Adaptive Hybrid Composition Based Super-Resolution Network via Fine-Grained Channel Pruning

DOI: 10.1007/978-3-030-67070-2_7
Handle: http://hdl.handle.net/1942/33257

ECCV #52

Adaptive Hybrid Composition based Super-Resolution Network via Fine-grained Channel Pruning

Siang Chen[1[0000−0003−2846−781X]], Kai Huang[1[0000−0003−2295−5433]⋆], Bowen Li[1[0000−0001−7525−9672]], Dongliang Xiong[1[0000−0002−4882−7504]], Haitian Jiang[1[0000−0001−8215−166X]], and Luc Claesen[2[0000−0003−0405−6290]]

[1] Zhejiang University, Hangzhou, China
{11631032, huangk, 11631033, xiongdl, jianghaitian}@zju.edu.cn
[2] Hasselt University, 3590 Diepenbeek, Belgium
luc.claesen@uhasselt.be

Abstract. In recent years, remarkable progress has been made in single image super-resolution due to the powerful representation capabilities of deep neural networks. However, the superior performance is at the expense of excessive computation costs, limiting the SR application in resource-constrained devices. To address this problem, we firstly propose a hybrid composition block (HCB), which contains asymmetric and shrinked spatial convolution in parallel. Secondly, we build our baseline model based on cascaded HCB with a progressive upsampling method. Besides, feature fusion method is developed which concatenates all of the previous feature maps of HCB. Thirdly, to solve the misalignment problem in pruning residual networks, we propose a fine-grained channel pruning that allows adaptive connections to fully skip the residual block, and any unimportant channel between convolutions can be pruned independently. Finally, we present an adaptive hybrid composition based super-resolution network (AHCSRN) by pruning the baseline model. Extensive experiments demonstrate that the proposed method can achieve better performance than state-of-the-art SR models with ultra-low parameters and Flops.

Keywords: single image super-resolution, efficient model, channel pruning

1 Introduction

Single image super-resolution (SISR) is a classic computer vision task that reconstructs a high-resolution (HR) image from its degraded low-resolution (LR) version. It has broad applications in photo editing, medical imaging and object detection. Although numerous methods have been proposed for SISR, it is still an active and challenging task as an ill-posed problem.

⋆ Corresponding author

Recently, deep learning based methods have shown superior performance compared with previous example-based methods. After [3] first developed a convolutional neural network (CNN) to establish a mapping between LR and HR images, various networks have been proposed to boost the overall performance of image super-resolution. However, the significant improvement always comes at the expense of a large amount of parameters and high computation cost, which is not suitable to be deployed on resource-limited devices.

To tackle this problem, a natural idea is designing light-weight neural netorks. For example, FSRCNN [4] and ESPCN [26] reduce model size by building shallow network models, [1] and [26] utilize squeeze and group operations to construct efficient super-resolution blocks. Another trend is to use recursive operators or parameter sharing strategy, such as DRCN [18],DRRN [28].

In addition to designing efficient networks, compressing pre-trained deep neural networks is also helpful in deriving the optimal architectures. Pruning is an effective method to reduce the redundancy in networks by removing those unimportant individual neurons with negligible performance degradation. While doing this reduces the theoretical size of the model, it does not result in real computation cost or memory footprint reduction unless special hardware and software are designed. Therefore, channel pruning [23, 33, 15] is proposed to implement real speed up and memory footprint reduction on general hardwares(CPU/GPU) by removing the whole filters in networks. While this compression method has shown state-of-the-art accuracy on image classification problems, channel pruning has rarely been investigated for the efficient image super-resolution task. In addition, residual learning has been widely employed by SR models, which ease the task by learning only the residuals between input and output images, and alleviate the vanishing problem as well. However, pruning residual networks is challenging due to the constraints induced by the cross layer connections. Recent works [2, 7, 34] propose to assign channels connected by skip connections in the same group and prune them simultanesly, while solving the constraint problem, the pruning ratio on these troublesome filters is limited.

In this paper, we firstly propose a hybrid composition based super-resolution neural network (HCSRN). To leverage the efficiency of different kernel size and resolution, we design a hybrid composition block (HCB) which contains asymmetric convolution and shrinked spatial convolution, and then construct HCSRN based on HCB via a progressive upsampling method. Secondly, we take HCSRN as our baseline model and apply pruning on it to further reduce parameters and FLOPs. To solve the constraint that the pruning problem encounters when pruning residual blocks, we propose a novel fine-grained channel pruning strategy that allows any channel to be pruned independently, which breaks the monotonous design constraint in residual neural networks. Finally, we obtain an efficient pruned model called AHCSRN based on adaptive hybrid composition blocks with different weights for asymmetric convolution and shrinked spatial convolution blocks, as well as adaptive local feature fusion connections. Extensive experiments show that the proposed AHCSRN can achieve better

performance with ultra-low parameters and FLOPs compared to state-of-the-art methods.

In summary, the main contributions of this paper are as follows:

1)We propose HCSRN for image super-resolution, a basic neural network based on the hybrid composition modules (HCB), thanks to the efficient compositions in hybrid modules, our HSRN achieves high performance on SR task with a modest number of parameters.

