AsConvSR: Fast and Lightweight Super-Resolution Network with Assembled Convolutions

Jiaming Guo[1] Xueyi Zou[1] Yuyi Chen[1] Yi Liu[1] Jia Hao[2]
Jianzhuang Liu[1] Youliang Yan[1]
[1]Huawei Noah's Ark Lab
[2]HiSilicon (Shanghai) Technologies Co., Ltd
{guojiaming5, zouxueyi, chenyuyi1, liuyi113, hao.jia, liu.jianzhuang, yanyouliang}@huawei.com

Abstract

In recent years, videos and images in 720p (HD), 1080p (FHD) and 4K (UHD) resolution have become more popular for display devices such as TVs, mobile phones and VR. However, these high resolution images cannot achieve the expected visual effect due to the limitation of the internet bandwidth, and bring a great challenge for super-resolution networks to achieve real-time performance. Following this challenge, we explore multiple efficient network designs, such as pixel-unshuffle, repeat upscaling, and local skip connection removal, and propose a fast and lightweight super-resolution network. Furthermore, by analyzing the applications of the idea of divide-and-conquer in super-resolution, we propose assembled convolutions which can adapt convolution kernels according to the input features. Experiments suggest that our method outperforms all the state-of-the-art efficient super-resolution models, and achieves optimal results in terms of runtime and quality. In addition, our method also wins the first place in NTIRE 2023 Real-Time Super-Resolution - Track 1 (×2). The code will be available at https://gitee.com/mindspore/models/tree/master/research/cv/AsConvSR

1. Introduction

Super-resolution is widely used to improve the visual quality of images and videos [22, 23, 28] displayed on various devices like mobile phones, TVs and so on. In recent years, most media contents are produced and distributed in high resolution like 720p (HD), 1080p (FHD) and 4k (UHD), and display devices with higher resolutions have become more affordable and popular for the public. Therefore, super-resolution needs to process higher resolution images and videos , which significantly increase the processing time and memory bandwidth [46].

Previous works like RFDN [32] and RTSRN [17] have proposed efficient networks for super-resolution, but they mainly aim at lower input resolutions like 540p and 640p. Their real-time (30fps, below 33ms per image) performance cannot be guaranteed on higher input resolutions. To design real-time super-resolution networks for inputs with higher resolutions above 720p, the effectiveness of skip-connection, concatenation and other operations which are commonly used in existing methods needs to be re-evaluated.

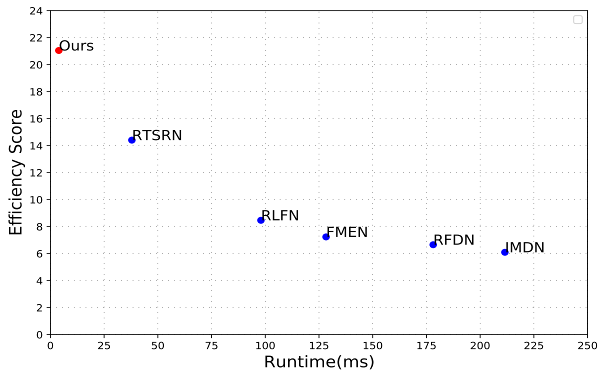

Figure 1. Illustration of efficiency score (defined in Section 4.1) and runtime of different efficient super-resolution models on DF2K dataset. Our proposed AsConvSR achieves the best performance.

Facing the aforementioned challenge, we redesign the basic structure of the network to achieve the goal of real-time super-resolution on high input resolutions like 720p and 1080p. We re-evaluate the performances of the network structures with complex topologies such as Enhanced Spatial Attention (ESA) and Residual Feature Distillation Block (RFDB) [32]. These structures can improve the performance of the SR network, but they also increase the model runtime. Therefore, a network with simple topology should be the best candidate to construct an efficient super-resolution model. We adopt a simple backbone that extracts features via sequential convolutions in a straight-forward topology, additionally with pixel-unshuffle and pixel-shuffle operations respectively at the beginning and the end of the network. We discard all intermediate skip-connection which bring additional computation overhead in practice, and only retain a global skip-connection to achieve high efficiency.

Considering the images as inputs for the super-resolution network we discuss above, we usually see patches or areas with different textures and contents from a great variety of objects like trees, buildings, human beings, etc. Intuitively, these patches with different patterns and texture complexity require different processing methods. Therefore, we propose the assembled convolution which enables the networks to adaptively apply different convolution kernels for different inputs. Compared with previous dynamic convolution [6, 44, 48], our assembled convolution is more flexible and effective as it calculates the optimal convolution kernel coefficients for each channel, and the design only brings a slight increase in computation cost and inference time for the entire network.

As demonstrated in Fig.1, our model outperforms all the state-of-the-art efficient super-resolution model in the efficiency score and runtime. With our compact design, our model is capable of real-time super-resolution on 720p and 1080p inputs with a single GPU (NVidia Tesla V100).

Our main contributions can be summarized as follows:

- We propose a fast and lightweight super-resolution network in a simple and straight-forward network topology, which can keep the restoration accuracy and significantly reduce the model runtime.
- We propose assembled convolution for our efficient super-resolution network. It expands the capacity of feature extraction and meanwhile retains efficiency by adapting convolution kernels based on the input.
- By applying the assembled convolution in the fast and lightweight super-resolution network, we propose AsConvSR which wins the first place in NTIRE 2023 Real-Time Super-Resolution - Track 1 (X2) [8].

