1×1 CONVOLUTION IS ALL YOU NEED FOR LIGHTWEIGHT IMAGE SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

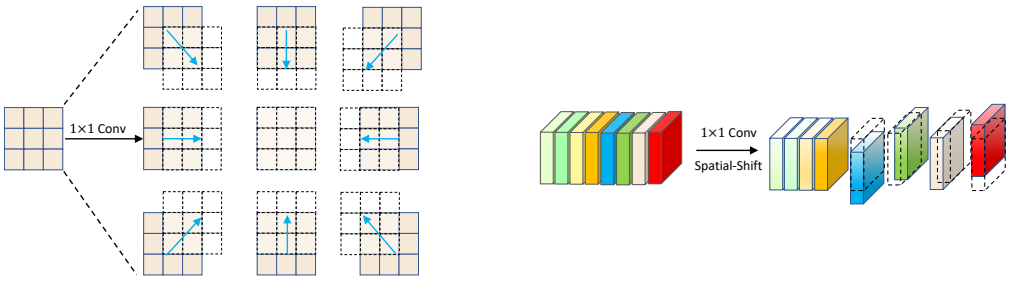
In resource-constrained environments, such as mobile devices, lightweight and efficient architectures are crucial for the deployment of single image super-resolution (SISR) deep models. Due to the advantage of achieving a good trade-off between model capacity and efficiency, 3×3 convolutions are widely utilized in current convolutional neural networks (CNN). Compared to the normal 3×3 convolution, 1×1 convolution involves less computation burden but lacks the ability to represent and aggregate spatial information. Accordingly, a common sense in the literature is that 1×1 solely cannot constitute a powerful SR network. In this paper, we revisit 1×1 in the lightweight scenario and demonstrate that the fully 1×1 convolutional network with strong learning ability can be achieved for SISR, thanks to the manual spatial-shift operation. We investigate the feature aggregation scheme in normal 3×3 convolution and analogously extend the 1×1 convolution with a parameter-free spatial-shift operation, simplified as the shift-conv layer. In the proposed SISR method, we propose the **S**hift-**C**onv-based **N**etwork (SCNet) by replacing all normal 3×3 convolutions with shift-conv layers. Extensive experiments demonstrate that SCNets with all 1×1 convolutions obtain even better results than SR models with normal 3×3 convolutions that have a larger model size.

1 INTRODUCTION

Single image super-resolution (SISR) aims at reconstructing a high-resolution (HR) image from its corresponding degraded low-resolution (LR) input. It has witnessed substantial advancements and gained more of the spotlight in research communities with the rapid development of deep learning. The pioneering work SRCNN (Dong et al., 2016a) proposes to learn the mapping from LR inputs to HR targets directly by a convolutional neural network (CNN) and outperforms traditional approaches. Subsequently, many CNN-based work explore more effective architectures (Ledig et al., 2017; Lim et al., 2017; Zhang et al., 2018b; Haris et al., 2018). Besides CNN-based architectures, transformer-based backbone (Liang et al., 2021) has been proposed and achieved SOTA performance.

However, the models mentioned above improve the SISR performance with very deep or complicated network architectures, leading to a heavy burden on parameter amounts and computational cost. As a result, the required substantial resources make them hard to be deployed in the resource-constrained environment, such as mobile or edge devices. Correspondingly, efficient and lightweight SR models are highly demanded. Many work have been proposed to reduce the amounts of parameters or floating-point operations (FLOPs) to achieve lightweight neural networks for SISR (Dong et al., 2016b; Ahn et al., 2018; Hui et al., 2019; Li et al., 2020; Wang et al., 2021; Zhang et al., 2021; Gao et al., 2022; Sun et al., 2022).

Due to the advantage in the trade-off between model capacity and computational cost, 3×3 convolution has become the most widely used operation in CNN-based models. A larger kernel can promote advancing performance but at the price that the parameter amount and computational cost increase rapidly (Liu et al., 2022; Ding et al., 2022). On the other hand, a smaller kernel with the size of 1×1 can reduce the amount of parameters but impairs the learning ability of SR network due to the lose in local feature aggregation (with neighborhood pixels). Naturally, a question comes to mind: *can we achieve the best of both worlds to build a lightweight yet effective SR model with fully* 1×1 *convolutions?*



(a) Feature aggregation scheme in normal 3×3 convolution (b) Feature aggregation scheme in Shift-Conv layer

Figure 1: Illustration of the feature aggregation in the normal 3×3 convolution and Shift-Conv layer.

To answer the question, let's dig into the feature aggregation scheme in normal 3×3 convolution. As illustrated in Figure 1(a), the normal 3×3 convolution can be separated into nine solid 1×1 convolutions to obtain nine projected feature maps. Then the projected features are shifted in different directions and added as the final feature. If we directly replace 3×3 convolution with 1×1 convolution, the number of parameters is reduced significantly, but the absence of a local feature aggregation impairs the model. Here, we extend 1×1 convolution with local feature aggregation by the spatial-shift operation against the channel dimension, as shown in Figure 1(b). It is worth noting that the spatial-shift operation is non-parametric, thus no extra FLOPs are evolved. We separate the input feature map into different groups along the channel dimension, and then the spatial-shift operation is employed in different directions among groups. In this way, each pixel in the shifted feature map is assembled around features along the channel dimension, which bridges the gap to the 3×3 convolution. The 1×1 convolution extended with local feature aggregation by the spatial-shift operation is noted as the Shift-Conv layer here (simplified as SC layer), which reduces the number of parameters significantly compared to the normal 3×3 convolution.

In light of the above findings, in this paper, we propose a lightweight yet effective SR model with extremely few parameters stacked by SC layers, which utilizes fully 1×1 convolutional layers. It is worth noting that the hyper-parameters of stride and direction in the SC layer correspond to the kernel in normal 3×3 convolution, i.e., when we take stride as 2 in around eight directions, the shift-conv is analogous to the normal 3×3 convolution. It is worth noting that we can take different spatial priors in local pixel selection. We can select different locations by setting hyper-parameters of the stride and direction in spatial-shift operation. Therefore, the SC layer can reduce parameters and extend the receptive fields in normal 3×3 convolution. Following the widely used residual connection block (Lim et al., 2017), we propose a shift-conv residual block, simplified as the SC-ResBlock, to replace all 3×3 convolutions with 1×1 convolutions. Furthermore, a lightweight shift-conv-based network is proposed, which is stacked by several SC-ResBlocks, named SCNet. In addition, we extend our SCNet with larger hidden dimensions and deeper layers to obtain a similar number of parameters as the existing SR models, and the three SCNet with different model sizes are introduced with different suffixes: tiny (T), base (B), and large (L), respectively. The performance of our SCNets on Manga109 testset ($\times 4$) compared to other models of different sizes is shown in Figure 2. It can be observed that our proposed SCNet obtains a better trade-off between accuracy and the amount of parameters.

