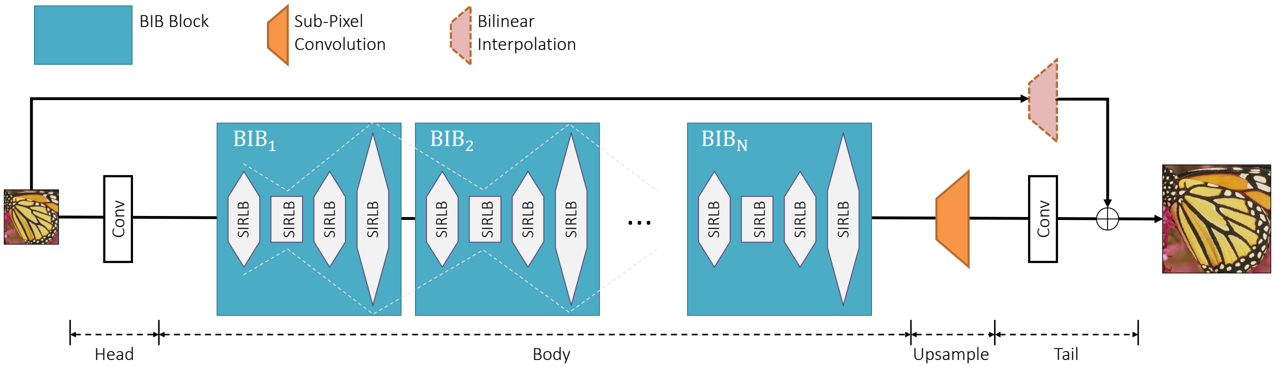


Fig. 3: Network Architecture of BIBNet. BIBNet-N-K is composed of N BIB blocks with K size of bottleneck.

TABLE I: **Comparison of BIB block with other residual methods** We change the body of BIBNet to the same number of other residual blocks. The base channel size K is adjusted to fit parameters and operations as closely as possible.

Model	Params	MultAdds	Urban100 PSNR/SSIM
BIBNet-1-16 (ours)	53K	12.4G	**30.84/0.9140**
BIBNet-1-16 + non linear bottleneck	53K	12.4G	30.75/0.9129
BIBNet-1-16 + constant expansion $t = 2.3$	53K	12.4G	30.74/0.9126
BIBNet-1-14 $BIB_{b=3}$	55K	12.8G	30.79/0.9133
Regular Residual $K = 22$	55K	13.0G	30.71/0.9127
Bottleneck Residual $K = 9, t = 4$	56K	13.5G	30.46/0.9094
MobileNet V1 based $K = 33$	54K	13.0G	30.65/0.9120
MobileNet V2 based $K = 26, t = 4$	53K	12.5G	30.71/0.9128

nonlinear transformation is performed at a much larger dimension than the actual input. Does it always require a higher-dimensional space? Depending on the feature, it may sometimes be sufficient to have a nonlinearity in a lower-dimensional space. Based on this assumption, we propose a method of gradually increasing and decreasing the size of the channel. Our proposed BIB does not increase the dimension of the channel continually but keeps the size of the bottleneck, and gradually increases and decreases the size of the expansion layer. This constitutes an outer bottleneck, as shown in Fig 1, which allows nonlinear transformations in various dimensions while reducing computation.

We construct a BIB with four SIRLB blocks with different expansion factors. Suppose that a SIRLB block with weight ω, expansion factor t, and input feature map X is $f_\omega(X; t)$, BIB is defined as

$$BIB(X; b) = f_{\omega 4}(f_{\omega 3}(f_{\omega 2}(f_{\omega 1}(X; b); 1); b); 2b). \quad (2)$$

If we set b as 2, the expansion factors of SIRLB constituting the BIB block are 2, 1, 2, and 4, respectively, in order. Stacking multiple BIB blocks repeats the outer bottleneck with a minimum expansion factor of 1 to a maximum of 4.

C. BIBNet Architecture

BIBNet is a super-resolution model that uses our proposed BIB blocks. Fig 3 shows the structure of BIBNet. BIBNet is composed of Head, Body, Upsample, and Tail. The Head is the first layer to extract features from input images with 5x5 convolution and ReLU activation. The Body is the essential component of BIBNet and contains N BIB blocks of size K in the bottleneck. The Upsample layer upsamples the input feature to the desired size, and it uses the sub-pixel convolution proposed by ESPCN [30]. The Tail converts the high-resolution feature map to 3 channel image space through 3x3 convolution, and performs bilinear upsampling on the input image, then adds both the final result.

