Adjustable Super-Resolution Network via Deep Supervised Learning and Progressive Self-Distillation

Juncheng Li[a,b], Faming Fang[a,*], Tieyong Zeng[b], Guixu Zhang[a], Xizhao Wang[c]

[a] *School of Computer Science and Technology, East China Normal University, Shanghai, China.*
[b] *Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, China.*
[c] *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.*

Abstract

With the use of convolutional neural networks, Single-Image Super-Resolution (SISR) has advanced dramatically in recent years. However, we notice a phenomenon that the structure of all these models must be consistent during training and testing. This severely limits the flexibility of the model, making the same model difficult to be deployed on different sizes of platforms (e.g., computers, smartphones, and embedded devices). Therefore, it is crucial to develop a model that can adapt to different needs without retraining. To achieve this, we propose a lightweight Adjustable Super-Resolution Network (ASRN). Specifically, ASRN consists of a series of Multi-scale Aggregation Blocks (MABs), which is a lightweight and efficient module specially designed for feature extraction. Meanwhile, the Deep Supervised Learning (DSL) strategy is introduced into the model to guarantee the performance of each sub-network and a novel Progressive Self-Distillation (PSD) strategy is proposed to further improve the intermediate results of the model. With the help of DSL and PSD strategies, ASRN can achieve elastic image reconstruction. Meanwhile, ASRN is the first elastic SISR model, which shows good results after directly changing the model size without retraining.

Keywords: Single-image super-resolution, SISR, elastic image reconstruction, deep supervised learning, progressive self-distillation.

1. Introduction

Single-image super-resolution (SISR) has received increasing attention in recent years, which aims to reconstruct a super-resolution (SR) image from its degraded low-resolution (LR) one. Despite its widespread application in various tasks, such as, video enhancement [1, 2], medical image reconstruction [3, 4], and image segmentation [5, 6],

*Corresponding author

Email addresses: cvjunchengli@gmail.com (Juncheng Li), fmfang@cs.ecnu.edu.cn (Faming Fang), zeng@math.cuhk.edu.hk (Tieyong Zeng), gxzhang@cs.ecnu.edu.cn (Guixu Zhang), xizhaowang@ieee.org (Xizhao Wang)

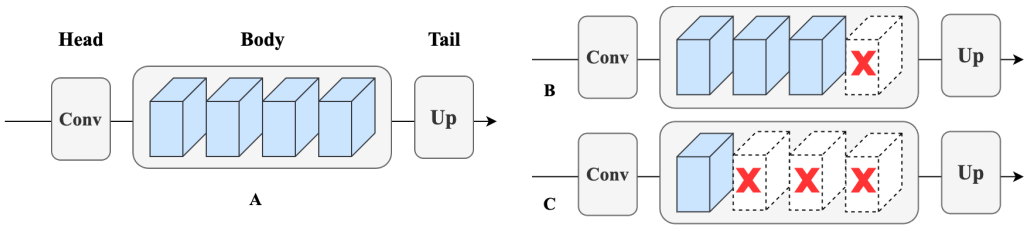


Fig. 1: An example of the modular network. This type of model can freely disassemble feature extraction blocks in the network.

it still is a challenging task since the mapping between LR and high-resolution (HR) images has multiple solutions.

To solve this problem, many methods have been proposed. Among them, Convolutional Neural Networks (CNNs) based methods [7, 8, 9, 10, 11, 12, 13, 14, 15, 16] show great potential. The majority of them is to design a specially neural network to learn the mapping between LR and HR images adaptively. For example, Dong et al. proposed the first CNN-based SISR model in 2014, named SRCNN [7]. Since then, CNN-based SR methods have been blooming and constantly refreshing the best result. In 2017, Ledig et al. proposed the SRResNet [17] by using a series of residual blocks [18], which launched the trend of network modularity. Although these models perform well, their application scenarios are limited due to numerous parameters. To address this problem, many lightweight SISR models [19, 20, 21] have been proposed in recent years, which greatly alleviates the problem of model size being too large.

The models mentioned above depict two main branches of the SISR field, each reflecting the different demands of the market. One pursues model performance but ignores the model size, while the other is dedicated to reducing model size but also leads to model performance degradation. In different application scenarios, we have different requirements for the size and performance of the model. However, there is no model that can dynamically adjust its size to adapt to different sizes of platforms and maintain excellent performance.

As shown in Fig. 1 (A), existing mainstream SISR models can be simplified to this modular structure, which includes head, body, and tail parts. Among them, the body part often consists of a series of feature extraction blocks. The biggest advantage of this type of model is that the depth of the network can be easily changed by adjusting the number of feature extraction blocks. However, this type of model still requires its structure to be completely consistent during training and testing. For example, if we remove some feature extraction blocks (B, C) in model A so that it can run on a small size platform, the model performance will be extremely degraded. This will greatly limit the flexibility and application scenarios of the model.

Recently, diverse smart devices are getting popular, such as laptops, tablets, mobile phones, and terminal devices. These devices have different storage space, memory, and computing power. This means that the size of the model that can be run on these platforms is limited. However, designing and training specialized models for different

sizes of platforms requires a lot of manpower and material resources, including design costs, training costs, storage costs, and time costs. As a result, it is crucial to develop a method that can adjust the model size without retraining. To achieve this, we propose a lightweight Adjustable Super-Resolution Network (ASRN) for SISR. ASRN consists of a set of Multi-scale Aggregation Blocks (MABs), which are lightweight and efficient feature extraction modules. Meanwhile, to build an adjustable model, we introduced the Deep Supervised Learning (DSL) strategy to guarantee that the intermediate outputs of the network are still acceptable. Meanwhile, a novel Progressive Self-Distillation (PSD) strategy is also offered to improve the intermediate outcomes even further. In summary, our ASRN can achieve elastic image reconstruction with the help of DSL and PSD strategies. In other words, the model size of ASRN can be easily changed during testing to meet different requirements without redesigning and retraining. The main contributions of this paper are as follow:

(i) We propose a lightweight and efficient Multi-scale Aggregation Block for feature extraction, which is the most important module for model building.

(ii) We propose a powerful Elastic Reconstruction Technology (ERT). To achieve this, the Deep Supervised Learning (DSL) mechanism is introduced for multi-task learning and the weight sharing strategy is used in the upsampling module. Therefore, the model can achieve elastic image reconstruction.

(iii) We propose a novel Progressive Self-Distillation (PSD) Strategy to further improve the intermediate results of the model to reduce the negative impact of multi-task learning. Therefore, the model can make full use of the neighboring deep features and achieve self-distillation elegantly.

(iv) We propose a lightweight and Adjustable Super-Resolution Network (ASRN), which can flexibly adjust the size and complexity of the model without retraining.

The rest of this paper is organized as follows. Related works are reviewed in Sec. 2. A detailed introduction of the proposed method is presented in Sec. 3. Furthermore, we give a series of experiments, ablation analyses, and discussions in Sec 4, 5, and 6, respectively. Finally, we draw a conclusion in Sec. 7.

2. Related Works

2.1. Single-Image Super-Resolution (SISR)

Image super-resolution, especially single-image super-resolution has been greatly developed in the past few decades. Recently, CNN-based SISR methods can be roughly divided into two categories. One is dedicated to the pursuit of high performance, and the other is dedicated to exploring lightweight networks. For example, Ledig et al. proposed a SRResNet [17] by using a series of residual blocks [18]. Lim et al. proposed an Enhanced Deep Residual Network (EDSR [22]) based on SRResNet. Both SRResNet and EDSR are modular networks consisting of a series of residual blocks. Modular structure design can improve model performance, simplify

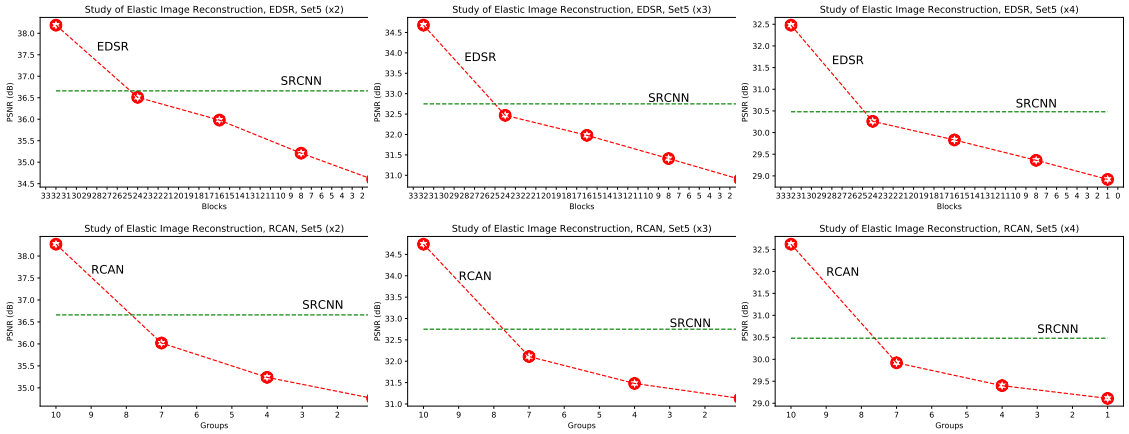
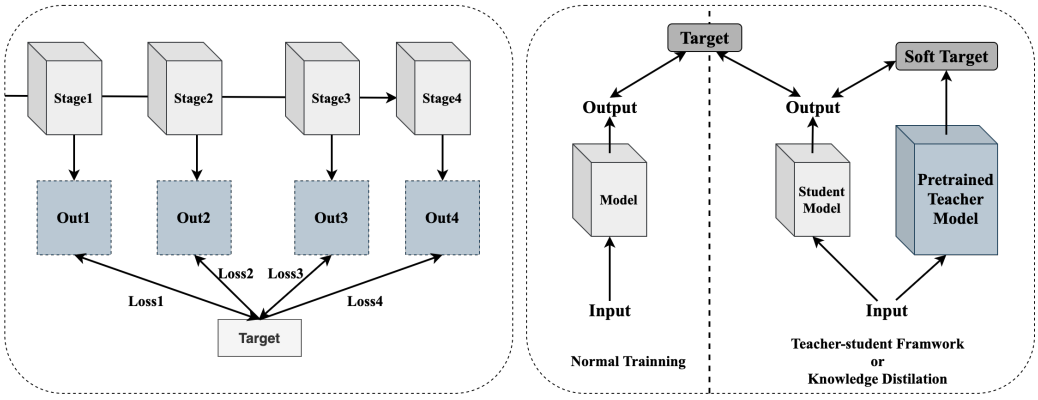


Fig. 2: As the model depth/size changes, the performance of EDSR [22] and RCAN [23] will be severely degraded, even worse than SRCNN [7].

the design process, and increase network scalability, which greatly accelerates the development of the entire field. After that, building SISR models by proposing efficient feature extraction blocks becomes mainstream, such as RDN [24], MSRN [25], and SAN [10]. Although these models achieve superior performance, they are often accompanied by a large number of parameters. To address this issue, many lightweight models [26, 27, 28, 29] have been proposed. For instance, Ahn et al. proposed a lightweight Cascaded Residual Network (CARN [19]) by using the cascade mechanism. Hui et al. proposed an Information Distillation Network (IDN [20]) and an Information Multi-Distillation Network (IMDN [21]) by using distillation and selective fusion strategies, respectively. All these models pay more attention to designing efficient feature extraction modules and learning strategies. More SISR modes can be found in [30] and [31].

