# A Single Residual Network with ESA Modules and Distillation

Yucong Wang  Minjie Cai
College of Computer Science and Electronic Engineering, Hunan University
1401121556@qq.com, caiminjie@hnu.edu.cn

## Abstract

*Although there are many methods based on deep learning that have superior performance on single image super-resolution (SISR), it is difficult to run in real time on devices with limited computing power. Some recent studies have found that simply relying on reducing parameters or reducing the theoretical FLOPs of the model does not speed up the inference time of the network in a practical sense. Actual speed on the device is probably a better measure than FLOPs. In this work, we propose a new single residual network (SRN). On the one hand, we try to introduce and optimize an attention mechanism module to improve the performance of the network with a relatively small speed loss. On the other hand, we find that residuals in residual blocks do not have a positive impact on networks with adjusted ESA. Therefore, the residual of the network residual block is removed, which not only improves the speed of the network, but also improves the performance of the network. Finally, we reduced the number of channels and the number of residual blocks of the classic model EDSR, and removed the last convolution before the long residual. We set this tuned EDSR as the teacher model and our newly proposed SRN as the student model. Under the joint effect of the original loss and the distillation loss, the performance of the network can be improved without losing the inference time. Combining the above strategies, our proposed model runs much faster than similarly performing models. As an example, we built a Fast and Efficient Network (SRN) and its small version SRN-S, which run 30%-37% faster than the state-of-the-art EISR model: a paper champion RLFN. Furthermore, the shallow version of SRN-S achieves the second-shortest inference time as well as the second-smallest number of activations in the NTIRE2023 challenge. Code will be available at https://github.com/wnxbwyc/SRN.*

## 1. Introduction

Single image super-resolution reconstruction is a relatively low-level task in computer vision. Its goal is to reconstruct a given degraded low-resolution image into a sharp high-resolution image as much as possible. There is a lot of detail loss in low-resolution images, so reconstruction is beyond imagination. However, in recent years, with the rapid development of deep learning, the reconstruction effect of neural network is amazing, and it greatly surpasses the traditional scheme such as A+ [27]. First, the convolutional neural network [5] [6] [1] [2] [24] [27] [28] [29] [30]produced superior results, and then the transformer-based model [11] [12] [31]shined. Unfortunately, the cutting-edge high-performance indicators often require a lot of computing power, and it is difficult to run in real time on devices with limited resources. In order to complete the lightweight model, it is necessary to reduce performance expectations and increase speed.

It is precisely because of the need to run in real time on resource-constrained devices that many works target efficient image super-resolution. Some early works used recurrent neural networks to save the amount of model parametersp [7] [13], but in fact the requirements for computing power still increase due to the increase in the number of recursions. Some works try to reduce the FLOPs of the model, such as depthwise separable convolution [32] and the blueprint separable residual network [33] that won the NTIRE2022 model complexity track. Although it tries to compress the model to the extreme, compared with the conventional model, the actual running speed does not decrease due to the reduction of FLOPs. This fully demonstrates that the number of parameters and FLOPs are widely used in theoretical analysis, and models with low FLOPs do not always run fast. That is, if we try to lower the theoretical metric, we will defeat the original goal of optimizing the runtime. Currently, it is especially important to develop a model with faster inference speed rather than low flops or parameters.

Another classic example of optimizing FLOPs is using feature fusion. Feature fusion usually uses 1x1 convolution, which is characterized by low FLOPs and high runtime. There are many efficient super-resolution models that use this strategy to improve performance. It usually adopts multi-level connections to facilitate the exchange of information between different locations. At the same time, it

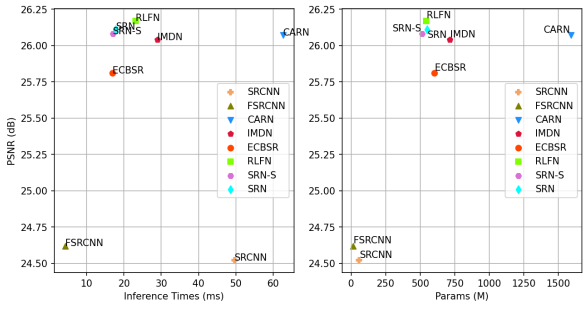


Figure 1. Illustration of PSNR, inference time and parameter numbers of different SISR models on the Urban100 dataset for 4x SR.

also reduces the difficulty of deep network training. For example, IMDN [8] proposes to build an information multi-distillation block IMDB, whose main idea is to combine channel split and feature fusion. IMDN also uses global fusion, which makes IMDN have a relatively strong representation ability. This model won the first place in the inference time and parameter quantity on the AIM2019 ESR track [36]. RFDN [34], the champion of AIM2020 ESR track [37], made some optimizations on the basis of IMDN to further reduce the complexity of the model. However, the basic architecture hasn't changed much. Multiple related feature maps stay in memory until aggregated, which accomplishes huge memory consumption. The 1x1 convolution and frequent memory access introduced by the aggregation will cause a decrease in the inference speed. As the second place in the ESR track of NTIRE2022 [21], FMEN [35] pointed out that serial networks should be used instead of feature aggregation when designing network backbones. Coincidentally, RLFN [10], the first place in the ESR track of NTIRE2022, also adopted this serial design when designing the network.

