s-LWSR: Super Lightweight Super-Resolution Network

Biao Li, Jiabin Liu, Bo Wang, Zhiquan Qi, and Yong Shi

Abstract—Deep learning (DL) architectures for super-resolution (SR) normally contain tremendous parameters, which has been regarded as the crucial advantage for obtaining satisfying performance. However, with the widespread use of mobile phones for taking and retouching photos, this character greatly hampers the deployment of DL-SR models on the mobile devices. To address this problem, in this paper, we propose a super lightweight SR network: s-LWSR. There are mainly three contributions in our work. Firstly, in order to efficiently abstract features from the low resolution image, we build an information pool to mix multi-level information from the first half part of the pipeline. Accordingly, the information pool feeds the second half part with the combination of hierarchical features from the previous layers. Secondly, we employ a compression module to further decrease the size of parameters. Intensive analysis confirms its capacity of trade-off between model complexity and accuracy. Thirdly, by revealing the specific role of activation in deep models, we remove several activation layers in our SR model to retain more information for performance improvement. Extensive experiments show that our s-LWSR, with limited parameters and operations, can achieve similar performance to other cumbersome DL-SR methods.

Index Terms—super-resolution, lightweight, multi-level information, model compression, activation operations.

Fig. 1. Visual SR results with 4X enlargement on "img-074" in benchmark dataset Urban100 [15]. In this comparison, s-LWSR$_{16}$ (Ours) only uses $144k$ parameters to obtain similar performance to way larger models. Besides, if properly adding more channels in our model (s-LWSR$_{32}$), the final performance will surpass the others. The compared methods include: Bicubic, IDN [16], and CARN-M [11].

I. INTRODUCTION

HOW to recover super-resolution (SR) image from its low-resolution counterpart is a longstanding problem in image processing regime [1], [2], [3], [4]. In this paper, we focus on the problem called single image super-resolution (SISR), which widely exists in medicine [5], security and surveillance [6], [7], as well as many scenarios where high-frequency details are extremely desired.

Recently, thanks to the emergence of convolutional neural networks (CNNs), specially designed SR neural networks [8], [9], [10], [11], [12], [13] as an example-based SR method, has achieved impressive performance in terms of model accuracy. Particular, these new deep learning (DL) algorithms strive to

B. Li, Z. Qi, and Y. Shi are with the School of Economics and Management, University of Chinese Academy of Sciences, Beijing 101408, China.

B. Li, Jiabing Liu, Z. Qi, and Y. Shi are also with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China, and also with the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China (e-mail: libiao17@mails.ucas.ac.cn; liujiabin008@126.com; qizhiquan@foxmail.com; yshi@ucas.ac.cn).

B. Wang is with the School of Information Technology and Management, University of International Business and Economics, Beijing 100029, China. He is currently a visiting scholar in the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA (e-mail:wangbo@uibe.edu.cn).

Y. Shi is also with the College of Information Science and Technology, University of Nebraska, Omaha, NE 68182, USA.
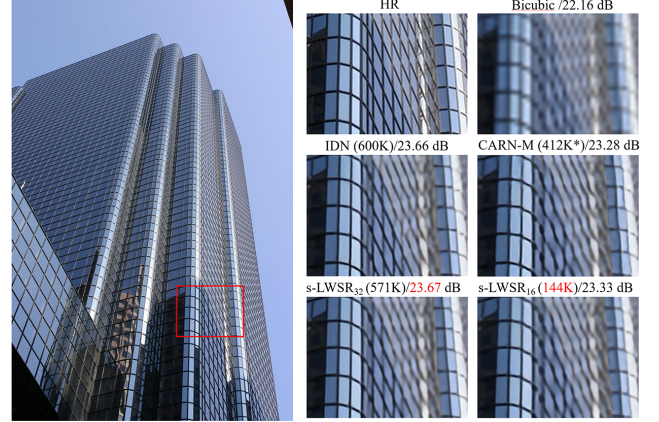
Correspond author: Zhiquan Qi

generate satisfactory SR images with super-high peak-signal-noise-ratio (PSNR) scores than their traditional competitors [14], [1].

To the best of our knowledge, the cutting edge of SISR is CNN-based methods, normally equipped with specifically designed convolutional blocks and sophisticated mechanisms, such as global residual [17], self-attention [10], Densenet [18]. In particular, the first convolutional SR network (SRCNN) is propose by Dong et al. [8], based on three simple convolutional layers, where the model is trained with an interpolated low-resolution (LR) image. Although the network is not consummately design, its performance is still significantly better than almost all traditional SR algorithms. However, shallow convolutional structure constrains the model's learning ability, and the pre-processed input causes huge computation and operation cost. Hence, along with the great development of CNNs in other Computer Vision (CV) tasks, in order to leverage more information from LR inputs, SRResNet [9] presents a new network by stacking 16 residual blocks, which are learnt from ResNet [19]. Later, EDSR [12] leverages 32 modified residual blocks with 256 channels to build an enormous SR model. Eventually, EDSR proves its super generating ability by winning the NTIRE2017 Super-Resolution Challenge [20] . As far as we know, RCAN [10] is currently the best CNN-based SR method (according to PSNR), which employs complicated residual in residual (RIR) block and self-attention mechanism.

However, as smart phones develop into regular tools for

taking photos or retouch images on daily basis, CNN-based SR algorithms, which are innately designed with tremendous number of parameters, are not suitable for lightweight delivery of the model, especially as a built-in application in mobile devices. The contradiction between accuracy and efficiency raises a demanding challenge: How to deploy a CNN-based SR model on these civil-use mobile devices with a comparable performance. In other words, designing a lightweight SR network while maintaining the advanced ability in image processing becomes a rather tough yet promising computer vision task.

Generally speaking, an appropriate model architecture with well-designed hyper-parameters is needed in order to build an accurate and fast lightweight model, which is attributed to well arrangement of two principal factors therein: parameters and operations. Hence, to promote the application of SR methods on mobile devices, the essential issue will be focusing on reducing the number of parameters and operations, while keeping satisfying performance. In terms of parameters decrease, one widespread idea is to slim the network by parameters sharing among different blocks/modules. For example, the DRCN [21] and DRRN [22] recursively employ certain basic block with same parameters.