2.2. Elastic Image Reconstruction

In this paper, we aim to explore a new method that can use the adjusted new model to directly reconstruct SR images without retraining. As discussed in Fig. 1, the easiest way to change the model size is to adjust the number of feature extraction blocks in the model. In Fig. 2, we provide the results of elastic image reconstruction of some classical models like EDSR [22] and RCAN [23]. Both EDSR and RCAN have a modular structure thus we can easily change the model size. According to Fig. 2, we can clearly observe that as the number of feature extraction blocks/groups decreases, the model performance is greatly reduced, even worse than SRCNN [7]. It is worth noting that the adjusted model still has more parameters than SRCNN, but the performance is much lower than SRCNN. This is because the structure of these models must be consistent during training and testing. In this paper, we aim to explore an adjustable SISR model.



(A) Deep Supervised Mechanism (B) Knowledge Distillation Strategy

Fig. 3: Schematic diagram of deep supervised mechanism and knowledge distillation strategy.

2.3. Deep Supervised Learning

The deep supervised mechanism was first proposed in the Deeply Supervised Net (DSN [32]). As shown in Fig. 3 (A), the key idea of this mechanism is to add additional losses in the middle part of the model. Furthermore, all of these losses share the same target and are optimized overall. Nowadays, the deep supervised mechanism has been proved to be helpful with the directness and transparency of the hidden layer learning process. Therefore, more and more research introduce it to improve model performance. However, they are mostly restricted to image recognition, image detection, and image segmentation. In this paper, we try to introduce the deep supervised mechanism into the SISR task to explore an elastic reconstruction technology.

2.4. Knowledge Distillation

Knowledge distillation (KD), also known as the teacher-student framework, was first proposed by Hinton et al. [33]. The knowledge transfer from a complex model (Teacher) to another lightweight model (Student) is called knowledge distillation, which aims to solve the problem of model parameter redundancy. According to the types of knowledge transfer, knowledge distillation can be roughly divided into two categories: output transfer [33, 34] and feature transfer [35, 36]. Among them, output transfer aims to pass the output of the large model as knowledge to the small model. As shown in Fig. 3 (B), this strategy takes the output of the pre-trained teacher network as the soft target and uses it as a part of the total loss to induce the training of the student network. Feature transfer regards the output of the hidden layer as the learning object and aims to make the feature maps of the student and teacher networks as similar as possible. Recently, some research also introduced the knowledge distillation strategy into image restoration tasks [37, 38]. For example, Hong et al. [38] proposed a KD-based method for image dehazing. However, these models rigidly introduce additional external models as the Teacher to guide the model training, which will make the training process more complicated. In this paper, we

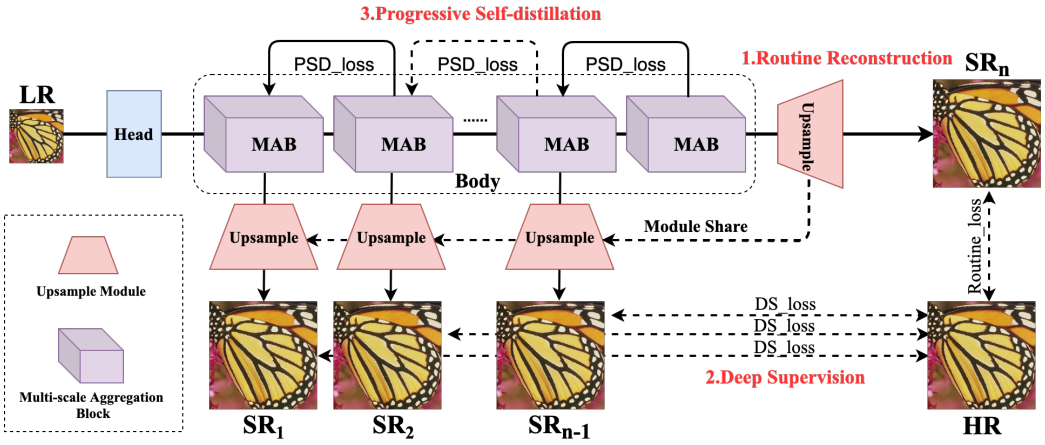


Fig. 4: The complete architecture of ASRN. ASRN consists of three modules: head, body, and upsample. The training process can be divided into three parts: routine reconstruction, deep supervision, and progressive self-distillation. Meanwhile, all upsample modules are weight sharing.

aim to explore a more elegant knowledge distillation strategy without introducing additional models.

3. Adjustable Super-Resolution Network (ASRN)

In this paper, we propose a lightweight Adjustable Super-Resolution Network (ASRN). As shown in Fig. 4, ASRN consists of three parts: head, body, and upsample modules. The head module contains two convolutional layers, which are used to transform the input image to the high dimension space. The body module consists of a series of Multi-scale Aggregation Blocks (MAB), which are used to extract image features for image reconstruction. The upsample module contains two convolutional layers and one deconvolutional layer, which takes the extracted features for the final SR image reconstruction. In addition, we propose a new training strategy for ASRN thus it can realize elastic image reconstruction.

Define I_{LR}, I_{SR}, and I_{HR} as the input, output, and label of ASRN, respectively. I_{low} and I_{high} denote the input and output of the body module. Following previous works, we use the head module to progressive upgrade the input image to the high dimension space

$$I_{low} = F_{head}(I_{LR}), \tag{1}$$

where I_{low} is the extracted low-level image features and also severed as the input of the body module for high-level feature extraction

$$I_{high} = F_{body}(I_{low}), \tag{2}$$

where $F_{body}(\cdot)$ is an elastic architecture that contains N MABs. It is worth noting that the number (N) of MABs can be easily changed during training and testing

according to actual needs. After that, all extracted image features are sent to the upsample module for SR images reconstruction

$$I_{SR} = F_{upsample}(I_{high}), \tag{3}$$

where $F_{upsample}(\cdot)$ represents the upsample module, which uses the deconvolutional layer to upscale the feature maps I_{high} into the final SR image.

Therefore, given a training dataset $\{I^i_{LR}, I^i_{HR}\}^M_{i=1}$, we aim to solve

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}^{\mathcal{SR}}(I^i_{SR}, I^i_{HR}), \tag{4}$$

where θ denotes the parameter set of the proposed ASRN and $\mathcal{L}^{\mathcal{SR}}(\cdot)$ is the specially designed elastic loss function.

3.1. Training Strategy

In this paper, we aim to explore a model that can dynamically adjusts the size and complexity of the model without retraining. In order to achieve this goal, the depth/size of ASRN must be adjustable, and the adjusted model should maintain excellent performance. Therefore, we design a new training strategy for ASRN, which makes ASRN become an elastic model and can be suitable for the proposed elastic reconstruction technology. As shown in Fig. 4, the training strategy can be divided into three parts: routine reconstruction, deep supervision, and progressive self-distillation.

3.1.1. ***Routine Reconstruction***

In this work, routine reconstruction is still the first and the most important step. Like most existing SISR models, ASRN takes the LR image as input and the data stream flows into the head, body, and upsample modules in sequence. Besides, the MSE loss is applied to train the model in this step

$$\mathcal{L}_{Routine} = \|F(I_{LR}) - I_{HR}\|_2 , \tag{5}$$

where $F(\cdot)$ and $F(I_{LR})$ denote ASRN and reconstructed SR image, respectively.

Multi-scale Aggregation Block (MAB): Extracting useful image features is crucial for SISR. In order to extract useful features, we propose a new module named Multi-scale Aggregation Block (MAB). MAB is the basic component of ASRN, which is inspired by MSRB [25], ShuffleNet [39], and Res2Net [40]. In 2018, Li et al. [25] pointed out that multi-scale image features are beneficial for SISR and proposed a multi-scale residual block (MSRB) for features extraction. However, MSRB contains a lot of parameters, which is not conducive to building a lightweight model. Therefore, we introduce the channel split, channel shuffle [39], and residual-like connection [40] mechanisms into the module to make MAB can extract rich multi-scale features with fewer parameters. As shown in Fig. 5, we first use a 1×1 convolutional layer to

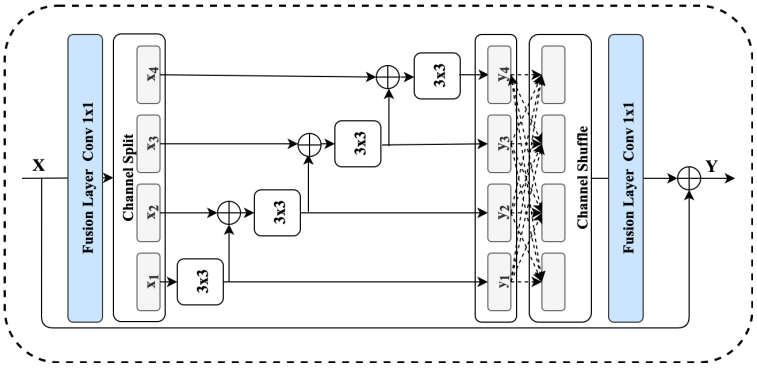


Fig. 5: The complete architecture of the proposed Multi-scale Aggregation Block (MAB).

fusion the input features. Then, we apply the channel split operation to divide the features into 4 groups $[x_1, x_2, x_3, x_4]$. In each group, we use a 3×3 convolutional layer to extract image features. Meanwhile, we add the extracted image features in the current group to the next group to make these layers have different receptive fields thus achieving multi-scale feature extraction

$$y_i = \begin{cases} K_i(x_i) & i = 1 \\ K_i(x_i + y_{i-1}) & 1 < i \leq 4 \end{cases}, \tag{6}$$

where x_i, y_i, and $K_i(\cdot)$ represent the received features, output features, and convolutional layer in the i-th group, respectively. After that, we contact all the extracted features and apply the channel shuffle to overcome the side effects brought by the channel split. Meanwhile, we introduce a fusion layer with 1×1 kernel at the tail of the block to achieve features aggregation. Finally, we introduce local residual learning [18] into our MAB to further improve the information flow. At the same time, the input and output channels of each MAB are set to 96.

