IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 19, 2020, accepted March 18, 2020, date of publication March 24, 2020, date of current version April 8, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2983079*

# Lightweight Image Super-Resolution With Mobile Share-Source Network

**JUAN DU, WENLAN WEI, CIEN FAN, LIAN ZOU, JIAWEI SHEN, ZIYU ZHOU, AND ZEZONG CHEN**
School of Electronic Information, Wuhan University, Wuhan 430072, China

Corresponding author: Cien Fan (fce@whu.edu.cn)

**ABSTRACT** Within the development of the deep convolutional neural network, the great achievements had been made in the single-image super-resolution (SISR) task. However, the higher performance always comes with the deeper layers which also brings larger numbers of network operations and parameters that make it hard to implement in practice. In our work, a lightly super-resolution, named Mobile Share- Source Network (MSSN), is purposed to address these practical issues. In MSSN, a high-efficiency block, the mobile adaptive weighted residual unit, is designed to fulfill the need for the reduction in both parameters and the Mult-Adds while maintaining the performance with importing the deep separable convolution. Moreover, it brings into the Adaptive Weighted Share-Source Skip Connection, getting abundant information from the shallow layer which helps reconstruct better images. The experimental results show that our network has fewer numbers of parameters and operations than the state-of-the-art lightweight network while maintaining high reconstruction quality comparing with many state-of-the-art super-resolution methods in terms of both peak signal-to-noise ratio (PSNR) and structural similarity index metrics (SSIM).

**INDEX TERMS** Super-resolution, depthwise separable convolution, adaptive weighted module, lightweight network.

## I. INTRODUCTION

Single image super-resolution (SR) has recently received great attention. Super-resolution is a typical computer vision task, it reconstructs the low-resolution (LR) image into the high-resolution (HR) image generally. After the introduction of the CNN into the field, plenty of related methods had been proposed. From 2014, Dong et al. proposed convolutional neural networks based SRCNN [8] algorithm that outperformed other traditional works in SISR such as interpolated-based and model-based methods. By focusing on the wider and deeper architectures, the CNN-based networks obtained remarkable progress. Represented by EDSR [22], it improved the structure of the deep residual network [19], removed the batch normalization (BN) layer [18] and then deepened the network layers which had achieved excellent performance in image reconstruction, however, the drawbacks was the large amounts of parameters and high computational costs which fundamentally limited the application in practical scenarios. To address this problem, the lightweight SISR networks are proposed.

One simple strategy is to reduce the parameters such as constructing a shallow network model, like the ESPCN [27] and FSRCNN [9] algorithms. Another way is to share parameters through a recursive mechanism. DRRN [28] and DRCN [10] are two typical SISR networks with fewer network parameters but better performance. However, as mentioned above, the number of operations is also an important factor when operates on portable devices, the battery capacity is heavily dependent on the number of computation operations. From the perspective of user experience, the speed of the system is also crucial.

To address the question, we propose our lightweight model MSSN with three principal ingredients: a feature extraction module, a nonlinear mapping module, and an image reconstruction module. Our MSSN achieves the highly competitive performance for SR, with fewer network parameters at the same time, as shown in Figure 1. In addition, our MSSN also has advantages in speed, as shown in the figure 2.

The main contributions of this work are twofold:

1) The Adaptive Weighted Share-Source module is proposed, a module consists of six cascading blocks and attains more information by using the

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani.

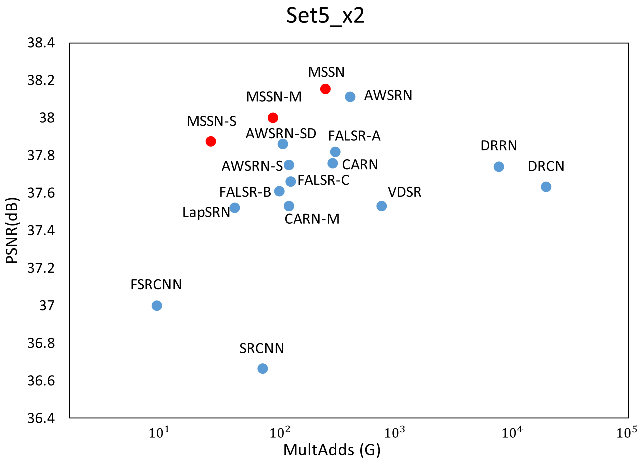


**FIGURE 1.** The performance of the MSSN family (red dots) is compared with that of other most advanced lightweight networks (blue dots) on the Set5 test set. Mult-Adds represents the number of operations.

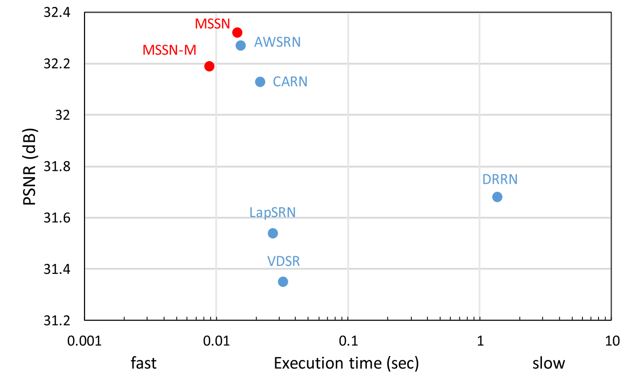


**FIGURE 2.** Speed and accuracy trade-off. The average PSNR and the average inference time for upscaling ×4 on Set5.The MSSN is faster than other methods and achieves the best performance at the same time.

share-source skip. This also eases our training for a deep learning network.

2) The newly designed cascading blocks is proposed, stacks four Mobile adaptive weighted residual units (MAWRU) could greatly reduce the network parameters and calculations while maintaining performance. The proposed network has at least 1/4 reduction in parameters and calculations while shows comparable performance against the state-of-art networks.

## II. RELATED WORK

### A. LIGHTWEIGHT SUPER-RESOLUTION NETWORKS

After VDSR [19] introduced the idea of the deep network and residual learning into SR, it is a trend to obtain better image reconstruction performance by deepening the network layers and learning residual mapping. MDSR [22] network used improved residual units to stack more than 160 layers. Similarly, both RCAN [35] and EDSR [22] had huge parameters. To implement the network in practical scenarios, lightweight networks become a heated focus.