2. Related Work

2.1. Efficient Image Super-Resolution

In recent years, many approaches have been proposed to devise efficient super-resolution networks [2, 15, 18, 21, 25, 32, 39] that can run robustly on devices with limited computational resources with low latency. Rethinking the pioneering work SRCNN [13] that applies deep learning to SISR for the first time, FSRCNN [14] significantly accelerates the SISR network by adopting the original low-resolution as input without bicubic interpolation, smaller sizes of convolution kernels, and a deconvolution layer at the final stage of the network to perform upsampling. LapSRN [27] progressively reconstructs the sub-band residuals of high-resolution images using the Laplacian pyramid. CARN [2] further improves efficiency by its design of cascading residual networks with group convolution. IMDN [21] proposes information multi-distillation blocks with contrast-aware attention (CCA) layer based on the information distillation mechanism, while RFDN [32] refines the architecture of RFDN with feature distillation mechanism by proposing the residual feature distillation block. Following IMDN and RFDN, both RLFN and FMEN rethink the effectiveness of applying distillation and attention mechanism in this field, and consequently adopt fully sequential CNN network architecture. Specifically, RLFN [25] redesigns RFDB by adding more channels to compensate for discarded feature distillation branches to achieve higher inference speed and better performance with fewer parameters. FMEN [15] expands optimization space during training with re-parameterizable building blocks [12] without increasing extra inference time. SwinIR [29] proposes an efficient transformer-based SR model which fully explores the swin transformer structure, and it outperforms pure convolution networks with fewer parameters and FLOPs. Swin2SR [7] further improved the network structure by introducing the SwinV2 attention, and proposes auxiliary loss and high-frequency loss for the compressed images. However, all these methods cannot satisfy real-time performance.

2.2. Dynamic Convolution

Dynamic Convolution [4, 5, 9, 10, 24, 34, 37, 38, 40–42, 45, 49] has gained growing interests for its potential to learn more powerful features by generating convolution kernels adaptively based on the inputs. Consequently, this method successfully increases capacity and flexibility of convolutional neural networks (CNN) without adding redundant computation cost, and can be readily applied in various CNN architectures. This method usually aggregates several fixed convolution kernels as introduced in [6, 44, 48], which also carry out experiments to demonstrate its effectiveness on typical computer vision tasks including image classification, object detection and segmentation. Dynamic convolution is also powerful on low-level vision tasks, such as denoising [35], and single image super-resolution (SISR) [43]. Inspired by the above research of dynamic convolution, we design the assembled convolution block for our efficient super-resolution network. It differs from the previous dynamic convolution, and gains higher performance as well as efficiency in super-resolution task.

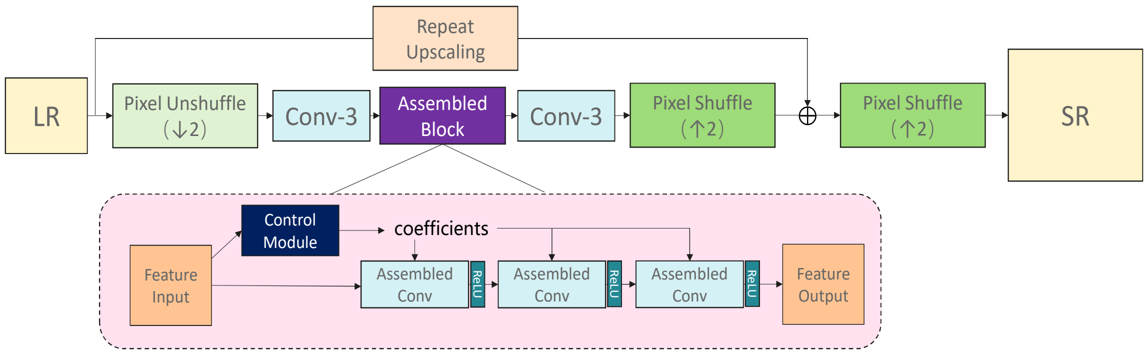

Figure 2. Architecture of fast and lightweight super-resolution network with assembled convolutions (AsConvSR).

3. Method

In this section, we first describe the network architecture of AsConvSR in Section 3.1. In Section 3.2, we analyze the dynamic convolutions and propose our assembled convolutions.

3.1. Network Architecture

Processing time and memory bandwidth of a neural network are highly related to the resolution of input images. RTSRN [17] can achieve real-time performance for a 360p input on a Tesla V100 GPU. However, runtime of this model would increase to 37.91 ms (see Section 4.2) when the input resolution is 1080p. The definition of floating point operations per second (FLOPs) is described as follows:

$$FLOPs = (\overbrace{C_i \times K^2}^{multiplications} + \overbrace{C_i \times K^2 - 1}^{additions}) \times H \times W \times C_o, \quad (1)$$

where C_i indicates the number of input channels, C_o is the number of output channels, H and W represent the height and width of the input feature map, and K is the size of the convolution kernel. Note that this formula represents the FLOPs for a convolution layer without bias. According to this equation, given a fixed FLOPs, larger H and W mean smaller C_i and C_o. However, in practice, the runtime cannot be reduced by cutting down the channel size when the number of channels is smaller than a certain level (16 or 32) for most computing devices. Small channel size also limits the flexibility of design and the performance of the SR network.