Upsample Module: As shown in Fig. 6, upsampling module often contains some convolutional layers, shuffle operation (also named sub-pixel convolutional layer), or deconvolutional layers. Due to the efficiency and no additional parameters will be introduced, the sub-pixel convolutional layer has becomes the most widely used upsample method. Correspondingly, Fig. 6 (A) becomes the most widely used upsample module in recent works. However, it is worth noting that due to the characteristics of shuffle operation, the output channel of its previous convolutional layer must be S^2 (S is the upsample factor) times of the input channel in order to keep the input and output dimensions consistent. This means that module A will still accompanied by a large number of parameters. Taking S = 2 as an example, module A has 334K parameters, which even exceeds the sum of our ASRN parameters (227K). Therefore, directly using module A in a lightweight model will cause the upsample module occupying with a larger parameter proportion in the entire network. However, we found that if the parameters of the feature extraction module are fewer than the upsample module, it is not conducive to fully extracting image features. Consequently, the

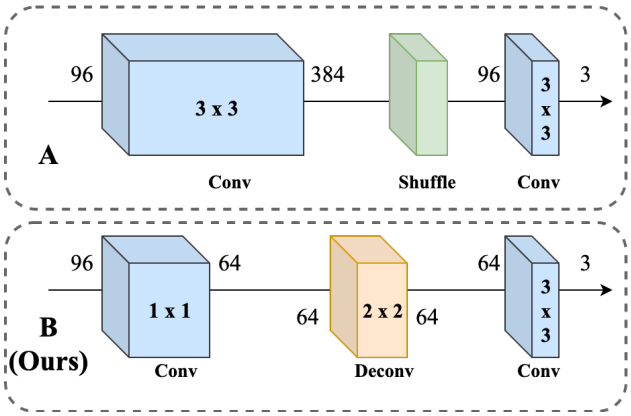


Fig. 6: Architecture comparison between regular upsample module and our used upsample module.

model performance is limited. Therefore, we redesign the parameter ratio of each module and propose module B. As shown in the figure, module B uses the deconvolutional layer instead of the shuffle operation. Taking S = 2 as an example, module B only needs 24K parameters, which is 1/14 of module A. This allows ASRN to apply more parameters for feature extraction and enables it to be a really lightweight model.

3.1.2. ***Deep Supervised Learning (DSL)***

In order to build an adjustable model, we need to guarantee that the intermediate results of the network are still acceptable. To achieve this, we introduce the Deep Supervised Learning (DSL) mechanism in the second step. As shown in Fig. 4, we add the upsampling module behind each MAB to reconstruct the intermediate SR results. This means that each branch can constitute a new SR model, which can be regarded as a simplified version of ASRN. Therefore, a total of $N-1$ SR images $[I_{SR}^{1}, I_{SR}^{2}, ..., I_{SR}^{N-1}]$ will be reconstructed and all sub-model will be supervised by the HR image. Therefore, the deep supervised loss can be defined as

$$\mathcal{L}_{DSL} = \sum_{i=1}^{N-1} \left\| F_{Sub}^{i}(I_{LR}) - I_{HR} \right\|_2 = \sum_{i=1}^{N-1} \left\| I_{SR}^{i} - I_{HR} \right\|_2 , \tag{7}$$

where N represents the number of MAB, $F_{sub}^{i}(\cdot)$ is the operation of the i-th subnet, and I_{SR}^{i} denotes the output of the i-th subnet. With the help of DSL mechanism, each subnet can work independently or operate as a sub-component of the larger model. This allows each MAB to be fully trained and can improve the robustness of the model. Meanwhile, this can maximize the performance of each subnet thus directly applying the elastic reconstruction technology can still achieve good results.

It is worth noting that all upsample modules in ASRN are weight sharing. This is because (1) module sharing makes ASRN will not bring additional parameters, which is beneficial for lightweight model building; (2) module sharing makes ASRN no longer simply aggregate multiple SR models together but achieve an adjustable structure;

(3) module sharing can fully exploit the performance of the upsample module and increase the robustness of the model.

3.1.3. ***Progressive Self-Distillation (PSD)***

In this third step, we aim to use the complex model (Teacher) to guide the lightweight model (Student) for knowledge transfer. However, specifically designing and training a suitable teacher model requires more computing resources. Rethinking the characteristics of ASRN, we found that the model consists of multiple sub-models and each one has different depths. Therefore, using the deeper sub-model in ASRN to guide the shallow sub-model directly is a more efficient method. To achieve this, we propose a self-distillation strategy, which is essentially a self-guidance process that uses the deeper sub-models to guide the shallow sub-models in ASRN.

Different from most knowledge distillation methods [33, 41, 34] that use the output as the transfer target, we choose to transfer the learned image features since there is no soft target in the SISR task. As we mentioned in the relate works, knowledge distillation can be roughly divided into two categories: output transfer and feature transfer. In high-level tasks, most knowledge distillation-based methods can directly use the output of the teacher model as the transfer target since the output provide a soft label for the model. However, in low-level tasks like SISR, directly using the output of the teacher model as the target of the shallow model will limit the model performance. This is because in SISR, the output of the teacher model is a deterministic reconstructed SR image, and the result is worse than ground-true (GT) image. Therefore, using the output of the teacher model as the transfer target will limit the performance of the student model. To address this issue, we choose to transfer the learned image features rather than the final output. However, we also noticed that the features produced by each MAB have too many channels, which will cost a lot of computational resources to calculate the distillation loss. To solve the first problem, we used the attention transfer [36] in this work. Attention transfer is also a type of feature transfer, which tries to encode the most focused spatial regions of the image features to determine its output decisions. In short, attention transfer mechanism aims to find the most critical regions to implement the distillation loss, thus reducing the computational overhead. As shown in Fig. 7, we consider the output of MAB as the tensor $A \in R^{C\times H\times W}$, which has C channels with spatial dimension $H \times W$. Following [36], we apply an activation-based mapping function $\mathcal{F}$ to change the tensor A to a spatial attention map

$$\mathcal{F} : R^{C\times H\times W} \rightarrow R^{H\times W}. \tag{8}$$

More specially, we calculate the sum of absolute values of each channel as the activation-based spatial attention maps

$$F_{sum}(A) = \sum_{i=1}^{C} |A_i| . \tag{9}$$

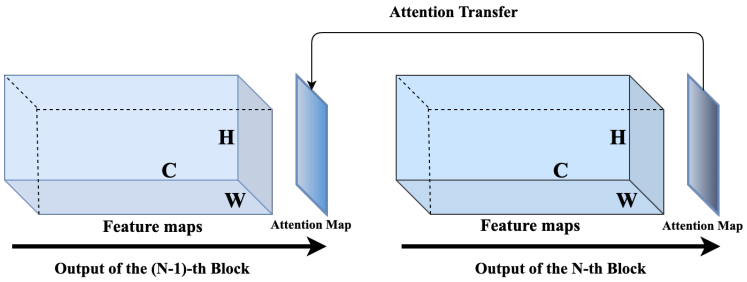


Fig. 7: Schematic diagram of the proposed Progressive Self-Distillation (PSD) mechanism.

In addition, we find that the output attention map of each MAB is completely different. This means that directly using the output of the last MAB to guide all previous MABs is not the best method since large gaps will make these sub-models difficult to work together and hard to converge. In other words, this is equal to letting one teacher model guide multiple student models, it is difficult to ensure each student model learns well. Inspired by the feedback mechanism, we transform this problem into a multi-teacher and multi-student learning problem. Therefore, each student model only needs to learn the most relevant knowledge from the next-level teacher model, which will greatly reduce the difficulty of learning. To this end, we propose a Progressive Self-Distillation (PSD) strategy, which uses the attention map produced by the next MAB to guide the current MAB. The progressive self-distillation loss can be defined as follows

$$\mathcal{L}_{PSD} = \sum_{n=2}^{N} \left\| \frac{Q_{n-1}}{\|Q_{n-1}\|_2} - \frac{Q_n}{\|Q_n\|_2} \right\|_2, \tag{10}$$

where $Q_{n-1} = \boldsymbol{vec}(F(A_{n-1}))$ and $Q_n = \boldsymbol{vec}(F(A_n))$ represent the n-th pair of student and teacher attention maps, respectively. $\boldsymbol{vec}(\cdot)$ is the vectorized operation.

Overall speaking, we propose an Adjustable Super-Resolution Network (ASRN). Meanwhile, we design a special training strategy for ASRN, which contains three steps: routine reconstruction, deep supervision, and progressive self-distillation. For each step, we propose corresponding loss functions and all these losses make up the final elastic loss function $\mathcal{L}^{\mathcal{SR}}(\cdot)$

$$\mathcal{L}^{SR} = \mathcal{L}_{Routine} + \mathcal{L}_{DSL} + \lambda \mathcal{L}_{PSD}, \tag{11}$$

where λ is a super-parameter used to control the proportion of $\mathcal{L}_{PSD}$. During training, these three steps are coordinated and performed in an end-to-end manner.

3.2. Elastic Reconstruction Technology (ERT)

As we discussed in Sec. 2.2, the method of using the adjusted new model to directly reconstruct the final image without retraining is called Elastic Image Reconstruction (ERT). In this part, we introduce the proposed model with the powerful ERT in detail.