In addition to architecture modification, some methods attempt to reduce the operations along with the parameters through unusual convolutional layer (e.g., depth-wise separable convolution [23]), cascading structure [19], or even neural architecture search (NAS) [24]. Regarding lightweight SISR, to our knowledge, CARN [11] and FALSR [25] achieve state-of-the-art results by appropriately balancing between SR restoration accuracy and model simplicity. Although these advanced compression methods have made a great progress on decreasing model size and operations, there is still a huge space for improvement.

In this paper, we propose an adjustable super lightweight SR network called s-LWSR to promote bette balance between accurate and model size than former SR methods. The contribution of this paper is mainly threefold:

- Inspired by U-Net [26], we build an SR model with symmetric architecture, possessing an assistant information pool. The skip connection mechanism greatly promotes learning ability. By further combination of multi-level information from chosen layers, we build the information pool to transmit features to high-dimensional channels. Experiments show that our new architecture does well in extracting accurate information. This new information pool enforces better features transmission between the first and the second half of the model.
- We propose a comparatively flexible SR model compared with existing methods. Normally, the most effective factor of model size is channel numbers in intermediate layers. Here, we also modify the model size by different setting of channel numbers. Nevertheless, number change results in reduplicated model variation. Hence, by introducing a novel compression module (the inverted residual block originally borrowed from MobileNet V2 [27]), the model size can be reduced by partly replacing normal residual blocks. In this way, we can control the total number of

parameters within the ideal size by properly choosing the channel number and replacing specific layers with the new compression module.
- According to our observation, when performing the non-linear mapping in some activation layers (e.g., ReLU), useful information is likely to be partly discarded. As a result, we remove some activation operations to retain object details in our lightweight model. Experiments prove that this minor modification improves the performance of our lightweight SR model.

II. RELATED WORK

With the development of deep learning, a bunch of achievements on SR has been obtained [8], [10], [9], [11], [25], [28], [12], [13], [19], [17], [27], [22], [29]. There are many detailed reviews about SR development in these papers. Based on these surveys, we firstly present a brief introduction about DL-SR algorithms. Additionally, literature study addresses model compression will be given in Section II-B.

A. Deep Single Image Super-Resolution (SISR)

The first deep SISR model that surpasses almost all former traditional methods is SRCNN [8]. In this end-to-end network, three convolutional layers are employed to produce HR images from their interpolated LR counterparts. Then, Dong et al. push the envelope further by introducing a new architecture FSRCNN [30]. The model replaces the pre-upsampling layer at the beginning of the network with a learnable scale-up layer at the end of the network. Because of training with smaller patches in most intermediate layers, the computational and operational costs greatly drop.

Subsequently, more sophisticated and powerful approaches have been proposed. For instance, by using 20 convolutional layers and a global residual, VDSR [17] obtains a shocking result that satisfies various applications. Meanwhile, DRCN [21] proposes a deeper recursive architecture with fewer parameters. In particular, several identical layers are stacked recursively in DRCN. At the same time, recursive-supervision and skip-connection are applied to ease the problem of mis-convergence.

Besides, benefiting from ResNet [19], SRResNet [9] improves the model efficiency by stacking several residual blocks. Based on SRResNet, Lim et al. propose the EDSR [12], which removes the batch normalization [31] module and expends the width of channels. However, there are still more than 40 million parameters in this model. Recently, a very deep residual network RCAN is proposed [10], which introduces a novel local block and the channel attention mechanism. As described in the paper, the attention mechanism further facilitates learning in high-frequency information. Although these methods receive the state-of-the-art results on PSNR, too many parameters ($\sim 30 - 40$ million parameters) make them hard to run on common CPU-based computers, not to mention any mobile devices/phones.

On the other hand, although most SR algorithms persist in obtaining SOTA results in pixel level, it is still controversial that high PSNR or SSIM guarantees satisfying and realistic

feeling in visual. Based on this consideration, some former researches focus on how to generate perceptual satisfying images. For example, SRGAN [9] leverage the generative adversarial networks (GANs) [32] with SRResNet as the generator to produce photo-realistic images. Similar to SRGAN, EnhanceNet [33] produces automated texture synthesis in a GANs framework. Although GAN-based SR models work well on perceptual generation, they act poorly on PSNR or SSIM accuracy.

In this paper, we mainly focus on how to obtain more accurate SR images in pixel level. However, our perspective is to properly balance between the pixel level fidelity and the model size.

B. Model Compression

Recently, how to make deep models be capable in running on mobile devices has received much attention. In this section, we provide a brief survey on compression methods, especially in SR relevant models. Firstly, most compression methods try to compress the model by modifying the network structure, such as [34], [35], [36], [37]. In MobileNetV1 [36], it reduces the number of parameters through utilizing depth-wise separable convolutions [23]. Since convolution operation are separated into two steps, the total number of parameters is reduced in a large margin, accompanying with the learning ability decline. In order to maintain the accuracy as reducing the model size, MobileNetV2 [37] proposes a novel layer module: the inverted residual with linear bottleneck. A scale factor is introduced to add more channels into the compression module. As a result, we can obtain better performance by reducing the compression level. In addition, a new compression pattern: neural architecture search (NAS) [38], which searches architecture by genetic algorithms, reinforcement learning, and Bayesian optimization, has received much attention. In this paper, we employ a similar mechanism as MobileNetV2 to build an efficient lightweight model.

For SR compression, Kim et al. introduce the recursive layers to share parameters in different blocks. They propose a very deep convolutional network (DRCN) [21] consisting of 16 identical intermediate layers. In this way, the number of parameters can be controlled when more layers are added. Similar to DRCN, DRRN [22] utilizes both global and local residual learning to further optimize the method. Using these recursive blocks, DRRN with 52 recursive layers surpasses former methods in performance. Recently, Ahn et al. design an efficient and lightweight model called CARN [11]. Their compression strategies include the residual-E (similar to MobileNetV1 [36]) and the recursive layers in the cascading framework. Finally, the CARN-M achieves comparable accuracy to other CNN-based SR methods, with fewer parameters and operations than CARN. Besides, the NAS strategy (like [38]) is proposed in FALSR [25]. Unsurprisingly, its result is comparable with CARN or CARN-M with appropriate model size. However, the generated architecture is extremely complex and hard to explain. Besides, Ma et al. make efforts to use binary weights and operations, compared with general 16-bit or 32-bit float operations, to address the over-parametrization in [39].