EDSR [1] is a standard serial design model. It removes unnecessary BN parts and tries to expand the feature channels of the model, which enhances the expressiveness of the model. We try to further optimize the EDSR based model. In recent years, models based on attention mechanism have emerged continuously. Enhanced Spatial Attention [3] is one of the better attention mechanisms. We try to introduce a separate ESA into the improved lightweight version of EDSR. In order to reduce the speed impact of introducing the ESA, we further simplify the ESA. Considering that the original intention of the residual network is to solve the problem that the deep network is difficult to learn, for a lightweight shallow network, the speed improvement brought by not introducing the residual is more attractive than the performance reduction. We tentatively removed the residuals and found that the model introducing a simplified version of ESA achieved an unexpected performance improvement after removing the residuals. This means that the original residuals limit the representation ability of ESA. Finally, we try to use the model distillation strategy to make the student model learn from the teacher model, so as to improve the performance of the student model without compromising the speed. It can be seen from Fig. 1 and Table 1 that our SRN has significantly better performance than the ECBSR [9] with a similar speed and adopts structural reparameterization technique. Compared with the NTIRE2022 ESR [21]champion model RLFN [10], the running speed is 30% to 37% faster under the premise of similar performance. Please note that SRN-S and SRN in the X2 part of the table are not trained with a distillation strategy, while the X4 part is trained with a distillation strategy. Our contributions can be summarized as follows:

1. According to our analysis, although the introduction of the Enhanced Spatial Attention has brought performance improvements, it has also led to a corresponding decrease in speed. In order to reduce the loss of speed, we optimized the Enhanced Spatial Attention module.

2. Through comparative experiments, we found that the model with a simplified version of Enhanced Spatial Attention can not only speed up the model after removing the residual, but also free the model from the constraints imposed by the residual, thus improving performance. This is where our model SRN comes from.

3. We try to gain benefits in model distillation. By designing a suitable teacher network, it can be found that the student model can benefit from the output of the teacher model through comparative experiments.

## 2. Related Work

### 2.1. Efficient Image Super-Resolution

SRCNN [5] is the first to apply deep learning algorithms to the field of single image super-resolution. It consists of only three layers of neural networks and performs bicubic interpolation on the input before passing it through the network. But performing inference directly on large-resolution images will undoubtedly slow down the model. The improved version FSRCNN [6] of SRCNN solves this problem very well. It achieves a large speedup by moving upsampling to the end of the model, and shows that processing at low resolutions is not only faster but may result in better accuracy. DRCN [7] introduces a recurrent neural network to reduce the number of parameters, but the actual amount of computation has not been reduced due to the reduction of parameters. IMDN [8] proposes a lightweight information multi-distillation network by constructing cascaded information multi-distillation blocks, using the information

distillation mechanism (IDM) to gradually extract hierarchical features. DRN [24] fuses images of different sizes to achieve lightweight models. ECBSR [9] adopts a novel structural reparameterization technique method to sacrifice the training time of the model to obtain better performance. RLFN [10] designs a novel residual structure, which enables it to achieve an excellent balance between speed and performance. RLFN won the NTIRE2022 efficient super-resolution champion [21]. Besides, [26] points out that the number of iterations and patch size can also significantly improve model performance.

### 2.2. Attention Mechanism in SR

Other computer vision tasks also have some inspiration for super-resolution tasks. SENet [38] has achieved excellent results in the field of image classification by applying the attention mechanism. It automatically obtains the importance of each feature channel by learning, promotes useful features and suppresses less useful features. Since SENet was proposed, various attention mechanisms have been applied in super-resolution. RCAN [2] applies the channel attention mechanism in the residual block. RNAN [39] uses both residual non-local attention blocks and residual local attention blocks to adaptively adjust hierarchical features. SAN [40] is a second-order attention network that can use more powerful feature representation and feature correlation learning. DRN [24] uses RCAN's basic residual block RCAB to stack residual blocks and tries to fuse and gain benefits at multiple scales. HAN [41] combines a Layer Attention Module (LAM) with a Channel Spatial Attention Module (CSAM) to selectively capture more informative features. Transformer-based methods such as SwinIr [11] outperform convolutional neural networks in the field of SR. On the one hand, SwinIR uses a moving window mechanism and supports long-range dependency modeling. On the other hand, it has a local attention mechanism that gives it the advantage of CNN for processing large-scale images. However, compared with the model of the convolutional neural network, the model of the transformer class is temporarily difficult to run in real time.

### 2.3. Model Optimization

A common SR model optimization method is model compression. Recursive models such as [7] reduce the size of the model and the amount of model parameters by sharing the weight of the network, but the recursive model still has a long inference time. Model quantization can reduce memory storage by converting weights to lower-bit storage [42]. However, since the super-resolution model is very sensitive to quantization, full quantization will cause a significant decrease in accuracy, so the method adopted is often partial quantization. Some methods also use full-precision activation [42]in order to avoid loss of precision, which makes the speed improvement brought by quantization not ideal. PAMS [43] proposes a parameterized maximum scale, using trainable truncation parameters to adaptively explore the upper limit of the quantization range, but it takes some time for input quantization and output dequantization, so that the actual speed of the model is limited. DDTB [44] proposes a layer-by-layer quantizer with trainable upper and lower bounds and a dynamic gate controller to overcome the sharply changing activation range of different samples. But the introduction of quantizers and gate controllers slows things down while addressing the loss of precision. This makes the model quantification strategy still have limited advantages. Another common approach is to use depthwise separable convolutions [32] or similar for model compression. However, the decrease in the number of model parameters does not mean the increase in computing speed.