The width of the BIBNet depends on the dimension of the base feature, the channel size of the bottleneck. The number of BIB blocks determines the depth of BIBNet. Therefore, we can define BIBNet as *BIBNet-N-K* with the number of BIB blocks N and the base feature size K.

Previous research has shown that the proper balance of depth and width has a notable impact on performance [31]. In addition to performance, it has the following effect on BIBNet: Narrow and deep structures (small K and large N) can reduce the memory requirement, but make it hard to be processed parallelly. This means that even if the system has enough resources, it is difficult to utilize properly. On the other hand, wide and shallow structures have opposite characteristics. Therefore, it is important to select the appropriate model according to the system resources. The user can choose the proper BIBNet for the system with the parameters N and K. Experimental results of the session IV show that BIBNet is highly efficient at various depths and widths.

IV. Experiments

We have experimented on the multiple super-resolution datasets to verify the followings:
1. Is BIB more efficient than conventional lightweight CNN architectures?
2. The effect of depth and width on super-resolution task.
3. Can various sized models of BIBNet achieve competitive performance with fewer parameters and operations?

A. Datasets and Metric

Datasets. For training, we use 800 training images from DIV2K [32] dataset. DIV2K dataset contains high-quality

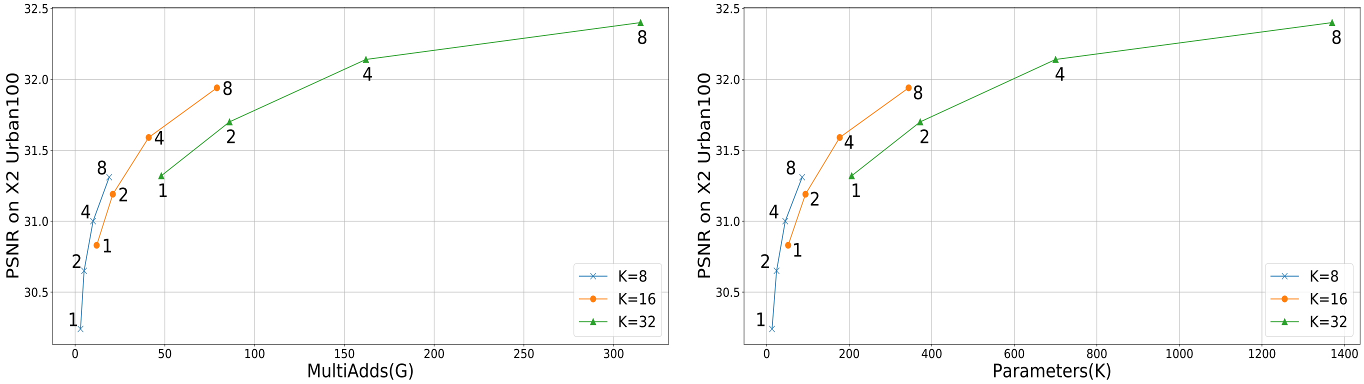


Fig. 4: **Depth vs Width on BIBNet** This result shows the performance of BIBNet for various parameters, N and K. If the model is shallow, N is 1 or 2, the performance is somewhat lower than that of deeper models with similar operations or parameters.

images. Because of the richness of this dataset, many SR models [11], [12], [14], [19] also use DIV2K recently. We use four standard benchmark datasets, Set5 [2], Set14 [3], B100 [33] and Urban100 [34] for testing.

Evaluation Metrics. We transform SR result images to YCbCr colorspace, and evaluate with PSNR and SSIM on Y channel.

B. Implementation Details

In training, we use randomly sampled patches of size 128×128 from low resolution images with RGB colorspace as input. For optimizer, we use ADAM optimizer [35] with parameter $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ in 6×10^5 steps. We also use minibatch with size 16, and set learning rate 5×10^{-4} on start and is halved every 1.5×10^5 steps. We use L1 loss in our experiments, since L1 loss shows better performance and faster conversion than L2 loss. We use Pytorch to implement our models with 2×Nvidia RTX 2080 TI GPUs.