Though these lightweight SR models have achieved great success, there is still huge improvement space in how to obtain a better balanced and more flexible SR model. This is the start point of our research.

III. METHODOLOGY

In this section, we present the technical details of s-LWSR, which consists of five parts: basic residual blocks, symmetric connection frame, information pool, model compression, and activation removal mechanism. The first part, residual blocks, is the fundamental unit used to sufficiently extract information from the LR image (i.e., I^{LR}). The second and third parts work as the backbone of the network, functioning as the fusion of multi-level information among intermediate layers. In the fourth part, we further introduce a compression module to decrease the number of parameters and operations, so that the model size can be controlled within an ideal range. In the last part, selected activation layers are removed from the pipeline to retain more information in inner layers. The architecture of our s-LWSR is shown in Fig. 2.

A. Basic Residual Block

We firstly introduce the basic cell of s-LWSR: the residual block ($\mathcal{R}$)[19], which plays the fundamental role in our model. It leads to excellent extracting ability as learning from the LR inputs (I^{LR}). The i_{th} cell is defined as:

$$R^i = \alpha \cdot Conv(Conv(\mathcal{F}(R^{i-1}))) \cdot +\mathcal{F}(R^{i-1}), \qquad (1)$$

where R^i refers to final output of the i_{th} residual block. As shown in Fig. 3, the starting activation operation ($\mathcal{F}$) is utilized to process initial input to all following operations. In the branch part, two convolutional layers are cascaded like other residual setting. A scale factor: α is introduced to control the effect of residual branch. Both of them are used to extract useful information and increase dimensions. Inspired by the EDSR, we remove all batch normalization layers from the original residual block to enhance final performance, as well as reducing the redundant operations.

B. Symmetric Connection Frame

Inspired by U-Net [26], we propose a novel symmetric architecture which is depicted in Fig. 2. Like most SR models, the whole process of s-LWSR contains three sub-procedures: original feature extraction, detailed information learning, and SR image restoration. The RBG inputs (I_{LR}) are firstly operated by original feature extraction part. Then, pre-processed layers go through a series of well-designed blocks which are used to act accurate information. Finally, SR images (I_{SR}) are generated from the last outputs containing abundant features by the SR image restoration block, where HR images (I_{HR}) supervise the quality of generations.

In s-LWSR, experiments prove a trade-off between accuracy and model size: the more channels involved, the better performance achieved. In order to flexibly adjust the model size, we set the channel number of all residual blocks, n-feats (β), as the primary factor of model size. In Fig. 2, the channel number
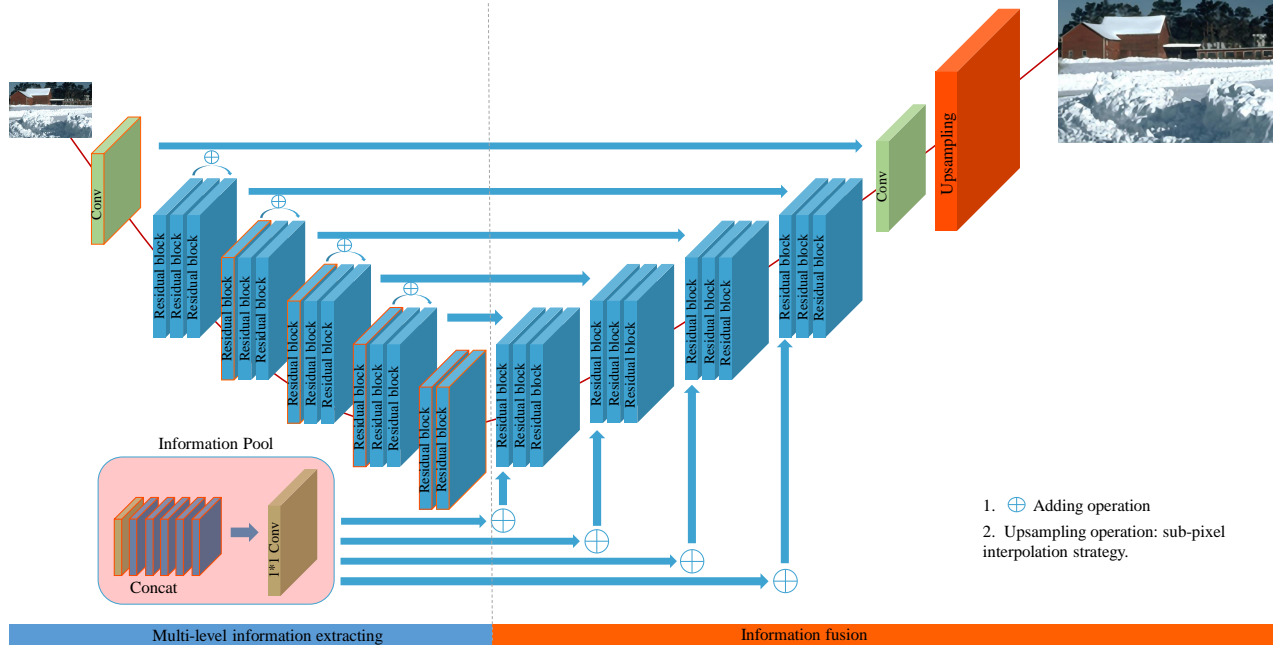
Fig. 2. The architecture of s-LWSR. The blue one is the basic residual block with all chosen blocks for information pool marked in red. Convolutional layers appear in green color. The information pool and the path of information are also marked with arrow lines.
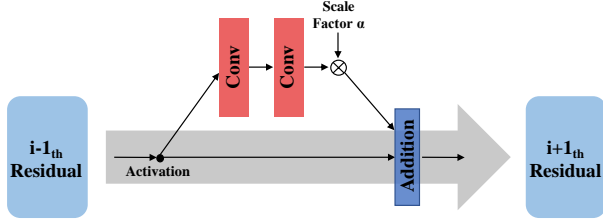

Fig. 3. The proposed residual block in s-LWSR. There are two separate information flows. A scale factor is used to control the magnitude of the information introduced in the branch. Features from two flows are element-wise added as the input of the next block.

is chosen from $[16, 32, 64, 128]$. With the increasing of β, the model size enlarges diploid. More experimental details about models with different channels are shown in Section IV-A.