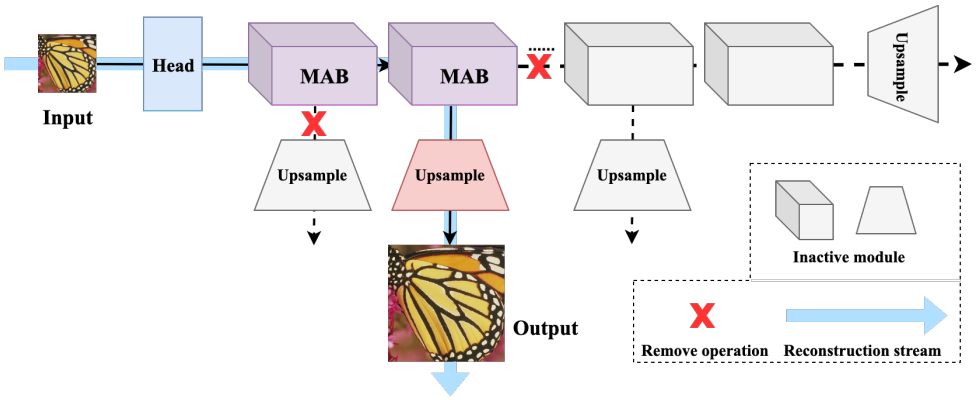


Fig. 8: Schematic diagram of elastic image reconstruction. ASRN is an Adjustable model, which can flexibly adjust its size according to the actual application platforms or requirements. As shown in this structure, the gray modules represent the inactivated modules, which do not participate in the final SR image reconstruction. The colored areas represent the final structure of the model and the blue arrow denotes the reconstruction stream.

In Fig. 8, we show the complete process of elastic image reconstruction. As we can see, the gray part represents the inactivated or removed modules, which means the module will not participate in SR image reconstruction. The colored module and the blue arrow denote the final architecture and the reconstruction stream. Specifically, if the computing platform is large enough for the large size model, we can use the complete ASRN for deployment and image reconstruction. However, if the model needs to be deployed on a small platform (e.g., mobile), we can directly remove some MABs in ASRN to build a shallow model to meet the requirements. It is worth noting that with the help of the DSL and PSD strategies, the adjusted model can still reconstruct high-quality SR images. In summary, the proposed ASRN is an adjustable model that can dynamically adjust the model size to adapt to different needs.

4. Experiments

4.1. Datasets

DIV2K [42] is a high-quality image dataset, which is widely used in the SISR task. Following previous works, we use DIV2K (1-800) as the training dataset. For testing, we choose Set5 [43], Set14 [44], BSD100 [45], Urban100 [46], and Manga109 [47]. These five datasets are the most widely used benchmark test datasets in the SISR task, which contains 328 different images that can effectively verify the model effect.

4.2. Implementation Details

Model Setting: In this paper, we propose an adjustable model, named **ASRN**. The core part of ASRN is the body module, which consists of N MABs. In the final model, we set $N = 5$ and the input/output channels of each MAB are set to 96. Meanwhile, the kernel size of all the convolutional layers is set as 3×3 except for the

Table 1: Quantitative comparisons on **BI** mode. The best and the second-best results are highlighted with red and blue, respectively. Obviously, our ASRN (SS) and ASRN can achieve competitive results with fewer parameters.

Model	Scale	Param.	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	×2	–	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
FSRCNN	×2	13K	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9710
SCN	×2	42K	36.52/0.9530	32.42/0.9040	31.24/0.8840	29.50/0.8960	35.51/0.9670
SRCNN	×2	57K	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
ESPCN	×2	57K	37.00/0.9559	32.75/0.9098	31.51/0.8939	29.87/0.9065	36.21/0.9694
CNF	×2	337K	37.66/0.9590	33.18/0.9136	31.91/0.8962	-/-	-/-
DWSR	×2	373K	37.42/0.9568	-/-	31.85/0.8944	30.46/0.9162	37.27/0.9719
VDSR	×2	665K	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140	37.22/0.9750
LapSRN	×2	812K	37.52/0.9591	33.08/0.9130	31.80/0.8950	30.41/0.9101	37.27/0.9740
WSDSR	×2	-	37.16/0.9583	32.57/0.9108	31.49/0.8914	30.23/0.9066	-/-
DNCL	×2	-	37.65/0.9599	33.18/0.9141	31.97/0.8971	30.89/0.9158	-/-
ASRN	×2	227K	37.67/0.9594	33.19/0.9144	31.95/0.8970	31.20/0.9186	37.79/0.9753
ASRN (SS)	×2	227K	37.69/0.9595	33.26/0.9149	31.98/0.8974	31.30/0.9199	37.84/0.9754
Bicubic	×3	–	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
FSRCNN	×3	13K	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
SCN	×3	42K	32.62/0.9080	29.16/0.8180	28.33/0.7830	26.21/0.8010	30.22/0.9140
SRCNN	×3	57K	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
ESPCN	×3	57K	33.02/0.9135	29.49/0.8271	28.50/0.7937	26.41/0.8161	30.79/0.9181
CNF	×3	337K	33.74/0.9221	29.90/0.8322	28.82/0.7980	-/-	-/-
DWSR	×3	373K	33.75/0.9209	-/-	28.80/0.7972	27.22/0.8293	32.14/0.9323
VDSR	×3	665K	33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290	32.01/0.9340
WSDSR	×3	-	33.45/0.9196	29.39/0.8302	28.59/0.7934	26.91/0.8204	-/-
DNCL	×3	-	33.95/0.9232	29.93/0.8340	28.91/0.7995	27.27/0.8326	-/-
ASRN	×3	248K	33.84/0.9223	29.97/0.8348	28.86/0.7990	27.41/0.8342	32.63/0.9364
ASRN (SS)	×3	248K	33.95/0.9238	30.01/0.8359	28.89/0.8000	27.45/0.8366	32.70/0.9383
Bicubic	×4	–	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
FSRCNN	×4	13K	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
SCN	×4	42K	30.39/0.8620	27.48/0.7510	26.87/0.7100	24.52/0.7250	27.39/0.8570
SRCNN	×4	57K	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
ESPCN	×4	57K	30.66/0.8646	27.71/0.7562	26.98/0.7124	24.60/0.7360	27.70/0.8560
CNF	×4	337K	31.55/0.8856	28.15/0.7680	27.32/0.7253	-/-	-/-
DWSR	×4	373K	31.39/0.8829	-/-	27.27/0.7246	25.27/0.7552	29.01/0.8855
VDSR	×4	665K	31.35/0.8830	28.02/0.7680	27.29/0.7267	25.18/0.7540	28.83/0.8870
LapSRN	×4	812K	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560	29.09/0.8900
WSDSR	×4	-	31.29/0.8821	27.59/0.7659	27.12/0.7215	25.11/0.7492	-/-
DNCL	×4	-	31.66/0.8871	28.23/0.7717	27.39/0.7282	25.36/0.7600	-/-
ASRN	×4	244K	31.65/0.8867	28.28/0.7733	27.34/0.7279	25.42/0.7616	29.59/0.8935
ASRN (SS)	×4	244K	31.73/0.8888	28.32/0.7748	27.38/0.7293	25.48/0.7648	29.60/0.8962

fusion layer and bottleneck layer, whose kernel size is 1×1. In the experiment and analysis parts, in order to verify the effectiveness of the introduced deep supervised learning (DSL) mechanism and progressive self-distillation (PSD) strategy, we trained two different versions of ASRN: **ASRN (SS)** and **ASRN (DSL)**. Among them, **ASRN (SS)** denotes the specifically trained model without DSL and PSD, **ASRN (DSL)** represents a model trained using only the DSL mechanism. Moreover, in order

Table 2: Quantitative comparisons on **BD** and **DN** modes. The best and the second-best results are highlighted with red and blue, respectively.

Mode	Methods	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
BD	Bicubic	28.78/0.8308	26.38/0.7271	26.33/0.6918	23.52/0.6862	25.46/0.8149
	SRCNN [7]	32.05/0.8944	28.80/0.8074	28.13/0.7736	25.70/0.7770	29.47/0.8924
	VDSR [8]	33.25/0.9150	29.46/0.8244	28.57/0.7893	26.61/0.8136	31.06/0.9234
	IRCNN_C [48]	33.17/0.9157	29.55/0.8271	28.49/0.7886	26.47/0.8081	31.13/0.9236
	IRCNN_G [48]	33.38/0.9182	29.63/0.8281	28.65/0.7922	26.77/0.8154	31.15/0.9245
	ASRN (Ours)	33.92/0.9229	30.04/0.8353	28.92/0.7997	27.40/0.8338	32.84/0.9379
DN	Bicubic	24.01/0.5369	22.87/0.4724	22.92/0.4449	21.63/0.4687	23.01/0.5381
	SRCNN [7]	25.01/0.6950	23.78.0.5898	23.76/0.5538	21.90/0.5737	23.75/0.7148
	VDSR [8]	25.20/0.7183	24.00/0.6112	24.00/0.5749	22.22/0.6096	24.20/0.7525
	IRCNN_G [48]	25.70/0.7379	24.45/0.6305	24.28/0.5900	22.90/0.6429	24.88/0.7765
	IRCNN_C [48]	27.48/0.7925	25.92/0.6932	25.55/0.6481	23.93/0.6950	26.07/0.8253
	SRMDNF [49]	27.74/0.8026	26.13/0.6974	25.64/0.6495	24.28/0.7092	26.72/0.8424
	ASRN (Ours)	28.16/0.8073	26.32/0.7006	25.78/0.6523	24.32/0.7098	27.32/0.8424

to compare with large SISR models, we trained some larger versions, which contain 15, 25, 35, and 45 MABs respectively.

Training Setting: Following previous works [22, 25, 23], we use RGB image as input and augment the image by flipping horizontally and vertically during training. The learning rate is initialized as 10^{-4} and the batch size is set to 16. In addition, $1,000$ iterations of back-propagation constitute an epoch and the learning rate halved every 200 epochs. Meanwhile, we set $\lambda = 0.1$ to balance the proportion of $\mathcal{L}_{PSD}$ and the model is updated with the Adam optimizer. ASRN is implemented by Pytorch of 0.4.0, Python of 3.6, and Ubuntu of 16.04. All codes run on a server with a CPU of Intel i7- i7-5930K, two RAMs of 16G, and two GPUs of Nvidia Titan Xp. The four GPUs can be accelerated by Nvidia CUDA of 9.0 and CuDNN of 7.5.