The main contributions of our work are highlighted as follows:

- This paper sheds new light on the designing of lightweight architecture for SISR, and proposes a novel **S**hift-**C**onv **N**etwork (SCNet) with fully 1×1 convolutional layers, which produces an extremely small amount of parameters.
- We investigate the feature aggregation in normal 3×3 convolution and extend 1×1 convolution with local feature aggregation by a manual spatial-shift operation against the channel dimension. To demonstrate the effectiveness of our SCNet, we modify the basic residual blocks by replacing normal 3×3 convolutional layers with our SC layers. The three SCNets are introduced with different model capacities.
- Extensive experimental results are offered to verify the superiority of the proposed SCNet, and detailed ablation studies are evolved to help understand the impact of various components and the scalability of our SCNet.

2 RELATED WORK

Recently, deep learning methods have achieved dramatic improvements in SISR tasks (Jing & Tian, 2021; Anwar et al., 2020; Li et al., 2021). Especially for CNN-based models, various well-designed CNN architectures explore to further improve the SISR performance (Kim et al., 2016b; Tai et al., 2017a; Lim et al., 2017). Besides, attention mechanism like the channel attention has been introduced to SISR task as well (Zhang et al., 2018a; Dai et al., 2019; Niu et al., 2020). Most recently, vision transformers have attracted great attention (Dosovitskiy et al., 2021; Liu et al., 2021) and many work have been proposed to explore transformer-based architectures that achieve new SOTA performance (Chen et al., 2021; Liang et al., 2021).

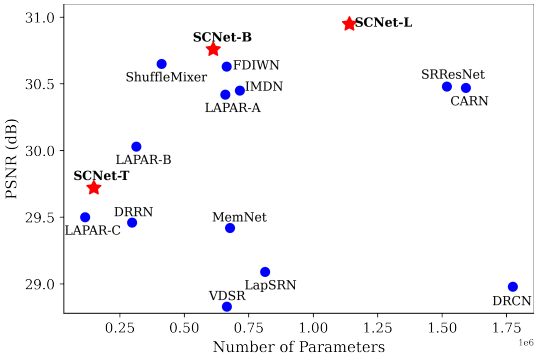


Figure 2: **PSNR** vs. **Parameters**. Comparisons with most recent efficient SISR models on Manga109 ($\times 4$) test dataset.

In contrast to achieving advancing performance with a rapidly increased number of parameters and computational cost, many lightweight SISR models have been explored by reducing parameters, especially for resource-limited devices (Ahn et al., 2018; Hui et al., 2019; Li et al., 2020; Wang et al., 2021; Zhang et al., 2021; Gao et al., 2022). They commonly leverage the normal 3×3 convolutions and try to develop well-designed blocks to promote the performance.

In the last year, several work investigated some modern CNN-based architectures (Liu et al., 2022; Ding et al., 2022). Liu *et al.* explored a modern CNN-based architecture and introduced larger kernels that utilize 9×9 kernel size. Ding *et al.* further brought the kernel size up to 31. Larger kernels bring larger receptive fields that significantly improve the capabilities of CNN-based networks compared to normal 3×3 convolution. It is foreseen that large kernel convolution will also improve the low-level tasks because receptive fields play a key role as well.

However, as those mentioned above, lightweight and efficient architectures are crucial to the real-world application of single image super-resolution (SISR) models, especially for edge devices. In contrast to exploring the larger kernel or deeper models, we pay more attention to the basic 1×1 convolution which introduces the fewest parameters.

3 METHODS

3.1 ARCHITECTURAL DESIGN

As shown in Figure 3, numerous basic SR-ResBlocks stack the main backbone of our SCNet followed by up-scaling layers to reconstruct high-resolution (HR) results.

Given the LR image $I^{LR} \in \mathbb{R}^{C \times H \times W}$ where H, W, and C are image height, width, and channel number, respectively. Firstly, a normal 1×1 convolution is utilized as our shallow feature extractor to map image space to a latent space. The shallow extractor is noted as N_{head} and latent feature is $f_{head} = N_{head}(I^{LR}) \in \mathbb{R}^{C_{latent} \times H \times W}$ where C_{latent} is the channel dimension of the latent space.

Our main backbone N_{main} is stacked by numerous basic SC-ResBlocks that are implemented by the shift-conv and 1×1 convolutional layers replacing the 3×3 convolutional layers in the normal residual block (Lim et al., 2017). Here main backbone N_{main} takes shallow features f_{head} as input and extracts deep features $f_{main} = N_{main}(f_{head})$.

Then given the extracted deep feature f_{main}, the up-scaling module is utilized to reconstruct HR results. Here we take the SC layer, ReLU, 1×1 convolution, and the pixelshuffle operation to build our basic up-scaling module N_{rec}, and a normal 1×1 convolution is utilized to map the up-scaledd feature into the output with 3 channels. In addition, we add the up-scaledd LR images by bilinear interpolation and the super-resolved output is $I^{SR} = N_{rec}(f_{main}) + \text{Bilinear}(\text{I}^{\text{LR}})$. Finally, the SR network is trained by minimizing L_1 loss.

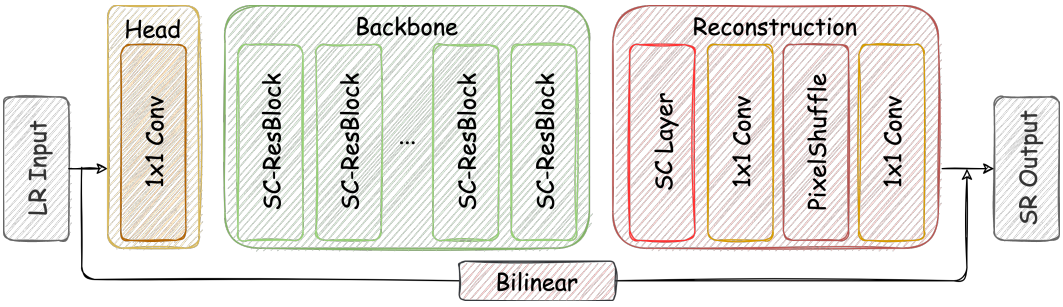


Figure 3: The architecture of our SCNet which is simply stacked by numerous basic residual blocks.