2)We propose the adaptive hybrid composition based super-resolution network (AHCSRN) with ultra-low parameters and FLOPs while still keeping high performance. By applying fine-grained channel pruning (FCP) on HCSRN, we not only reduce channels, but also derive adaptive hybrid modules with different weights on asymmetric and shrinked spatial convolutions. Moreover, the proposed FCP avoids the misalignment problem for pruning residual networks, and results in a novel efficient residual architecture.

3)Experimental results show that the proposed lightweight AHCSRN achieves superior performance than the state-of-the-art methods with ultra-low parameters and computation cost.

2 Related Work

2.1 Deep Learning Based Super-Resolution

Convolutional Neural Network (CNN) has shown great success in image super-resolution. [3] firstly employ CNN with tree layers (SRCNN) to learn the SR task, which achieves superior performance than previous example-based methods [5, 6, 30]. After that, various improved algorithms have been proposed. [17] explore a deeper network named VDSR with 20 convolution layers, whichs show noticeable progress than SRCNN. [22] utilize residual network to ease the training of deep networks and make the neural network go deeper which is denoted as EDSR. RCAN [36] even built an SR network with more than 400 layers with channel attention mechanisms to further improve performance. However, with the networks going deeper, the number of parameters and Flops are also dramatically increasing, which limits the real-world applications on resource-constrained devices. Therefore, there is an urgent need to design light-weight SR networks.

FSRCNN [4] reduces the computation cost of SRCNN by removing the preprocessing bicubic interpolation and upscales the image at the end of the network. DRRN [28], MemNet [29] share parameters through recursive mechanism to avoid introducing new parameters while improving the reconstruction quality. [1] propose a cascading residual network (CARN) to learn the LR-HR mapping more efficiently. Recently, [16] propose a light-weight information multi-distillation network by constructing the cascaded information multi-distillation blocks, which shows a better tradeoff between computation cost and performance. And [31] design an architecture that makes full use of the features by adaptive weighted residual connections.

2.2 Channel Pruning

Model pruning is a predominant approach in learning compressed light-weight neural networks by removing unimportant neurons. Early works [9, 8] propose to remove individual weight values, despite the deep compression of parameters, such pruning strategy results in non-structured sparsity in the network, and practical runtime acceleration cannot be achieved unless special custom software and hardware are designed. Therefore recent works focus more on filter pruning (a.k.a. channel pruning) which is a universal technique that can be applied to various types of CNN models, and benefits from inference speedup and memory saving as well. Some leverage heuristic metric to evaluate the importance such as the magnitude of filters [12], the average percentage of zero activations [14], and the geometric median criterion [13]. Some methods add a regularization function such as L_1 [24], Group LASSO [32] to the loss function to induce sparsity. However, due to the cross layer connections in residual neural networks, methods for pruning plain networks such as VGG and AlexNet cannot be applied directly. To address the misalignment problem of feature maps in the shortcut connection, several solutions have been proposed. [11] only prune internal channels layers in residual blocks. [24] place a channel selection layer before the first convolution in each residual block to mask out insignificant channels, and leave the last convolution layer unpruned, which only works for pre-activation networks. [20] use a mixed block connectivity to avoid redundant computation. Recently, [2, 7, 34] propose to assign the layers connected by pure skip connections into the same group, thus the filters in the same group can be pruned simultaneously. However, although the above methods avoid the misalignment problem, pruning ratios on these troublesome filters are still limited, which results in non-optimal neural network structures.

3 Approach

3.1 Compression Flow

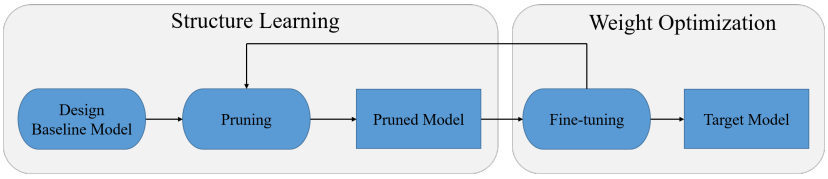


Fig. 1. The overall flow to obtain the light-weight model.

The goal of efficient super-resolution challenge in the 2020 ECCV AIM workshop is to devise a network that reduces one or several aspects such as runtime, parameters, FLOPs, activations, and depth while at least maintaining PSNR of

MSRResNet. Directly applying pruning on MSRResNet to reduce model size is not a reasonable choice, because the performance of the pruned model will be lower than the baseline model especially when the pruning ratio is large. Therefore, as shown in Figure 1 we should firstly design a larger but more efficient baseline model before pruning, then employ the pruning method to compress the model to get a better tradeoff between performance and model size. To reduce the model size as much as possible while keeping the performance higher than MSRResNet during validation, we utilize an iterative pruning and fine-tuning strategy to get the final model.