Another possible approach is to use model distillation. Knowledge distillation aims to transfer the knowledge learned by a large model or multiple models (teacher model) to another single lightweight model (student model). In fact, there have been some works applying model distillation in the field of super-resolution in recent years. [45] simultaneously compress and accelerate SR models using a contrastive self-distillation framework. On the one hand, we construct a channel segmentation super-resolution network from the teacher network and use it as a compact student model. On the other hand, a contrastive loss is introduced to improve the learning ability of the student model through explicit knowledge transfer. A novel distillation framework is proposed in [46]. Consisting of teacher and student networks can significantly improve the performance of FSRCNN. It uses ground-truth high-resolution (HR) images as privileged information, and lets the encoder of the teacher model imitate the process of loss learning degradation, while the decoder in the student and teacher model has the same network architecture as FSRCNN to distill features transfer to students. We did not use the above options. We train a simplified version of EDSR as our teacher model and set SRN as the student model. We try to derive yield from the output of the trained teacher model.

## 3. Method

In this section, we first introduce motivation in Section 3.1, our proposed SRN Framework in Section 3.2. In Section 3.3, we introduce distillation loss.

### 3.1. Motivation

In recent years, many deep neural network structures have been proposed in single image super-resolution research. An enhanced deep super-resolution network (EDSR) [1] with performance exceeding those of state-of-the-art SR methods before 2017. It proposes to use the L1

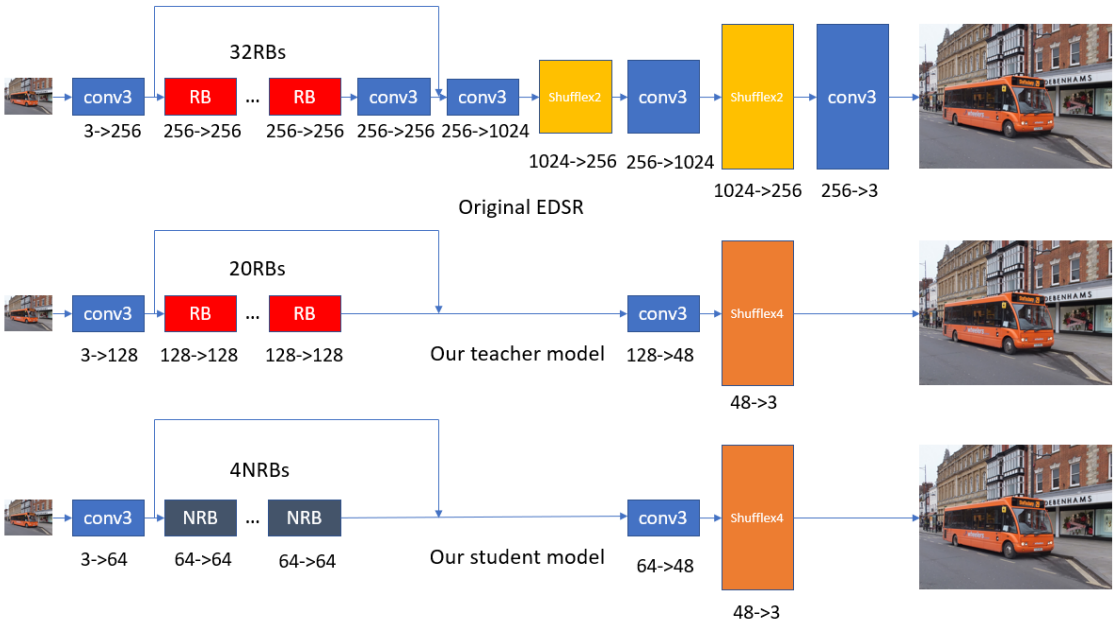


Figure 2. Original EDSR and our models

loss function instead of the L2 loss function and remove the batch normalization, which still makes sense today.

Another significant thing is the addition of attention mechanisms such as Residual Channel Attention Networks (RCAN) [2]. The proposal of enhanced spatial attention (ESA) [3] brings the attention mechanism to a higher level. The ESA block is designed to be lightweight and efficient although adding ESA will make the network train slower. So the teacher model we use does not contain ESA to speed up training. Considering that the student model is generally small and shallow, the impact of adding ESA for student model on training time is more acceptable.

Previously, some scholars used residual learning to solve the learning problem of deep networks [4]. In order to make the model run faster, considering that the small model is relatively shallow and easy to train, the performance degradation of removing the residual connection in the residual block should be within an acceptable range. In FMEN [35], ERB with residuals removed is used instead of RB, and it is found that the performance of ERB relative to RB has declined but is acceptable. Among them, ERB uses a structural reparameterization technique. Therefore, it is feasible to remove unnecessary residual connections in the residual block and only keep one residual connection. However, using a structural reparameterization technique will greatly increase the training time of the model. To reduce the slowdown caused by the introduction of ESA, we have retuned ESA to run faster. Related comparative experiments can be seen in Table 2. Through comparative experiments, it is found that the residual block introduced into ESA can not only increase the speed but also improve the performance after removing the residual, which may be that the residual limits the expressive ability of ESA which can be seen in Table 3. We call the new non-residual block NRB (Non-Residual Block), as shown in the figure. Details of our SRN model can be seen in Fig. 2. See Fig. 3 and Fig. 4 for more information. In order to speed up the training, we do not use structural reparameterization technique.

### 3.2. Framework

Our SRN mainly consists of three parts: the first feature extraction convolution, Non-Residual Blocks(NRBs), and the reconstruction module.

Specifically, the initial feature extraction is implemented by a $3 \times 3$ convolution to generate coarse features from the input LR image. Given the input x, this procedure can be expressed as

$$F_0 = h(x) \tag{1}$$

where $h$ denotes the coarse feature extraction function and $F_0$ is the extracted features.