Up to now, plentiful strategies have been adopted. Tai *et al.* [28] proposed DRRN using the residual network (ResNet) [12] which adopted a weight sharing strategy to reduce parameters. Although parameters were partly reduced, objectively, a large number of operations was still needed. Lai *et al.* proposed a Laplacian pyramid super-resolution network (LapSRN) [21] to solve the contradiction between speed and performance of the SR tasks. The network took the original LR image as input and reconstructs the HR image step by step. Ahn *et al.* proposed a cascading residual network (CARN) [2] to achieve lightweight and efficient reconstruction. CARN used group convolution to achieve the trade-off between computational complexity and model performance. Unfortunately, directly the appliance of the group convolution to the SISR did not turn out well. As the HUNG *et al.* came up with SSNet-M, he proposed an image super-resolution network with extremely small parameters and operations, with the recursive blocks being used flexibly [15], [16]. The Deep Differential Convolutional network has been proposed by Peng tried to modify the loss function [23]. Also, the addition of the Wavelets and color guidance block was impressive [1]. [4] used a two-stage fusion to improve processing speed. However, they all had a certain degree of performance deficiencies. Recently, Neural Architecture Search (NAS) had emerged as an emerging method for automatically designing efficient networks [38], which was subsequently introduced into SR tasks such as the FALSR algorithm [5] which only had fewer calculations. However, due to search space and policy constraints, the performance of NAS-based networks was also limited and it requires huge training costs.

Although the number of calculations and the performance could hardly be achieved at the same time, all of this works showed great possibilities that lightweight SR networks are able to achieve a good balance between the quality of the reconstruction and the number of network parameters.

### B. PARAMETER-EFFICIENT NEURAL NETWORK

According to the urgent needs from industry and academia, plenty of methods for compression and acceleration of convolutional neural networks (CNNs) have been proposed. So far, the methods of compression and acceleration for CNN are mainly the following two types:

1) compression pre-training network
2) lightweight network structure

In terms of compressing the pre-training model, Han *et al.* [11] proposed a deep compression technique that included pruning, weight quantization, and Huffman coding to reduce the size of a pre-trained network. The BinaryConnect [6] from Matthieu Courbariaux *et al.*, the INQ quantification method proposed by Intel China Research Institute [36], and the DoReFa-Net [37] quantification method proposed by Zhou *et al.* had different degrees of compression effects.

To obtain a lightweight network structure, a popular way is to design more efficient network calculation methods. In 2016, SqueezeNet [17] compressed and reduced the input feature map with a convolution kernel size of 1 × 1. ShuffleNet [34] achieved more efficient convolution

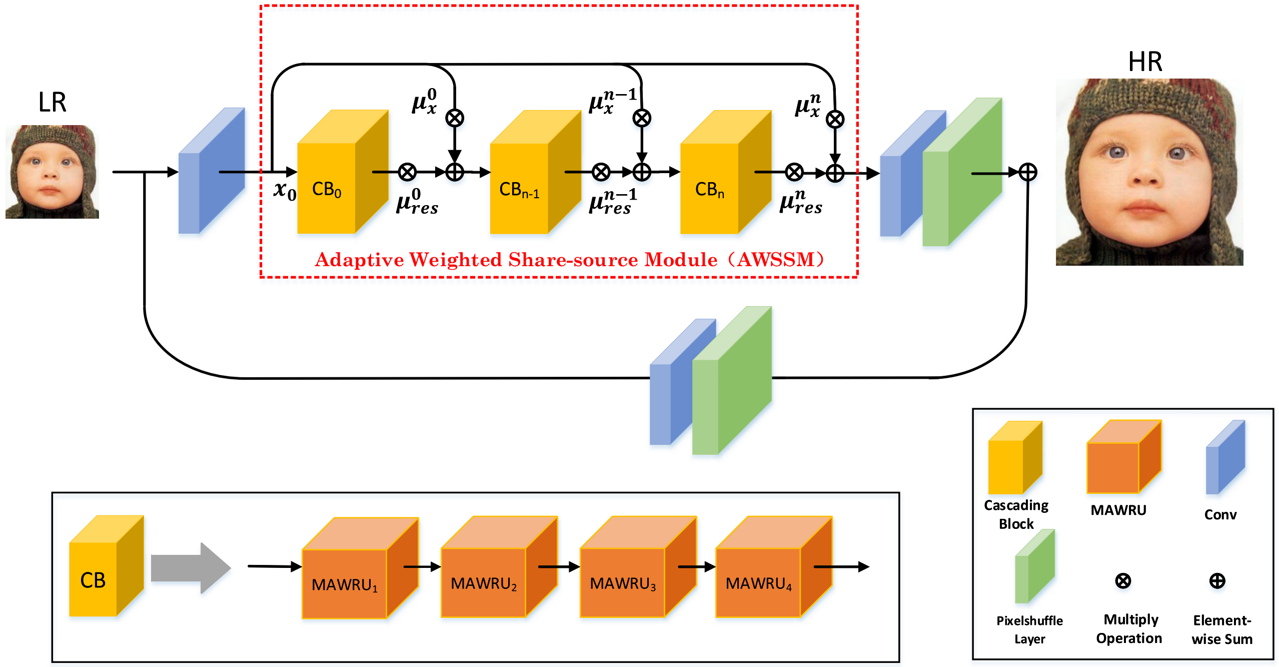


**FIGURE 3.** Network architecture of the proposed MSSN.

calculations by using group convolution and channel shuffling. In the same year, Google's MobileNet [13] designed a new type of deep separable convolution, which decomposed common convolution calculations into channel-by-channel depthwise convolution and pointwise convolution, which brought us inspiration, we apply the idea of depthwise convolution and pointwise convolution and up-dimensional dimensionality reduction for target detection model compression to the SR task.

## III. METHOD

### A. NETWORK FRAMEWORK

As shown in Figure 3, our proposed Mobile Share-Source Network consists of a feature extraction module, a nonlinear mapping module, and a reconstruction module. The input and output of the MSSN are $I_{LR}$ and $I_{SR}$ respectively. We only use convolutional layers with a kernel size of 3 × 3 to extract features $F_0$ from the LR input.

$$F_0 = f_e(I_{LR}) \tag{1}$$

$f_e$ is the feature extraction function of the $I_{LR}$ image, and then the extracted feature $F_0$ is used for the nonlinear mapping module based on the Adaptive Weighted Share-Source Module (AWSSM).

$$F_{NM} = f_{AWSSM}(F_0) \tag{2}$$

where $f_{AWSSM}$ represents a non-linear mapping module based on AWSSM, which includes n Cascading Blocks and n Adaptive Weighted Share-Source Skip Connections, with more details on AWSSM in the section III-B. We can control the increase in the number of CBs to achieve deep depth and thus provide very large receptive field size. More details about Cascading Block will be provided in the section III-C. The extracted feature $F_{NM}$ is then reconstructed by the reconstruction module.

$$I_{SR} = f_{up}(F_{NM}) + f_{up}(I_{LR}) \tag{3}$$

$f_{up}$ is the upsampling function. We extract all features in the low resolution stage and then upsample, unlikes some models which insert one or more convolution layers after upsampling.