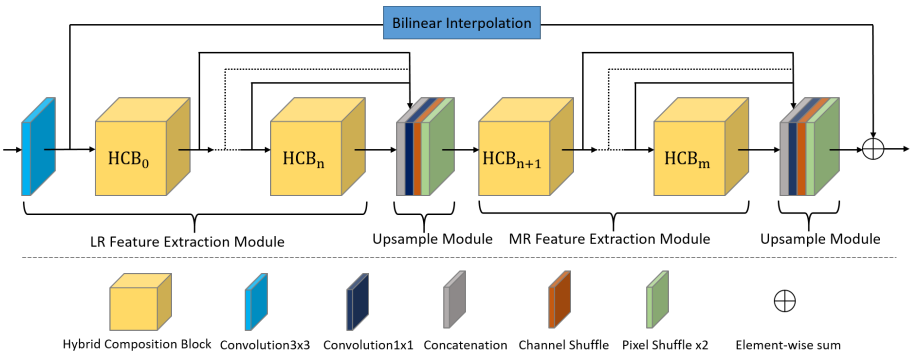


Fig. 2. The architecture of hybrid composition based super-resolution neural network (HCSRN)

3.2 Architecture of HCSRN

In this section, we describe our proposed baseline model, a hybrid composition based super-resolution neural network (HCSRN) in detail. Figure 2 shows the architecture of HCSRN. We employ the progressive upsampling strategy that decomposes the image space in HCSRN into low-resolution (LR, H × W), middle-resolution (MR, 2H × 2W) and high resolution (HR, 4H × 4W), which is divided by two upsampling modules. HCSRN consists of five modules, namely the feature extraction module, the LR hybrid composition module, the first upsampling module, the MR hybrid composition module and the second upsampling module.

The feature extraction module is a convolution layer with kernel size of 3×3, which can be formulated as

$$LHBF_0 = f_{FE}(I_{LR}) \tag{1}$$

where I_{LR} is the input LR image, f_{FE} denotes the feature extraction function, and $LHBF_0$ is the output feature map from the first convolution layer.

In the LR hybrid composition module (LRHCM), there are n numbers of the proposed sequential hybrid composition blocks (HCB), the function can be

expressed as

$$LHBF_i = f_{LHB_i}(LHBF_{i-1}) \tag{2}$$

where i is 1,2,...,n, $LHBF_i$ is the output of ith hybrid block, f_{LHB_i} denotes the corresponding function, of which the details will be described in section 3.3.

For the upsampling module, assume the input feature size to be H × W × C, and scaling factor to be s^2, the first upsampling module reshapes the image size to be sH × sW × C.

$$MHBF_0 = f_{LRUM}(LHBF_1, LHBF_2, ..., LHBF_n) \tag{3}$$

where $MHBF_0$ is the upscaled feature, f_{LRUM} denotes the function of the LR upsampling module (LRUM). Specifically, we firstly employ the feature fusion method that concates all of the previous feature maps output by each LHBF in the channel dimension. Secondly, we perform a convoltuion with kernel size 1x1 that reduces the channel number from n × C to s^2 × C. Thirdly, the channel shuffle layer proposed in ShuffleNet [35] is used to perform a channel reorder operation. Finally, the pixel shuffle layer upscales feature maps to sH × sW × C.

In the MR hybrid composition module (MRHCM), there are m hybrid blocks, and the architecture is the same as LRHCM except that the feature map is of size sH × sW × C.

$$MHBF_i = f_{MHB_i}(MHBF_{i-1}) \tag{4}$$

where i is 1, 2,...,m.

To further compress the model size, instead of upsampling feature maps and follows a reconstruction module that reshapes the image to the final HR size, our MR upsampling module (MRUM) upscales features of size sH × sW × C to the final HR image size directly. In addition, we apply the global residual learning and bilinear upsampling operator, the output of the HCSRN is the element-wise sum of MRUM's output and the interpolated image.

$$I_{SR} = f_{MRUM}(MHBF_1, MHBF_2, ..., MHBF_m) + f_{Bilinear}(I_{LR}) \tag{5}$$

where f_{MRUM} and $f_{Bilinear}$ are the function of the MR upsampling module and bilinear upsampling operator respectively, and I_{SR} denotes the output of HCSRN.

3.3 Basic Hybrid Block

As depicted in Figure 3, our hybrid composition block is constructed by three parallel blocks, of which the element-wise sum operation is utilized for local feature fusion. The whole block adopts the residual connection. The main idea of this module is extracting useful features by different efficient blocks.