Degradation Modes: Different degradation modes will produce different LR images, which is a great challenge for SR models. In order to demonstrate the effectiveness of ASRN, we use three different degradation modes (**BI**, **BD**, and **DN**) to simulate LR images. **BI** is the most widely used degradation mode for LR images generation. It is essentially a bicubic downsampling operation that adopts the Matlab function *imresize* with the option of *Bicubic*. **BD** mode first uses a Gaussian kernel of size 7×7 with a standard deviation of 1.6 to blur the HR image and then downsamples the blurred image with scaling factor ×3. **DN** first downsamples the HR image with scaling factor ×3 and then applies Gaussian noise with $level = 30$. In order to fully verify the effectiveness of LegoNet, we evaluate our ASRN on all these three modes although most of the previous works were only tested on the **BI** mode.

4.3. Comparison with Lightweight SISR Models

Although plenty of SISR models have been proposed recently, we only focus on the lightweight models in this part since the proposed ASRN is a lightweight model.

To verify the effectiveness of ASRN, we compare it with more than 9 SISR methods, including Bicubic, SRCNN [7], SCN [50], FSRCNN [51], ESPCN [52], CNF [53], VDSR [8], DWSR [54], LapSRN [55], WSDSR [56], and DNCL [57]. All these models are classic lightweight models in the SISR field. Besides, all the SR results are evaluated with PSNR and SSIM on the Y channel of the transformed YCbCr space.

Results of BI Degradation Mode: BI mode is the most widely used degradation mode. In Table 1, we show the performance and parameters comparison between ASRN and other SISR models. According to the table, we can observe that: (i). Compared with tiny SR models (e.g., SRCNN, SCN, FSRCNN, and ESPCN), the performance of ASRN has been significantly improved; (ii). Compared with other lightweight SISR models (e.g., VDSR, DWSR, and LapSRN), ASRN can achieve better performance with fewer parameters. All these results fully demonstrate that ASRN is an efficient and lightweight SISR model, which achieves a better balance between model size and performance. It is worth noting that the performance of ASRN is slightly worse than ASRN (SS). This is because the introduced DSL strategy makes ASRN become a multi-task learning model, so it needs to share some resources for the intermediate results learning. Fortunately, this performance degradation is negligible. More analysis will be provided in Sec. 5.2.

In Fig. 9, we show the visual comparison on $\times 2$, $\times 3$, and $\times 4$, respectively. Obviously, compared with other methods, our ASRN and ASRN (SS) can reconstruct more accurate SR images. This is because the proposed multi-scale aggregation block can extract rich multi-scale features, which is beneficial for SR image reconstruction.

Results of BD and DN Degradation Modes: In Table 2, we show the SR results on the **BD** and **DN** degradation modes. According to the table, we can clearly observe that: (i). Compared with other SISR models, the performance of ASRN has been significantly improved; (ii). Compared with these methods, the performance improvement of ASRN on BD/DN is significantly higher than that on BI mode. This is because BD and DN degradation modes are more complex than BI, so more effective models are needed. Fortunately, with the help of the proposed MAB, ASRN can extract rich features for high-quality image reconstruction even the LR images are seriously degraded. This further validates the effectiveness of ASRN.

4.4. Elastic Image Reconstruction

The main contribution of this paper is the proposed adjustable network, which can achieve elastic image reconstruction with the help of DSL and PSD strategies. In this part, we exhibit the elastic image reconstruction results for different version of ASRN. In Fig. 10, we provide the trend graph of the performance changes of ASRN and ASRN (SS) as the number of MAB decreases. Specifically, we gradually remove MAB in the pre-trained models and then use the adjusted model to reconstruct SR images without retraining. According to the figure, we can clearly observe that when the number of MAB is gradually decreased, the performance of ASRN (SS) experiences a cliff-like decline, even worse than Bicubic. This severely limits the

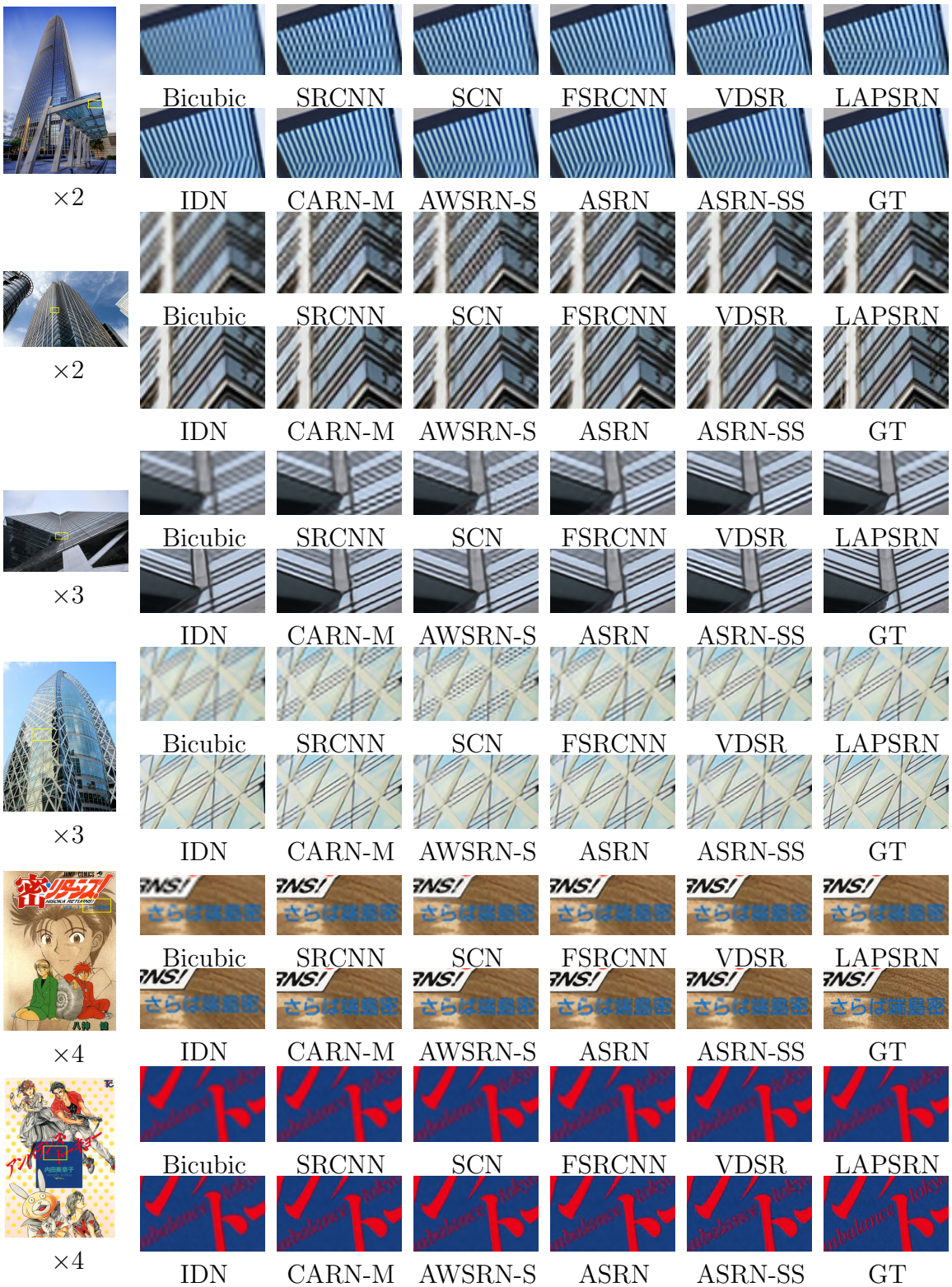


Fig. 9: Visual comparisons with other lightweight SISR models. Obviously, our ASRN and ASRN-SS achieve excellent reconstruction results.

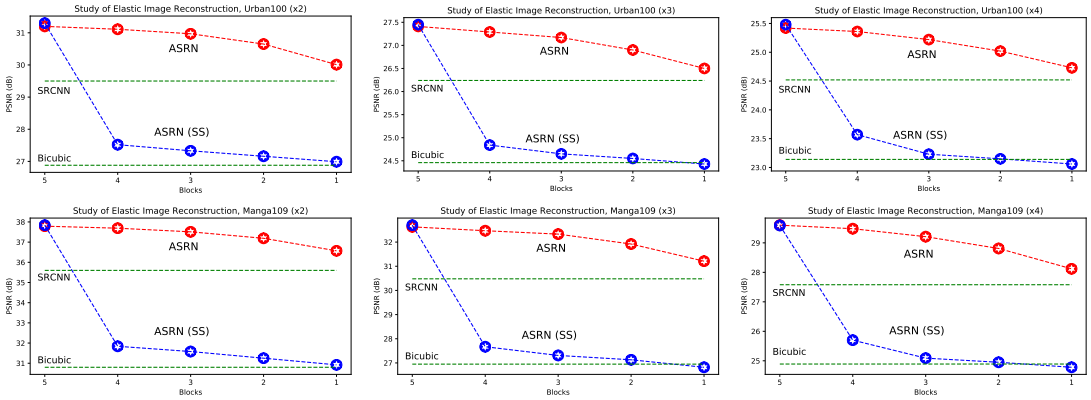


Fig. 10: Elastic image reconstruction results for different version of ASRN. Obviously, when the number of MAB is gradually decreased, the performance of ASRN (SS) experiences a cliff-like decline while the performance of ASRN is still stable.