### B. ADAPTIVE WEIGHTED SHARE-SOURCE MODULE (AWSSM)

Our Adaptive Weighted Share-Source Module (AWSSM), which contains $M$ cascading blocks with Adaptive Weighted Share-Source Skip Connection.

It has been verified that the accumulation of residual blocks helps to form a deep CNN. However, it can cause training difficulties and restricting performance [7] due to the disappearance and exploding of gradients flow in deep networks with several layers. Inspired by the work of [13], [22], we propose the Cascading Block (CB) as the basic unit. It is well known that simply stacks same CBs will not achieve better performance. In order to solve this problem, AWSSM introduces Adaptive Weighted Share-Source Skip Connection (AWSSC), which absorbs richer low-level information.

Define Cascading Block as the $f_{CB}$ function, which can be expressed as

$$x_n = \mu_{res}^n f_{CB}^n(\mu_{res}^{n-1} f_{CB}^{n-1}(\ldots \mu_{res}^0 f_{CB}^0(x_0) + \mu_x^0 x_0 \ldots) + \mu_x^{n-1} x_0) + \mu_x^n x_0) \tag{4}$$

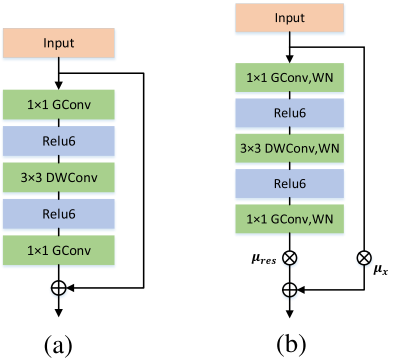


**FIGURE 4.** (a) mobilenetv2 without BN layer is basic RU (b) MAWRU with WN layer and adaptive weighted scale is proposed by us.

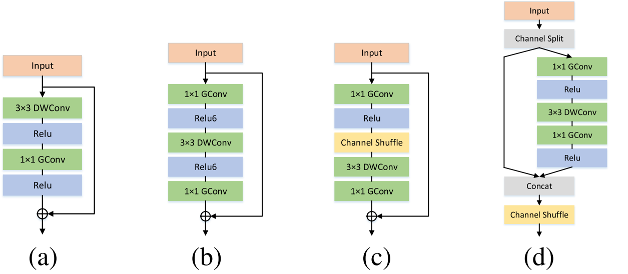


**FIGURE 5.** (a) Mobilenetv1 block: removes the BN layer from Mobilenetv1
(b) Mobilenetv2 block: removes the BN layer from Mobilenetv2
(c) Shufflenetv1 block: removes the BN layer from Shufflenetv1
(d) Shufflenetv2 block: removes the BN layer from Shufflenetv2.

where $x_n$ is the output of the nth Cascading Block, $\mu_{res}^n$ and $\mu_x^n$ are adaptive weighted scale. The scale of adaptive weights can help the flow of information and gradients to be more efficient [30], while the output includes features from multiple layers for more low-level information where global cascading connections can quickly propagate information among layers, greatly increasing efficiency.

### C. HIGH EFFICIENT CASCADING BLOCK

Considering the balance of speed and performance, we propose the Cascading Block and introduce the idea of depthwise convolution and pointwise convolution. The Mobilenetv2 [26], as shown in Figure 4 (a), was used as the basic residual unit (basic RU). This unit allows you to extract more information without adding parameters by narrowing the input dimensions and extending the internal dimensions before ReLU and finally reducing the output dimensions. Interior dimensions (channels) are controlled by expanding ratio, the details of which are discussed in section IV-C. Based on this, a Mobile Adaptive Weighted Residual Unit (MAWRU), based on basic RU, is proposed. As shown in Figure 4 (b), MAWRU differs from mobilenetv2 in which we remove the BN layer and add a WN layer [25] to each convolution. The benefits of removing the BN layer and adding the WN layer have been evidenced in EDSR and WDSR [32]. In addition to this, MAWRU also contains two independent weight scales, which can be adaptively learned after given initial values. A Cascading Block is a cascade of four MAWRUs.

Assuming that $x_{m-1}$ and $x_m$ are the input and output of MAWRU, then

$$x_m = \mu_{res}^m f_{pw}(f_{dw}(f_{pw}(x_{m-1}))) + \mu_x^m x_{m-1} \tag{5}$$

$f_{pw}$ is a pointwise convolution function, $f_{dw}$ is a depthwise convolution function, $\mu_{res}^m$ and $\mu_x^m$ are the weight values for two branches of the residual unit.

Due to the introduction of mobilenetv2, the efficiency of MAWRU is self-evident. After adding the WN layer and adaptive weight scale, it can improve the learning efficiency of training, speed up the convergence, and improve the accuracies of performance.

## IV. EXPERIMENTS

### A. DATASETS

DIV2K [29], is implemented to train the MSSN: The DIV2K dataset consists of 800 training images, 100 validation images, and 100 test images. We use standard benchmark data sets such as Set5 [3], Set14 [33], B100 [24] and Urban100 [14] for testing and benchmarking. The SSIM is aim to evaluates the human view and luminance is being regarded as the most important part in human vision leading the mainstream to calculate the SSIM under the YCbCr space [31]. For a fair comparison, we test PSNR and SSIM on the Y channel of the images that are transformed into the YCbCr space.