One of the methods to depress the computation of networks is reducing the kernel size of convolutions. We adopt the same idea as [27] that employ the

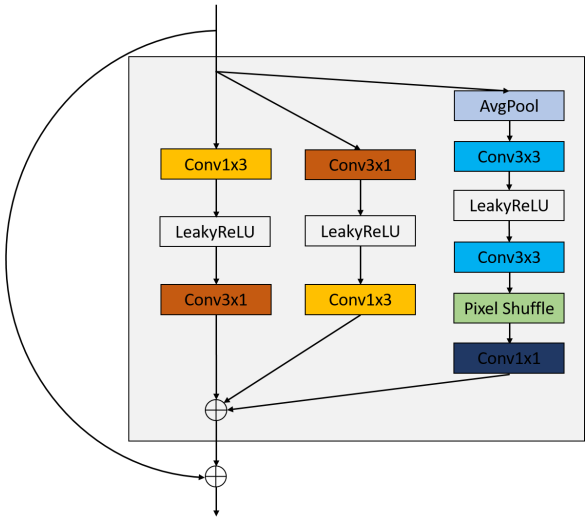


Fig. 3. The architecture of our proposed hybrid composition block (HCB).

asymmetric kernel. However, instead of using 5×1 and 1×5 kernels, we aggressively use smaller kernel size that factorizes the normal 3×3 Conv into an 3×1 Conv followed by a 1×3 Conv, thus the parameters and operations decrease dramatically from $O(3^2)$ to O(2×3). To extract different features without significant performance drop, two asymmetric blocks with inversed kernel orders are utilized.

Another way is reducing the scale of features. Assume the input feature is of size H × W × C, an average pooling layer with kernel size 2×2 is firstly adopted to shrink the feature from H × W × C to (H/2) × (W/2) × C. Then we use the same sequential Conv-LeakyRuLU-Conv structure to extract features. In order to do element-wise sum operations with the other two parallel blocks, sub-pixel convolution is used for reconstructing the high-resolution image of size H × W × (C/4) due to its efficiency. Finally, a convolution with kernel 1×1 is adopted to expand the channel number back to C. While reducing the computation cost, scaling the feature also expands the receptive field to obtain more context information, which is helpful to extract different features for local fusion.

3.4 Fine-grained Channel Pruning

Although the efficient blocks in HCB provide different features for local fusion, it's hard to determine the weights for these features manually especially when the model size is limited. In addition, which are the most important features that the upsampling module needs remains a question. Therefore, we utilize pruning method to remove those channels that contribute little to the quality of the reconstruction image.

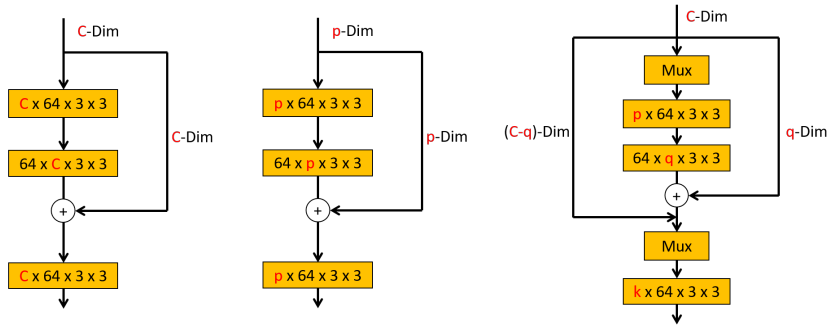


Fig. 4. The difference between our FCP and group pruning. **(a)**: baseline structure. **(b)**: structure pruned by group strategy. **(c)**: structure pruned by our fine-grained strategy. The red letter denotes the channel number.

We reduce the redundancy in upsampling modules and convolutions in HCB by asserting the gating function, which can be expressed as

$$g(\alpha) = \begin{cases} 0, & IS(\alpha) < T \\ \alpha, & otherwise \end{cases} \tag{6}$$

where α is a scaling factor that multiplied on each channel, $IS(\alpha)$ denotes the importance score of each channel, T is the global score threshold that depends on the pruning ratio. For the importance criterion, we utilize the algorithm in [25, **?**] that estimate the change in loss function caused by setting α to zero, which can be easily computed during back-propagation.

The difference between our fine-grained channel pruning strategy and the previous grouping method is shown in Figure 4. Grouping method assigns the channels connected by the skip connection into a group, and importance scores for channels in the same group are accumulated which makes these troublesome filters harder to be pruned. Instead of only considering the output channel of each convolution, we try to assert the gating function before and after each convolution, and allow each channel to be pruned independently. To avoid the misalignment problem between the convolution and the skip connection, we do not prune the skip connections. Figure 5(a) shows the possible structures pruned by our proposed FCP. For representational simplicity, we only show structures of one path, while the condition is the same for pruning the other two paths and the upsampling module. Note that since there is no difference for pruning channels inner residual blocks between FCP and other methods, these channels are not considered in Figure 5(a).

1)Only prune the input channel of the first convolution in residual blocks. As shown in Figure 5(b), channel 0 will totally skip Resblock_l, and directly perform the element-wise operation with the output channel of Resblock_l.