As shown in Fig. 2, a sequence of basic residual blocks consecutively connected, aiming at learning the feature map between I^{LR} and I^{SR}. Similar to U-Net, our model equips the skip-connection between corresponding structure channels, and the entire mid-procedure is separated into nine bunches of local blocks (LB_s). Separated by function, the first five LB_s serve as the multi-level information extractor for the information pool, and rest blocks are information fusion part. Inspired by RDN [40], we further introduce local residual learning (LRL) to fuse features from different dimension. Given any LB_s^i in the information extractor, the information propagating process runs as follows:

$$LB_{output}^i = LB_{R3}^i(LB_{R1}^i(LB_{output}^{i-1}) + LB_{R2}^i(LB_{R1}^i)). \quad (2)$$

Benefit from the skip-connection and LRL, the I^{LR} can be sufficiently processed in local spatial architecture with multi-level information.

For the latter half of LB_s, the sum of features from the information pool and skip connection of former layers form their input. To coordinate the proportion, we set 0.5 as weight for either source. As a result, s-LWSR is not only fully extracting multi-level information from information pool, but also fully utilizes specific features of its corresponding former layers.

C. Information Pool

For combining detailed multi-layer information, specific layers in the former five LB_s are chosen as sources of the information pool. As shown in Fig. 2, we mark these layers with red border. All chosen layers are firstly concatenated, and then followed with a 1×1 convolutional layer which is used to reduce these five times concatenated layers to original input numbers. Finally, the output of information pool contains the same number of channels as other residual blocks. To be specified, equal layers is the basic processing for adding operations at any point of the network. In general, the whole process of information pool can be described as:

$$IP_{output} = Conv^*(Cat[conv_1, R_2^1, R_3^1, R_4^1, R_5^1, R_5^2]), \quad (3)$$

where $Conv^*$ denotes the 1×1 convolution, and R_j^i represents the i_{th} residual block in the j_{th} block bunch.

In fact, a similar structure has been introduced in DRCN [27], where all predictions from different layers are weighted combined in the last layer. The intention therein is to train the network in a supervised way. The output of inner blocks is summed with an extra weight factor w. Then, the output I^{SR} is determined by the learning ability of middle blocks,

and parameter sharing is employed in all the learning blocks of DRCN for reducing the number of parameters.

Hence, although the information pool utilizes the similar structure as DRCN, the underlying mechanism is fairly different. Instead of adding every generation in the halfway blocks, some specified dimensional layers chosen by experiments are concatenated in the information pool, which considerable alleviate the over-fitting problem. We choose the channel concatenation because it can maintain more multi-level information within the channels, whereas the channel addition operation will change the value in the tensor. Totally, the information pool introduced here is distinct from the existing information fusion strategies.

D. Model Compression of s-LWSR

In deep learning architectures, the function of how to count parameters in each convolutional layer is like:

$$Para_{sum} = F_{kernel} \times F_{kernel} \times C_{input} \times C_{output} + C_{output}, \quad (4)$$

where F_{kernel} is the kernel size and C is the channel number. In particular, when the channel number reduces by half, both of C_{input} and C_{output} decrease by half, which further cause that the total number of parameters approximately decreases to one quarter of its full size. On the other hand, in order to endow the model size with the flexibility, we further compress s-LWSR with a novel module: The inverted residual with linear bottleneck, which is originally introduced in MobileNetV2 [37]. This paper demonstrates that this compression module improves the performance in a large margin, compared with the depth-wise separable convolution in MobileNetV1 [36]. Details of the module are illustrated in Fig. 4. In our model, some basic blocks are changed with this new module to progressively reduce the model size to the ideal range.
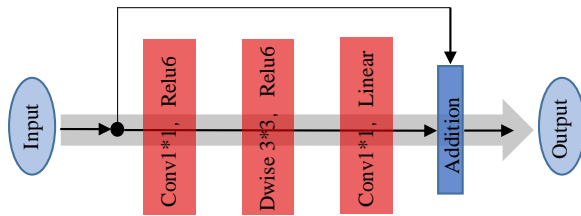

Fig. 4. Illustration of the inverted residual module which is introduced in MobileNetV2. Three convolutional layers and a residual connection are involved. To improve the learning ability of the module, the channels of middle layers are increased by 1×1 convolution. Compared with MobileNetV1 [36], the number of inner layers affects the performance and total parameters.

E. Activation Removal

To maintain more information when performing model compression, we modify s-LWSR with activation layers removal mechanism. Unlike high-level CV tasks, such as object detection YOLOV3 [41] and semantic segmentation [42], the SR task requires to recover information from the I^{LR} as much as possible. Thus, maintaining the comprehensive details flow from the original input is essential to the following

processing on features. However, the activation operations, e.g., ReLU, alter the details in feature map in order to realize the non-linearity, which may undermine the fidelity of useful information [43]. The learning ability of SR models inevitably suffers from the model compression module in a certain degree. Hence, removing some activation layers could be a proper strategy to offset the information loss brought by the model compression, and retain important feature information. Meanwhile, this operation can further reduce the computational complexity. However, it is still an open question that how many activation layers should be removed, and we strive for making it clear through looking at the influence arisen by this removal with our multi-level ablation analysis in Section IV-B.

IV. EXPERIMENTS

A. Implementation and Training Details

To fair compare our approach with other DL-SR methods, we conduct the training process on a widely used dataset, DIV2K [44], which contains 800 LR-HR image pairs. Then, we investigate the performance of different algorithms upon four standard datasets: $Set5$ [45], $Set14$ [46], $B100$ [47], and $Urban100$ [15]. Besides, the generated SR images are transformed into $YCbCr$ space, where we compute the corresponding PSNR and SSIM [48] on the Y channel.

In detail, the data augmentation is firstly adopted to the training data to improve the generalization ability. During the training process, our algorithm extracts features with 48×48 patches from the I^{LR}, and the objective is optimized with the ADAM ($\beta_1 = 0.9, \beta_2 = 0.999$) [49]. Besides, most filters in the pipeline are designed with the same size 3×3, except some 1×1 layers for channels reduction, and the learning rate is set as 1×10^{-4}, halved every 200 epochs. We implement s-LWSR on Pytorch with a Titan Xp GPU. Our code is availabe on https://github.com/Sudo-Biao/s-LWSR.