NRB consists of two 3x3 convolutions with a leaky_ReLU and ESA module, which can be seen in Fig. 3. Compared with RB, it has no residual.

The next part of SRN is multiple NRBs that are stacked in a chain manner to gradually refine the extracted features. This process can be formulated as

$$F_k = H_k(F_{k-1}), k = 1, ..., n \tag{2}$$

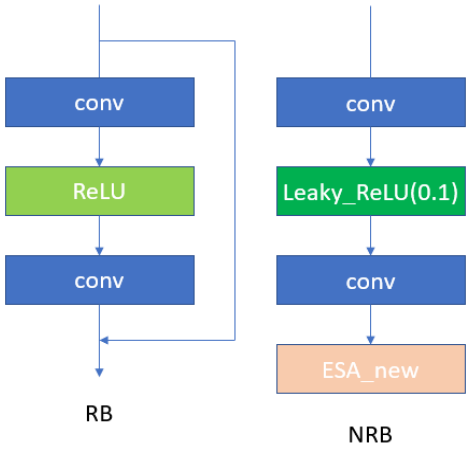


Figure 3. RB and NRB

where $H_k$ denotes the k-th NRB function, $F_{k-1}$ and $F_k$ represent the input feature and output feature of the k-th NRB, respectively.

Finally, the SR images are generated through the reconstruction as follows

$$y = R(F_n + F_0) \tag{3}$$

where R denotes the reconstruction function and y is the output of the network. The reconstruction process only consists of a 3 × 3 convolution and a non-parametric sub-pixel operation.

Our SRN model's feature channels is set to 64 while the channel number of ESA is set to 16. For NTIRE2023 ESR challenge, we set NRBs to 4, otherwise we set it to 6. The last convolution before the long residual is removed, which constitutes the SRN-S. The model used in NTIRE2023 ESR challenge is SRN-S.

### 3.3. Distillation

Model distillation has become a hot topic since this [25] post came out. Super-resolution distillation may be a viable solution but we found that it is difficult for the student model to benefit from the features of the middle layer of the teacher model. Therefore, we only append the loss function between the output of the teacher model and the student model on the basis of the primary loss function. Details can be seen in the Fig. 5.

Furthermore, We use a lightweight version of EDSR and remove the last single convolutional layer included in the long residual connection. We use this as our teacher model. The difference between the original EDSR and our teacher model can be seen in the Fig. 2. Considering our limited resources, we did not use the original channel size of 256 and 32 residual blocks [1]. The number of feature channels

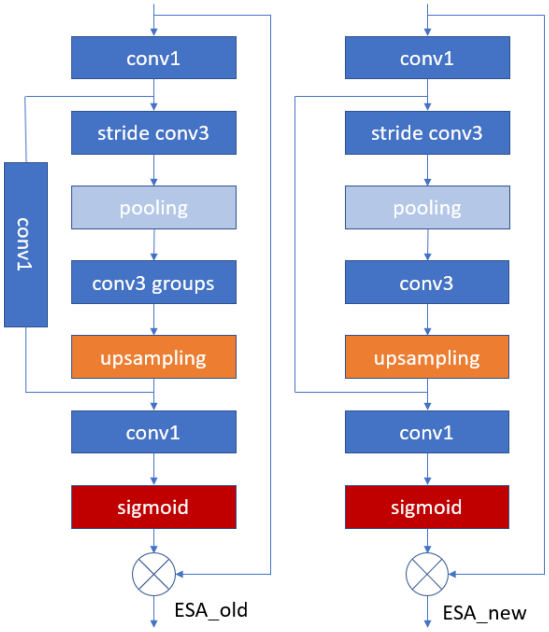


Figure 4. ESA modify

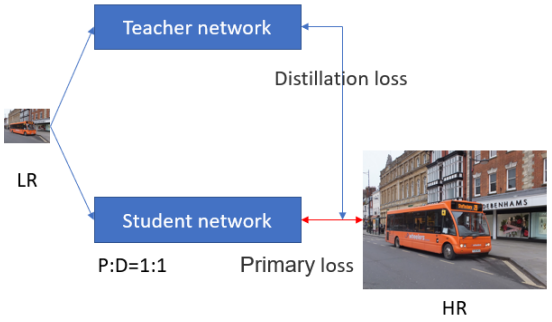


Figure 5. Loss

is set to 128, the residual blocks is set to 20. The residual scaling factor is set to 1.

We combine the original loss with the distillation loss as:

$$\begin{aligned} L(\theta) =& \lambda_P ||H_{SRN}(I^{LR}) - I^{HR}||_1 + \\ & \lambda_D ||H_{SRN}(I^{LR}) - H_T(I^{LR})||_1 \end{aligned} \tag{4}$$

where $H_{SRN}$ epresents the function of our proposed network,$H_T$ epresents the function of our teacher network, $\theta$ indicates the learnable parameters of SRN and $||\cdot||_1$ is the $l_1$ norm. $I^{LR}$ and $I^{HR}$ are the input LR images and the corresponding ground-truth HR images, respectively. $\lambda_P$ means the coefficient of the main loss function and $\lambda_D$ means the coefficient of the distillation loss. we set them to 1.