2)Only prune the output channel of the last convolution in residual blocks. As shown in Figure 5(c), the output channel 0 of Resblock_l is removed, however,

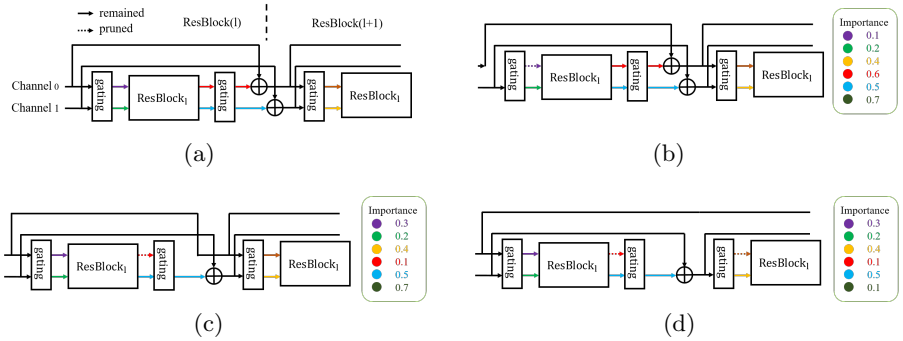


Fig. 5. Illustration of possible architectures pruned and reconstructed by FCP in channel-wise view, the dotted lines denote pruned channels. **(a)**: structure before pruning. **(b)**: only prune input channel. **(c)**: only prune output channel. **(d)**: prune both input and output channels

we do not prune the input channel 0 of Resblock$_{l+1}$, previous channel 0 will bypass the element-wise operation and become the input of Resblock$_{l+1}$.

3)Prune both input and output channels. Figure 5(d) shows the condition, although the channel 0 between Resblock$_l$ and Resblock$_{l+1}$ is removed, the input channel 0 of Resblock$_l$ will bypass these two blocks and flow into Resblock$_{l+2}$, which leverages the full use of residual information.

4 Experiments

4.1 Datasets and metric

We use the DIV2K and Flicker2K datasets as our training set, which contains 800 and 2650 high-resolution images respectively. The HR images are cropped into small images with size 480 × 480 and we downscale the HR images using bicubic interpolation to produce LR images. The LR patches with size of 96 × 96 are randomly cropped from LR images as the input of our model. Data augmentation is performed on the training set, such as random rotations of 90, 180, 270 and horizontal flips. For evaluation, we use five standard benchmark datasets: Set5, Set14, BSD100, Urban100 and Managa109. We evaluate the performance of the SR images using the peak signal-to-noise ratio(PSNR) and structure similarity index (SSIM). The results are calculated on Y channel of transformed YCbCr space, and the scaling factor is ×4 in all our experiments.

4.2 Implementation details

Training Baseline model. In all our experiments, we set the number of LR HCB (n) and MR HCB (m) to be 12 and 4 respectively. The model is trained by L1 loss with cyclic cosine annealing schedule. The restart learning rate is set

Table 1. Quantitative results of evaluated methods for ×4 SR

Method	Params	Flops	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	-	-	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
D-DBPN [10]	10426K	5925.3G	32.47/0.8980	28.82/0.7860	27.72/0.7400	26.38/0.7946	30.91/0.9137
RCAN [36]	15592K	1042.4	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	31.22/0.9173
SRCNN [3]	57K	59.9G	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCNN [4]	13K	5.2G	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [17]	668K	43.8G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
LapSRN [19]	818K	172.3G	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
DRRN [28]	302K	19.8G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.45/0.8946
MemNet [29]	677K	709.4G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
EDSR [22]	1518K	130.2G	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
CARN [1]	1592K	103.6G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.04/0.7838	30.45/0.9073
IMDN [16]	715K	46.7G	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
MSRResNet	1517K	166.7G	32.19/0.8943	28.64/0.7821	27.58/0.7356	26.12/0.7864	30.49/0.9079
HCSRN (Ours)	2216K	147.2G	32.43/0.8967	28.83/0.7867	27.71/0.7402	26.56/0.7999	31.10/0.9146
AHCSRN1 (Ours)	487K	36.7G	32.24/0.8949	28.70/0.7834	27.62/0.7371	26.23/0.7897	30.72/0.9105
AHCSRN2 (Ours)	354K	27.3G	32.18/0.8942	28.65/0.7824	27.59/0.7360	26.12/0.7860	30.58/0.9087

to 2×10^{-4}, while the minimum learning rate is 10^{-7}. Optimizer is configured as ADAM with $\beta_1 = 0.9$, $\beta_2 = 0.99$. Note that when training with multiple GPUs, we multiply the learning rate and mini-batch size with the number of GPUs. For example, when using 4 GPUs, the restart and minimum learning rate should be modified to 8×10^{-4} and 4×10^{-7}. If not stated otherwise, all the configurations are described in the 4 GPUs condition in the following. The mini-batch is set to 64 and we train 500000 iterations totally with 8 cosine annealing cycles (each cycle 62500 iterations).

Pruning. We employ an iterative pruning and finetuning strategy. The importance scores of channels are estimated every 800 iterations, and 2% of total channels are pruned away each time. After performing 10 times of such pruning, we finetune the model by 8000 iterations to recover the performance with learning rate linearly decreasing from 8×10^{-4} to 4×10^{-7}. When the compression ratio meets the requirement, we finetune the model for another 250000 iterations with 4 cosine annealing cycles, learning rate is set to the same as training baseline model.

