

Deep Neural Network for Fast and Accurate Single Image Super-Resolution via Channel-Attention-based Fusion of Orientation-aware Features

Du Chen*, Zewei He*, Yanpeng Cao†, Member, IEEE, Jiangxin Yang, Yanlong Cao, Michael Ying Yang, Senior Member, IEEE, Siliang Tang, and Yueting Zhuang

arXiv:1912.04016v1 [eess.IV] 9 Dec 2019

Abstract—Recently, Convolutional Neural Networks (CNNs) have been successfully adopted to solve the ill-posed single image super-resolution (SISR) problem. A commonly used strategy to boost the performance of CNN-based SISR models is deploying very deep networks, which inevitably incurs many obvious drawbacks (e.g., a large number of network parameters, heavy computational loads, and difficult model training). In this paper, we aim to build more accurate and faster SISR models via developing better-performing feature extraction and fusion techniques. Firstly, we proposed a novel Orientation-Aware feature extraction and fusion Module (OAM), which contains a mixture of 1D and 2D convolutional kernels (i.e., 5×1 , 1×5 , and 3×3) for extracting orientation-aware features. Secondly, we adopt the channel attention mechanism as an effective technique to adaptively fuse features extracted in different directions and in hierarchically stacked convolutional stages. Based on these two important improvements, we present a compact but powerful CNN-based model for high-quality SISR via Channel Attention-based fusion of Orientation-Aware features (SISR-CA-OA). Extensive experimental results verify the superiority of the proposed SISR-CA-OA model, performing favorably against the state-of-the-art SISR models in terms of both restoration accuracy and computational efficiency. The source codes will be made publicly available.

Index Terms—Single Image Super-Resolution, Channel Attention, Orientation-aware, Feature Extraction, Feature Fusion

I. INTRODUCTION

Single image super-resolution (SISR) restores a high-resolution (HR) image containing abundant details and textures based on its low-resolution (LR) version. It provides an effective technique to increase the spatial resolution of optical sensors and thus has attracted considerable attention from both the academic and industrial communities. In last decades, many machine learning SISR algorithms have been developed,

* The first two authors contributed equally to this work.

This work was supported in part by National Natural Science Foundation of China (No. 51605428, 51575486).

D. Chen, Z. He, Y. Cao, J. Yang and Y. Cao are with State Key Laboratory of Fluid Power and Mechatronic Systems and Key Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China (e-mail: caoyp@zju.edu.cn).

M. Y. Yang is with the Scene Understanding Group in University of Twente. (e-mail: michael.yang@utwente.nl).

S. Tang and Y. Zhuang are with College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China (e-mail: siliang@zju.edu.cn).

† Corresponding authors: Yanpeng Cao.

such as sparse coding [1], [2], local linear regression [3] and random forest [4]. However, SISR remains a challenging ill-posed problem because one specific LR input can correspond to many possible HR versions, and the mapping space is too vast to explore.

In recent years, Convolutional Neural Networks (CNNs) have been successfully adopted to solve the SISR problem by implicitly learning the complex nonlinear LR-to-HR mapping relationship based on numerous LR-HR training image pairs. SRCNN [5] proposed a three-layer CNN model to learn the nonlinear LR-to-HR mapping function. It is the first time that deep learning technique is applied to tackle the SISR problem. Compared with the many traditional machine-learning-based SISR methods, the lightweight SRCNN model achieved significantly improved image restoration results. Since then, many CNN-based models have been proposed to achieve more accurate SISR results [6]–[14]. Note that a common practice to improve the performance of CNN-based SISR models is either increasing the depth of the network or deploying more complex architectures [9], [15]. For instance, VDSR [6] is a 20-layer deep super-resolution convolutional network (VDSR), and more recent DRRN [7], SRDenseNet [8], and MemNet [9] SISR models contain 52, 68, and 80 layers, respectively. However, deploying very deep CNN models for SISR comes with many obvious drawbacks such as difficult model training due to the gradient vanishing problem, slow running time, and a large number of model parameters [16]–[18]. In this paper, our motivation is to explore alternative techniques to improve the performance of SISR in terms of both accuracy and computational load. More specifically, we look into (1) designing better-performing feature extraction modules and (2) exploring more effective schemes for multiple feature fusion.