C. BIB Block

We study the basic structure of our BIB(Bottleneck-In-Bottleneck) block. Recently, many Super-resolution(SR) models [10], [11] have used variation of residual block [25]. To evaluate the performance of our proposed BIB structure, we compare performance of blocks on the same model framework. Fig 3 shows network architecture of BIBNet. We fix the Head, Upsampling Layers, and Tail of the network, then switch the Body to 5 different types of the residual block. For a fair comparison, we keep the number of residual blocks same, and adjust only dimension to ensure all the networks have almost similar number parameters and Multi-Adds.

In Table I, our network with 1 BIB block and 16 base features (53K params, 12.5G Multi-Adds) outperforms other blocks with some margin. BIBNet-1-16 reaches the PSNR 30.84 dB on Urban100 with 2× scaling factors, which is relatively higher than the results of other block types. BIBNet-1-14 $BIB_{b=3}$ denotes BIBNet with small bottleneck and bigger expansions with base factor $b = 3$. This model also shows high performance, but slightly lower than the $b = 2$ model. This comparison shows the efficiency of our proposed BIB block as a simple architecture for Super-resolution.

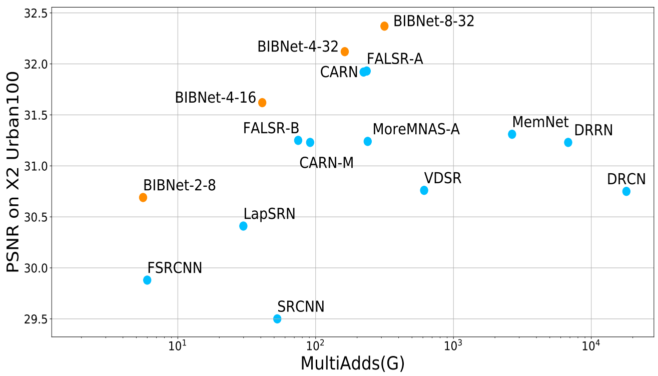


Fig. 5: BIBNet(orange) vs Others. MultiAdds is calculated based on the HR image with 1280x720 resolution.

D. Depth vs. Width

Given the structure of BIBNet, the network can be varied by using a different number of BIB blocks (depth) and feature size(width). Each model shows different performance by these factors. We experiment with a various number of BIB blocks and feature size to compare performance between deep BIB network and wide BIB network. In this case we need to build BIB networks with different N and K (where N denotes the total number of blocks and K denotes size of features). We train several network with different deepening factor $N\in$[2,4,8,16], and widening factor $K\in$[8,16,32]. Each network has one of deepening and widening factors; thus, there are $4\times3=12$ models to be evaluated.

We evaluate all networks with Urban 100 with 2× scaling factor and results are shown in Fig 4. Each line represents network group that have same feature size(K), but different number of BIB blocks(N). On increasing N, each of the