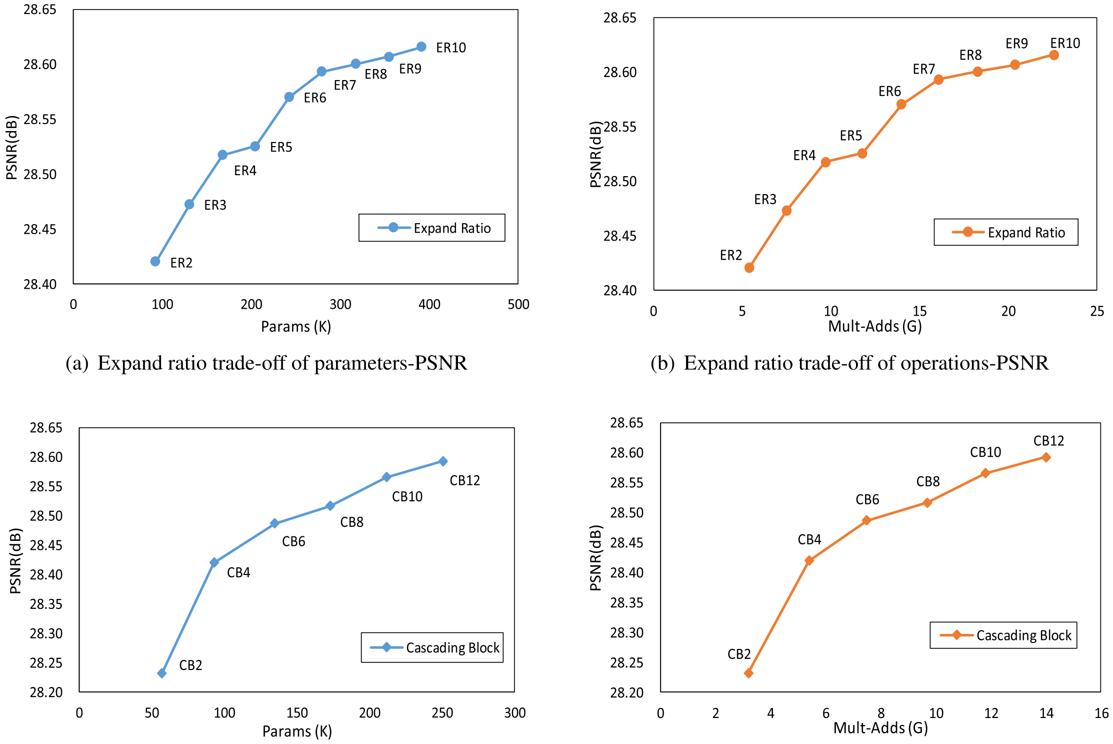
### B. IMPLEMENTATION AND TRAINING DETAILS

For training, we use the RGB input patches of size 48 × 48 from LR image with the corresponding HR patches. We train our model with ADAM optimizer [20] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We set minibatch size as 16. The learning rate is initialized as $10^{-3}$ and is halved at every $2 \times 10^5$ minibatch update. For the hyperparameter, $M$ is set to 6, expand ratio is set to 4, the value of the input and output channels are all set to 64, and the initial value of all adaptive weights is 1.

We train our network with L1 losses instead of L2. Minimizing L2 is usually preferred because it maximizes PSNR. However, the L1 loss has a better convergence than L2 [22]. We implemented our model using PyTorch framework with an NVIDIA 1080Ti GPU.The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

### C. MODEL ANALYSIS

In order to further study the performance of these methods, our model was analyzed with various experiments. How deeply separable convolution and parameter quantization affect MSSN performance was shown and also the examination on the trade-off between performance and parameters and operations was taken.



(a) Expand ratio trade-off of parameters-PSNR

(b) Expand ratio trade-off of operations-PSNR

(c) CBs trade-off of parameters-PSNR

(d) CBs trade-off of operations-PSNR

**FIGURE 6. Results of the trade-off of PSNR vs parameters(left), and PSNR vs operations(right) in relation to the number of Cascading Block and expand ratio. We evaluate all models on Set14 with 4 scale. ER represents expand ratio of mobile adaptive weighted residual unit and CB means the number of Cascading Block.**

**TABLE 1. Performance comparison of different blocks.**

| | M_v1 | S_v1 | S_v2 | M_v2_32 | M_v2_64 |
|---|---|---|---|---|---|
| *Params* | 108K | 46K | 71K | 93K | 314K |
| *MultAdds* | 6.2G | 2.7G | 4.1G | 5.4G | 18.1G |
| *PSNR* | 28.39 | 28.04 | 28.32 | 28.42 | 28.57 |

### 1) MOBILE ADAPTIVE WEIGHTED RESIDUAL UNIT

To demonstrate the effectiveness of MAWRN, two variants were conducted. Mobilenetv1 block, Mobilenetv2 block, Shufflenetv1 block and Shufflenetv2 block as shown in (a), (b), (c) and (d) in Figure 5 are used to replace the mobile adaptive weighted residual unit in the CB for comparison. The Adaptive Weighted Share-Source Skip Connection is also removed to guarantee the uniqueness of the test. The expand ration of the mobilenetv2 is set to a factor of 2. To verify the effectiveness of different modules, we compare the MSSN and its variants on the Set14 with the scale factor of 4.

From Table 1, mobilenetv2 with 32 input and output channels reaches PSNR = 31.88 dB on Set5 with ×4 scale which has a better performance in the case of equivalent parameters and operands. The main reason lies in that pointwise convolution has more channels to extract more features at the beginning of the unit and reduce the dimension of convolution layers in feature extraction module.

**TABLE 2. Comparison of whether Adaptive Weighted Share-Source Skip Connection is added to the original network.**

| | WDSR-A | WDSR-A + AWSSC |
|---|---|---|
| Params diff | 0 | +8 |
| PSNR | 32.29 | 32.33 |

| | WDSR-B | WDSR-B + AWSSC |
|---|---|---|
| Params diff | 0 | +8 |
| PSNR | 32.08 | 32.18 |

**TABLE 3. Effects of the MAWRN and AWSSC measured on the Set5 ×4 dataset. MSSN-NA represents MSSN without AWSSC and MSSN-NM without MAWRN. MSSN is our final model.**

| | Baseline | MSSN-NA | MSSN-NM | MSSN |
|---|---|---|---|---|
| MAWRN | × | √ | × | √ |
| AWSSC | × | × | √ | √ |
| PSNR | 32.07 | 32.15 | 32.14 | 32.19 |

**TABLE 4. The proportion of reduction in the number of parameters and operations of MSSN compared with AWSRN.**

| Scale | Params | Mult-Adds |
|---|---|---|
| ×2 | -38.1% | -38.8% |
| ×3 | -40.1% | -41.6% |
| ×4 | -44.0% | -44.7% |

### 2) THE IMPORTANCE OF AWSSM

In order to prove the high efficiency of AWSSM, the Adaptive Weighted Share-Source Skip Connection. It was added alone