The first and most important improvement is incorporating an orientation-aware mechanism to the feature extraction modules in CNN-based SISR models. Our key observation is that image structures/textures are the complex combinations of features extracted in different directions (e.g., horizontal, vertical, and diagonal). Thus the optimal way of reconstructing missing image details should also be orientation-dependent. However, the existing CNN-based SISR models (e.g., SRCNN [5], VDSR [6], DRRN [7], SRDenseNet [8], and MemNet [9]) typically utilize standard 3×3 or 5×5 convolutional kernels, which are square-shaped and orientation-independent,

to extract feature maps for the following super-resolution reconstruction. One possible solution to achieve orientation-aware SISR is to deploy convolutional kernels of various shapes in a single feature extraction module. In this paper, we proposed a novel Orientation-Aware feature extraction and fusion Module (OAM), which contains the mixture of 1D and 2D convolutional kernels (i.e., 5×1 , 1×5 , and 3×3) for extracting orientation-aware features.

The second improvement is to optimize the fusion scheme for integrating multiple features extracted in different directions and at various convolutional stages. Inspired by the channel attention mechanism for re-calibrating channel-wise features [19], we firstly propose to incorporate local channel attention (LCA) mechanism within each orientation-aware feature extraction module. It performs the scene-specific fusion of multiple outputs of orientation-dependent convolutional kernels (e.g., horizontal, vertical, and diagonal) to generate more distinctive features for SISR. Moreover, we utilize global channel attention (GCA) mechanism as an effective technique to adaptively fuse low-level and high-level features extracted in hierarchically stacked OAMs. Finally, we experimentally evaluate a number of design options to identify the optimal way to utilize the CA mechanism for re-calculating channel-wise weights for the concatenated orientation-aware and hierarchical features.

Based on the above improvements (**a.** the orientation-aware feature extraction and **b.** the channel attention-based feature fusion), we present a compact but powerful CNN-based model for high-quality SISR via Channel Attention-based fusion of Orientation-Aware features (SISR-CA-OA). The proposed SISR-CA-OA model shows superior performance over the state-of-the-art SISR methods on multiple benchmark datasets, achieving more accurate image restoration results and faster running speed. Overall, the contributions of this paper are mainly summarized as follows:

- We present a novel feature extraction module (OAM) containing a number of well-designed 1D and 2D convolutional kernels (5×1 , 1×5 , and 3×3) to extract orientation-aware features.
- We design channel attention-based fusion schemes (LCA and GCA), which can adaptively combine features extracted in different directions and in hierarchically stacked convolutional stages.
- We present a powerful SISR-CA-OA model for high-quality SISR, achieving higher accuracy and faster running time compared with state-of-the-art deep learning-based SISR approaches [6], [7], [9], [20]–[26].

In this paper, we make the following substantial extensions to our preliminary research work [27]. Firstly, we perform ablation studies to systematically validate the effectiveness of the proposed orientation-aware feature extraction technique. Secondly, we utilize the channel attention mechanism for the local fusion of features extracted in different directions and the global fusion of features extracted in hierarchical stages. Moreover, we investigate a number of design options to identify the optimal way to utilize the channel attention mechanism for feature fusion in SISR tasks. Thirdly, we significantly extend

the experiments, comparing the proposed SISR-CA-OA model with a number of recently published SISR methods [6], [7], [9], [17], [20]–[26] using various benchmark datasets (Set5 [28], Set14 [29], B100 [30], Urban100 [31], and Manga109 [32]).

The remainder of this paper is organized as follows. We first review a number of learning-based SISR models and different feature extraction/fusion techniques in Sec. II. Then Sec. III provides details of important components in our proposed SISR-CA-OA model. Qualitative and quantitative evaluation results are provided in Sec. IV, showing the superiority of our proposed method. Finally, Sec. V concludes this paper.