| Scale | Model | Params (K) | runtime (ms) | Set5 PNSR/SSIM | Set14 PSNR/SSIM | BSD100 PSNR/SSIM | Urban100 PSNR/SSIM |
|---|---|---|---|---|---|---|---|
| | SRCNN [5] | 57 | 49.44ms | 36.66 / 0.9542 | 32.45 / 0.9067 | 31.36 / 0.8879 | 29.50 / 0.8946 |
| | FSRCNN [6] | 12 | 10.02ms | 37.05 / 0.9560 | 32.66 / 0.9090 | 31.53 / 0.8920 | 29.88 / 0.9020 |
| | VDSR [14] | 666 | 220.77ms | 37.53 / 0.9587 | 33.03 / 0.9124 | 31.90 / 0.8960 | 30.76 / 0.9140 |
| | CARN [15] | 1592 | 144.70ms | 37.76 / 0.9590 | 33.52 / 0.9166 | 32.09 / 0.8978 | 31.92 / 0.9256 |
| × 2 | IMDN [8] | 694 | 106.21ms | 38.00 / 0.9605 | 33.63 / 0.9177 | 32.19 / 0.8996 | 32.17 / 0.9283 |
| | ECBSR [9] | 596 | 58.21ms | 37.90 / 0.9615 | 33.34 / 0.9178 | 32.10 / 0.9018 | 31.71 / 0.9250 |
| | RLFN [10] | 527 | 84.53ms | 38.07 / 0.9607 | 33.72 / 0.9187 | 32.22 / 0.9000 | 32.33 / 0.9299 |
| | **SRN-S(ours)** | 492 | 61.59ms | 37.96 / 0.9604 | 33.62 / 0.9175 | 32.13 / 0.8990 | 32.15 / 0.9282 |
| | **SRN(ours)** | 529 | 64.37ms | 37.98 / 0.9605 | 33.65 / 0.9176 | 32.14 / 0.8992 | 32.19 / 0.9287 |
| | SRCNN [5] | 57 | 49.66ms | 30.48 / 0.8628 | 27.50 / 0.7513 | 26.90 / 0.7101 | 24.52 / 0.7221 |
| | FSRCNN [6] | 13 | 4.20ms | 30.72 / 0.8660 | 27.61 / 0.7550 | 26.98 / 0.7150 | 24.62 / 0.7280 |
| | VDSR [14] | 666 | 220.22ms | 31.35 / 0.8838 | 28.01 / 0.7674 | 27.29 / 0.7251 | 25.18 / 0.7524 |
| | CARN [15] | 1592 | 62.67ms | 32.13 / 0.8937 | 28.60 / 0.7806 | 27.58 / 0.7349 | 26.07 / 0.7837 |
| × 4 | IMDN [8] | 715 | 28.97ms | 32.21 / 0.8948 | 28.58 / 0.7811 | 27.56 / 0.7353 | 26.04 / 0.7838 |
| | ECBSR [9] | 603 | 16.92ms | 31.92 / 0.8946 | 28.34 / 0.7817 | 27.48 / 0.7393 | 25.81 / 0.7773 |
| | RLFN [10] | 543 | 23.10ms | 32.24 / 0.8952 | 28.62 / 0.7813 | 27.60 / 0.7364 | 26.17 / 0.7877 |
| | **SRN-S(ours)** | 513 | 17.05ms | 32.13 / 0.8940 | 28.60 / 0.7813 | 27.57 / 0.7355 | 26.08 / 0.7843 |
| | **SRN(ours)** | 550 | 17.82ms | 32.15 / 0.8942 | 28.60 / 0.7814 | 27.58 / 0.7356 | 26.11 / 0.7849 |

Table 1. Quantitative results of the state-of-the-art models on four benchmark datasets.The best and second-best results are marked in red and blue colors, respectively.

# 4. Experiments

## 4.1. Setup

**Datasets and Metrics**. We train our models on DIV2K [16] and LSDIR [22] datasets. We test the performance of our models on four benchmark dataset: Set5 [17], Set14 [18], BSD100 [19] and Urban100 [20]. We evaluate the PSNR and SSIM on the Y channel of YCbCr space.

**Training Details**. The proposed SRN has 6 NRBs (Non-Residual Blocks), in which the number of feature channels is set to 64 while the channel number of ESA is set to 16. In total we used two datasets: DIV2K [16] and LSDIR. SRN_S means removing the last convolution before the end of the residual network which makes network running fast.

To train the models with images, we augment training dataset with geometric transforms: vertical/horizontal flips and 90-degree rotation in order to enhance the comprehensive ability of the model.This also makes the model have higher performance in dealing with non-training set data.

For teacher model:

1. At the first stage, the model is trained from scratch. HR patches of size 192 × 192 are randomly cropped from HR images, and the mini-batch size is set to 16. The teacher model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to $2 \times 10^{-4}$. The total number of epochs is 20000.(Only use DIV2K dataset). The learning rate decay is following cosine annealing with $T_{max} = total\_epochs, eta_{min} = 1 \times 10^{-7}$.

2. At the second stage, the model is initialized with the pretrained weights. The initial learning rate is set to $1 \times 10^{-4}$.In this stage, we use LDSIR dataset. The total number of epochs is 200. Other settings are the same as in the previous step.

3. At the last stage, the model is initialized with the pretrained weights.HR patches of size 256 × 256 are randomly cropped from HR images. The initial learning rate is set to $2.5 \times 10^{-5}$. Now the teacher model is trained by minimizing L2 loss function with Adam optimizer. The total number of epochs is 50. Other settings are the same as in the previous step. After training, we freeze the parameters of the teacher model.

Since we only trained the teacher network at x4 scale, the distillation scheme was not used in x2 scale.