Pixel unshuffle. We use pixel-unshuffle operation to downscale the input image and increase the number of channels. It can reduce the computational cost of

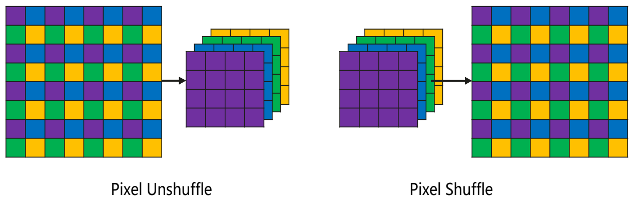

Figure 3. Illustrations of pixel-unshuffle and pixel-shuffle.

the network without loss of information volume. As demonstrated in Fig.3, pixel-unshuffle is the reverse of pixel shuffle. It rearranges the elements from input and outputs the data with smaller resolution and larger channel size. Take N times pixel-unshuffle as an example, the resolution of input image downscales to $\left(\frac{H}{N}, \frac{W}{N}\right)$ while the channel size becomes $C \times N^2$. According to Eq.1, adopting pixel-unshuffle do not change the FLOPs of the first convolution layer if the output channel size remains unchanged. From the second convolution, FLOPs decreases by a factor of N^2 since the resolution is downscaled. The performance usually degrades if we simply downscale the input by only applying pixel-unshuffle, but this can be compensated by increasing the complexity of our network design by several options like increasing the channel size and so on.

Skip connection. Besides the convolution layers, skip connections also occupy a large amount of computational resources. They not only require additions of two feature maps, but also have to cache the previous feature map which increases the runtime of accessing the memory. Skip connections used in super-resolution are mainly divided into two types: one is the skip connection inside the network structure, such as the skip connection of the residual block, and the other is the global skip connection which adds up the network output and the LR upscaled by classical interpolation like bicubic or bilinear. Skip connections in the network structure can be removed according to the experiments (Section 4.4). We cannot simply remove the global skip connection because it can stabilize the training of the network and accelerate the convergence (Section 4.4). So we replace the classical interpolation algorithm in the global skip connection by repeating the LR 4 times according to ABPN [16] which is much faster than the classical interpolation.

Based on the preceding analysis, we design a fast and lightweight super-resolution network. Given an input LR image, the resolution is converted to channel dimension by pixel-unshuffle layer. By using a 3x3 convolution, the channel of feature map is converted to the target size and then fed into the assembled block. The assembled block can adaptively apply different convolution kernels for different inputs. The details about the assembled block are described in the next section. After the assembled block, a 3×3 convolution layer is used to convert the channel size to 48 so that the feature map can be restored to target resolution after the pixel-shuffle layer. Noting that a low resolution image repeated in the channel dimension can also be restored to the high resolution with a pixel-shuffle layer, we divide the final pixel shuffle into two steps in order to introduce the global skip connection to the network.

3.2. Assembled Block

The idea of divide-and-conquer is widely used in image processing domain from classical methods to deep learning algorithms. For example, remove the noise and compression effect on flat areas, sharpen the edges in edge-dominant areas, and generate more fine details in rich-textured areas. These intuitions underlie the idea of patched-based super-resolution. Kong et al. [26] propose ClassSR with a classification network to determine whether the patches go to the simple sub-network to save FLOPs or the complex ones to get a better performance. However, according to our experiments, the process of splitting the images and recombining them to patches also significantly increases the total runtime of the network. Take a 1080p image as input, the process to assemble the patches costs 7ms in total, while the entire runtime budget is only 30ms. On the other hand, dynamic convolution causes only a small increase in runtime because the dynamic overhead is at the weight level. The major computational cost of the dynamic convolution is still the 3x3 convolution, making it perfect for super-resolution task.

The assembled block contains a control module and three assembled convolutions. As shown in Fig.4, the control module can generate the assembled coefficients from the input feature map:

$$coeff = F_{control}(f_{in}), \tag{2}$$

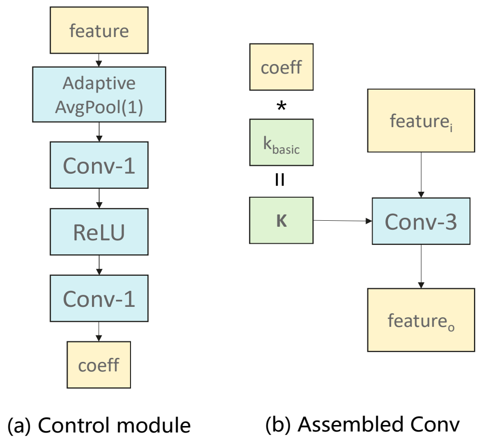

Figure 4. Details of the assembled block. Conv-1 indicates a convolution whose kernel size is 1×1.

where f_{in} is the input feature, $f_{in} \in R^{B\times C_i\times H\times W}$, $coeff$ is the output coefficient, $coeff \in R^{B\times C_o\times E}$, B represent the batch size, C_i denotes the number of input channels, C_o is the number of output channels, and E is the number of candidate convolution kernels. The control module converts features into coefficients through pooling and convolution. Let

$$K = coeff \otimes k_{basis}, \tag{3}$$

$$f_{out} = conv(f_{in}; K), \tag{4}$$

where $k_{basis} \in R^{E\times C_i\times ks\times ks}$ is the candidate convolution kernels, K is the assembled kernels for the convolution and ks is the kernel size. Coefficient $coeff$ and candidate kernels k_{basis} are first reshaped to $R^{(B\cdot C_o)\times E}$ and $R^{E\times(C_i\cdot ks\cdot ks)}$ respectively. Then, matrix multiplication is performed over the coefficient $coeff$ and candidate convolution kernels k_{basis}, to generate a final convolution kernel $K \in R^{B\times C_o\times C_i\times ks\times ks}$. Different batches of data require different convolution kernels, the batch dimension of the feature map is reshaped to the channel dimension and the group convolution is used to calculate the output feature maps.