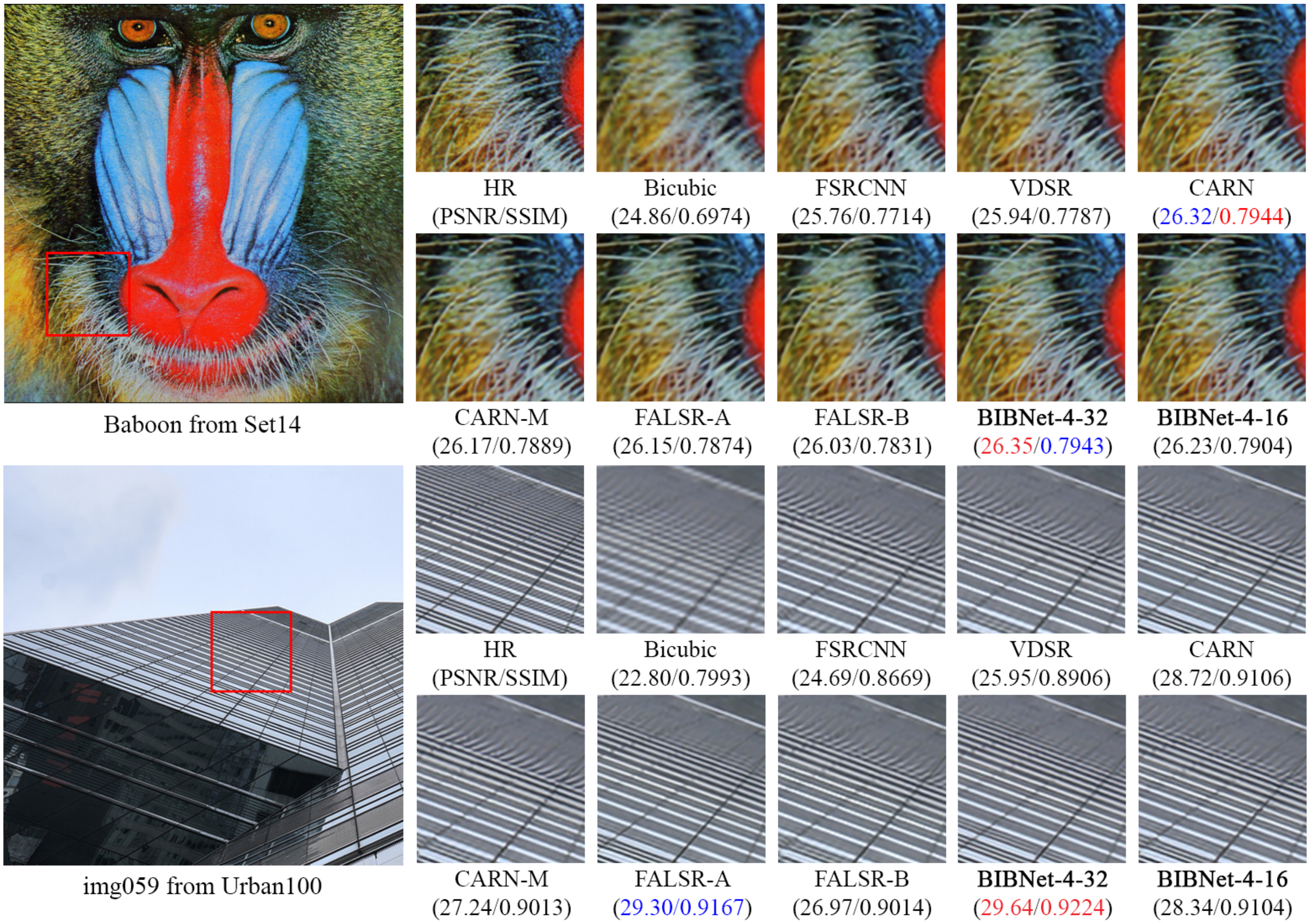
TABLE II: Quantitative Results of BIBNet and Other SR Methods. Red denotes best, and blue is second best.