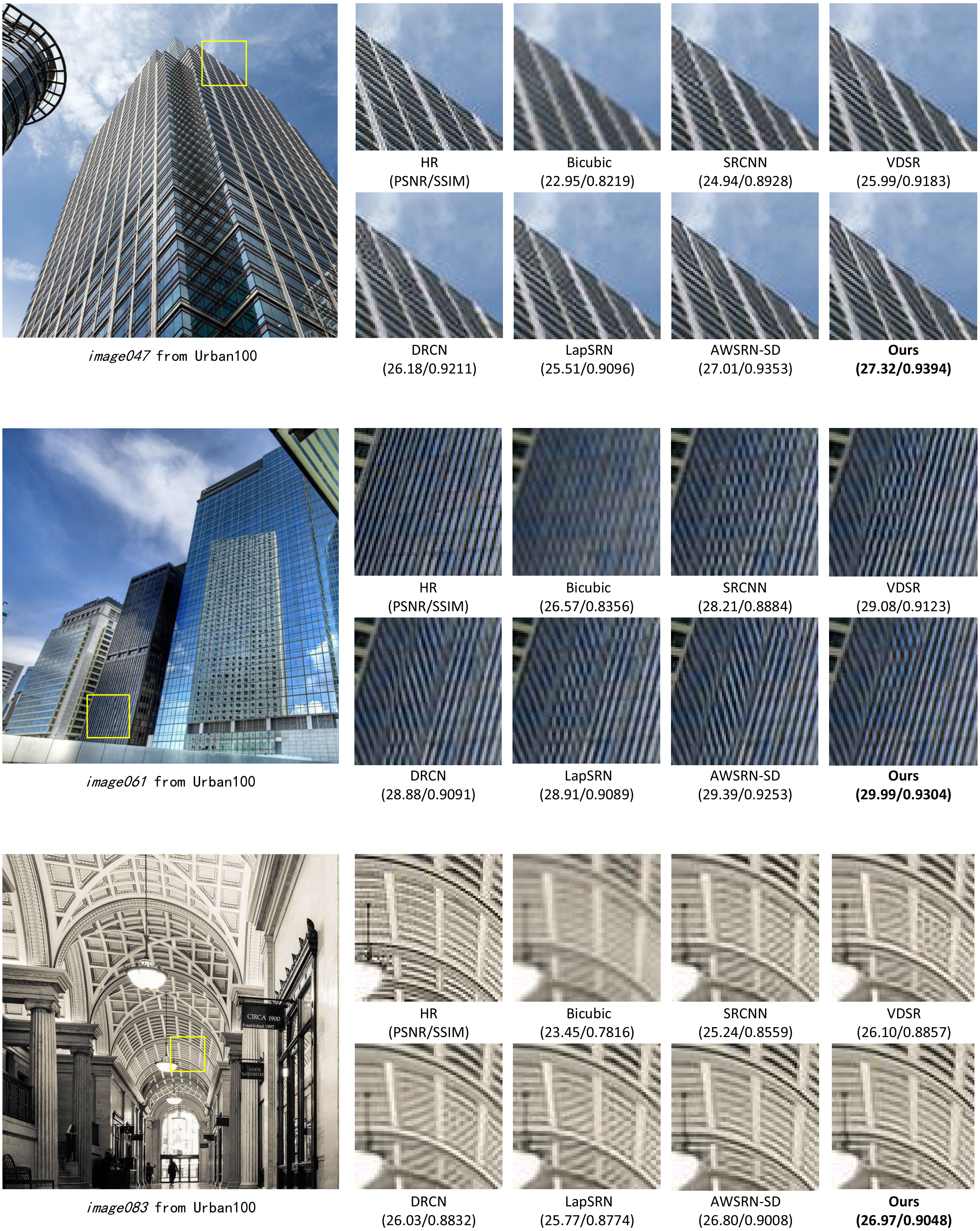


**FIGURE 7. Visual qualitative comparison on ×2 scale datasets.**

on the original network, WDSR-A and WDSR-B with n_feats is 32 to calculate the difference in parameters and PSNR.

From Table 2, we could see that the structure with Adaptive Weighted Share-Source Skip Connection performs better in PSNR. Specifically, the PSNR of WDSR-A is improved by 0.04 dB and 0.1 dB for WDSR-B. This is because the $1 \times 1$ convolution in MAWRN hinders the bypass of information, resulting in the high-complexity from the optimization