Algorithm 1 PyTorch-style pseudocode for spatial-shift operation.

```
# F: torch.nn.functional
def spatial_shift(f, steps, pad):
    """
    f [torch.Tensor]: input feature in (B, C, H, W)
    steps [Tuple(Tuple(int, int))]: parameters of the spatial-shift steps
    pad [int]: padding size
    """
    shift_groups = len(steps)
    B, C, H, W = f.shape
    group_dim = C//shift_groups
    f_pad = F.pad(f, pad)
    output = torch.zeros_like(f)

    for idx, step in enumerate(steps):
        s_h, s_w = step[0], step[1]
        output[:, idx*group_dim: (idx+1)*group_dim, :, :] = \
            f_pad[:, idx*group_dim:(idx+1)*group_dim, pad+s_h:pad+s_h+H, pad+s_w:pad+s_w+W]
    return output
```

3.2 Shift-Conv Residual Block

Spatial-Shift Operation. Let us note the shift direction as $d \in \{1, 0, -1\}$, and take d_h and d_w for each side, respectively. Correspondingly, the strides are noted as s_h and s_w. Then we can obtain the spatial-shift steps by combining direction and stride as $step = (d_h * s_h,\ d_w * s_w)$, and the set of spatial-shift steps is $S = \{step_i, i = 1, \ldots, n\}$ where n is the number of assembled features and $step_i$ presents the step for the ith local pixel-wise feature. If we want to take 8 local pixels around like the normal 3×3 convolution, the set of our spatial-shift steps can be defined as $\{(0,\ 1),\ (0, -1),\ (1,\ 0),\ (1,\ 1),\ (1, -1),\ (-1,\ 0),\ (-1,\ 1),\ (-1, -1)\}$. We utilize the $step_i$ to locate the target pixel feature and we can leverage pixels anywhere even with a long distance (just assign a large stride value). In addition, we can take different local aggregation schemes by setting different spatial-shift steps. For fair comparison and evaluating the effectiveness of our fully 1×1 convolutional SCNet, we take the local 8 pixels around like the normal 3×3 convolutional layer as the default.

Given the input feature f, we uniformly split it into n groups along the channel dimension where $n = |S|$, and n thinner tensors $f^i \in \mathbb{R}^{\frac{C_{latent}}{n} \times H \times W}, i = 1, \ldots, \frac{C_{latent}}{n}$ are obtained. Then each separated feature group is shifted by the given step parameters and the shifted feature f_{shift} is obtained. Each pixel feature in f_{shift} contains local features around it along the channel dimension. The pseudocode of our spatial-shift operation is shown in Algorithm 1.

Shift-Conv Layer. Since 1×1 convolutional operation works on the single pixel feature which impairs the modeling, here we explore the local feature aggregation explicitly by a simple spatial-shift operation that involves no parameters and FLOPs. The Shift-Conv layer (simplified as the SC layer) is stacked by a 1×1 convolutional layer and the spatial-shift operation, thus the SC layer extends the normal 1×1 convolution with local feature aggregation as well as the fewer parameters.

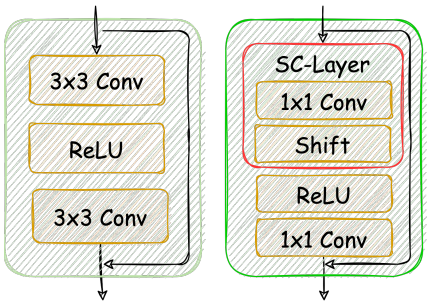
Shift-Conv Residual Block. As illustrated in Figure 4(a), the residual block proposed in (Lim et al., 2017) is widely used in SR networks. For a fair comparison, We modify and introduce our SC-ResBlock based on this basic residual connection. As illustrated in Figure 4(b), the SC-ResBlock

contains a SC layer, ReLU, and a 1×1 convolution. Compared with the 3×3 convolution-based residual block, our SC-ResBlock significantly reduces the number of parameters and computational cost by adopting only 1×1 convolution.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Training Settings. We crop the image patches with the fixed size of 64×64 for training, and the counterpart LR patches are downsampled by Bicubic interpolation. All the training patches are augmented by randomly horizontally flipping and rotation. We set the batch size to 32 and train our model using ADAM (Kingma & Ba, 2015) optimizer with the settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$. The initial learning rate is set as 2×10^{-4}.



(a)ResBlock (b)SC-ResBlock

Figure 4: Comparison between basic ResBlock our Shift-Conv-based ResBlock.

Datasets and Metrics. We take 800 images from DIV2K (Timofte et al., 2017) and 2650 images from Flickr2K for training. Datasets for testing include Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), B100 (Martin et al., 2001), Urban100 (Huang et al., 2015), and Manga109 (Matsui et al., 2017) with the up-scaling factor of 2,3, and 4. For comparison, we measure PSNR and SSIM (Wang et al., 2004) on the Y channel of transformed YCbCr space.

Comparison methods. We compare the proposed SCNet with representative efficient SR models, including SRCNN (Dong et al., 2016a), VDSR (Kim et al., 2016b), LapSRN (Lai et al., 2017), DRRN (Tai et al., 2017a), MemNet (Tai et al., 2017b), CARN (Ahn et al., 2018), IMDN (Hui et al., 2019), LAPAR (Li et al., 2020), SMSR (Wang et al., 2021), ECBSR (Zhang et al., 2021), FDIWN (Gao et al., 2022), and ShuffleMixer (Sun et al., 2022) on $\times 2$, $\times 3$, and $\times 4$ up-scaling tasks.

4.2 MAIN RESULTS

Quantitative Evaluation. The performance comparison of different SR models on five test datasets with scales 2, 3, and 4 are reported in Table 1. In addition to PSNR/SSIM, we also report the number of parameters. When the number of parameters is less than 400k, our SCNet-T outperforms all the tiny models except LAPAR-B (Li et al., 2020), which contains much more parameters. Our SCNet-T adopts the plain residual architectures and demonstrates that the fully 1×1 convolutional network is effective for SISR. When the number of parameters is between 500k and 900k, one can find that our SCNet-B outperforms some larger models such as IMDN (Hui et al., 2019), and LAPAR-A (Li et al., 2020) on all scales. More specially, the proposed SCNet-B achieves SOTA performance for scale 4 on all test datasets. In addition, when it comes to DRCN (Kim et al., 2016a), CARN (Ahn et al., 2018), SRResNet (Ledig et al., 2017), SMSR (Wang et al., 2021) and SCNet-L that contain more parameters, it can be found that our SCNet-L achieves SOAT performance in all cases. Benefiting from the extremely few parameters in 1×1 convolution, we can explore larger latent dimensions and deeper neural networks simply stacked by basic SC-Resblocks. The proposed SCNet-L obtains remarkable gains **0.26/0.0047** and **0.28/0.0062** in the term of PSNR/SSIM compared to IMDN and SRResNet, respectively. In general, SCNets with all 1×1 convolutions obtain even better results than SR models with normal 3×3 convolutions with a larger model size, demonstrating the effectiveness of the proposed SCNets. In this regard, there are more opportunities to exploit efficient architectures of the lightweight SR network because of the few parameters in the 1×1 convolution.