4.3 Comparison with state-of-the-arts

Table. 1 shows the results of our baseline HCSRN, pruned model AHCSRN and other state-of-the-art SR models. For calculating Flops and parameters, we utilize the open-source tool THOP[3], and input image is of size $1\times3\times256\times256$.

Firstly, we compare our proposed models with the performance-oriented models. Although D-DBPN and RCAN achieve very high performance, the superior PSNR and SSIM are at the expanse of increased network depth and additional blocks, which result in too much computation costs. We notice that our baseline

[3] https://github.com/youzhonghui/pytorch-OpCounter

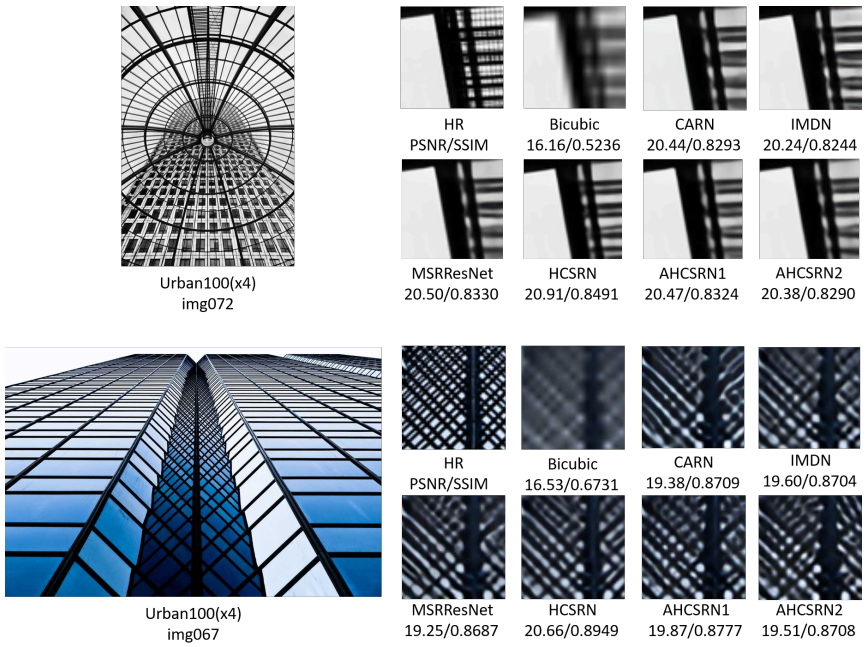


Fig. 6. Visual comparisons of HCSRN/AHCSRN with other SR methods.

model HCSRN has similar performance with D-DBPN, worse than D-DBPN on Set5 and BSD100 but better on Set14, Urban100 and Manag109, however, the parameters of D-DBPN is nearly 4.7 times of HCSRN, and the Flops is even 40.25 times of HCSRN. Therefore, the quantitative results show that our baseline model HCSRN archives better tradeoff among parameters, Flops and fidelity.

Secondly, we compare our pruned models with other light-weight methods. We show two variants of the pruned model (AHCSRN1 and AHCSRN2) of different pruning ratio. LapSRN also adopts the progressive upsampling strategy, which increases the computation cost in HR image space, therefore LapSRN has fewer parameters but more Flops than MSRResNet. In spite of employing progressive upsampling strategy, AHCSRN eliminates unnecessary computation in HR space by combining efficient blocks and pruning together, thus parameters and Flops of AHCSRN1 are both much less than MSRResNet while the performance is even higher. SRCNN, VDSR, CARN and DRRN have fewer parameters and Flops than MSRResNet, but these lightweight models all sacrifice performance to achieve such computation reduction. IMDN keeps the similar PSNR and SSIM as MSRResNet, and can reduce the parameters and Flops to 715K and 46.7G respectively. Our AHCSRN is even more efficient than IMDN, AHCSRN1 has much higher performance than IMDN with fewer parameters and Flops, the more lightweight version AHCSRN2 performs little worse on Set5, but

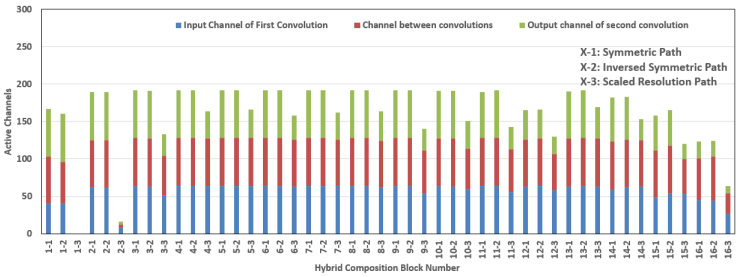
Table 2. Speed-up analysis on pruned models

Model	Params	FLOPs	Time (ms/img)	Realistic Speed-up(%)	Theoretical Speed-up(%)
HCSRN	2216K	147.2G	0.1659	-	-
AHCSRN1	487K	36.7G	0.1334	19.6	75.1
AHCSRN2	354K	27.3G	0.1180	28.9	81.5

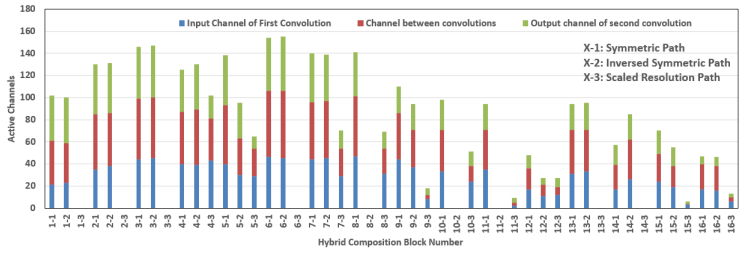
much better on Other four test datasets than IMDN. These experiment results validate the effect of pruning on HCSRN, and show that AHCSRN is more efficient than other state-of-the-art light-weight SR models.