Comparison between dynamic convolution and assembled convolution is shown in Fig.5. Dynamic convolution generates the whole convolution kernel (all channels) in a linear combination of the bases, which can be expressed as:

$$K_{dynamic} = \sum_{i=0}^{E} coeff_i \times k_i^{dy},\ k_i^{dy} \in R^{C_o\times C_i\times ks\times ks}, \tag{5}$$

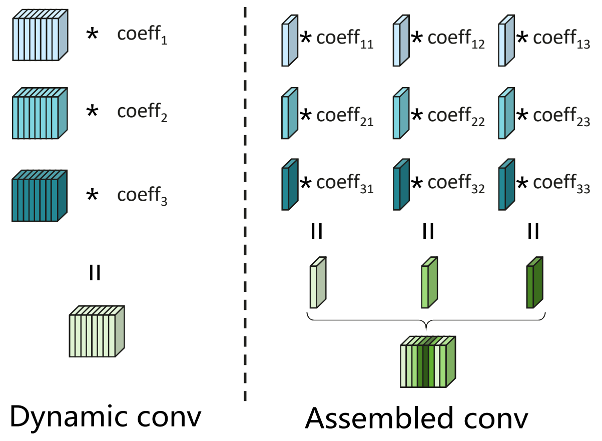

Figure 5. Comparison between dynamic convolution and assembled convolution.

where $K_{dynamic}$ is the kernel for convolution, $K_{dynamic} \in R^{C_o \times C_c \times ks \times ks}$, and k_i^{dy} is the candidate kernel which has the same dimension as $K_{dynamic}$. Assembled convolution generates an optimal convolution kernel coefficient for each channel, which can be expressed as:

$$K_j^{rs} = \sum_{i=0}^{E} coeff_{i,j} \times k_i^{rs},\ k_i^{rs} \in R^{C_i \times ks \times ks}, \qquad (6)$$

$$K_{assembled} = cat\left(K_0^{rs}, K_1^{rs}, ..., K_{C_o}^{rs}\right), \qquad (7)$$

where K_j^{rs} is the kernel for the output channel j, $K_j^{rs} \in R^{C_c, ks, ks}$, k_i^{dy} is the basis of the assembled kernel. $K_{assembled}$ for the convolution is assembled by concatenating all the K_j^{rs}. Compared with dynamic convolution, the assembled convolution we propose has finer granularity and higher flexibility in parameter generation, leading to a better performance in super-resolution.

4. Experiments

4.1. Setup

We use 3450 images in DIV2K [1] and Flick2K [30] (DF2K) datasets for training, and test the performance of our model on five benchmark datasets: DF2K (100 testing images), Set5 [3], Set14 [47], BSD100 [33] and Urban100 [20]. We evaluate PSNR and SSIM on RGB color space. Following the ranking criterion of NTIRE 2023 Real-Time Super-Resolution challenge, we measure the efficiency of a model by the following score function:

$$Score = \frac{2^{PSNR-bicubic} \cdot 2}{C \cdot \sqrt{runtime}}, \qquad (8)$$

where 'bicubic' means the PSNR of the bicubic interpolation, C is a constant set to 0.1 in our experiments. We adopt Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.9999$ to train our model. The learning rate is 5×10^{-4} in the initial stage and is halved for every 2×10^5 iterations. The entire training process takes 1×10^6 iterations to minimize the charbonnier loss. We randomly crop HR and LR patchs of sizes 256×256 and 128×128 from the training set. The mini-batch size is 32. We also augment the dataset by rotating and flipping the patches of the training pairs.

4.2. Quantitative Results

Method	FLOPs(G)	Params(M)	FP16
IMDN [21]	1808.98	0.87	211.52
RFDN [32]	824.04	0.42	178.15
RLFN [25]	592.36	0.30	97.98
FMEN [15]	671.36	0.32	128.29
RTSRN [17]	400.65	0.19	37.91
AsConvSR-L	36.67	5.21	24.61
AsConvSR	9.06	2.35	3.91

Table 1. FP16 indicates the runtime of a model running on half-precision floating-point in milliseconds (ms). And these runtimes are measured on a V100 GPU with 1920×1080 images as inputs. We also record the FLOPs and the number of parameters of each model

In this section, we compare AsConvSR with the state-of-the-art efficient super-resolution models. Specifically, IMDN [21], RFDN [32], RLFN [25], FMAN [15], and RTSRN [17] are chosen for experiments. We train two versions of our model, AsConvSR-L denoting the model with a larger scale and AsConvSR a smaller one. Specifically, AsConvSR-L has two assembled blocks, each of which has 128 channels and 128 candidate kernels. And AsConvSR has only one assembled block with 32 channels.

As shown in Tab.1, none of the methods can achieve the real-time performance with 1080p input except ours. AsConvSR has only 2.26% of RTSRN's FLOPs and 10.33% of RTSRN's runtime. Because assembled convolution has a large number of bases of weights, the number of parameters of our model is multiple times of the other model. However, our model is still much faster than other models, which indicates that the assembled convolution does not occupy many computing resources, and most of the computing budget is still consumed by the conventional convolution on the feature maps.