For student model:

For x4 scale, HR patches of size 256 × 256 are randomly cropped from HR images. For x2 scale, HR patches of size 128 × 128 are randomly cropped from HR images. The mini-batch size is set to 32. The student model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to $2 \times 10^{-4}$. The total number of epochs is 80000. The

| Model | last_conv | ESA | runtime | Set5 | | Set14 | | BSD100 | | Urban100 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | PNSR | Δ | PSNR | Δ | PSNR | Δ | PSNR | Δ |
| baseline | ✓ | ✗ | 13.81ms | 31.7287 | - | 28.3271 | - | 27.3810 | - | 25.4518 | - |
| baseline | ✗ | ✗ | 13.20ms | 31.6948 | -0.0339 | 28.3187 | -0.0084 | 27.3657 | -0.0153 | 25.4193 | -0.0325 |
| ESAN_old | ✓ | ✓ | 19.62ms | 31.9248 | +0.1961 | 28.4501 | +0.1230 | 27.4446 | +0.0636 | 25.6798 | +0.2280 |
| ESAN_old | ✗ | ✓ | 18.91ms | 31.8563 | +0.1276 | 28.3996 | +0.0725 | 27.4101 | +0.0291 | 25.5809 | +0.1291 |
| ESAN_new | ✓ | ✓ | 18.95ms | 31.9015 | +0.1728 | 28.4341 | +0.1070 | 27.4386 | +0.0576 | 25.6428 | +0.1910 |
| ESAN_new | ✗ | ✓ | 18.28ms | 31.8333 | +0.1046 | 28.3769 | +0.0498 | 27.3980 | +0.0170 | 25.5464 | +0.0946 |

Table 2. Introduction of ESA module. Runtime is tested on DIV2K validation set.

| Model | last_conv | residual | runtime | Set5 | | Set14 | | BSD100 | | Urban100 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | PNSR | Δ | PSNR | Δ | PSNR | Δ | PSNR | Δ |
| ESAN_new | ✓ | ✓ | 18.95ms | 31.9015 | - | 28.4341 | - | 27.4386 | - | 25.6428 | - |
| ESAN_new | ✗ | ✓ | 18.28ms | 31.8333 | -0.0682 | 28.3769 | -0.0572 | 27.3980 | -0.0406 | 25.5464 | -0.0964 |
| ESAN_new | ✓ | ✗ | 17.82ms | 31.9730 | 0.0715 | 28.4622 | 0.0281 | 27.4613 | 0.0227 | 25.7253 | 0.0825 |
| ESAN_new | ✗ | ✗ | 17.05ms | 31.9039 | 0.0024 | 28.4196 | -0.0145 | 27.4322 | -0.0064 | 25.6683 | 0.0255 |

Table 3. Limiting effect of residuals. Runtime is tested on DIV2K validation set.

learning rate decay is following cosine annealing with $T_{max} = total\_epochs, eta_{min} = 1 \times 10^{-7}$.

### 4.2. Quantitative Results

In this scetion, we compare our model against several advanced efficient super-resolution models with upsampling factors of 2 and 4. This includes SRCNN [5], FSRCNN [6] , VDSR [14], CARN [15], IMDN [8], ECBSR [9], RLFN [10]. The quantitative performance comparison of several benchmark datasets is shown in Table 1. The inference time in Table 1 is the average speed in milliseconds on the DIV2K [16] validation set on an NVIDIA 3080 GPU. Compared with other state-of-the-art models, the proposed SRN-S and SRN still have gaps in terms of PSNR and SSIM compared with the 2022 SOTA model RLFN, but the gaps are within acceptable limits. Even models can be sub-optimal in the table. SRN-S doesn't have the last convolution before the end of the large residual compared with SRN. It is worth mentioning that our SRN-S is about 36% faster than RLFN. SRN and can be about 30% faster than RLFN. Compared with ECBSR, which uses structural reparameterization technique, the speed is almost the same, but the performance index is far beyond. In other words, the speed is far better than RLFN, and the performance is far better than ECBSR. It can be considered that our model has achieved a good balance between speed and performance.

### 4.3. Ablation Study

When doing ablation experiments, we use DIV2K [16] and Flickr2K datasets. We consider whether there is a last convolution before a large residual as a variable. The absence of the last convolution can make the network run faster. The baseline network has 6 RBs without the attention mechanism.

**Effectiveness of a simplified version of enhanced spatial attention.** DIV2K and Flickr2K datasets are used. As shown in Fig. 4, compared to the original ESA model, we replace the convolution group with a single convolution and remove a 1x1 convolution. The model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to $2 \times 10^{-4}$. The total number of epochs is 1250. The learning rate decay is following cosine annealing with $T_{max} = total\_epochs, eta_{min} = 1 \times 10^{-7}$. The experimental results can be seen in the Table 2.

We can roughly draw the following conclusions:

- The absence of the last convolution will slightly reduce the performance of the network, but make network runs faster.
- The introduction of ESA can greatly improve network performance but will also greatly affect the speed.
- Compared with the normal version, our simplified ESA model does not perform too badly, but it is faster than the normal version.

In the end this made us decide to use a simplified version of the ESA module to do more optimization.

**Effectiveness of Non-Residual Blocks.** DIV2K [16] and Flickr2K datasets are used. The training is the same as before. Shown in Fig. 3, NRB (Non-Residual Block) means that the residual blocks in the network have no residuals. Finally, we call this network mainly composed of NRBs SRN. The experimental results can be seen in the Table 3.