Then we compare the visual results of HCSRN/AHCSRN with other state-of-the-art methods. We take images '067' and '072' from Urban100 dataset as examples, from Figure 6 we can see that image details of HCSRN, AHCSRN1 are recovered better than others, and AHCSRN2 has similar qualitative results with MSRResNet.

Table 2 shows the realistic speedup of our prune model. We measure the forward time with one RTX2028Ti GPU on the DIV2K validation dataset with batch size set to 1. The gap between theoretical and realistic models may come from the limitation of IO delay, buffer switch and efficiency of BLAS libraries.



(a)



(b)

Fig. 7. Channel number allocation of pruned model AHCSRN with parameters of **(a)**: 1503K. **(b)**: 354K.

Figure 7(a) and 7(b) shows the architecture of AHCSRN with 1503K and 534K parameters respectively. Apparently, the three parallel paths in hybrid composition block have different weights in different layers, the shrinked spatial convolution paths always have the smallest channel numbers while the other two paths have similar weights. However, as the pruning ratio goes deeper, we can see that in some layers such as layer 8/10/11 in Figure 7(b), the channel number in the inversed symmetric path is 0 and the shrinked spatial path is more important. In addition, channel numbers between each residual convolutions are also different, which validates the effectiveness of our proposed FCP.

4.4 Ablation Study of FCP

Table 3. Quantitative results of evaluated methods for x4 SR

Method	Params	FLOPs	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	-	-	28.63/0.8138	26.21/0.7087	26.04/0.6719	25.07/0.7904
EDSR	43090K	2894.5G	32.46/0.8968	28.80/0.7876	27.71/0.7420	31.02/0.9148
MSRResNet	1517K	146.0G	32.22/0.8952	28.63/0.7826	27.59/0.7357	30.48/0.9089
CARN	1592K	90.8G	32.13/0.8937	28.60/0.7806	27.58/0.7349	30.45/0.9073
Li et al.[21]	861K	78.69G	32.03/0.8931	28.54/0.7803	27.53/0.7346	30.23/0.9056
FPGM[13]	859K	83.94G	31.95/0.8917	28.48/0.7790	27.48/0.7332	30.03/0.9033
GBN[34]	863K	75.76G	32.09/0.8944	28.58/0.7815	27.56/0.7356	30.36/0.9075
Ours (60%)	973K	90.29G	32.18/0.8947	28.61/0.7823	27.58/0.7362	30.44/0.9084
Ours (50%)	799K	75.73G	32.15/0.8946	28.58/0.7816	27.57/0.7358	30.40/0.9080

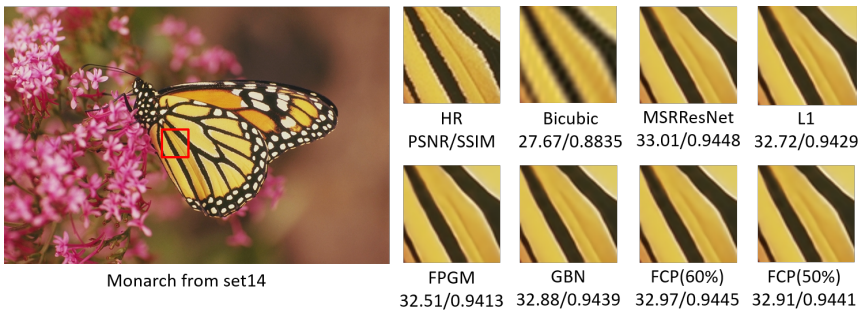


Fig. 8. Visual results of different pruning methods.

To validate the effect of our proposed fine-grained channel pruning, we compare with other state-of-the-art pruning methods on MSRResNet. Table 3 shows

the quantitative results, FCP can reduce more parameters and FLOPs while maintaining higher PSNR and SSIM on all datasets than other approaches. Specifically, we can achieve nearly 64% parameters and 62% computation cost of the baseline MSRResNet with negligible performance drop, and SSIM on dataset BSD100 can be even better than the original model. GBN adopts the same pruning criterion and applies group pruning strategy, but our 50% pruned model has less computation costs while the performance is higher. These results show that our FCP can compress the model size into a smaller one while still keeping high performance.