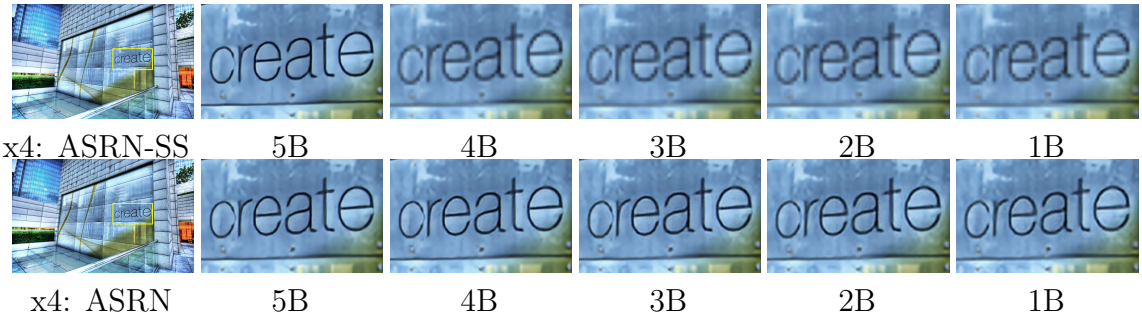


Fig. 11: Visual comparisons between ASRN (SS) and ASRN under different numbers of MAB. Obviously, as the model depth/size changes, the quality of SR images reconstructed by ASRN (SS) will be severely compromised, while ASRN can still reconstruct high-quality SR images.

application scenarios of the model. On the contrary, the performance of ASRN is stable and the degradation is acceptable. This means that the proposed model can easily adjust the model size according to actual requirements. Meanwhile, we also provide the SR images reconstructed by ASRN (SS) and ASRN with different numbers of MABs in Fig. 11. Obviously, as the model depth/size changes, the quality of SR images reconstructed by ASRN (SS) severely compromised. However, the SR images reconstructed by ASRN still show excellent visual effects. This further verifies the feasibility of elastic image reconstruction with the help of DSL and PSD strategies.

5. Investigations

5.1. Effectiveness of Multi-scale Aggregation Block (MAB)

Multi-scale Aggregation Block (MAB) is the basic component of ASRN, which is the key to building a modular model. In this section, we provide a series of experiments to illustrate the effectiveness of MAB.

Table 3: Ablation study of the importance of each component in MAB.

Case Index		1	2	3	4	5	6	7
MAB	Fusion Lay	×	√	√	√	√	√	√
	Residual-like connection	√	×	√	√	√	√	√
	Channel Shuffle	√	√	×	√	√	√	√
	Residual learning	√	√	√	×	√	√	√
MAB Number		5	5	5	5	5	5	10
Channel Number		64	64	64	64	64	96	96
PSNR (Urban100, x3)		26.85	27.08	27.13	26.33	27.19	27.45	**27.92**

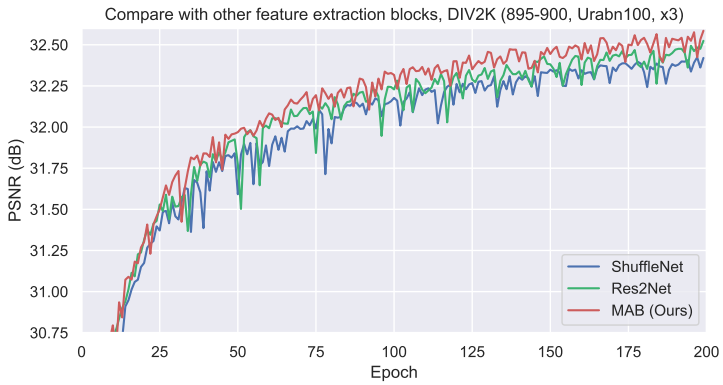


Fig. 12: Compare with the core block in ShuffleNet and Res2Net.

(1) MAB is essentially a multi-scale feature extraction block, which integrates the feature fusion, channel split, residual-like connection, channel shuffle, and residual learning mechanisms. Among them, channel split is the prerequisite for all operations and acts as the key operation for constructing a lightweight model. For other mechanisms, we provide a series of ablation studies in Table 3 to investigate their effectiveness. According to the table, we can clearly observe that (a). When the residual learning is removed, the performance of the model will be greatly reduced (case 4). This is because residual learning can accelerate model convergence and facilitate the information flow thus improving the model performance; (b). Removing any mechanism, the model performance will be degraded (cases 1,2,3); (c). Increasing the number of channels or MABs can effectively improve the model performance (cases 6,7). All these experiments prove the effectiveness and necessity of the introduced mechanisms, which together constitute the efficient MAB.

(2) MAB can be considered as an improved version of ResBlock [18], which is inspired by ShuffleNet [39] and Res2Net [40]. To further demonstrate the effectiveness of MAB, we compare MAB with the core feature extraction block in ShuffleNet and Res2Net. For a fair comparison, we use the same backbone as the infrastructure of these models, and all these blocks have a similar number of parameters. In Fig. 12,

Table 4: Study of the deep supervised learning mechanism. Best results are highlighted with red.

Scale	Blocks	Parameters	Method	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
x3	5B	248K (100%)	ASRN (SS)	33.95/0.9238	30.01/0.8359	28.89/0.8000	27.45/0.8366	32.70/0.9383
			ASRN (DSL)	33.81/0.9221	29.92/0.8346	28.81/0.7985	27.34/0.8331	32.55/0.9361
	4B	209K (84%)	ASRN (SS)	30.84/0.8942	28.00/0.7929	27.46/0.7555	24.84/0.7582	27.67/ 0.8789
			ASRN (DSL)	33.71/0.9211	29.86/0.8834	28.79/0.7979	27.24/0.8306	32.42/0.9341
	3B	170K (69%)	ASRN (SS)	30.64/0.8768	27.83/0.7852	27.35/0.7482	24.65/0.7472	27.31/0.8682
			ASRN (DSL)	33.65/0.9198	29.81/0.8319	28.74/0.7962	27.06/0.8258	32.19/0.9312
	2B	131K (53%)	ASRN (SS)	30.50/0.8726	27.72/0.7807	27.27/0.7437	24.55/0.7415	27.13/0.8631
			ASRN (DSL)	33.40/0.9175	29.71/0.8301	28.66/0.7943	26.81/0.8189	31.81/0.9274
	1B	92k (37%)	ASRN (SS)	30.28/0.8678	27.56/0.7749	27.17/0.7389	24.43/0.7354	26.82/0.8551
			ASRN (DSL)	33.04/0.9134	29.51/0.8265	28.50/0.7908	26.48/0.8099	31.12/0.9213
x4	5B	244K (100%)	ASRN (SS)	31.73/0.8888	28.32/0.7748	27.38/0.7293	25.48/0.7648	29.59/0.8962
			ASRN (DSL)	31.55/0.8855	28.25/0.7725	27.32/0.7270	25.33/0.7580	29.47/0.8915
	4B	205K (84%)	ASRN (SS)	29.04/0.8325	26.57/0.7259	26.28/0.6863	23.57/0.6854	25.70/0.8178
			ASRN (DSL)	31.45/0.8831	28.21/0.7713	27.29/0.7260	25.27/0.7558	29.34/0.8888
	3B	166K (68%)	ASRN (SS)	28.52/0.8170	26.19/0.7109	26.02/0.6730	23.23/0.6660	25.09/0.7969
			ASRN (DSL)	31.36/0.8819	28.13/0.7695	27.24/0.7246	25.16/0.7517	29.12/0.8849
	2B	127K (52%)	ASRN (SS)	28.41/0.8131	26.11/0.7071	25.97/0.6697	23.15/0.6611	24.95/0.7916
			ASRN (DSL)	31.13/0.8771	27.99/0.7662	27.15/0.7221	24.95/0.7439	28.68/0.8771
	1B	88k (36%)	ASRN (SS)	28.22/0.8068	25.98/0.7014	25.88/0.6674	23.06/0.6553	24.78/0.7853
			ASRN (DSL)	30.76/0.8698	27.76/0.7606	27.00/0.7177	24.68/0.7329	28.11/0.8671

we show the performance of these feature extraction blocks during training. Obviously, our model is more stable and achieves the best results. The is because the introduced residual-like connection can extract image features with different scales and the channel shuffle can overcome the side effects brought by the channel split. Therefore, MAB shows stronger feature extraction ability and stability.

5.2. Effectiveness of Deep Supervised Learning (DSL)

As mentioned in Section 4.2, ASRN (SS) denotes the specifically trained model without DSL and PSD mechanisms, and ASRN (DSL) is the improved version of ASRN (SS) with DSL. In Table 4, we provide the elastic image reconstruction results of the model on the adjusted versions. Among them, the results of '5B' and '1B-4B' represent the performance of the complete model and the adjusted version, respectively. In other words, '1B-4B' indicates the number of MABs remaining after some MABs in the pre-trained model are removed. According to the table, we can clearly observe that the performance of ASRN (SS) is severely degraded when the number of MABs changed. Contrastly, with the help of the DSL mechanism, the performance of ASRN (DSL) still achieves good results. Specifically, when the number of MABs reduced from 5 to 1, the performance degradation of ASRN (DSL) is slight and acceptable (less 1.4dB) while the performance degradation of ASRN (SS) is significant (4.8dB-7dB). This fully demonstrates that the introduced DSL mechanism is reasonable and effective, which makes elastic image reconstruction feasible. However, it is worth noting that the performance of the complete model (5B) of ASRN (DSL) is slightly worse (0.12dB-0.21dB) than ASRN (SS). This is because the introduced DSL strategy makes ASRN became a multi-task learning model. Since multi-task learning

Table 5: Study of the progressive self-distillation mechanism. Best results are highlighted with red.