In the experiment, we can find that simply removing the residuals not only makes the network inference faster, but also improves the network performance. This shows that this is a lossless optimization.

| Model | last_conv | distillation | runtime | Set5 | | Set14 | | BSD100 | | Urban100 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | PNSR | Δ | PSNR | Δ | PSNR | Δ | PSNR | Δ |
| teacher | ✗ | ✗ | - | 32.3928 | - | 28.8507 | - | 27.7372 | - | 26.7776 | - |
| SRN | ✓ | ✗ | 17.82ms | 31.9314 | - | 28.4481 | - | 27.4463 | - | 25.6931 | - |
| SRN | ✗ | ✗ | 17.05ms | 31.8728 | -0.0586 | 28.3795 | -0.0686 | 27.4184 | -0.0279 | 25.6250 | -0.0681 |
| SRN | ✓ | ✓ | 17.82ms | 31.9390 | +0.0076 | 28.4581 | +0.0100 | 27.4724 | +0.0261 | 25.7605 | +0.0674 |
| SRN | ✗ | ✓ | 17.05ms | 31.8937 | -0.0377 | 28.4159 | -0.0322 | 27.4584 | +0.0121 | 25.6713 | -0.0218 |

Table 4. The utility of distillation. Runtime is tested on DIV2K validation set.

| Team Name | PSNR [Val] | PSNR [Test] | Ave Time [ms] | Params [M] | FLOPs [G] | Acts [M] | Mem [M] | Conv |
|---|---|---|---|---|---|---|---|---|
| MegSR | 29.04 | 26.95 | 18.30 | 0.243 | 14.9 | 72.97 | 495.91 | 39 |
| Zapdos(ours) | 28.96 | 27.03 | 18.59 | 0.352 | 21.97 | 63.01 | 420.5 | 26 |
| DFCDN | 29.00 | 27.08 | 18.71 | 0.245 | 15.49 | 82.76 | 376.99 | 39 |
| KaiBai_Group | 28.95 | 27.01 | 20.49 | 0.272 | 16.76 | 65.1 | 296.45 | 35 |
| RIP_ShopeeVideo | 28.97 | 27.04 | 20.65 | 0.255 | 16.16 | 74.97 | 439.6 | 35 |

Table 5. Runtime results of NTIRE 2023 efficient SR challenge. Only the top five methods are included.

**Effectiveness of distillation.** Only DIV2K [16] datasets is used. The model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to $2 \times 10^{-4}$. The total number of epochs is 5000. The learning rate decay is following cosine annealing with $T_{max} = total\_epochs, eta_{min} = 1 \times 10^{-7}$. The distillation method can generally be seen from the Fig. 5. The experimental results can be seen in the Table 4.

The improvement brought by distillation is not as obvious as removing the residual directly, but it is also a positive effect because it can improve accuracy without affecting inference speed. At the same time, distillation can reduce the gap between SRN-S and SRN on some data sets such as BSD100.

### 4.4. SRN for NTIRE 2023 challenge

We include the top five methods in Table 5 [23]. The speed of our team's method achieves the second-shortest inference time as well as the second-smallest number of activations in the NTIRE 2023 efficient super-resolution challenge. At the same time, our method contains the least number of convolutions.

The training strategy in challenge is as follows. For teacher model, it is the same as previous method. But for student model, the model structure and training strategy are slightly different from the above:

1. We only use the DIV2K dataset. We use main loss and distillation loss to train our student model. The proposed SRN has 4 NRBs instead of 6. Besides, in order for the network to run faster, we remove the last convolution before the end of the single residual.

2. At the first stage, the model is trained from scratch. HR patches of size 256 × 256 are randomly cropped from HR images, and the mini-batch size is set to 32. The student model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to $2 \times 10^{-4}$. The total number of epochs is 80000. The learning rate decay is following cosine annealing with $T_{max} = total\_epochs, eta_{min} = 1 \times 10^{-7}$.

3. At the second stage, the model is initialized with the pretrained weights, and trained with the same settings as in the previous step.

4. At the last stage, the model is initialized with the pretrained weights. HR patches of size 640 × 640 are randomly cropped from HR images, and the mini-batch size is set to 32. The student model is trained by minimizing L2 loss function with Adam optimizer. The initial learning rate is set to $5 \times 10^{-5}$.The total number of epochs is 4000.

## 5. Conclusion

In this paper, we propose a single residual network for efficient SISR. By introducing and optimizing the ESA model, the accuracy of the network is reasonably improved. We then revisit the limitations of residuals on the capabilities of models incorporating ESA, and removing residuals greatly improves model performance and speed. We also propose a model distillation strategy that can effectively improve performance without compromising speed. Our method achieves a large advantage in running time and number of activations, and our model achieves a good balance between speed and performance.

## References


[1] Bee,Lim, et al. “Enhanced Deep Residual Networks for Single Image Super-Resolution.” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1132–1140. 1, 2, 3, 5

[2] Yulun,Zhang, et al. ”Image super-resolution using very deep residual channel attention networks.” Proceedings of the European conference on computer vision (ECCV). 2018. 1, 3, 4

[3] Jie,Liu, et al. ”Residual feature aggregation network for image super-resolution.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. 2, 4

[4] Kaiming,He, et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. 4

[5] Chao,Dong, et al. ”Learning a deep convolutional network for image super-resolution.” Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13. Springer International Publishing, 2014. 1, 2, 6, 7

[6] Chao,Dong, Chen Change Loy, and Xiaoou Tang. ”Accelerating the super-resolution convolutional neural network.” Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer International Publishing, 2016. 1, 2, 6, 7