Qualitative Evaluation. We compare the visual quality of SR results by our SCNet-L and five representative models, including LapSRN (Lai et al., 2017), VDSR (Kim et al., 2016b), DRCN (Kim et al., 2016a), CARN (Ahn et al., 2018) and IMDN (Hui et al., 2019) on $\times 2$, $\times 3$, and $\times 4$ up-scaling task. The $\times 4$ SR results are shown in Figure 5, and we can limpidly see that the results of CARN and IMDN are blurry and contain more artifacts, while our SCNet-L can recover the main structures with clear and sharp textures. Besides, more visual comparison results can be found in Appendix A.2.

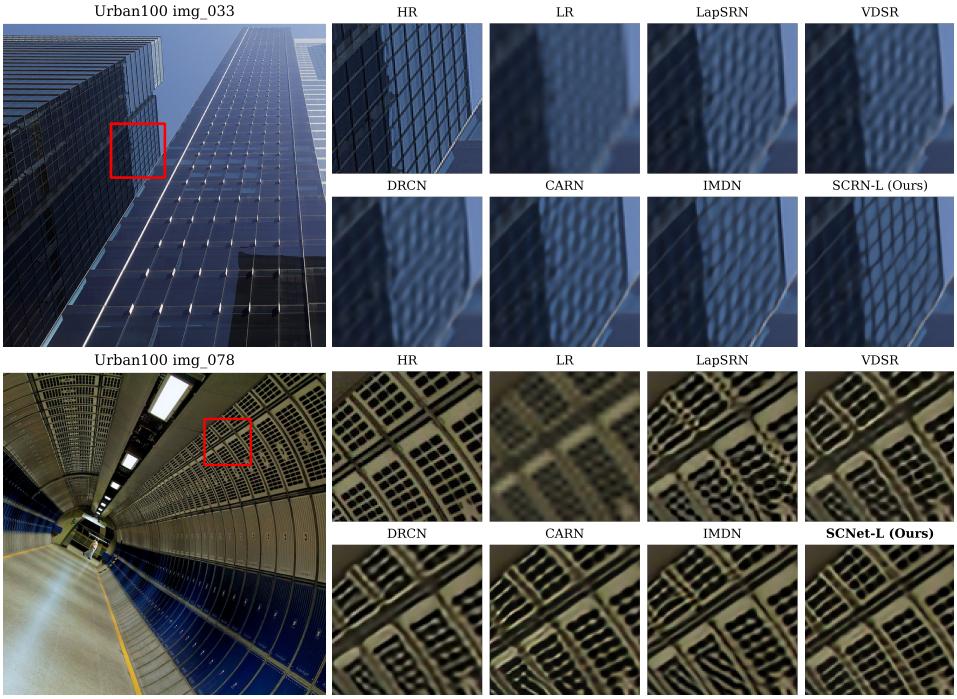


Figure 5: Visual comparisons on images with fine details on Urban100 test dataset (**Zoom in for more details**).

4.3 Analysis and Discussion

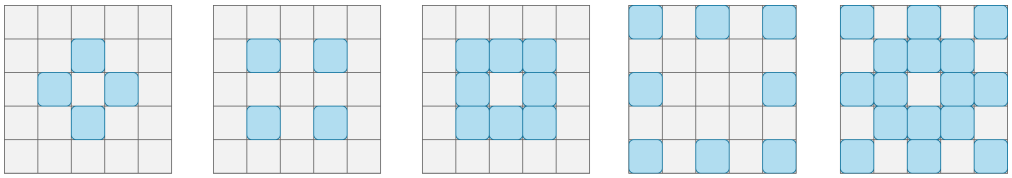
The core contribution in this paper is to propose a fully 1×1 convolutional network for SISR. To better understand the impact of different components of our SCNet, comprehensive ablation studies are presented in this section.

The Impact of Steps in SC Layer. Compared to the normal 3×3 convolution, there is an absence of spatial feature aggregation in 1×1 convolution. To aggregation the local features, the spatial-shift operation is exploited. The hyper-parameter shift step, which determines the aggregated local pixels, plays a key role in the spatial-shift operation. To better understand the impact of the shift step, the SCNet with 16 SC-ResBlocks and 128 channel dimensions is our basic model, and we re-train it with five different shift step settings, as shown in Figure 6. The first and the second patterns contain four local positions from the horizontal and vertical directions (noted as Shift4-Cross) and the diagonal directions (Shift4-Diag). The rest are dense 8 pixels around (Shift8), dilated 8 pixels (Shift8-Dilated), and 16 pixels that combine the Shift8 and Shift8-Dilated (Shift16), respectively. Results are summarized in Table 2. Furthermore, we utilize LAM (Gu & Dong, 2021) to visualize the reception fields of different spatial steps, as shown in Figure 7. By combining Table 2 and Figure 7, we observe that the local feature aggregation is of great significance in the following three aspects.

Neighborhood Feature Aggregation. The model adopting Shift4 with different steps is inferior compared to that with the default Shift8, demonstrating that the feature aggregation patterns in Shift4-Cross and Shift4-Diag are complementary, and the aggregation of neighborhood pixels around, like the normal 3×3 convolution, is essential to SR network. Furthermore, we can find that the Shift4-Diag can make the SR network learn successfully, but the worst performance is obtained here. We think it is due to the loss of information in spatial-shift operation. Since we take the constant value 0 for padding, the shift in diagonal directions removes the double number of pixels along two sides than Shift4-Cross.