B. Model Analysis

Most DL-SR algorithms can be separated into three parts: feature extraction, feature learning, and up-sampling. For the first part in our method, a $conv(3, n-feats)$ layer is implemented to primarily learn the comprehensive features, which are the inputs of the next layer, the information pool, and the global residual unit. In order to maintain more details from the input and deliver them to the following operation layers, we only use one $conv(3, n-feats)$ to achieve channel number change. We will explicitly illustrate the feature learning part in Section IV-C. For the up-sampling part, we adopt the sub-pixel shuffling strategy, which is commonly used by other outstanding DL-SR methods.

As we mentioned in Section III-D, the channel number is a crucial factor with a great effect on the model size and accuracy performance. In our experiment, we firstly use 16 channels for the simplicity of the desirable lightweight model. Then, channels in all modules are $2\times$ added for better learning ability, like 32 and 64. For the flexible parameter modification, we utilize the inverted residual module and remove some activation layers. Further analysis on the trade-off between

Fig. 5. The comparison of s-LWSR with different model settings. The test images are from $Set14$ [46]. In the comparison, we choose three models: s-LWSR$_{16}$, s-LWSR$_{32}$, and compressed s-LWSR$_{32}$ (s-LWSR32C). The final performance suggests that the number of channels is a crucial factor to SR results, while the use of compression modules significantly decreases the learning ability of the model.

the number of parameters and the model accuracy is provided in Fig. 5.

Channel Size. To demonstrate the learning ability of our model, we build several models with different $n-feats$: $16\times$, $32\times$, and $64\times$. The total number of parameters ranges from $140K$ to $2277K$. Referring to the $4\times$ SR task, s-LWSR$_{16}$ leverages an extremely small network to learn the feature map between I^{LR} and I^{SR}, and the final result is comparable to some DL-SR methods with several times larger in parameters as shown in Table II. Hence, s-LWSR$_{16}$ is the specific model that perfectly solves the mobile device implementation issue aforementioned. More visual detail comparisons are illustrated in Fig. 1.

The numerical comparison can be found in Table I. Experiments clearly demonstrate that the PSNR value can be significantly improved with additional parameters. However, the comparison with former leading methods proves that our model can achieve similar performance with considerable fewer parameters. In detail, we first compare our smallest model (s-LWSR$_{16}$), which is equipped with a deeper but thinner network, with other outstanding methods. Our method contains fewer parameters and operations than that of Lap-SRN, VDSR, and DRCN, while receiving even higher PSNR values in the final results. To be specific, for $4\times$ SR task on $Set5$, s-LWSR$_{16}$ achieves 31.62 dB, which is respectively 0.08 dB, 0.27 dB, and 0.09 dB higher than LapSRN, VDSR, and DRCN. On the other hand, the model parameter size of s-LWSR$_{16}$ is respectively 17.7%, 21.7%, and 8.1% of those state-of-the-art DL-SR methods. Meanwhile, the decrease of operations is even much greater, which are 5.6% of LapSRN, 1.4% of VDSR, and 0.085% of DRCN, respectively.

Besides, if we double the $n-feats$ to generate a bigger model: s-LWSR$_{32}$, it achieves the best performance of all SOTA DL-SR methods that with $< 1000K$ parameters on datasets: $Set5$, $B100$, and $Urban100$. Compared with s-LWSR$_{16}$, s-LWSR$_{32}$ is four times larger, which leads to 0.42 dB improvement in the final result on $Set5$. Besides, compared with former leading lightweight methods: CARN-M and IDN [16], our $32n-feats$ model performs better with 0.12 dB and 0.22 dB higher in PSNR for $4\times$ SR task on $Set5$ respectively. However, there is no data to compete with FALSR-A due to the lack of available code in public. Hence, we follow the

allegation in the paper that their results are comparable to CARN-M. In particular, CARN-M proposes a single model for $2\times$, $3\times$, and $4\times$ SR images at the same time. However, when calculating the parameter and multi-adds, they divide the total parameters number by 3. Our s-LWSR$_{32}$ contains less than half of the number of parameters and multi-adds in the entire CARN-M model, while obtaining better performance. In general, the generations of s-LWSR$_{32}$ verify the promising learning ability of the proposed set of mechanisms in our SR structure. To further study the relationship between the number of channels and the performance in our method, we increase the channel numbers to 64, that is, s-LWSR$_{64}$. We conduct additional experiments to affirm the expected capacity of the proposed unit. The final results are displayed in Table II.

Further Compression. The former comparison of s-LWSR with different $n-feats$ verifies the effectiveness and efficiency of our network. When designing a model for a practical SR problem, the number of $n-feats$ is determined by the computation resource. In addition, parameters decrease in three quarters when the $n-feat$ is halved down. There is still a huge space for the better trade-off between the number of parameters and the final accuracy. To address the issue, we introduce the inverted residual blocks derived from the MobileNetV2 in our model. When the basic residual blocks are replaced by this compression unit, the number of parameters is further reduced in a relatively small degree compared with channel changing. Taking s-LWSR$_{32}$ for an example, the total number of parameters reduces from 571K to 124K when all layers are replaced with this new module, which is a similar size as that of s-LWSR$_{16}$. We show the setting details in Table. I. On the other hand, experiments also demonstrate that the reverse residual block is less capable in extracting features than the original residual block. For example, the PSNR value of the entire compressed s-LWSR$_{32}$ is 0.4 dB less than that of s-LWSR$_{16}$ on the condition of similar model size. The comparison is shown in Fig. 5. As a result, the number of compressed blocks involved in the model should be elaborately determined to balance the model size and performance.

Activation Removal. In addition to the compression block, we further remove several activation layers to retain more details in the very model with small size. Note that the thinner channel design of the small model limits its learning ability.

TABLE I
THE COMPARISON OF THE ORIGINAL s-LWSR AND TWO DERIVATIVES
TRANSFORMED IN THE DEPTH OR THE WIDTH. THE CHANGES OF
PARAMETERS AND PSRN ARE ILLUSTRATED.