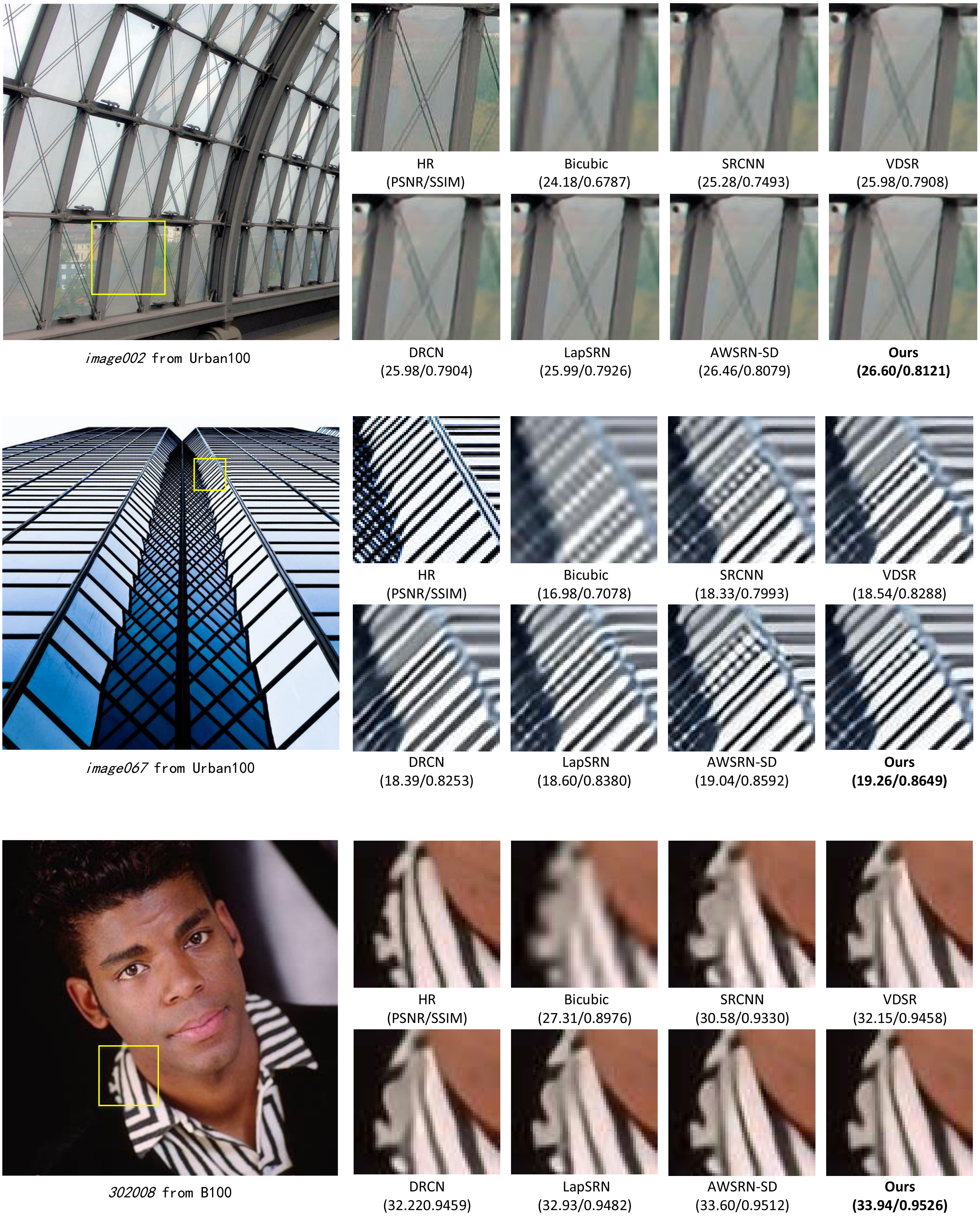


**FIGURE 8. Visual qualitative comparison on ×4 scale datasets.**

process. AWSSC, with global cascading, can bypass abundant information from the lower layer while extract features from multiple layers enable the network to improve efficiency notably.

### 3) THE IMPORTANCE OF MAWRN AND AWSSC

To evaluate the performance of the MAWRN and AWSSC components, a baseline consisting of 16 Mobilenetv2 blocks was built. Then, we constructed an MSSN-NA model only

**TABLE 5.** Quantitative results of deep learning-based SR algorithms. Red/blue text: best/second-best.Bold indicates the results of our model.

| Scale | Model | Params | MultAdds | Set5<br>PSNR/SSIM | Set14<br>PSNR/SSIM | B100<br>PSNR/SSIM | Urban100<br>PSNR/SSIM |
|---|---|---|---|---|---|---|---|
| 2 | SRCNN | 57K | 52.7G | 36.66/0.9542 | 32.42/0.9063 | 31.36/0.8879 | 29.50/0.8946 |
| | FSRCNN | 12K | 6.0G | 37.00/0.9558 | 32.63/0.9088 | 31.53/0.8920 | 29.88/0.9020 |
| | VDSR | 665K | 612.6G | 37.53/0.9587 | 33.03/0.9124 | 31.90/0.8960 | 30.76/0.9140 |
| | DRCN | 1,774K | 17,974G | 37.63/0.9588 | 33.04/0.9118 | 31.85/0.8942 | 30.75/0.9133 |
| | LapSRN | 813K | 29.9G | 37.52/0.9590 | 33.08/0.9130 | 31.80/0.8950 | 30.41/0.9100 |
| | DRRN | 297K | 6,796.9G | 37.74/0.9591 | 33.23/0.9136 | 32.05/0.8973 | 31.23/0.9188 |
| | CARN | 1,592K | 222.8G | 37.76/0.9590 | 33.52/0.9166 | 32.09/0.8978 | 31.92/0.9303 |
| | CARN-M | 412K | 91.2G | 37.53/0.9583 | 33.26/0.9141 | 31.92/0.8960 | 30.83/0.9233 |
| | FALSR-A | 1,021K | 234.7G | 37.82/0.9595 | 33.55/0.9168 | 32.12/0.8987 | 31.93/0.9256 |
| | FALSR-B | 326k | 74.7G | 37.61/0.9585 | 33.29/0.9143 | 31.97/0.8967 | 31.28/0.9191 |
| | FALSR-C | 408k | 93.7G | 37.66/0.9586 | 33.26/0.9140 | 31.96/0.8965 | 31.24/0.9187 |
| | AWSRN-S | 397K | 91.2G | 37.75/0.9596 | 33.31/0.9151 | 32.00/0.8974 | 31.39/0.9207 |
| | AWSRN-SD | 348K | 79.6G | 37.86/0.9600 | 33.41/0.9161 | 32.07/0.8984 | 31.67/0.9237 |
| | AWSRN | 1,397K | 320.5G | 38.11/0.9608 | 33.78/0.9189 | 32.26/0.9006 | 32.49/0.9316 |
| | **MSSN-S(Ours)** | **80K** | **18.4G** | **37.87/0.9590** | **33.37/0.9149** | **32.03/0.8971** | **31.39/0.9198** |
| | **MSSN-M(Ours)** | **290K** | **66.9G** | **38.00/0.9595** | **33.57/0.9164** | **32.16/0.8986** | **31.89/0.9249** |
| | **MSSN(Ours)** | **865K** | **196.2G** | **38.15/0.9600** | **33.81/0.9189** | **32.26/0.9001** | **32.38/0.9296** |
| 3 | SRCNN | 57K | 52.7G | 32.75/0.9090 | 29.28/0.8209 | 28.41/0.7863 | 26.24/0.7989 |
| | FSRCNN | 12K | 5.0G | 33.16/0.9140 | 29.43/0.8242 | 28.53/0.7910 | 26.43/0.8080 |
| | VDSR | 665K | 612.6G | 33.66/0.9213 | 29.77/0.8314 | 28.82/0.7976 | 27.14/0.8279 |
| | DRCN | 1,774K | 17,974G | 33.82/0.9226 | 29.76/0.8311 | 28.80/0.7963 | 27.15/0.8276 |
| | DRRN | 297K | 6,796.9G | 34.03/0.9244 | 29.96/0.8349 | 28.95/0.8004 | 27.53/0.8378 |
| | CARN | 1,592K | 118.8G | 34.29/0.9255 | 30.29/0.8407 | 29.06/0.8034 | 28.06/0.8493 |
| | CARN-M | 412K | 46.1G | 33.99/0.9236 | 30.08/0.8367 | 28.91/0.8000 | 27.55/0.8385 |
| | AWSRN-S | 477K | 48.6G | 34.02/0.9240 | 30.09/0.8376 | 28.92/0.8009 | 27.57/0.8391 |
| | AWSRN-SD | 388K | 39.5G | 34.18/0.9273 | 32.21/0.8398 | 28.99/0.8027 | 27.80/0.8444 |
| | AWSRN | 1,476K | 150.6G | 34.52/0.9281 | 30.38/0.8426 | 29.16/0.8069 | 28.42/0.8580 |
| | **MSSN-S(Ours)** | **86K** | **8.7G** | **34.13/0.9235** | **30.12/0.8368** | **28.94/0.8004** | **27.55/0.8370** |
| | **MSSN-M(Ours)** | **300K** | **30.7G** | **34.36/0.9256** | **30.28/0.8403** | **29.06/0.8034** | **27.94/0.8460** |
| | **MSSN(Ours)** | **875K** | **88.0G** | **34.55/0.9272** | **30.43/0.8433** | **29.16/0.8060** | **28.37/0.8555** |
| 4 | SRCNN | 57K | 52.7G | 30.48/0.8628 | 27.49/0.7503 | 26.90/0.7101 | 24.52/0.7221 |
| | FSRCNN | 12K | 4.6G | 30.71/0.8657 | 27.59/0.7535 | 26.98/0.7150 | 24.62/0.7280 |
| | VDSR | 665K | 612.6G | 31.35/0.8838 | 28.01/0.7674 | 27.29/0.7251 | 25.18/0.7524 |
| | DRCN | 1,774K | 17,974G | 31.53/0.8854 | 29.76/0.8311 | 27.23/0.7233 | 25.14/0.7510 |
| | LapSRN | 813K | 149.4G | 31.54/0.8850 | 28.19/0.7720 | 27.32/0.7280 | 25.21/0.7560 |
| | DRRN | 297K | 6,796.9G | 31.68/0.8888 | 28.21/0.7720 | 27.38/0.7284 | 25.44/0.7638 |
| | CARN | 1,592K | 90.9G | 32.13/0.8937 | 28.60/0.7806 | 27.58/0.7349 | 26.07/0.7837 |
| | CARN-M | 412K | 32.5G | 31.92/0.8903 | 28.42/0.7762 | 27.44/0.7304 | 25.62/0.7694 |
| | AWSRN-S | 588K | 33.7G | 31.77/0.8893 | 28.35/0.7761 | 27.41/0.7304 | 25.56/0.7678 |
| | AWSRN-SD | 444K | 25.4G | 31.98/0.8921 | 28.46/0.7786 | 27.48/0.7368 | 25.74/0.7746 |
| | AWSRN | 1,587K | 91.1G | 32.27/0.8960 | 28.69/0.7843 | 27.64/0.7385 | 26.29/0.7930 |
| | **MSSN-S(Ours)** | **93K** | **5.4G** | **31.88/0.8889** | **28.42/0.7768** | **27.43/0.7306** | **25.56/0.7671** |
| | **MSSN-M(Ours)** | **314K** | **18.1G** | **32.19/0.8930** | **28.58/0.7808** | **27.54/0.7343** | **25.90/0.7777** |
| | **MSSN(Ours)** | **889K** | **50.4G** | **32.32/0.8953** | **28.70/0.7842** | **27.64/0.7379** | **26.26/0.7892** |