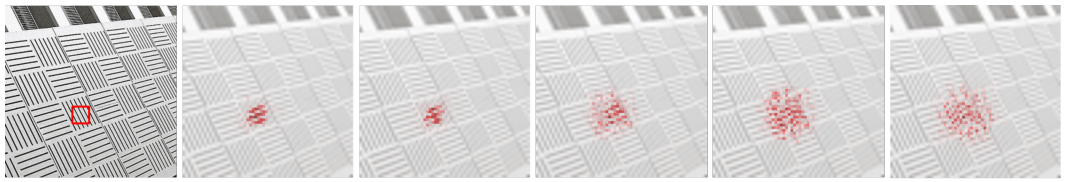
Table 1: Quantitative comparison with some representation SR approaches on five widely used benchmark datasets. Comparison methods are grouped according to the number of parameters and the best and our results are highlighted in underline and **bold** correspondingly.

Scale	Method	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
×2	SRCNN	57K	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
	LapSRN	251K	37.52/0.9591	32.99/0.9124	31.80/0.8952	30.41/0.9103	37.27/0.9740
	DRRN	298K	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.88/0.9749
	ECBSR-M10C32	95K	37.76/0.9609	33.26/0.9146	32.04/0.8986	31.25/0.9190	35.68/0.9421
	LAPAR-C	87K	37.65/0.9593	33.20/0.9141	31.95/0.8969	31.10/0.9178	37.75/0.9752
	LAPAR-B	250K	37.87/0.9600	33.39/0.9162	32.10/0.8987	31.62/0.9235	38.27/0.9764
	SCNet-T	159K	**37.85/0.9600**	**33.39/0.9161**	**32.06/0.8981**	**31.50/0.9187**	**38.29/0.9764**
	VDSR	666K	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
	MemNet	678K	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
	CARN-M	412K	37.53/0.9583	33.26/0.9141	31.92/0.8960	31.23/0.9193	35.62/0.9420
	IMDN	694K	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
	LAPAR-A	548K	38.01/0.9605	33.62/0.9183	32.19/0.8999	32.10/0.9283	38.67/0.9772
	FDIWN	629K	38.07/0.9608	33.75/0.9201	32.23/0.9003	32.40/0.9305	38.85/0.9774
	ShuffleMixer	394K	38.01/0.9606	33.63/0.9180	32.17/0.8995	31.89/0.9257	38.83/0.9774
	SCNet-B	557K	**38.07/0.9607**	**33.72/0.9188**	**32.23/0.9003**	**32.24/0.9296**	**38.95/0.9777**
	DRCN	1,774K	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	37.55/0.9732
	CARN	1,592K	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9765
	SRResNet	1,370K	38.05/0.9607	33.64/0.9178	32.22/0.9002	32.23/0.9295	38.05/0.9607
	SCNet-L	1,157K	**38.12/0.9609**	**33.90/0.9206**	**32.28/0.9009**	**32.46/0.9315**	**39.14/0.9781**
×3	SRCNN	57K	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
	DRRN	298K	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.71/0.9379
	LAPAR-C	99K	33.91/0.9235	30.02/0.8358	28.90/0.7998	27.42/0.8355	32.54/0.9373
	LAPAR-B	276K	34.20/0.9256	30.17/0.8387	29.03/0.8032	27.85/0.8459	33.15/0.9417
	SCNet-T	147K	**34.03/0.9244**	**29.99/0.8381**	**28.93/0.8017**	**27.65/0.8413**	**32.84/0.9403**
	VDSR	666K	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
	LapSRN	502K	33.81/0.9220	29.79/0.8325	28.82/0.7980	27.07/0.8275	32.21/0.9350
	MemNet	678K	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
	IMDN	703K	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
	LAPAR-A	594K	34.36/0.9267	30.34/0.8421	29.11/0.8054	28.15/0.8523	33.51/0.9441
	FDIWN	645K	34.52/0.9281	30.42/0.8438	29.14/0.8065	28.36/0.8567	33.77/0.9456
	ShuffleMixer	415K	34.40/0.9272	30.37/0.8423	29.12/0.8051	28.08/0.8498	33.69/0.9448
	SCNet-B	589K	**34.44/0.9276**	**30.43/0.8437**	**29.15/0.8063**	**28.31/0.8556**	**33.86/0.9462**
	DRCN	1,774K	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276	32.24/0.9343
	CARN	1,592K	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440
	SRResNet	1,554K	34.41/0.9274	30.36/0.8427	29.11/0.8055	28.20/0.8535	33.54/0.9448
	SMSR	993K	34.40/0.9270	30.33/0.8412	29.10/0.8050	28.25/0.8536	33.68/0.9445
	SCNet-L	1,107K	**34.53/0.9284**	**30.49/0.8452**	**29.20/0.8076**	**28.47/0.8588**	**34.08/0.9475**
×4	SRCNN	57K	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221	27.66/0.8505
	DRRN	297K	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.46/0.8960
	ECBSR-M10C32	98K	31.66/0.8911	28.15/0.7776	27.34/0.7363	25.41/0.7653	29.98/0.8281
	LAPAR-C	115K	31.72/0.8884	28.31/0.7740	27.40/0.7292	25.49/0.7651	29.50/0.8951
	LAPAR-B	313K	31.94/0.8917	28.46/0.7784	27.52/0.7335	25.85/0.7772	30.03/0.9025
	SCNet-T	149K	**31.82/0.8904**	**28.36/0.7764**	**27.39/0.7309**	**25.59/0.7696**	**29.72/0.9000**
	VDSR	665K	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8809
	LapSRN	813K	31.54/0.8850	29.19/0.7720	27.32/0.7280	25.21/0.7560	29.09/0.8845
	MemNet	677K	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
	CARN-M	412K	31.92/0.8903	28.42/0.7762	27.44/0.7304	25.62/0.7694	25.62/0.7694
	SRFBN-S	483K	31.98/0.8923	28.45/0.7779	27.44/0.7313	25.71/0.7719	29.91/0.9008
	IMDN	715K	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
	LAPAR-A	659K	32.15/0.8944	28.61/0.7818	27.61/0.7366	26.14/0.7871	30.42/0.9074
	ECBSR-M16C64	603K	31.92/0.8946	28.34/0.7817	27.48/0.7393	25.81/0.7773	30.15/0.8315
	FDIWN	664K	32.23/0.8955	28.66/0.7829	27.62/0.7380	26.28/0.7919	30.63/0.9098
	ShuffleMixer	411K	32.21/0.8953	28.66/0.7827	27.61/0.7366	26.08/0.7835	30.65/0.9093
	SCNet-B	578K	**32.26/0.8959**	**28.70/0.7844**	**27.64/0.7382**	**26.28/0.7917**	**30.76/0.9119**
	DRCN	1,774K	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.98/0.8816
	CARN	1,592K	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
	SRResNet	1,518K	32.17/0.8951	28.61/0.7823	27.59/0.7365	26.12/0.7871	30.48/0.9087
	SMSR	1,006K	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
	SCNet-L	1,140K	**32.37/0.8973**	**28.79/0.7861**	**27.70/0.7400**	**26.44/0.7962**	**30.95/0.9137**



(a) Shift4-Cross (b) Shift4-Diag (c) Shift8 (d) Shift8-Dilated (e) Shift16

Figure 6: Illustration of different spatial-shift steps.