Scale	Model	Params	MultAdds	Set5	Set14	B100	Urban100
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
2	SRCNN [9]	57K	52.7G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
	FSRCNN [18]	12K	6.0G	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020
	VDSR [10]	665K	612.6G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140
	DRCN [16]	1,774K	17,974.3G	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133
	LapSRN [13]	813K	29.9G	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100
	DRRN [17]	297K	6,796.9G	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188
	MemNet [15]	677K	2,662.4G	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195
	SelNet [36]	974K	225.7G	37.89/0.9598	33.61/0.9160	32.08/0.8984	-
	CARN [19]	1,592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
	CARN-M [19]	412K	91.2G	37.53/0.9583	33.26/0.9141	31.92/0.8960	31.23/0.9193
	MoreMNAS-A [37]	1,039K	238.6G	37.63/0.9584	33.23/0.9138	31.95/0.8961	31.24/0.9187
	FALSR-A [38]	1,021K	234.7G	37.82/0.9595	33.55/0.9168	32.12/0.8987	31.93/0.9256
	FALSR-B [38]	326k	74.7G	37.61/0.9585	33.29/0.9143	31.97/0.8967	31.28/0.9191
	BIBNet-8-32 (ours)	1,371K	315.6G	38.01/0.9597	33.76/0.9185	32.23/0.8998	32.37/0.9300
	BIBNet-4-32 (ours)	706K	162.7G	37.92/0.9595	33.59/0.9169	32.17/0.8990	32.12/0.9279
	BIBNet-4-16 (ours)	178K	41.0G	37.74/0.9588	33.41/0.9156	32.04/0.8975	31.62/0.9227
	BIBNet-2-8 (ours)	24K	5.6G	37.28/0.9573	32.97/0.9115	31.77/0.8939	30.69/0.9125
3	SRCNN [9]	57K	52.7G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989
	FSRCNN [18]	12K	5.0G	33.16/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080
	VDSR [10]	665K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
	DRCN [16]	1,774K	17,974.3G	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
	DRRN [17]	297K	6,796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
	MemNet [15]	677K	2,662.4G	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376
	SelNet [36]	1,159K	120.0G	34.27/0.9257	30.30/0.8399	28.97/0.8025	-
	CARN [19]	1,592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493
	CARN-M [19]	412K	46.1G	33.99/0.9236	30.08/0.8367	28.91/0.8000	27.55/0.8385
	BIBNet-8-32 (ours)	1,417K	145.4G	34.40/0.9265	30.35/0.8422	29.14/0.8057	28.36/0.8562
	BIBNet-4-32 (ours)	752K	77.5G	34.35/0.9260	30.29/0.8410	29.08/0.8040	28.15/0.8515
	BIBNet-4-16 (ours)	189K	19.6G	34.06/0.9239	30.15/0.8381	28.98/0.8017	27.79/0.8441
	BIBNet-2-8 (ours)	27K	2.9G	33.55/0.9196	29.81/0.8312	28.75/0.7952	27.09/0.8268
4	SRCNN [9]	57K	52.7G	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221
	FSRCNN [18]	12K	4.6G	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280
	VDSR [10]	665K	612.6G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
	DRCN [16]	1,774K	17,974.3G	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510
	LapSRN [13]	813K	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560
	DRRN [17]	297K	6,796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
	MemNet [15]	677K	2,662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630
	SelNet [36]	1,417K	83.1G	32.00/0.8931	28.49/0.7783	27.44/0.7325	-
	CARN [19]	1,592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
	CARN-M [19]	412K	32.5G	31.92/0.8903	28.42/0.7762	27.44/0.7304	25.62/0.7694
	BIBNet-8-32 (ours)	1,408K	88.0G	32.21/0.8947	28.66/0.7828	27.60/0.7366	26.20/0.7894
	BIBNet-4-32 (ours)	742K	49.8G	32.08/0.8931	28.58/0.7809	27.56/0.7349	26.07/0.7842
	BIBNet-4-16 (ours)	187K	12.7G	31.82/0.8899	28.43/0.7772	27.47/0.7318	25.78/0.7751
	BIBNet-2-8 (ours)	26K	2.1G	31.35/0.8829	28.11/0.7689	27.24/0.7239	25.18/0.7535

groups show an increase in performance, though the performance enhancement is not always the same. Note that there is difference on network architecture between BIBNet-4-16 and BIBNet-1-32. BIBNet-4-16 has deeper architecture than BIBNet-1-32. By contrast, BIBNet-1-32 has wider architecture than BIBNet-4-16. In spite of different network arhcitecture, BIBNet-1-32 and BIBNet-4-16 have similar computational cost(Multi-Adds). However, BIBNet-4-16 outperforms by a margin of 0.27 PSNR which is quite big. This result shows that increasing depth(N) improves performance more efficiently than increasing width(K) in BIBNet architecture.

E. Comparison with Other SR Methods

We compare the proposed BIBNet with other SR Methods. We use two metrics: PSNR and SSIM, which are the most commonly-used image quality metrics. Note that we use Multi-Adds to represent the number of multiply-accumulate operations based on 1280×720 input size. We choose four different BIBNet models: BIBNet-8-32, BIBNet-4-32, BIBNet-4-16, and BIBNet-2-8 to compare performance on various Multi-Adds range. We show the comparison in Fig 5, where we have compared with only those models that have less than 5M parameters. The PSNR is evaluated on Urban100 $\times 2$ dataset. We exclude heavy SR models, such as EDSR [11], RCAN [12], since BIBNet targets lightweight SR methods. Here we observe that BIBNet clearly outperforms other SR models that have similar Multi-Adds. For x2 super-resolution, BIBNet-4-32 outperforms the PSNR for CARN [19] by a margin of 0.2, in spite of having just 162.7G Multi-Adds which is less than 222.8G of CARN. BIBNet-2-8, which is the lightest model of ours, only have 5.6G of Multi-Adds. However, it shows comparable performance against computationally heavy models. For example, VDSR [10] has



Baboon from Set14

HR (PSNR/SSIM) | Bicubic (24.86/0.6974) | FSRCNN (25.76/0.7714) | VDSR (25.94/0.7787) | CARN (26.32/0.7944)

CARN-M (26.17/0.7889) | FALSR-A (26.15/0.7874) | FALSR-B (26.03/0.7831) | **BIBNet-4-32** (26.35/0.7943) | **BIBNet-4-16** (26.23/0.7904)

img059 from Urban100

HR (PSNR/SSIM) | Bicubic (22.80/0.7993) | FSRCNN (24.69/0.8669) | VDSR (25.95/0.8906) | CARN (28.72/0.9106)

CARN-M (27.24/0.9013) | FALSR-A (29.30/0.9167) | FALSR-B (26.97/0.9014) | **BIBNet-4-32** (29.64/0.9224) | **BIBNet-4-16** (28.34/0.9104)

Fig. 6: Visual comparison with other methods. It is compared on x2 dataset due to the FALSR has x2 model only.