containing MAWRN to observe the reconstruction effect of MAWRN. On the other hand, we also built the model MSSN-NM only containing the AWSSC without WN and adaptive weighted scale (AWS) to observe the performance of AWSSC. As shown in Table 3, MSSN is obviously superior to MSSN-NA and MSSN-NM with the improvement of 0.12 dB comparison with the 0.08 dB and 0.07 dB when contrasts with the baseline, which just verifies the validity of the structure we proposed.

### 4) EFFICIENCY TRADE-OFF

Figure 6 depicts the trade-off study between PSNR and parameters, PSNR and operations in relation to the number of Cascading Block and expand the ratio of the Mobile adaptive weighted residual unit. In an appropriate range of parameters and FLOPS, we evaluate the influence of the number of the cascading blocks and the factor of expanding ratio on the parameters and FLOPS. In the Figure 6, blue points represent the versus between the PSNR and params, orange dots represent the versus between the Mult-Adds and PSNR. While the round points represent the data from the MSSN-S with diverse expand ratio and the diamond dots represent the MSSN-S with different numbers of CBs.

There is no doubt that the more CBs and the higher expand ratio result in the wider receptive field however the great expansion of parameters and FLOPS is unacceptable. When the expand ratio is higher than 6, the improvement from the expand ration reduces. When the factor of ER is 10, the PSNR only improved by 0.05 dB compared with the factor of 6 but the operations and FLOPS are almost doubled. The effect of increasing the ER from 4 to 6 on improving the performance is not as much when the ER is under 4.

From Figure 6, the feature could not be extracted with too little CBs. On the other hand, the improvement of performance decreased when the number of CBs is over 4. We could choose an appropriate number of CBs according to the specific task.

### D. COMPARISON WITH STATE-OF-THE-ART METHODS

We compared the proposed MSSN with the most advanced SR method on the commonly used datasets for super-resolution with the PSNR and SSIM as metrics by convention. To be fair, the models which are too large or too deep are not in our consideration. One thing to note is that we use Mult-Adds as a metric for the FLOPS. The HR image size is 720p (1280 × 720) to calculate Mult-Adds.

For a comprehensive comparison, we have designed three models, MSSN, MSSN-M and MSSN-S, MSSN-M and MSSN-S both have 4 CBs. The difference is that MSSN-M's MAWRN has {64,128,64} channels. MSSN-S's MAWRN has {32,64,32} channels. MSSN has 6 CBs with the MAWRN has {64,256,64} channels.

Table 5 shows the PSNR and SSIM results of different algorithms on four benchmark data sets.

On the scale of ×2, ×3, and ×4, we compare the proposed MSSN, MSSN-M and MSSN-S with many lightweight and efficient SR methods, including SRCNN [8], FSRCNN [9], VDSR [19], DRCN [10], LapSRN [21], CARN [2], AWSRN [30], FALSR [5], noted that FALSR has only a ×2 result. The results show that MSSN-M has achieved good performance on various scaling factors. Compared with SRCNN and FSRCNN, although they have fewer FLOPS, MSSN has a sharper result with more details. Compared with DRCN, LapSRN, CARN-M, FALSR-A, FALSR-B, FALSR-C, CARN, AWSRN-SD, it is not only on FLOPS and total amounts of parameters but also a mightily advantage on the reconstruction quality. Compared with AWSRN, we achieved much fewer FLOPS and parameters with comparable results. MSSN-M has a large advantage in the model with the parameterless than 1M and the total operations less than 100G.