(a) Reference (b) Shift4-Cross (c) Shift4-Diag (d) Shift8 (e) Shift8-Dilated (f) Shift16

Figure 7: LAM (Gu & Dong, 2021) comparisons between different shift step settings.

Table 2: Results of different selected positions. Based on SCNet-B for scale 4, we replace the default dense Shift8 steps with different settings as shown in Figure 6

Scale	Steps	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
×4	Shift4-Cross	612K	32.14/0.8946	28.61/0.7819	27.58/0.7360	26.05/0.7836	30.48/0.9086
	Shift4-Diag	612K	31.83/0.8898	28.39/0.7769	27.44/0.7314	25.65/0.7705	29.90/0.9015
	Shift8	612K	32.16/0.8949	28.65/0.7830	27.60/0.7368	26.16/0.7864	30.58/0.9100
	Shift8-Dilated	612K	32.19/0.8953	28.67/0.7832	27.60/0.7369	26.14/0.7868	30.61/0.9102
	Shift16	612K	32.10/0.8941	28.57/0.7812	27.55/0.7355	26.02/0.7833	30.34/0.9075

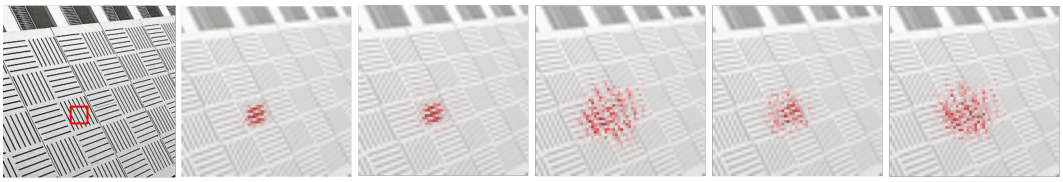
Receptive Field. Based on the default Shift8 step, we extend it into Shift8-Dilated, as shown in Figure 6(d). The dilated SCNet obtains slightly better performance than the default besides the Urban100. According to Figure 7, one can find that a larger receptive field can be obtained by the Shift8-Dilated, which shows that different feature aggregation patterns can be obtained by spatial-shift steps like the normal dilated convolution.

Group Dimension. Furthermore, we combine the default Shift8 with the Shift8-Dilated and obtain the Shift16, shown in Figure 6(e). Compared to the Shift8 and Shift8-Dilated, SCNet with Shift16 obtains an even larger receptive field but has worse performance, as summarized in Figure 7 and Table 2. We attribute this to the fewer feature dimensions of each shift group, which hampers the feature extraction. Since the dimension of the latent feature is fixed, the number of the shift group dimension in Shift16 is reduced to half of that in Shift8 and Shift8-Dilated. As illustrated in Figure 7, we could observe that there are still large activating regions but smaller activating values.

The Impact of Model Capacity. Benefiting from the few parameters in the SC layer, there are opportunities to explore more depths and widths of SCNet. Here we exploit our SCNets stacked with different SC-ResBlocks to analyze the impact of the model capacity. As summarized in Table 3, we build our SCNets by SC-ResBlocks with different blocks (simplified as B) and channel dimensions (D). When comparing SCNets with the same channel dimensions, such as 64 channels, one can find that better results are obtained with deeper architectures. As illustrated in Figure 8, deeper structure brings larger receptive fields. When we take the B64D64 vs. B16D128, we can find that B64D64 obtains better performance with even fewer parameters. We think it is due to the field of local feature aggregation that B64D64 brings larger receptive fields and much more feature aggregation, while shallow architecture in B16D128 lacks. In addition, the largest SCNet with B32D128 obtains the best performance. As shown in Figure 8, one can find that more activated pixels are obtained in B32D128 than that in B32D64, which shows that the group dimension is of great significance to the feature aggregation as well. The trade-off between the depth and width (group dimension) can be further explored in the future. Moreover, detailed ablations about the deeper architecture and larger channel dimension are shown in Figure 9. One can find that our SCNet is scalable that better performance can be obtained with larger model capacity.

Table 3: Results of SCNets with different capacity. The number of the SC-Resblock and latent dimension are simplified as the B and D.

Scale	Model Size	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
×4	B16D64	149K	31.82/0.8904	28.36/0.7764	27.39/0.7309	25.59/0.7696	29.72/0.9000
	B32D64	312K	32.08/0.8939	28.59/0.7816	27.57/0.7357	26.01/0.7829	30.42/0.9079
	B64D64	579K	32.26/0.8959	28.70/0.7844	27.64/0.7382	26.28/0.7917	30.76/0.9119
	B16D128	612K	32.19/0.8949	28.65/0.7830	27.60/0.7368	26.16/0.7864	30.58/0.9100
	B32D128	1,140K	32.37/0.8973	28.79/0.7861	27.70/0.7400	26.44/0.7962	30.95/0.9137



(a) Reference (b) B16D64 (c) B32D64 (d) B64D64 (e) B16D128 (f) B32D128

Figure 8: LAM (Gu & Dong, 2021) comparisons between different architectures of SCNet.

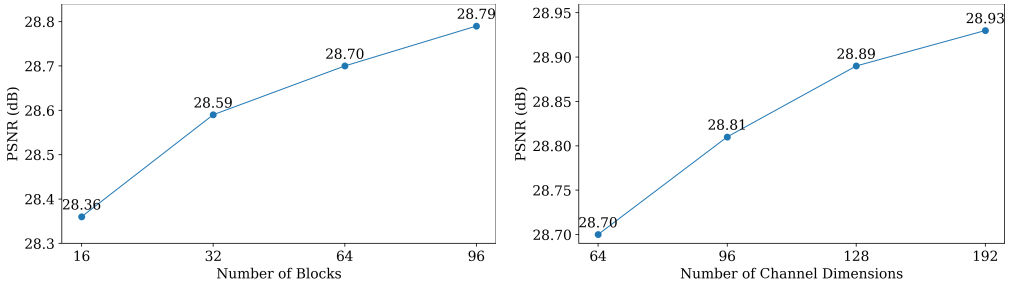


Figure 9: Results of SCNet on Set14 (×4) with different model capacities. (a) Increasing the number of SC-ResBlock with a fixing channel dimension 64. (b) Increasing the number of channel dimension with 64 SC-ResBlocks.