Options	s-LWSR(base line)	Depth	Width
Basic blocks	26	6	26
n-feats	32	32	16
Loss function	L1	L1	L1
Parameters	$571K$	$308K$	$144K$
PSNR(+)	32.15	31.93	31.78

How to retain more accurate information of input becomes a crucial factor regarding to better performance. Hence, we imply the strategy of removing some activation layers to keep more information.

To evaluate our opinion, some activation layers are discarded from the model-s-LWSR$_{16}$. More comparing experiments are done for the purpose that how the model change with the reduce of activation layers. Actually, we decrease activation operations with the setting: rare (only first and last convolutional layers) kept, 1/3 kept, 1/2 kept, 2/3 kept and all. It can be inferred from the results that the removal of the moderate number of activate layers brings the beneficial effect on the small SR model. Even with a few activation layers, our model can still achieve comparable results. What's more, with the increasing of parameters, on the contrary, the removing operation results in a worse performance. We can see from the chart that better performance is achieved in all middle setting(like 1/3, 2/1, or 2/3) compared with rare and all activation layers kept. The final outputs are illustrated in Fig. 6. Our final s-LWSR model imply half activation setting to obtain a better balance between PSNR and SSIM.

C. Ablation Study

In s-LWSR, is the newly introduced information pool really works for final performance? To answer this question, we design the ablation experiments. Besides, the chosen channels are evaluated by different setting to better evaluate effects.

For the purpose of acquiring multi-dimensional information of inputs, chosen layers of the front half model are concatenated as the information pool which provides hybrid features to latter layers. From the perspective of information utilization, the more details are involved, the better performance of model achieves. However, over recurrence leads to overfitting. We respectively compare the performance of different setting. Moreover, s-LWSR with 16, 32 and 64 channels are all involved for clarifying the effect of model size.

As shown in Table III, the existence of information pool slightly increases SSIM score and PSNR. We mark the best scores in red color. The benefit exists among all three settings and performs better with the increase of channel number. This trend is related to learning ability and more parameters. Because of minor filtered operations, former layers extract more accurate and useful information from input. As a result, chosen layers bring these better details into the information pool and are transferred to the latter layers. Because there are

skip connections without information pool, the improvements are limited in a rather small level.
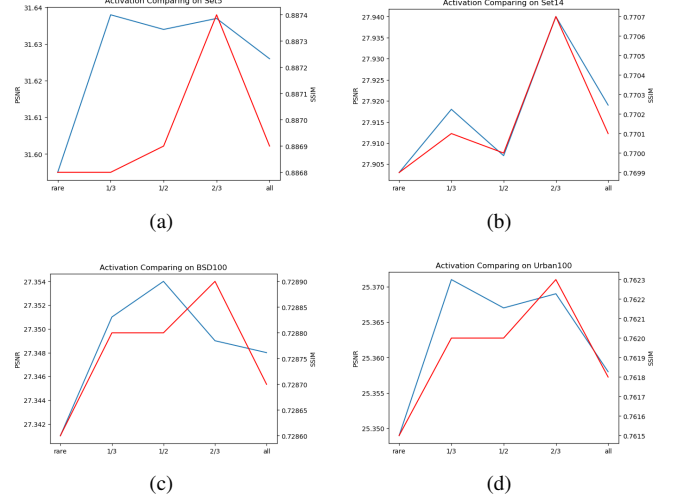

Fig. 6. The PSNR and the SSIM values of s-LWSR$_{16}$ with different ratios of activation layers removal on $Set5$ [45], $Set14$ [46], $BSD100$ [47], and $Urban100$ [15]. There are five settings: rare(only conv layers keep activation), 1/3 activation, half activation, 2/3 activation, and all activation. In details, the comparing results indicate that s-LWSR$_{16}$ with part activation layers ablation possesses significant advantage in both PSNR and SSIM.

We further make contrast experiments in s-LWSR$_{16}$ to check out the effect of layers involved in the information pool. Note that skip connections play equal influence as the information pool, we just compare three extreme conditions of front half: all involved, half, and none. In Table III, all SR results are shown in III. From the table, we can inform that s-LWSR$_{16}$ obtains better generations in mostly datasets where the only exception is marked in blue color. Even though, SR generations achieve equal PSNR score, the SSIM provides additional evidence of the effect. We attribute the advantage of s-LWSR$_{16}$ to reasonable using of the information. To be specific, s-LWSR$_{16}$ without pool transfers information by the skip-residual mechanism which transmits given layer to fixed ones. However, our pool block gathers layers from various channels, which concatenates multi-dimensions information. Referring to s-LWSR$_{16}$ with all former layers, repetitive features of adjacent layers lead to overfitting.

D. Comparison with State-of-the-art Models

To confirm the learning ability of our proposed network, we compare our model with several state-of-the-art methods: SRCNN [8], FSRCNN [30], CARN[11], VDSR [17], MemNet [50], IDN [16], LapSRN [28], DRCN [21], DBPN [13], and EDSR [12]. We conduct the evaluation experiments through two frequently-used image quality metrics: the PSNR and the SSIM. Most pre-trained models are directly based on the $DIV2K$. Here, it is noting that that the DBPN and the CARN are trained with extra images as they declaring in their papers. Accordingly, test datasets are $Set5$, $Set14$, $B100$, and $Urban100$. In this paper, all methods are only performed for the $4\times$ SR task.