Quantitative comparisons of our model and other efficient SR models are demonstrated in Tab.2. Our method achieves the best scores on every benchmark dataset. AsConvSR takes only 3.91 ms, which is close to the bicubic interpolation, and achieves an improvement on PSNR for more than 1dB. Our AsConvSR is the only

Method	DF2K PSNR/SSIM/score	Set5 PSNR/SSIM/score	Set14 PSNR/SSIM/score	BSD100 PSNR/SSIM/score	Urban100 PSNR/SSIM/score
bicubic	29.81/0.8573	29.96/0.8576	27.38/0.7987	27.67/0.8022	24.98/0.7965
IMDN [21]	31.96/0.8933/6.10	32.21/0.8934/6.52	29.37/0.8409/5.43	29.27/0.8404/4.16	28.30/0.8730/13.71
RFDN [32]	31.96/0.8932/6.66	32.27/0.8941/7.39	29.32/0.8404/5.73	29.27/0.8397/4.54	28.36/0.8735/15.57
RLFN [25]	31.88/0.8915/8.47	32.14/0.8923/9.10	29.19/0.8376/7.05	29.20/0.8374/5.84	28.16/0.8696/18.21
FMEN [15]	31.84/0.8915/7.24	32.20/0.8935/8.34	29.22/0.8387/6.32	29.20/0.8389/5.11	28.07/0.8694/15.05
RTSRN [17]	31.52/0.8871/10.67	31.93/0.8886/12.67	28.96/0.8335/9.68	29.01/0.8345/8.20	27.49/0.8581/18.50
AsConvSR-L	31.62/0.8872/14.12	31.95/0.8888/15.98	29.01/0.8337/12.43	29.05/0.8351/10.46	27.65/0.8616/25.68
AsConvSR	30.87/0.8766/21.05	31.33/0.8797/26.00	28.37/0.8202/19.96	28.60/0.8249/19.23	26.50/0.8365/28.98

Table 2. Quantitative results on DF2K [1,30], Set5 [3], Set14 [47], BSD100 [33] and Urban100 [20] ($\times 2$). Red indicates the best and blue indicates the second best. The scores in this table are calculated by using the Eq.8. And the runtime of each model is obtained from Tab.1

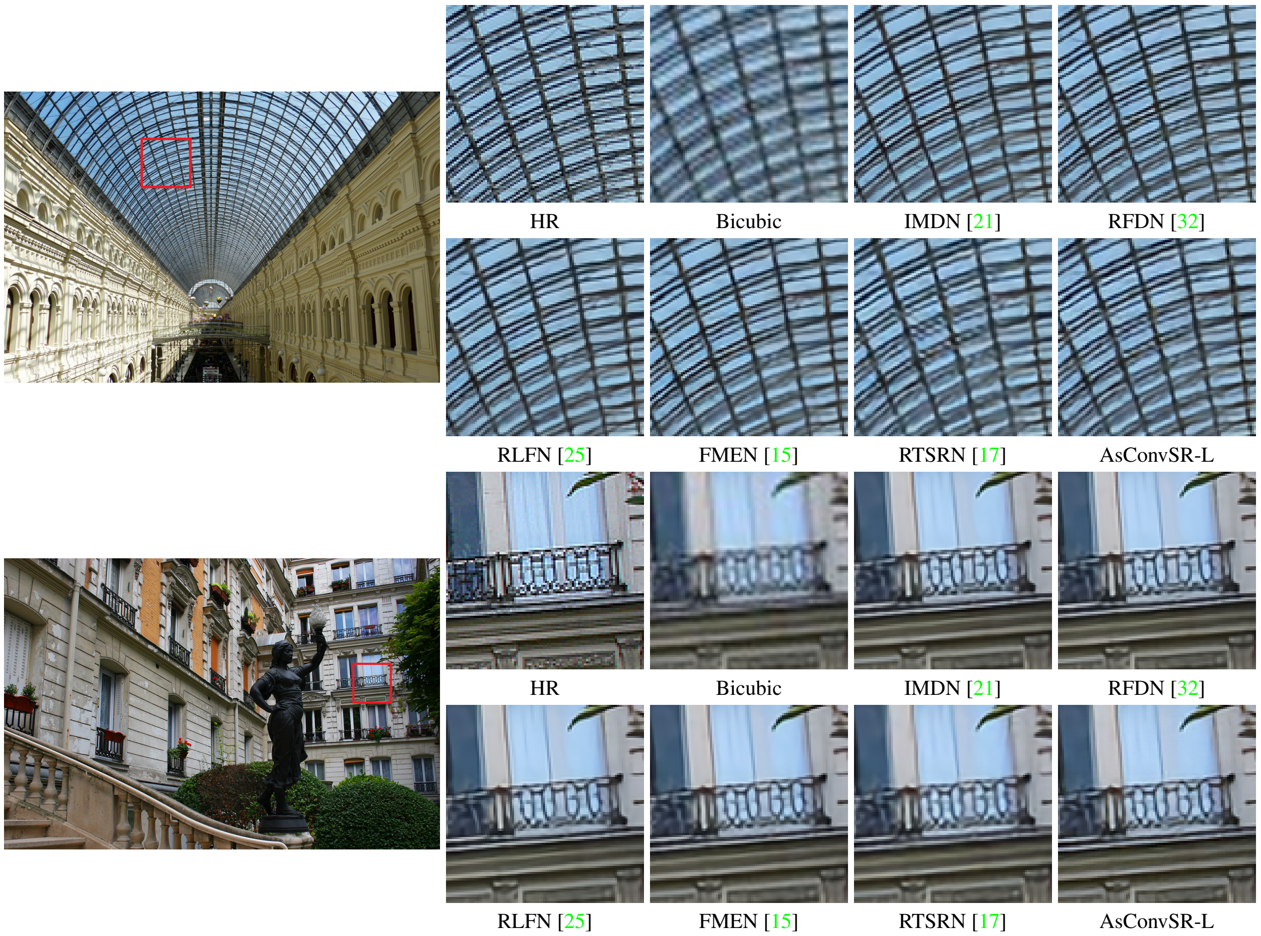

Figure 6. Visual comparison of efficient SR models. Best viewed by zooming.

method that achieves real-time performance with 1080p inputs. As demonstrated in Tab.1, AsConvSR-L exceeds RTSRN in PSNR with only 64.90% of runtime.