612.6G Multi-Adds which are almost hundred times more Multi-Adds than BIBNet-2-8, but there is only margin of 0.07 PSNR between BIBNet-2-8 and VDSR.

Table II shows quantitative results. We compare our BIBNet models against other SR methods over the benchmark datasets. Our BIBNet shows comparable performance with lesser number of Multi-Adds. Notably, we would like to accentuate the comparison between BIBNet and FALSR [38]. MoreMNAS [37] and FALSR are Network Architecture Search(NAS) based networks. These models use the various size of convolution and complex skip connection between blocks and have shown efficient performance. In contrast to NAS based networks, BIBNet uses more simple and intuitive architecture but shows more efficient performance on SR. For example, BIBNet-4-32 model achieves relatively high PSNR than FALSR-A [38] with a small size of Multi-Adds and parameters on Urban 100 2×. Fig 6 shows the qualitative results of our method. The overall subjective quality of BIBNet models are better than those of other conventional ones. In particular, BIBNet4-3 model gives more distinguishable white lines compared with others for img059 from Urban 100.

V. Conclusion

In this paper, we propose a novel lightweight efficient model architecture for single image super-resolution(SR), Bottleneck-In-Bottleneck(BIB). Our method reduces the number of computations, while at the same time possessing the ability to handle computations in higher dimensional feature space. We also looked at which elements of the lightweight architecture are suitable for the super-resolution task. Through our technique, we can easily control the depth and width of the network with parameters N and K. Experimentally, we also examined the effects of depth and width differences in BIBNet. These results show that the proper balance of depth and width is also essential in the SR domain, as with the previous studies. BIBNet shows high performance with fewer parameters and operations than the complicated models found through Network Architecture Search (NAS). If we extend the search space of NAS to our proposed BIB structure, we expect to find a more efficient model.

References

[1] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., volume 1, pages I–I. IEEE, 2004.

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[3] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[4] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.

[5] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE international conference on computer vision*, pages 1920–1927, 2013.

[6] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian conference on computer vision*, pages 111–126. Springer, 2014.

[7] Samuel Schulter, Christian Leistner, and Horst Bischof. Fast and accurate image upscaling with super-resolution forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3799, 2015.

[8] Jordi Salvador and Eduardo Perez-Pellitero. Naive bayes super-resolution forest. In *Proceedings of the IEEE International conference on computer vision*, pages 325–333, 2015.

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[10] 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*, pages 1646–1654, 2016.

[11] 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 IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[12] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[13] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[14] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[15] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.

[17] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.

[18] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016.

[19] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–268, 2018.

[20] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[22] Shizhan Zhu, Sifei Liu, Chen Change Loy, and Xiaoou Tang. Deep cascaded bi-network for face hallucination. In *European conference on computer vision*, pages 614–630. Springer, 2016.

[23] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *European conference on computer vision*, pages 353–369. Springer, 2016.

[24] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE international conference on computer vision*, pages 370–378, 2015.

[25] K He, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition. computer vision and pattern recognition (cvpr). In *2016 IEEE Conference on*, volume 5, page 6, 2015.

[26] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[27] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[29] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5927–5935, 2017.

[30] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 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*, pages 1874–1883, 2016.

[31] F. Wang Z. Hu Z. Lu, H. Pu and L. Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, pages 6231–6239, 2017.

[32] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[33] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Iccv Vancouver:, 2001.

[34] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[36] Jae-Seok Choi and Munchurl Kim. A deep convolutional neural network with selection units for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 154–160, 2017.

[37] Xiangxiang Chu, Bo Zhang, Ruijun Xu, and Hailong Ma. Multi-objective reinforced evolution in mobile neural architecture search. *arXiv preprint arXiv:1901.01074*, 2019.

[38] 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.