Discussion. In this section, we explore the impact of the spatial-shift steps and the model capacity. As summarized above, local feature aggregation is essential to the SR network, and a larger range of feature aggregations can achieve further improvement. Moreover, different model sizes are evaluated. Since there are few parameters in the 1×1 convolution, more architectures are able to be explored. The deeper architecture and the larger group dimension are advantageous to the proposed SCNet. In addition, more ablations about the up-scaling module and the scalability of our SCNet can be found in Appendix A.1.

5 Conclusion

In this paper, we propose a lightweight SISR model with fully 1×1 convolutions, named SCNet. Compared to the normal 3×3 convolution, 1×1 convolution contains fewer parameters and less computational cost, but the local feature aggregation is missing, which is essential to CNN-based SR networks. To bridge the gap, we investigate the feature aggregation in the normal 3×3 convolution and take one step further from the design convention, extending the 1×1 convolution into the shift-conv with the spatial-shift operation. It has no extra computational cost and involves the local feature aggregation along the channel dimension. Then the Shift-Conv-based Network (SCNet) is proposed for SISR task. Extensive experiments demonstrate the effectiveness of our SCNet, which obtains comparable and even outperforms the existing SR models. Furthermore, detailed ablation studies are conducted to better understand the impact of different components. We expect that our SCNet can further help the community explore the combination of the local feature aggregation with the long-range or global information in the near future.

REFERENCES

Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018.

Saeed Anwar, Salman H. Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Comput. Surv.*, 2020.

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.

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 *CVPR*, 2021.

Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019.

Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. *CoRR*, abs/2203.06717, 2022.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307, 2016a.

Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016b.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

Guangwei Gao, Wenjie Li, Juncheng Li, Fei Wu, Huimin Lu, and Yi Yu. Feature distillation interaction weighting network for lightweight image super-resolution. In *AAAI*, 2022.

Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In *CVPR*, 2021.

Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, 2018.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015.

Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM Multimedia*, 2019.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*. IEEE, 2016a.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016b.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep Laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

Juncheng Li, Zehua Pei, and Tieyong Zeng. From beginner to master: A survey for deep learning-based single-image super-resolution. *CoRR*, abs/2109.14335, 2021.

Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. LAPAR: linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. In *NeurIPS*, 2020.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCVW*. IEEE, 2021.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CPVR*, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 9992–10002. IEEE, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CoRR*, abs/2201.03545, 2022.

David R. Martin, Charless C. 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 *ICCV*, 2001.

Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multim. Tools Appl.*, 76 (20):21811–21838, 2017.

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 *ECCV*, 2020.

Long Sun, Jinshan Pan, and Jinhui Tang. Shufflemixer: An efficient convnet for image super-resolution. *arXiv preprint arXiv:2205.15175*, 2022.

Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*. IEEE, 2017a.

Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017b.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, et al. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017.

Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In *CVPR*, 2021.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 2004.

Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010.

Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In *ACM Multimedia*, 2021.

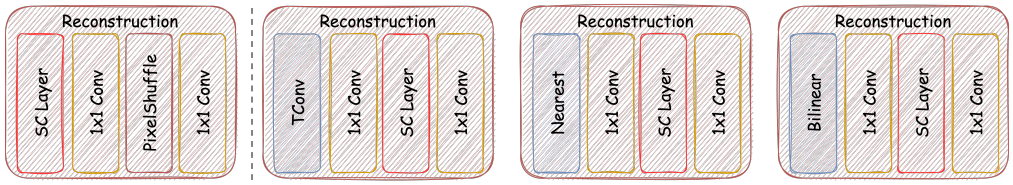
Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018a.

Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018b.

A APPENDIX

This appendix contains additional details for the paper "1×1 *Convolution is All You Need for Lightweight Image Super-Resolution*", including additional ablation studies and more visual results.

A.1 ABLATIONS



(a) (b)

Figure 10: Ablation studies about the reconstruction block with different up-scaling modules. (a) Our default reconstruction block with pixelshuffle. (b) Our reconstruction block with three different up-scaling modules.

Table 4: Results about the impact of up-scaling modules.

Scale	Up-Scaling	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
$\times 2$	PixelShuffle	159K	37.85/0.9600	33.39/0.9161	32.06/0.8981	31.50/0.9187	38.29/0.9764
	Nearest	146K	37.76/0.9597	33.37/0.9151	31.99/0.8974	31.30/0.9197	38.14/0.9760
	Bilinear	146K	37.78/0.9597	33.31/0.9152	32.00/0.8974	31.24/0.9193	38.12/0.9759
	TConv	151K	37.80/0.9598	33.40/0.9153	32.02/0.8977	31.40/0.9207	38.18/0.9761

The Impact of Up-Scaling Modules. In this section, we investigate the impact of different up-scaling approaches in our SCNet. For fair comparison, we take the SCNet-T as the default model, and modify the reconstruction module with different up-scaling strategies as illustrated in Figure 10. Here we evaluate four widely utilized up-scaling strategies: transport convolution, convolution with pixelshuffle, bilinear interpolation with convolution, and the nearest interpolation with convolution, and they are shortly noted as TConv, PixelShuffle, Bilinear, and Nearest, respectively. Results about the $\times 2$ super-resolution are summarized in Table 4. As summarized in Table 4, one can find that the pixelshuffle module with a bit more parameters achieves the best performance on all test datasets. Specially, SCNet with pixelshuffle obtain 0.10 dB and 0.11 dB improvement on Urban100 and Manga109 against the second.

Table 5: Results of SCNet-T with different attention modules.

Scale	Attention Module	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
$\times 4$	-*	159K	31.82/0.8904	28.36/0.7764	27.39/0.7309	25.59/0.7696	29.72/0.9000
	SA	188K	31.90/0.8912	28.40/0.7763	27.43/0.7308	25.67/0.7716	29.84/0.9004
	SPA	179K	31.89/0.8913	28.45/0.7779	27.46/0.7318	25.71/0.7727	29.95/0.9020
	PA	245K	31.94/0.8924	28.50/0.7791	27.49/0.7329	25.81/0.7757	30.10/0.9038

* - presents our default SCNet-T without attention module.