4.3. Visual Comparison

Fig.6 demonstrates the visual results of other efficient SR models and our AsConvSR-L. AsConvSR-L outperforms RTSRN in restoring compressed textures, and shows competitive performance in sharpness compared with RLFN [25] and FMEN [15]. Fig.7 shows the visual results of AsConvSR. The visual performance of AsConvSR significantly surpasses bicubic interpolation by using only 3.91 ms, and achieves similar quality to RTSRN. In general, both AsConvSR-L and AsConvSR keep the restoration accuracy and significantly reduce the model runtime.

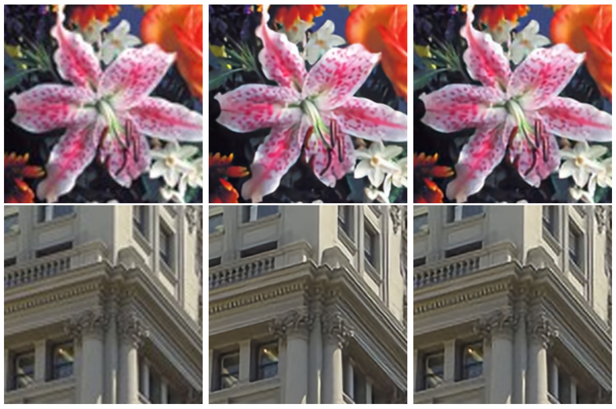
(a) Bicubic (b) RTSRN (c) AsConvSR

Figure 7. Visual results of AsConvSR and RTSRN on Set14 [47] and Urban100 [20].

4.4. Ablation Study

In this section, we evaluate the relationship between fidelity improvement and runtime consumption of various network structures. The ablation study is performed on the DF2K dataset, and Eq.8 is used as the efficiency criterion.

Method	Runtime	PSNR	Score
RLFN	97.98	31.88	8.47
w/o ESA	77.91	31.83	9.17
repeat upscaling	75.78	31.82	9.21

Table 3. Ablation study on the Enhanced Spatial Attention (ESA) module and repeat upscaling.

We verify the efficiency of ESA and repeat upscaling using RLFN as the baseline. As demonstrated in Tab.3, adding ESA does improve network performance on PSNR, but it also increases the running time by 20.07 ms, which is 25.72% of the runtime for the whole model. This phenomenon shows that ESA is not efficient enough with large resolutions inputs. Then we replace the bicubic interpolation in the model with repeat upscaling, which reduces the runtime by 2.13 ms. Given the standard of real-time 30 ms, using repeat upscaling can save 7.1% of runtime, indicating that repeat upscaling is practical and efficient.

Method	Runtime	PSNR	Score
Pixel-Unshuffle(1)	22.58	31.39	12.53
Pixel-Unshuffle(2)	15.72	31.33	14.46
Pixel-Unshuffle(3)	15.49	31.23	13.30
Pixel-Unshuffle(4)	12.56	31.05	13.31

Table 4. Comparison of the pixel-unshuffle factor. Values in the brackets indicate different factors.

The comparisons of the pixel-unshuffle factor are given in Tab.4. 'Pixel-Unshuffle(1)' means taking images of the original resolution as the input for the network, without using pixel-unshuffle to sample extra channel data from height and width. 'Pixel-Unshuffle(x)' means to sample pixels with interval of x from the height and width dimensions to the channel dimension. A larger x results in a larger number of channels. Therefore, for the experiments in Tab.4, the number of channels for the corresponding networks are 64, 128, 192, and 256 respectively, which can ensure the consistence of FLOPs in each experiment.

As demonstrated in the Tab.4, although the number of FLOPs is consistent in each experiment, the runtime and PSNR both decrease when the pixel-unshuffle factor increases. First, the performance in pixel-unshuffle experiment indicates that convolutions with a larger channel size and a smaller resolution run faster in NVIDIA GPU even the FLOPs remains unchanged. Second, pixel-unshuffle disrupts the spatial distribution of features and deteriorate the performance of the model. Finally, we choose 2 as the pixel-unshuffle factor in the following experiments.

Method	Runtime	PSNR	Score
residual	15.72	31.33	14.46
w/o residual	15.21	31.34	14.79
w/o global skip	15.16	31.24	13.83
w/o residual&bias	14.14	31.35	15.43

Table 5. Ablation study on the residual structure, bias and global skip connection.

As demonstrated in Tab.5, we re-evaluate the efficiency of residual structure and bias of convolution. For the experiments without residual structure, we increase the values of certain kernel in convolution by 1 in the initialization phase. According to the theory proposed by RepOpt [11], such modification on initialization is equivalent to adding a skip connection to the corresponding convolution layer without explicitly adding this structure.