Scale	Blocks	Parameters	Method	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
x3	5B	248K (100%)	ASRN (DSL)	33.81/0.9221	29.92/0.8346	28.81/0.7985	27.34/0.8331	32.55/0.9361
			ASRN	33.84/0.9223	29.97/0.8348	28.86/0.7990	27.41/0.8342	32.63/0.9364
	4B	209K (84%)	ASRN (DSL)	33.71/0.9211	29.86/0.8834	28.79/0.7979	27.24/0.8306	32.42/0.9341
			ASRN	33.75/0.9213	29.91/0.8337	28.82/0.7983	27.29/0.8317	32.47/0.9346
	3B	170K (69%)	ASRN (DSL)	33.65/0.9198	29.81/0.8319	28.74/0.7962	27.06/0.8258	32.19/0.9312
			ASRN	33.67/0.9205	29.86/0.8328	28.78/0.7973	27.17/0.8285	32.33/0.9329
	2B	131K (53%)	ASRN (DSL)	33.40/0.9175	29.71/0.8301	28.66/0.7943	26.81/0.8189	31.81/0.9274
			ASRN	33.43/0.9179	29.74/0.8306	28.69/0.7949	f26.90/0.8216	31.92/0.9286
	1B	92k (37%)	ASRN (DSL)	33.04/0.9134	29.51/0.8265	28.50/0.7908	26.48/0.8099	31.12/0.9208
			ASRN	33.06/0.9136	29.53/0.8267	28.53/0.7910	26.50/0.8107	31.21/0.9213
x4	5B	244K (100%)	ASRN (DSL)	31.55/0.8855	28.25/0.7725	27.32/0.7270	25.33/0.7580	29.47/0.8915
			ASRN	31.65/0.8867	28.28/0.7733	27.34/0.7279	25.42/0.7616	29.60/0.8935
	4B	205K (84%)	ASRN (DSL)	31.45/0.8831	28.21/0.7713	27.29/0.7260	25.27/0.7558	29.34/0.8888
			ASRN	31.57/0.8855	28.23/0.7724	27.31/0.7270	25.36/0.7593	29.48/0.8912
	3B	166K (68%)	ASRN (DSL)	31.36/0.8819	28.13/0.7695	27.24/0.7246	25.16/0.7517	29.12/0.8849
			ASRN	31.41/0.8824	28.15/0.7701	27.26/0.7253	25.22/0.7540	29.21/0.8858
	2B	127K (52%)	ASRN (DSL)	31.13/0.8771	27.99/0.7662	27.15/0.7221	24.95/0.7439	28.68/0.8771
			ASRN	31.21/0.8782	28.02/0.7669	27.17/0.7228	25.02/0.7465	28.81/0.8787
	1B	88k (36%)	ASRN (DSL)	30.76/0.8698	27.76/0.7606	27.00/0.7177	24.68/0.7329	28.11/0.8671
			ASRN	30.79/0.8701	27.77/0.7611	27.03/0.7180	24.73/0.7356	28.12/0.8674

needs to coordinate the training of multiple subnets at the same time, which makes its performance is slightly lower than that of the single-task model. Fortunately, the gap is small and acceptable. Meanwhile, we introduce the PSD strategy to improve the performance of elastic reconstructed results thus further reducing this gap.

5.3. Effectiveness of Progressive Self-Distillation (PSD)

In order to further improve the model performance and reduce the negative impact of multi-task learning gaps, we propos the PSD strategy. This strategy aims to use the deep sub-models in ASRN to guide the shallow sub-model in ASRN, which is essentially a learning task with multiple teachers and studies. In this part, we provide more experiments and ablation studies to prove its effectiveness.

As defined in Sec. 5.2, ASRN (DSL) is the model with DSL but not PSD while ASRN is the final model with DSL and PSD. In Table 5, we provide the performance comparison between ASRN (DSL) and ASRN. According to the table, we can clearly observe that with the help of PSD, the model performance can be further improved whether the complete model (5B) or the adjusted models (1-4B). This means that the performance gap between ASRN and the model specially trained for different numbers of MABs will be further reduced. This also proves the effectiveness of the PSD strategy, which can further improve the results of elastic image reconstruction.

5.4. Comparison with Large SISR Models

Increasing the depth of the model is the easiest way to improve model performance. Therefore, various large-size models have been proposed in recent years, e.g., MADNet [58], DRCN [59], SRMDNF [49], CARN [19], DCAE [60], MSRN [25],

Table 6: Quantitative comparisons between our ASRN and other SISR models.

Model	Scale	Param.	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
DRCN	×3	1.8M	33.85/0.9215	29.89/0.8317	28.81/0.7954	27.16/0.8311	32.31/0.9328
MADNet	×3	0.9M	34.16/0.9215	30.21/0.8398	28.98/0.8023	27.77/0.8439	-/-
SRMDNF	×3	1.5M	34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
CARN	×3	1.6M	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440
EDSR-base	×3	1.6M	34.37/0.9270	30.28/0.8417	29.09/0.8052	28.15/0.8527	33.45/0.9439
DCAE	×3	3.0M	34.12/0.9251	30.02/0.8353	28.98/0.8016	27.59/0.8386	-/-
MSRN	×3	6.3M	34.48/0.9276	30.40/0.8436	29.13/0.8061	28.31/0.8560	33.56/0.9451
SeaNet	×3	7.5M	34.55/0.9282	30.42/0.8444	29.17/0.8071	28.50/0.8594	33.73/0.9463
CRN	×3	9.5M	34.60/0.9286	30.48/0.8455	29.20/0.8081	28.62/0.8620	-/-
CFSRCNN	×4	1.2M	34.24/0.8256	30.27/0.8410	29.03/0.8035	28.04/0.8496	-/-
ACNet	×4	1.3M	34.14/0.9247	30.19/0.8398	28.98/0.8023	27.97/0.8482	-/-
ASRN(25B)	×3	1.0M	34.34/0.9268	30.28/0.8418	29.07/0.8047	28.11/0.8515	33.41/0.9438
ASRN(45B)	×3	**1.8M**	**34.51/0.9280**	**30.40/0.8436**	**29.15/0.8068**	**28.44/0.8580**	**33.67/0.9458**
DRCN	×4	1.8M	31.56/0.8810	28.15/0.7627	27.24/0.7150	25.15/0.7530	28.98/0.8816
MADNet	×4	1M	31.95/0.8917	28.44/0.7780	27.47/0.7327	25.76/0.7746	-/-
SRMDNF	×4	1.6M	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
CARN	×4	1.6M	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
EDSR-base	×4	1.5M	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
DCAE	×4	3.0M	31.72/0.8884	28.27/0.7733	27.40/0.7288	25.55/0.7660	-/-
MSRN	×4	6.3M	32.25/0.8958	28.63/0.7833	27.61/0.7377	26.20/0.7905	30.57/0.9103
SeaNet	×4	7.4M	32.33/0.8970	28.72/0.7855	27.65/0.7388	26.32/0.7942	30.74/0.9129
CRN	×4	9.5M	32.34/0.8971	28.74/0.7855	27.66/0.7395	26.44/0.7967	-/-
CFSRCNN	×4	1.2M	32.06/0.8920	28.57/0.7800	27.53/0.7333	26.03/0.7824	-/-
ACNet	×4	1.3M	31.83/0.8903	28.46/0.7788	27.48/0.7326	25.93/0.7798	-/-
ASRN(25B)	×4	1.0M	32.05/0.8931	28.54/0.7811	27.55/0.7353	25.99/0.7828	30.35/0.9066
ASRN(45B)	×4	**1.8M**	**32.30/0.8966**	**28.68/0.7843**	**27.62/0.7377**	**26.29/0.7938**	**30.61/0.9111**

RCAN [23], and EDSR [22], SeaNet [61], CRN [62], CFSRCNN [63], and ACNet [16]. Although these models achieve excellent performance, they are also accompanied by a large number of parameters that are 60 times or even 100 times of our ASRN. For a fair comparison, we trained some extended versions, including ASRN (25B) and ASRN (45B). In Table 6, we show the performance comparison between ASRN (25B), ASRN (45B), and other SISR models. According to the table, we can clearly observe that ASRN can obtain competitive results with fewer parameters. Meanwhile, ASRN (45B) can obtain very close or better results than MSRN, SeaNet, and CRN with only 1/4 or 1/5 parameters. This fully proves the excellence of the proposed ASRN. Moreover, we provide a more intuitive comparison in Fig. 13. In Fig. 13, we show the parameters and performance of these models in the form of dot plots. Since some models have a huge number of parameters, it is difficult to display them well on one figure, so we display them on two figures according to the parameter level of the model. In these two figures, red dots represent different versions of ASRN and blue dots denote other SISR models. It is worth noting that, in these two figures, the peripheral dots represent the model which achieves a better balance between model size and performance. In other words, in the figure, the closer to the upper left corner, the better the model. Obviously, even compared with very large SISR models (e.g., RCAN, RDN, and EDSR), our ASRN can still achieve competitive results. Therefore,

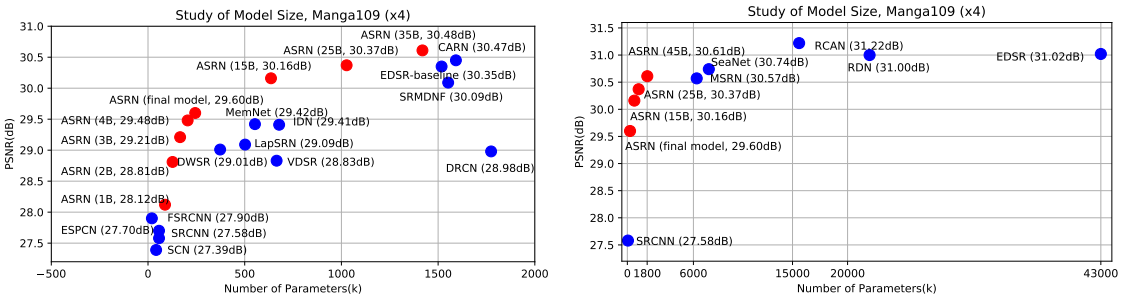


Fig. 13: Study of model size and performance. Red dots denote our proposed ASRNs.

Table 7: Comparison of the number of model parameters, execution time, and FLOPs (1024 * 1024).

Method	VDSR	CARN-M	CFSRCNN	LESRCNN	RDN	ASRN (Ours)	ASRN (25B, Ours)
Params	665K	412K	1200K	516K	22M	**244K**	1000K
Time (s)	0.212	0.033	0.029	0.023	0.914	**0.013**	0.025
FLOPs	10.9G	2.5G	11.08G	3.08G	130.8G	**2.1G**	8.3G

we can draw a conclusion that the proposed ASRN is a lightweight and efficient model which achieved a good trade-off between the size and performance of the model.

5.5. Model Complexity Studies

To further verify the validity of the model, we provide the running time and FLOPs comparison of several classic SISR models in Table 7. All these models execute on the same device for fair comparison. From this table, we can clearly observe that our ASRN has the shortest running time and lowest FLOPs. This is due to the fact that ASRN has few parameters and does not introduce complex operations such as the attention mechanism. This further verifies the effectiveness of ASRN.