5 Conclusions

In summary, we propose an adaptive hybrid composition based super-resolution network called AHCSRN for SISR. We take two steps to design an efficient super-resolution network with the resource constraints: 1) Design a baseline model and 2) Apply channel pruning. To leverage the efficiency of different kernel size and feature scale, we firstly propose a hybrid composition block which contains asymmetric convolution and shrinked spatial convolution blocks. And we construct our baseline model with cascaded hybrid block via a progressive upsampling method. Secondly, we propose a fine-grained channel pruning method to solve the misalignment problem in pruning residual networks, and apply it to our baseline model to get the AHCSRN. Extensive experiments have shown that the proposed method can achieve a better tradeoff between performance and computation costs than state-of-the-art models, and the proposed AHCSRN has the same performance as MSRResNet with ultra-low parameters and Flops.

Acknowledgement. This work is supported by the National Key R&D Program of China (2020YFB0906000, 2020YFB0906001).

References

1. Ahn, N., Kang, B., Sohn, K.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) In: European Conference on Computer Vision, ECCV. vol. 11214, pp. 256–272 (2018)
2. Ding, X., Ding, G., Guo, Y., Han, J.: Centripetal SGD for pruning very deep convolutional networks with complicated structure. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4943–4953 (2019)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) In: European Conference on Computer Vision, ECCV. vol. 8692, pp. 184–199 (2014)
4. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) In: European Conference on Computer Vision, ECCV

5. Freedman, G., Fattal, R.: Image and video upscaling from local self-examples. ACM Trans. Graph. **30**(2), 12:1–12:11 (2011)
6. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications **22**(2), 56–65 (2002)
7. Gao, S., Liu, X., Chien, L., Zhang, W., Alvarez, J.M.: VACL: variance-aware cross-layer regularization for pruning deep residual networks. In: In: International Conference on Computer Vision Workshops, ICCV Workshops. pp. 2980–2988 (2019)
8. Guo, Y., Yao, A., Chen, Y.: Dynamic network surgery for efficient dnns. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) In: Annual Conference on Neural Information Processing, NeurIPS. pp. 1379–1387 (2016)
9. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. CoRR **abs/1506.02626** (2015)
10. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1664–1673 (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 770–778 (2016)
12. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft filter pruning for accelerating deep convolutional neural networks. In: Lang, J. (ed.) In: International Joint Conference on Artificial Intelligence, IJCAI. pp. 2234–2240 (2018)
13. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4340–4349 (2019)
14. Hu, H., Peng, R., Tai, Y., Tang, C.: Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. CoRR **abs/1607.03250** (2016)
15. Huang, Q., Zhou, S.K., You, S., Neumann, U.: Learning to prune filters in convolutional neural networks. In: In: Winter Conference on Applications of Computer Vision, WACV. pp. 709–718 (2018)
16. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: Amsaleg, L., Huet, B., Larson, M.A., Gravier, G., Hung, H., Ngo, C., Ooi, W.T. (eds.) In: International Conference on Multimedia, MM. pp. 2024–2032 (2019)
17. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1646–1654 (2016)
18. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1637–1645 (2016)
19. Lai, W., Huang, J., Ahuja, N., Yang, M.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 5835–5843 (2017)
20. Lemaire, C., Achkar, A., Jodoin, P.: Structured pruning of neural networks with budget-aware regularization. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 9108–9116 (2019)
21. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: In: International Conference on Learning Representations, ICLR (2017)

22. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: In: Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops. pp. 1132–1140 (2017)
23. Lin, S., Ji, R., Li, Y., Wu, Y., Huang, F., Zhang, B.: Accelerating convolutional networks via global & dynamic filter pruning. In: Lang, J. (ed.) In: International Joint Conference on Artificial Intelligence, IJCAI. pp. 2425–2432 (2018)
24. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: In: International Conference on Computer Vision, ICCV. pp. 2755–2763 (2017)
25. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J.: Importance estimation for neural network pruning. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 11264–11272 (2019)
26. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1874–1883 (2016)
27. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 2818–2826 (2016)
28. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 2790–2798 (2017)
29. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: In: International Conference on Computer Vision, ICCV. pp. 4549–4557 (2017)
30. Timofte, R., Smet, V.D., Gool, L.V.: Anchored neighborhood regression for fast example-based super-resolution. In: In: International Conference on Computer Vision, ICCV. pp. 1920–1927 (2013)
31. Wang, C., Li, Z., Shi, J.: Lightweight image super-resolution with adaptive weighted learning network. CoRR **abs/1904.02358** (2019)
32. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) In:Annual Conference on Neural Information Processing Systems, NeurIPS. pp. 2074–2082 (2016)
33. Ye, J., Lu, X., Lin, Z., Wang, J.Z.: Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. In: In: International Conference on Learning Representations, ICLR (2018)
34. You, Z., Yan, K., Ye, J., Ma, M., Wang, P.: Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) In: Annual Conference on Neural Information Processing, NeurIPS. pp. 2130–2141 (2019)
35. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: In: Conference on Computer Vision and Pattern Recognition, CVPR. pp. 6848–6856 (2018)
36. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) In: European Conference on Computer Vision, ECCV. vol. 11211, pp. 294–310 (2018)