By removing the residual structure in the network, the performance is maintained and the runtime is decreased. On the contrary, global skip connection demonstrates its necessity as the PSNR drops greatly after the removal of global skip connection. In addition, training also becomes unstable after the removal of global skip connection. In conclusion, it is not preferable to remove the global skip connection from the super-resolution model. We also find that removing bias can improve the efficiency of the network. So we set up a new baseline without residual structure and bias for the following experiments.

We compare the efficiency of the model with different channel sizes in Tab.6. The FLOPs and runtime do not proportionally decrease or increase when the number of channels changes. The FLOPs of a 128-channel network

Method	Runtime	PSNR	Score
channels(128)	14.14	31.35	15.43
channels(64)	7.01	31.08	18.16
channels(32)	3.73	30.69	19.04
channels(16)	3.33	30.38	16.26

Table 6. Comparison of models with different channel sizes.

is almost four times that of a 64-channel network, but the runtime is only two times. However, considering the deterioration speed on PSNR, the 64-channel network still wins the competition in the efficiency score. In addition, the speed increase caused by reducing the number of channels has a limit. On the Tesla V100 platform, the reduction in runtime is no longer obvious, when the network is reduced to 16 channels. Given consideration on both the runtime and PSNR, we consider 32 as the optimal number of channels for our models.

Method	Runtime	PSNR	Score
w/o dynamic	3.73	30.69	19.04
dynamic	3.84	30.83	20.62
assembled	3.91	30.87	21.05

Table 7. Comparison between dynamic convolution and assembled convolution.

Comparisons on dynamic and assembled convolutions are shown in Tab.6. Replacing classical convolution with dynamic convolution only slightly increases the runtime (0.11 ms), but improves the PSNR by 0.14dB, which verifies the efficiency of the idea of divide-and-conquer. Compared with dynamic convolution, assembled convolution can keep the runtime basically unchanged while improving the final score by 0.43, achieving highest efficiency.

To compare the efficiency gains of each design, we collect all above experiments results into Fig. 8. Adopting pixel-unshuffle, removal of residual and bias does not deteriorate the performance of the model in PSNR, and reduces runtime at the same time. Reducing the number of channels in the network increases the score by greatly reducing the runtime, but also reduces the PSNR. Assembled convolution can improve the PSNR without increasing the runtime, resulting in a significant score improvement and achieving highest efficiency among all the models.

4.5. AsConvSR for NTIRE 2023 Real-Time Super-Resolution Challenge

Our model wins the first place in NTIRE 2023 Real-Time Super-Resolution - Track 1 ($\times 2$). The major difference in training the competition model is the training datasets. In the competition we use DIV2K [1], Flick2K [30], DIV8K [19], GTAV [36], and LIU4K-V2 [31] for training. Other hyperparameters are consistent with the above experiments. As demonstrated in Tab.8, our AsConvSR exceeds the second best model in the PSNR score by 0.39 dB and ranked first with the highest score.

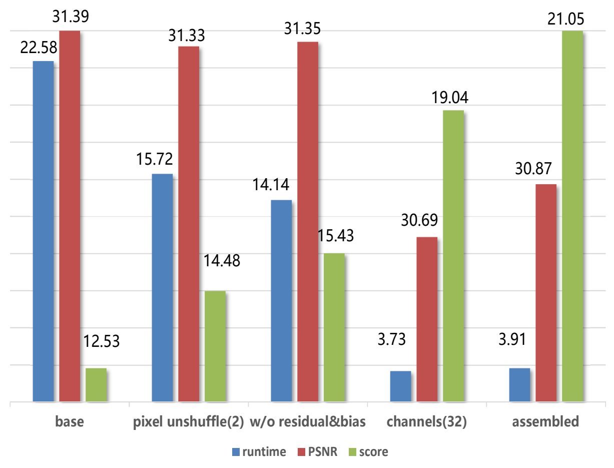

Figure 8. Comparison of models with different efficient designs. 'base' is a vgg-style model with 64 channels and repeat upscaling.

Method	Runtime	PSNR/SSIM /PSNR-Y	Score
Bicubic	–	33.92/0.8829/36.66	–
Baseline [8]	7.09	34.22/0.8854/–	9.27
DFCDN Team	4.67	34.63/0.8916/37.46	15.17
Team OV	2.91	34.62/0.8899/37.45	19.06
RTVSR	2.24	34.71/0.8910/37.50	23.13
ALONG	1.91	34.63/0.8906/37.38	23.81
AsConvSR	3.19	35.02/0.8957/37.74	24.13

Table 8. Ranking of the NTIRE 2023 Real-Time Super-Resolution track 1 [8]. Top 5 methods are included.

5. Conclusion

In this paper, we propose a fast and lightweight super-resolution network with assembled convolution for real-time super-resolution. We revisit the efficiency of several designs such as pixel-unshuffle, repeat upscaling, residual and bias removal. Furthermore, we design a lightweight block named assembled block which can adaptively assembles the convolution kernels according to the input features. By introducing these designs, our model runtime is significantly reduced while an excellent super-resolution performance is kept. Quantitative experiments demonstrate the competitive performance of our model.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 5, 6, 8

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision*, pages 252–268, 2018. 2

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*. BMVA press, 2012. 5, 6

[4] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pages 527–536. PMLR, 2017. 2

[5] Jin Chen, Xijun Wang, Zichao Guo, Xiangyu Zhang, and Jian Sun. Dynamic region-aware convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8064–8073, 2021. 2

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020. 2

[7] Marcos V Conde, Ui-Jin Choi, Maxime Burchi, and Radu Timofte. Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration. In *Proceedings of Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 669–687. Springer, 2023. 2