[7] Jiwon,Kim, Jung Kwon Lee, and Kyoung Mu Lee. ”Deeply-recursive convolutional network for image super-resolution.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. 1, 2, 3

[8] Zheng,Hui et al. ”Lightweight image super-resolution with information multi-distillation network.” Proceedings of the 27th acm international conference on multimedia. 2019. 2, 6, 7

[9] Xindong,Zhang, Hui Zeng, and Lei Zhang. ”Edge-oriented convolution block for real-time super resolution on mobile devices.” Proceedings of the 29th ACM International Conference on Multimedia. 2021. 2, 3, 6, 7

[10] Fangyuan,Kong, et al. ”Residual local feature network for efficient super-resolution.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022. 2, 3, 6, 7

[11] Jingyun, Liang, et al. ”Swinir: Image restoration using swin transformer.” Proceedings of the IEEE/CVF international conference on computer vision. 2021. 1, 3

[12] X.Chen, et al. ”Activating More Pixels in Image Super-Resolution Transformer. arXiv 2022.” arXiv preprint arXiv:2205.04437. 1

[13] Ying,Tai, Jian Yang, and Xiaoming Liu. ”Image super-resolution via deep recursive residual network.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. 1

[14] Jiwon,Kim, Jung Kwon Lee, and Kyoung Mu Lee. ”Accurate image super-resolution using very deep convolutional networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. 6, 7

[15] Namhyuk,Ahn, Byungkon Kang, and Kyung-Ah Sohn. ”Fast, accurate, and lightweight super-resolution with cascading residual network.” Proceedings of the European conference on computer vision (ECCV). 2018. 6, 7

[16] Eirikur,Agustsson, and Radu Timofte. ”Ntire 2017 challenge on single image super-resolution: Dataset and study.” Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017. 6, 7, 8

[17] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In BMVC, 2012. 6

[18] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In International Conference on Curves and Surfaces, pages 711–730. Springer, 2010. 6

[19] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(5):898–916, 2011. 6

[20] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In IEEE Conference on Computer Vision and Pattern Recognition, pages 5197–5206, 2015. 6

[21] Yawei Li, Kai Zhang, Luc Van Gool, Radu Timofte, et al. Ntire 2022 challenge on efficient super-resolution: Methods and results. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2022. 2, 3

[22] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Lsdir: A large scale dataset for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2023. 6

[23] Yawei Li, Yulun Zhang, Luc Van Gool, Radu Timofte, et al. Ntire 2023 challenge on efficient super-resolution: Methods and results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2023. 8

[24] Yong,Guo, et al. "Closed-loop matters: Dual regression networks for single image super-resolution." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. 1, 3

[25] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015). 5

[26] Lin, Zudi, et al. "Revisiting rcan: Improved training for image super-resolution." arXiv preprint arXiv:2201.11279 (2022). 3

[27] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In ACCV 2014. 1

[28] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In IEEE Conference on Computer Vision and Pattern Recognition, pages 11065–11074, 2019. 1

[29] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In European Conference on Computer Vision, pages 191–207, 2020. 1

[30] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image superresolution with non-local sparse attention. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3517–3526, 2021. 1

[31] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In IEEE Conference on Computer Vision and Pattern Recognition, pages 12299–12310, 2021. 1

[32] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 1, 3

[33] Z Li, Liu Y, Chen X, et al. Blueprint separable residual network for efficient image super-resolution[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 833-843. 1

[34] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In European Conference on Computer Vision, pages 41–55. Springer, 2020. 2

[35] Z Du, Liu D, Liu J, et al. Fast and memory-efficient network towards efficient image super-resolution[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 853-862. 2, 4

[36] Kai Zhang, Shuhang Gu, Radu Timofte, Zheng Hui, Xiumei Wang, Xinbo Gao, Dongliang Xiong, Shuai Liu, Ruipeng Gang, Nan Nan, et al. AIM 2019 challenge on constrained super-resolution: Methods and results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 3565–3574. IEEE, 2019. 2

[37] Kai Zhang, Martin Danelljan, Yawei Li, Radu Timofte, Jie Liu, Jie Tang, Gangshan Wu, Yu Zhu, Xiangyu He, Wenjie Xu, et al. AIM 2020 challenge on efficient super-resolution: Methods and results. In Proceedings of the European Conference on Computer Vision Workshops, pages 5–40. Springer, 2020. 2

[38] Jie, Hu, Li, Shen, Samuel, Albanie, Gang, Sun, Enhua, and Wu. Squeeze-and-excitation networks. IEEE transactions on pattern analysis and machine intelligence, 2019. 3

[39] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. 2019. 3

[40] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 3

[41] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H Shen. Single image super-resolution via a holistic attention network. 2020. 3

[42] Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. Efficient super resolution using binarized neural network. In CVPRW, 2019. 3

[43] H Li, Yan C, Lin S, et al. Pams: Quantized super-resolution via parameterized max scale[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. Springer International Publishing, 2020: 564-580. 3

[44] Zhong Y, Lin M, Li X, et al. Dynamic Dual Trainable Bounds for Ultra-low Precision Super-Resolution Networks[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII. Cham: Springer Nature Switzerland, 2022: 1-18. 3

[45] Wang Y, Lin S, Qu Y, et al. Towards compact single image super-resolution via contrastive self-distillation[J]. arXiv preprint arXiv:2105.11683, 2021. 3

[46] Lee W, Lee J, Kim D, et al. Learning with privileged information for efficient image super-resolution[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16. Springer International Publishing, 2020: 465-482. 3