MSSN has the best performance when we could have more parameters, it performs better than AWSRN with much smaller amounts of parameters and FLOPS, as shown in the Table 4. It is also worth to bear in mind that we could reach the balance of parameters and performance by adjusting the number of CBs and the factor of expanding ratio.

In Figure 1, we compare the performance of our proposed MSSN family (red dots) with other most advanced lightweight networks (blue dots) on the Set5 test (scaling factor ×2). While, Mult-Adds represents the number of operations, and the size of the output image is 1280 × 720.

In Figures 7 and 8, a visual comparison of the two data sets (B100, Urban100) ×2 and ×4 was shown. Our model is MSSN-M. We select some details from the image, and it can be seen that our model works better than other models, especially in the reconstruction of streaks and linear patterns, which is closer to the HR image.

## V. CONCLUSION

In summary, we propose a lightweight, efficient super-resolution SISR network. Our model has at least a 1/4 decrease in parameters and Mult-Adds while maintaining comparable performance with state-of-the-art algorithms. The efficiency of the algorithm mainly comes from the following two reasons:

1) MAWRU uses depthwise convolution, pointwise convolution to reduce parameters and operations and uses adaptive weights to maintain the efficient circulation of information and gradients.
2) AWSSM helps activate the low-level layers' features which improved efficiency, profiting from Adaptive Weighted Share-Source Skip connection in AWSSM.

In addition, we hope that this work can be applied to real-world applications. Not only can you reduce the storage space of the high-resolution images in your portable devices but also improve the quality of them.

## REFERENCES

[1] A. Ahmed, S. Kun, R. A. Memon, J. Ahmed, and G. Tefera, "Convolutional sparse coding using wavelets for single image super-resolution," *IEEE Access*, vol. 7, pp. 121350–121359, 2019.

[2] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 252–268.

[3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 135.1–135.10.

[4] J. Chaki, "A two-fold fusion fuzzy framework to restore non-uniform illuminated blurred image," *Optik*, vol. 206, Mar. 2020, Art. no. 164197.

[5] X. Chu, B. Zhang, H. Ma, R. Xu, J. Li, and Q. Li, "Fast, accurate and lightweight super-resolution with neural architecture search," 2019, *arXiv:1901.07261*. [Online]. Available: http://arxiv.org/abs/1901.07261

[6] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.

[7] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11065–11074.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 184–199.

[9] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 391–407.

[10] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 597–613.

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: http://arxiv.org/abs/1510.00149

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[14] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5197–5206.

[15] K.-W. Hung, C. Qiu, and J. Jiang, "Video super resolution via deep global-aware network," *IEEE Access*, vol. 7, pp. 74711–74720, 2019.

[16] K.-W. Hung, Z. Zhang, and J. Jiang, "Real-time image super-resolution using recursive depthwise separable convolution network," *IEEE Access*, vol. 7, pp. 99804–99816, 2019.

[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and >0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[19] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[21] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 624–632.

[22] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

[23] P. Liu, Y. Hong, and Y. Liu, "Deep differential convolutional network for single image super-resolution," *IEEE Access*, vol. 7, pp. 37555–37564, 2019.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Vancouver, BC, Canada, Jul. 2001, pp. 416–423.

[25] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 901–909.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[27] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.

[28] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3147–3155.

[29] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 114–125.

[30] C. Wang, Z. Li, and J. Shi, "Lightweight image super-resolution with adaptive weighted learning network," 2019, *arXiv:1904.02358*. [Online]. Available: http://arxiv.org/abs/1904.02358

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[32] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, "Wide activation for efficient and accurate image super-resolution," 2018, *arXiv:1808.08718*. [Online]. Available: http://arxiv.org/abs/1808.08718

[33] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.* Cham, Switzerland: Springer, 2010, pp. 711–730.

[34] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

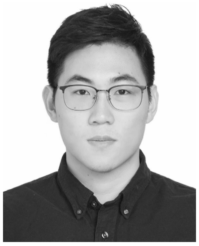
[35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 286–301.

[36] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless CNNs with low-precision weights," 2017, *arXiv:1702.03044*. [Online]. Available: http://arxiv.org/abs/1702.03044

[37] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," 2016, *arXiv:1606.06160*. [Online]. Available: http://arxiv.org/abs/1606.06160

[38] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

**JUAN DU** is currently pursuing the B.E. degree in electronic engineering with Wuhan University. Her research interests include image super-resolution, lightweight networks, and deep learning.



**WENLAN WEI** is currently pursuing the B.E. degree in electronic engineering with Wuhan University. His research interests include object tracking, image super-resolution, and deep learning.

**CIEN FAN** received the B.S. degree in electronic instruments and measurement technology, the M.S. degree in measurement technology and instruments, and the Ph.D. degree in radio physics from Wuhan University, Wuhan, China, in 1998, 2001, and 2012, respectively.

She is currently an Associate Professor with the Electronic Information School, Wuhan University. Her research interests include artificial intelligent, machine learning, and image processing.

**LIAN ZOU** was born in 1975. He received the B.S. degree in wireless communication from the Nanjing University of Posts and Telecommunications, in 1998, and the M.S. and Ph.D. degrees from Wuhan University, in 2000 and 2004, respectively. From 2005 to 2010, he was a Lecturer with the Electronic Information School, Wuhan University, and from 2011 to 2016, he became an Associate Professor. Since 2017, he has been a Professor with the Digital Signal Processing Laboratory, Electronic Information School, Wuhan University. His research interests include image analysis and understanding, object detection, image super resolution, and denoising.

**JIAWEI SHEN** is currently pursuing the B.E. degree in electronic engineering with Wuhan University. His research interests include instance segmentation, image super-resolution, and event camera.

**ZIYU ZHOU** is currently pursuing B.E. degree with the Electronic Information School, Wuhan University. Her current research interests include image super-resolution, lightweight networks, and deep learning.

**ZEZONG CHEN** received the B.S. and M.Eng. degrees in electronic engineering and the Ph.D. degree in radio physics from Wuhan University, Wuhan, China, in 1987, 1993, and 2005, respectively. Since 1993, he has been with the School of Electronic Information, Wuhan University, where he is currently a Professor carrying out teaching and research in electronic engineering. His current research interests include radar system, detect theory and signal process, application of high-frequency, and microwave radar in remote sensing of the marine environment.