[8] Marcos V Conde, Eduard Zamfir, Radu Timofte, et al. Efficient deep models for real-time 4k image super-resolution. ntire 2023 benchmark and report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 8

[9] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018. 2

[10] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6192–6201, 2019. 2

[11] Xiaohan Ding, Honghao Chen, Xiangyu Zhang, Kaiqi Huang, Jungong Han, and Guiguang Ding. Re-parameterizing your optimizers rather than architectures. *arXiv preprint arXiv:2205.15242*, 2022. 7

[12] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021. 2

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015. 2

[14] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Proceedings of Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 391–407. Springer, 2016. 2

[15] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 853–862, 2022. 2, 5, 6

[16] Zongcai Du, Jie Liu, Jie Tang, and Gangshan Wu. Anchor-based plain net for mobile image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2494–2502, 2021. 4

[17] Ganzorig Gankhuyag, Jingang Huh, Myeongkyun Kim, Kihwan Yoon, HyeonCheol Moon, Seungho Lee, Jinwoo Jeong, Sungjei Kim, and Yoonsik Choe. Skip-concatenated image super-resolution network for mobile devices. *IEEE Access*, 2022. 1, 3, 5, 6

[18] Guangwei Gao, Wenjie Li, Juncheng Li, Fei Wu, Huimin Lu, and Yi Yu. Feature distillation interaction weighting network for lightweight image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 661–669, 2022. 2

[19] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3512–3516. IEEE, 2019. 8

[20] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 5, 6, 7

[21] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2024–2032, 2019. 2, 5, 6

[22] Andrey Ignatov, Radu Timofte, Maurizio Denna, and Abdel Younes. Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2525–2534, 2021. 1

[23] Andrey Ignatov, Radu Timofte, Maurizio Denna, Abdel Younes, Ganzorig Gankhuyag, Jingang Huh, Myeong Kyun Kim, Kihwan Yoon, Hyeon-Cheol Moon, Seungho Lee, et al. Efficient and accurate quantized image super-resolution on mobile npus, mobile ai & aim 2022 challenge: report. In *Proceedings of Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 92–129. Springer, 2023. 1

[24] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion

compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3224–3232, 2018. 2

[25] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 766–776, 2022. 2, 5, 6

[26] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12016–12025, 2021. 4

[27] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 624–632, 2017. 2

[28] Yawei Li, Kai Zhang, Radu Timofte, Luc Van Gool, Fangyuan Kong, Mingxi Li, Songwei Liu, Zongcai Du, Ding Liu, Chenhui Zhou, et al. Ntire 2022 challenge on efficient super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1062–1102, 2022. 1

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 2

[30] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017. 5, 6, 8

[31] J. Liu, D. Liu, W. Yang, S. Xia, X. Zhang, and Y. Dai. A comprehensive benchmark for single image compression artifacts reduction. In *arXiv*, 2019. 8

[32] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *Proceedings of Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 41–55. Springer, 2020. 1, 2, 5, 6

[33] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 5, 6

[34] Mason McGill and Pietro Perona. Deciding how to decide: Dynamic routing in artificial neural networks. In *International Conference on Machine Learning*, pages 2363–2372. PMLR, 2017. 2

[35] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018. 2

[36] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016. 8

[37] Huiyu Wang, Aniruddha Kembhavi, Ali Farhadi, Alan L Yuille, and Mohammad Rastegari. Elastic: Improving cnns with dynamic scaling policies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2258–2267, 2019. 2

[38] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 409–424, 2018. 2

[39] Yan Wang. Edge-enhanced feature distillation network for efficient super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 777–785, 2022. 2

[40] Fei Wu, Feng Chen, Xiao-Yuan Jing, Chang-Hui Hu, Qi Ge, and Yimu Ji. Dynamic attention network for semantic segmentation. *Neurocomputing*, 384:182–191, 2020. 2

[41] Jialin Wu, Dai Li, Yu Yang, Chandrajit Bajaj, and Xiangyang Ji. Dynamic filtering with large sampling field for convnets. In *Proceedings of the European Conference on Computer Vision*, pages 185–200, 2018. 2

[42] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6556–6565, 2018. 2

[43] Yu-Syuan Xu, Shou-Yao Roy Tseng, Yu Tseng, Hsien-Kai Kuo, and Yi-Min Tsai. Unified dynamic convolutional network for super-resolution with variational degradations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12496–12505, 2020. 2

[44] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: conditionally parameterized convolutions for efficient inference. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 1307–1318, 2019. 2

[45] Zerui Yang, Yuhui Xu, Wenrui Dai, and Hongkai Xiong. Dynamic-stride-net: Deep convolutional neural network with dynamic stride. In *Optoelectronic Imaging and Multimedia Technology VI*, volume 11187, pages 42–53. SPIE, 2019. 2

[46] Eduard Zamfir, Marcos V Conde, and Radu Timofte. Towards real-time 4k image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1

[47] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proceedings of Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pages 711–730. Springer, 2012. 5, 6, 7

[48] Yikang Zhang, Jian Zhang, Qiang Wang, and Zhao Zhong. Dynet: Dynamic convolution for accelerating convolutional neural networks. *arXiv preprint arXiv:2004.10694*, 2020. 2

[49] Yin-Dong Zheng, Zhaoyang Liu, Tong Lu, and Limin Wang. Dynamic sampling networks for efficient action recognition in videos. *IEEE Transactions on Image Processing*, 29:7970–7983, 2020. 2