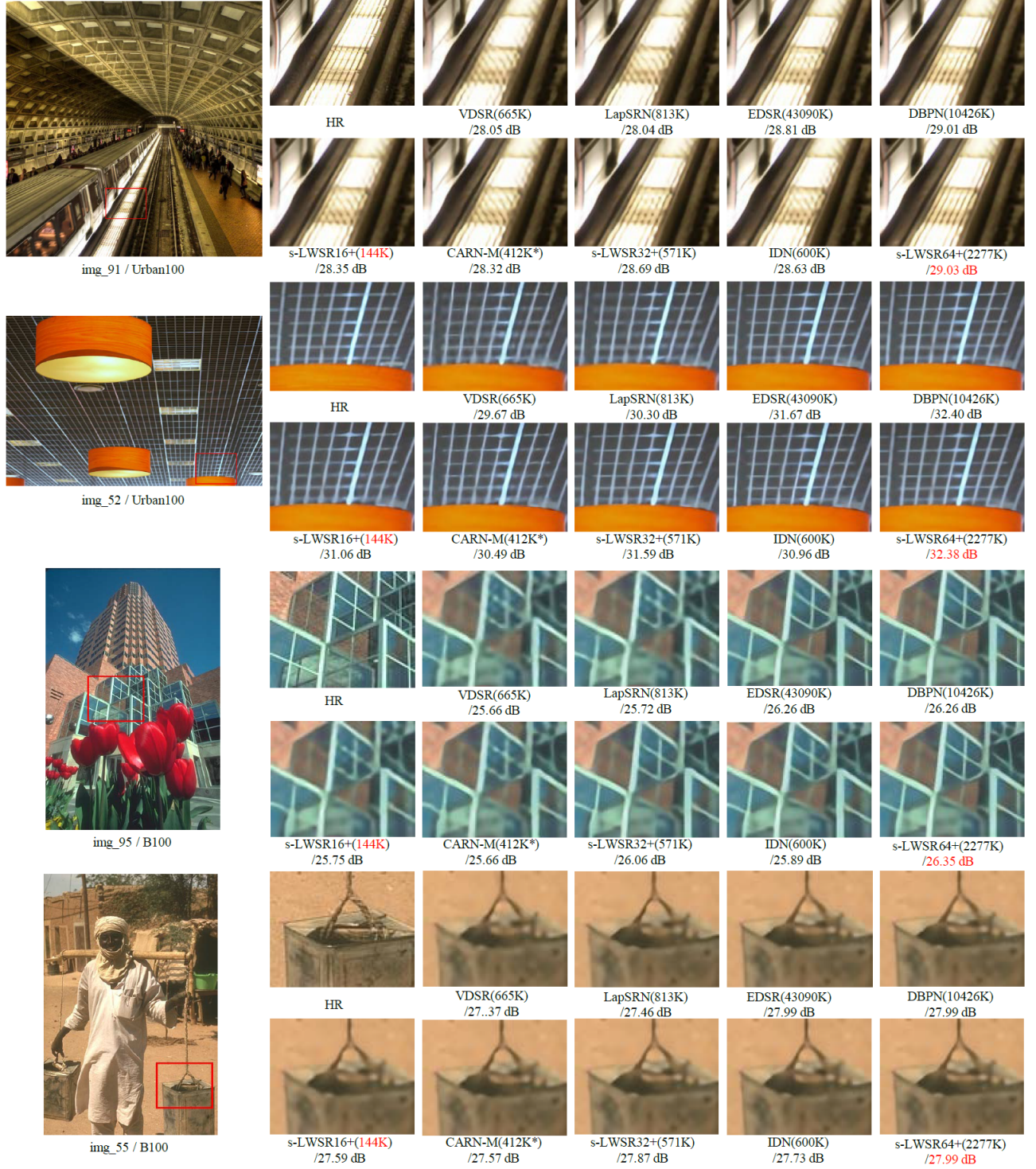For precise comparison, we separate these algorithms into three sections based on their sizes: $0-500K$, $500K-1000K$,

Fig. 7. Qualitative comparison with leading algorithms: VDSR, LapSRN, EDSR, DBPN, IDN, and CARN on 4× task. From the figure, we can point out that s-LWSR achieves outstanding performance when there is similar parameters. With the adding of more channels, s-LWSR show persistent increasing in learning ability. As shown, s-LWSR64 supass most of existing SR model in PSNR and SSIM on condition of less parameters.

TABLE II
THE COMPARISON OF S-LWSR AND OTHER STATE-OF-THE-ART METHODS: SRCNN [8], FSRCNN [30], CARN[11], VDSR [17], MEMNET [50], IDN [16], LAPSRN [28], DRCN [21], DBPN [13], AND EDSR [12] ON 4× ENLARGEMENT TASK. THE PSNR AND SSIM ARE COMPARED ACCORDING TO THE FINAL RESULTS. $s - LWSR+$ DENOTE SELF-ENSEMBLE VERSIONS OF S-LWSR.

Algorithm	Scale	Params (K)	Multi-Adds (G)	Set5 PSNR	Set5 SSIM	Set14 PSNR	Set14 SSIM	B100 PSNR	B100 SSIM	Urban100 PSNR	Urban100 SSIM
Bicubic	4	-	-	28.42	0.810	26.10	0.704	25.96	0.669	23.15	0.659
FSRCNN	4	12	4.6	30.71	0.866	27.59	0.753	26.98	0.715	24.62	0.728
SRCNN	4	57	52.7	30.48	0.863	27.49	0.750	26.90	0.710	24.52	0.722
s-LWSR$_{16}$(Ours)	4	144	8.3	31.62	0.886	27.92	0.770	27.35	0.729	25.36	0.762
s-LWSR$_{16}$+(Ours)	4	144	8.3	31.78	0.889	28.00	0.772	27.40	0.730	25.45	0.765
CARN-M	4	412*	18.3	31.92	0.890	28.42	0.776	27.44	0.730	25.63	0.769
s-LWSR$_{32}$(Ours)	4	571	32.9	32.04	0.893	28.15	0.776	27.52	0.734	25.87	0.779
s-LWSR$_{32}$+(Ours)	4	571	32.9	32.15	0.894	28.24	0.778	27.58	0.736	26.00	0.782
IDN	4	600	34.5	31.82	0.890	28.25	0.773	27.41	0.730	25.41	0.763
VDSR	4	665	612.6	31.35	0.884	28.01	0.767	27.29	0.725	25.18	0.752
MemNet	4	677	623.9	31.74	0.889	28.26	0.772	27.40	0.728	25.50	0.763
LapSRN	4	813	149.4	31.54	0.885	28.19	0.772	27.32	0.728	25.21	0.756
CARN	4	1592*	65.4	32.13	0.894	28.60	0.781	27.58	0.735	26.07	0.784
DRCN	4	1774	9788.7	31.53	0.885	28.02	0.767	27.23	0.723	25.14	0.751
s-LWSR$_{64}$(Ours)	4	2277	131.1	32.28	0.896	28.34	0.780	27.61	0.738	26.19	0.791
s-LWSR$_{64}$+(Ours)	4	2277	131.1	32.42	0.898	28.42	0.782	27.69	0.739	26.39	0.795
D-DBPN	4	10426	590.2	32.47	0.898	28.82	0.786	27.72	0.740	26.38	0.795
EDSR	4	43090	2482.0	32.46	0.897	28.80	0.788	27.71	0.742	26.64	0.803

TABLE III
THE COMPARISON OF THE ORIGINAL S-LWSR AND TWO DERIVATIVES TRANSFORMED ON THE DEPTH OR THE WIDTH. THE CHANGES OF PARAMETERS AND PSRN ARE ILLUSTRATED.