Extensive Attention Modules. As illustrated in Figure 11, we modify and extend our SC-ResBlock with some widely utilized attention modules. They are channel attention, spatial attention, and pixel-wise attention, and are shotly noted as CA, SPA, and PA, respectively. We add the extensive attention module at the end of the SC-ResBlock before the residual connection. Results are summarized in Table 5. From Table 5, one can find that our SCNet is scalable to attention modules as well. In addition, pixel-wise attention obtains the best performance, while it contains the most number of parameters. Both channel attention and spatial attention achieve further improvement. Specifically, spatial attention is more efficient because it obtains better performance with less parameters.

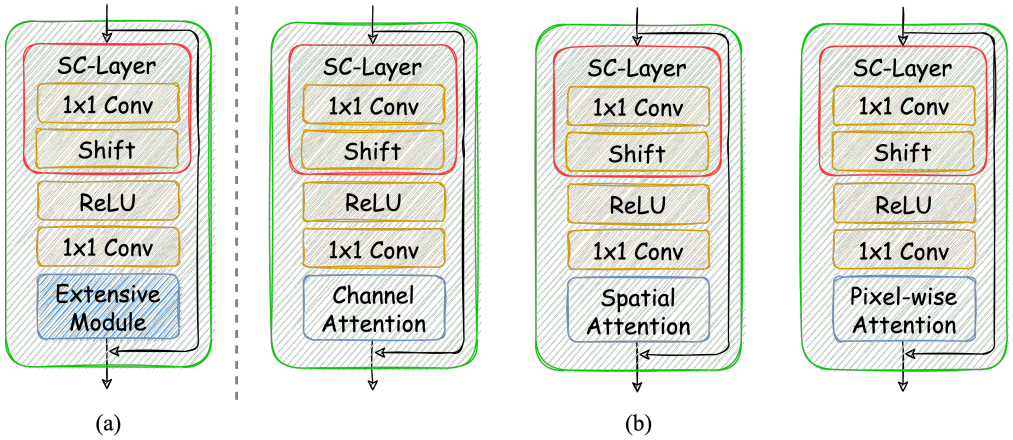


Figure 11: Ablation studies about the extensive attention modules. (a) The generally extended SC-ResBlock. (b) SC-ResBlock with different attention modules.

A.2 Qualitative Results

More visual comparison on Urban100 are shown in Figure 12. In addition, results of our different SCNets on Manga109 test datasets are shown in Figure 13.

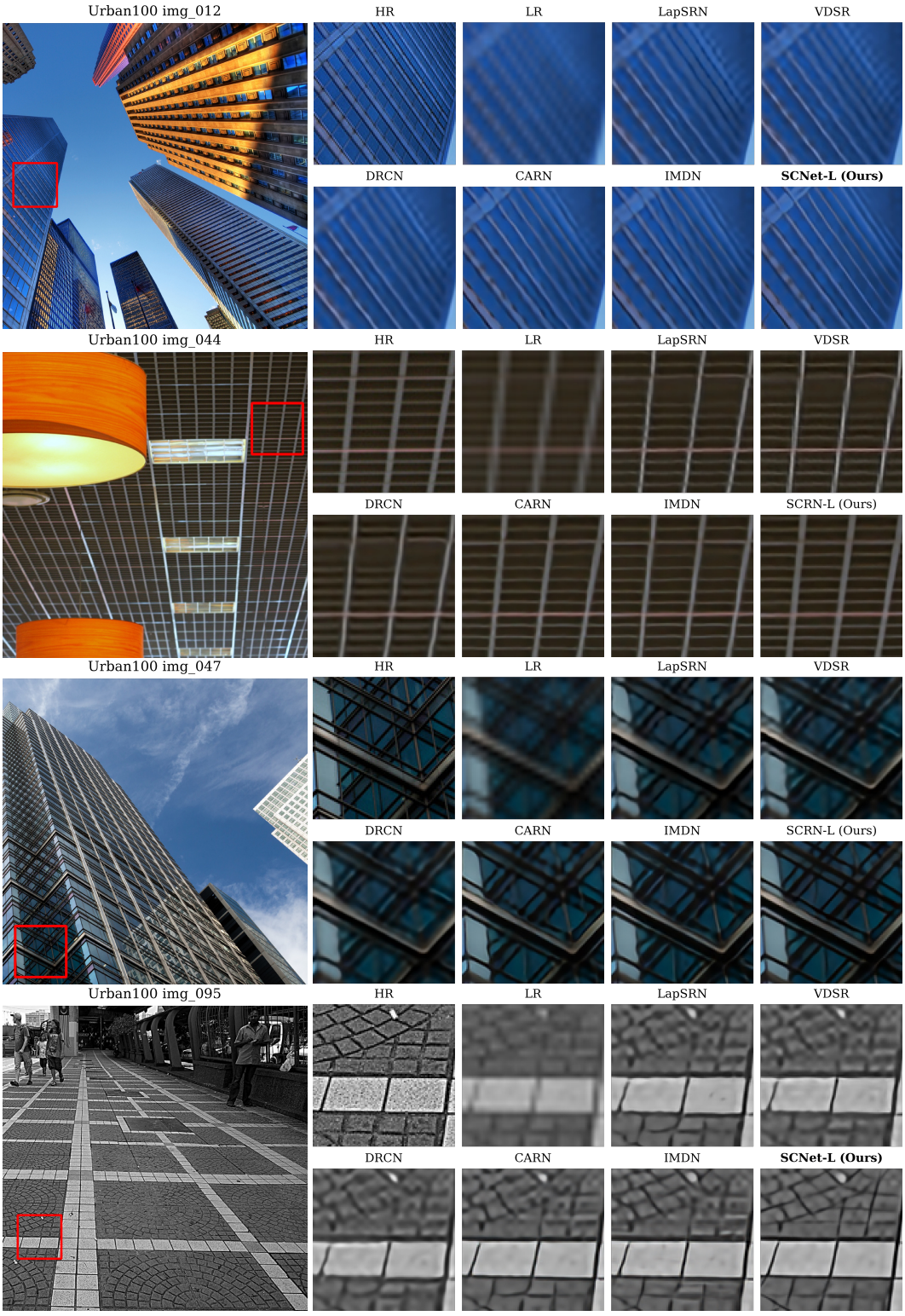


Figure 12: Visual comparisons on images with fine details on Urban100 test dataset. Results obtained by our SCNet-L are in **bold** that contain clearer and more accurate reconstructed textures with less artifacts (**Zoom in for more details**).

Figure 13: Visual comparisons on Manga109 test dataset. Results obtained by our SCNets are in **bold** (**Zoom in for more details**).