6. Discussion

Contribution of the method: Different from blindly pursuing model performance, we start from a new application perspective and propose an adjustable super-resolution network. The main advantages of ASRN are as follow: (1) The proposed model can achieve one model, one training, flexible deployment on different sizes of platforms; This is a new technical attempt in the field of image restoration; (2) The proposed method is a task-agnostic and model-agnostic method, which can suitable for other image restoration tasks (e.g., mage denoising, image deblurring, and image dehazing) and other deep modular networks; (3) The proposed PSD strategy is a new attempt of the knowledge distillation strategy in the field of image restoration.

Limitations of the method: Although ASRN has achieved outstanding results, as a new technical attempt, it still has some shortcomings: (1) In order to achieve elastic image reconstruction, we introduce the DSL mechanism during training. This mechanism makes ASRN become a multi-task learning model, which limit the model

performance. Although the introduced PSD strategy can alleviate this problem, the model performance still cannot surpass the specially trained models. (2) The proposed model can flexibly adjust the depth/size of the model to achieve elastic image reconstruction. However, we should notice that this method can only adjust the large model to a small one. Although this is a huge breakthrough, if a small model can be expanded to a large one without training, it will have more application prospects. These will be the focus of our future works.

7. Conclusion

In this paper, we proposed an Adjustable Super-Resolution Network (ASRN), which can flexibly adjust the depth/size of the model to adapt to different needs without retraining. In order to achieve elastic image reconstruction, a powerful Multi-scale Aggregation Blocks (MAB) was proposed to build the modular network, and the Deep Supervised Learning (DSL) mechanism was introduced during the training process to maximize the performance of each sub-network in ASRN. Moreover, we proposed a novel Progressive Self-Distillation (PSD) strategy to further improve the intermediate results of the model to alleviate the negative effects of multi-task learning. It is worth noting that this method is a task-agnostic method, which is suitable for other image restoration tasks, such as image denoising, image deblurring, and image dehazing. In future work, we will further verify the effectiveness of the method on other image restoration tasks.

References

[1] Y. Huang, W. Wang, L. Wang, Video super-resolution via bidirectional recurrent convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2017) 1015–1028.

[2] T. Xue, B. Chen, J. Wu, D. Wei, W. T. Freeman, Video enhancement with task-oriented flow, International Journal of Computer Vision (2018) 1–20.

[3] O. Schmitt, J. Modersitzki, S. Heldmann, S. Wirtz, B. Fischer, Image registration of sectioned brains, International Journal of Computer Vision 73 (2006) 5–39.

[4] L. Sun, Z. Fan, Y. Huang, X. Ding, J. Paisley, Compressed sensing mri using a recursive dilated network, in: AAAI, 2018.

[5] B. Shuai, Z. Zuo, B. Wang, G. Wang, Scene segmentation with dag-recurrent neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (6) (2017) 1480–1493.

[6] N. Zeng, H. Li, Z. Wang, W. Liu, S. Liu, F. E. Alsaadi, X. Liu, Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip, Neurocomputing 425 (2021) 173–180.

[7] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2) (2015) 295–307.

[8] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: CVPR, 2016, pp. 1646–1654.

[9] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: CVPR, 2017, pp. 2790–2798.

[10] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, L. Zhang, Second-order attention network for single image super-resolution, in: CVPR, 2019, pp. 11065–11074.

[11] B. Liu, D. Ait-Boudaoud, Effective image super resolution via hierarchical convolutional neural network, Neurocomputing 374 (2020) 109–116.

[12] J. Qin, Y. Huang, W. Wen, Multi-scale feature fusion residual network for single image super-resolution, Neurocomputing 379 (2020) 334–342.

[13] C. Tian, R. Zhuge, Z. Wu, Y. Xu, W. Zuo, C. Chen, C.-W. Lin, Lightweight image super-resolution with enhanced cnn, Knowledge-Based Systems 205 (2020) 106235.

[14] Y. Dun, Z. Da, S. Yang, X. Qian, Image super-resolution based on residually dense distilled attention network, Neurocomputing 443 (2021) 47–57.

[15] W. Li, J. Li, J. Li, Z. Huang, D. Zhou, A lightweight multi-scale channel attention network for image super-resolution, Neurocomputing 456 (2021) 327–337.

[16] C. Tian, Y. Xu, W. Zuo, C.-W. Lin, D. Zhang, Asymmetric cnn for image superresolution, IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[17] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: CVPR, 2017, pp. 4681–4690.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.

[19] N. Ahn, B. Kang, K.-A. Sohn, Fast, accurate, and lightweight super-resolution with cascading residual network, in: ECCV, 2018, pp. 252–268.

[20] Z. Hui, X. Wang, X. Gao, Fast and accurate single image super-resolution via information distillation network, in: CVPR, 2018, pp. 723–731.

[21] Z. Hui, X. Gao, Y. Yang, X. Wang, Lightweight image super-resolution with information multi-distillation network, in: ACMMM, 2019, pp. 2024–2032.

[22] B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, Enhanced deep residual networks for single image super-resolution, in: CVPR Workshop, 2017, pp. 1132–1140.

[23] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: ECCV, 2018.

[24] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: CVPR, 2018.

[25] J. Li, F. Fang, K. Mei, G. Zhang, Multi-scale residual network for image super-resolution, in: ECCV, 2018.

[26] X. Jin, Q. Xiong, C. Xiong, Z. Li, Z. Gao, Single image super-resolution with multi-level feature fusion recursive network, Neurocomputing 370 (2019) 166–173.

[27] Z. Hui, X. Gao, X. Wang, Lightweight image super-resolution with feature enhancement residual network, Neurocomputing 404 (2020) 50–60.

[28] Z. Wang, G. Gao, J. Li, Y. Yu, H. Lu, Lightweight image super-resolution with multi-scale feature interaction network, in: ICME, 2021, pp. 1–6.

[29] G. Gao, W. Li, J. Li, F. Wu, H. Lu, Y. Yu, Feature distillation interaction weighting network for lightweight image super-resolution, arXiv preprint arXiv:2112.08655.

[30] Z. Wang, J. Chen, S. C. Hoi, Deep learning for image super-resolution: A survey, arXiv preprint arXiv:1902.06068.

[31] J. Li, Z. Pei, T. Zeng, From beginner to master: A survey for deep learning-based single-image super-resolution, arXiv preprint arXiv:2109.14335.

[32] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Artificial intelligence and statistics, 2015, pp. 562–570.

[33] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531.

[34] Y. Zhang, T. Xiang, T. M. Hospedales, H. Lu, Deep mutual learning, in: CVPR, 2018, pp. 4320–4328.

[35] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, arXiv preprint arXiv:1412.6550.

[36] N. Komodakis, S. Zagoruyko, Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer, in: ICLR, 2017.

[37] W. Lee, J. Lee, D. Kim, B. Ham, Learning with privileged information for efficient image super-resolution, in: ECCV, 2020, pp. 465–482.

[38] M. Hong, Y. Xie, C. Li, Y. Qu, Distilling image dehazing with heterogeneous task imitation, in: CVPR, 2020, pp. 3462–3471.

[39] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: CVPR, 2018, pp. 6848–6856.

[40] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, P. H. Torr, Res2net: A new multi-scale backbone architecture, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[41] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, J. Lin, Distilling task-specific knowledge from bert into simple neural networks, arXiv preprint arXiv:1903.12136.

[42] E. Agustsson, R. Timofte, Ntire 2017 challenge on single image super-resolution: Dataset and study, in: CVPR Workshop, 2017, pp. 1110–1121.

[43] M. Bevilacqua, A. Roumy, C. Guillemot, M. L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: BMVC, 2012.

[44] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: ICCS, 2010, pp. 711–730.

[45] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 898–916.

[46] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: CVPR, 2015, pp. 5197–5206.

[47] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, K. Aizawa, Sketch-based manga retrieval using manga109 dataset, Multimedia Tools and Applications 76 (20) (2017) 21811–21838.

[48] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep cnn denoiser prior for image restoration, in: CVPR, 2017, pp. 3929–3938.

[49] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: CVPR, 2018, pp. 3262–3271.

[50] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: ICCV, 2015, pp. 370–378.

[51] C. Dong, C. C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: ECCV, 2016, pp. 391–407.

[52] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: CVPR, 2016, pp. 1874–1883.

[53] H. Ren, M. El-Khamy, J. Lee, Image super resolution based on fusing multiple convolution neural networks, CVPR Workshops (2017) 1050–1057.

[54] T. Guo, H. Seyed Mousavi, T. Huu Vu, V. Monga, Deep wavelet prediction for image super-resolution, in: CVPR Workshops, 2017, pp. 104–113.

[55] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep laplacian pyramid networks for fast and accurate super-resolution, in: CVPR, 2017, pp. 624–632.

[56] C. Cruz, R. Mehta, V. Katkovnik, K. O. Egiazarian, Single image super-resolution based on wiener filter in similarity domain, IEEE Transactions on Image Processing 27 (3) (2017) 1376–1389.

[57] C. Xie, W. Zeng, X. Lu, Fast single-image super-resolution via deep network with component learning, IEEE Transactions on Circuits and Systems for Video Technology 29 (12) (2018) 3473–3486.

[58] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, X. Luo, Madnet: A fast and lightweight network for single-image super resolution, IEEE transactions on cybernetics 51 (3) (2021) 1443–1453.

[59] J. Kim, J. Kwon Lee, K. Mu Lee, Deeply-recursive convolutional network for image super-resolution, in: CVPR, 2016, pp. 1637–1645.

[60] Y. Zhou, Y. Zhang, X. Xie, S.-Y. Kung, Image super-resolution based on dense convolutional auto-encoder blocks, Neurocomputing 423 (2021) 98–109.

[61] F. Fang, J. Li, T. Zeng, Soft-edge assisted network for single image super-resolution, IEEE Transactions on Image Processing 29 (2020) 4656–4668.

[62] R. Lan, L. Sun, Z. Liu, H. Lu, Z. Su, C. Pang, X. Luo, Cascading and enhanced residual networks for accurate single-image super-resolution, IEEE Transactions on Cybernetics 51 (1) (2021) 115–125.

[63] C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, C.-W. Lin, Coarse-to-fine cnn for image super-resolution, IEEE Transactions on Multimedia 23 (2020) 1489–1502.