Algorithm	Scale	Set5 PSNR	Set5 SSIM	Set14 PSNR	Set14 SSIM	B100 PSNR	B100 SSIM	Urban100 PSNR	Urban100 SSIM
s-LWSR$_{16}$(normal)	4	31.63	0.8869	27.92	0.7701	27.35	0.7287	25.36	0.7618
s-LWSR$_{16}$(no-pool)	4	31.63	0.8868	27.92	0.7696	27.35	0.7284	25.36	0.7616
s-LWSR$_{16}$(pool- former 11 layers)	4	31.63	0.8871	27.90	0.7698	27.34	0.7286	25.36	0.7616
s-LWSR$_{32}$(normal)	4	32.02	0.893	28.15	0.776	27.52	0.734	25.87	0.779
s-LWSR$_{32}$(no-pool)	4	31.97	0.892	28.12	0.776	27.51	0.734	25.86	0.779
s-LWSR$_{64}$(normal)	4	32.23	0.896	28.34	0.780	27.61	0.738	26.19	0.791
s-LWSR$_{64}$(no-pool)	4	32.23	0.896	28.32	0.780	27.61	0.738	26.13	0.790

and $1000K+$. It is worth noticing that the CARN actually contains three times parameters in the main network than that asserted in the single scale-up model. Here, we only compare with the asserted size. colorredTo maximize the performance of SR generations, we adopt the self-ensemble strategy which is widerly used in EDSR, RCAN. Moreover, to separate enhanced version with original SR, the $+$ is added behind initial name. In the first section, s-LWSR$_{16}$+ performs a little worse than CARN-M, while greatly surpasses SRCNN and FSRCNN. However, the total number of parameters and operations in s-LWSR$_{16}$+ is only half of the asserted value of the CARN-M. In the second section, s-LWSR$_{32}$+ outperforms all the competitors. It can be concluded from Table II that s-LWSR$_{32}$+ demonstrates great advantages on both model size and accuracy in a large margin. Referring to the last section, s-LWSR$_{64}$+ performs similarly with the DBPN and the EDSR. Meanwhile, the size of our model is distinctly smaller on both parameters and operations. Besides, the outputs of $4\times$ enlargement are visually exhibited in Fig. 7. In general, the comparison suggests that our model has a strong capability in the SR generation, weather in the lightweight model size or better accuracy.

V. CONCLUSION

In this paper, we propose a super lightweight SR network: s-LWSR. To facilitate the implementation on mobile devices, we compress our model to only $144K$ parameters while the s-LWSR achieves a satisfying performance. Base on the symmetric architecture, we propose an information pool with skip-connection mechanism to comprehensively incorporate the multi-level information. Besides, we further explore s-LWSR with more channels and remove certain ratios of activation layers to achieve comparable performance with leading SR models. In addition, we introduce a compression module to further reduce the model size to the ideal scale. The extensive experiments demonstrate that our model performs better than other state-of-the-art lightweight SR algorithms, with a relatively smaller model size.

REFERENCES

[1] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 349–356.

[2] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[3] J. Sun, Z. Xu, and H.-Y. Shum, "Gradient profile prior and its applications in image super-resolution and enhancement," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1529–1542, 2010.

[4] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[5] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. M. S. M. de Marvao, T. Dawes, D. ORegan, and D. Rueckert, "Cardiac image super-resolution with global correspondence using multi-atlas patchmatch," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2013, pp. 9–16.

[6] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau, "Eigenface-domain super-resolution for face recognition," *IEEE transactions on image processing*, vol. 12, no. 5, pp. 597–606, 2003.

[7] W. W. Zou and P. C. Yuen, "Very low resolution face recognition problem," *IEEE Transactions on image processing*, vol. 21, no. 1, pp. 327–340, 2011.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.

[9] C. Ledig, L. Theis, F. Huszar, J. Caballero, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Computer Vision and Pattern Recognition*, 2017.

[10] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[11] 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)*, 2018, pp. 252–268.

[12] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Computer Vision and Pattern Recognition Workshops*, 2017.

[13] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[14] H. Chang, D. Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[15] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[16] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 723–731.

[17] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125.

[21] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637–1645.

[22] T. Ying, Y. Jian, and X. Liu, "Image super-resolution via deep recursive residual network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," *Computer Science*, vol. 3559, pp. 501–515, 2014.

[24] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 784–800.

[25] X. Chu, B. Zhang, H. Ma, R. Xu, J. Li, and Q. Li, "Fast, accurate and lightweight super-resolution with neural architecture search," *arXiv preprint arXiv:1901.07261*, 2019.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[28] W. S. Lai, J. B. Huang, N. Ahuja, and M. H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[29] Y. Shi, B. Li, B. Wang, Z. Qi, and J. Liu, "Unsupervised single-image super-resolution with multi-gram loss," *Electronics*, vol. 8, no. 8, p. 833, 2019.

[30] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 391–407.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[33] M. S. Sajjadi, B. Scholkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4491–4500.

[34] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[35] S. Changpinyo, M. Sandler, and A. Zhmoginov, "The power of sparsity in convolutional neural networks," *arXiv preprint arXiv:1702.06257*, 2017.

[36] 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," *arXiv preprint arXiv:1704.04861*, 2017.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[38] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[39] Y. Ma, H. Xiong, Z. Hu, and L. Ma, "Efficient super resolution using binarized neural network," *arXiv preprint arXiv:1812.06378*, 2018.

[40] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[41] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[42] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[43] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[44] E. Agustsson and R. 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*, 2017, pp. 126–135.

[45] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *Proceedings of the 23rd British Machine Vision Conference (BMVC)*, 2012.

[46] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[47] 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 *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2001, pp. 416–423.

[48] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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