II. RELATED WORK

Over the past decades, developing effective SISR techniques to reconstruct a HR image from its corresponding single LR version has attracted extensive attention from both the academic fields and the industrial communities. In this work, we mainly focus on reviewing the existing CNN-based SISR methods which deploy various network architectures to construct distinctive feature representations for high-accuracy image restoration.

A. Deep-learning-based SISR

Dong et al. formulated the first 3-layer convolutional neural network model (SRCNN) to implicitly learn the end-to-end mapping function between the LR and HR images [5], [20]. Following this pioneering work, Kim et al. presented deeper networks (VDSR [6] and DRCN [21]) to generate more distinctive feature over larger image regions for more accurate image restoration. To alleviate the gradient vanishing problem that occurs when training a deep CNN model, they integrate a global residual learning architecture, which is firstly proposed by He et al. [33] into their SISR models. Dong et al. proposed to deploy a deconvolution layer to up-scale the feature maps at the end of the neural network to achieve faster speed and better reconstruction accuracy [34]. For the same purpose, Shi et al. proposed a pixel-shuffle operation for fast and accurate upscaling of the LR images via rearranging the feature maps [35].

It is noted that the most previously proposed SISR methods attempted to achieve more accurate restoration results by either increasing the depth of the network or deploying more complex architectures. For instance, Tai et al. developed a 52-layer DRRN model [7] which deploys local and global residual learning and recursive layers and a 84-layer MemNet model [9] which contains persistent memory units and multiple supervisions. More recently, some very deep CNN models such as RDN [26], D-DBPN [24], MSRN [23], and RCAN [15] are trained using the high-resolution DIV2K [36] dataset (containing 800 training images of 2K resolution), achieving the state-of-the-art SISR performance. However, their training process took a long time to complete and cannot deliver real-time processing speed. In this paper, we aim to develop better-performing feature extraction modules and more effective feature fusion schemes to improve the performance of SISR in terms of both accuracy and computational load.

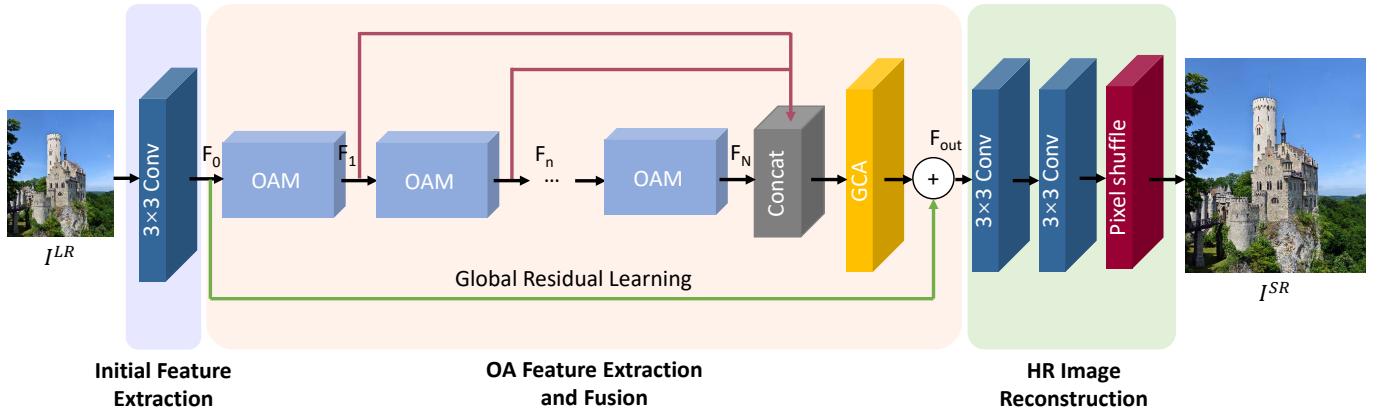


Fig. 1. The overall architecture of our proposed SISR-CA-OA model. Given the LR input image I^{LR} , we first employ a 3×3 convolutional layer to extract low-level features. Then a number of OAMs are hierarchically stacked to infer the non-linear LR-to-HR mapping function. Note here we incorporate a local channel attention (LCA) mechanism for the local fusion of features extracted in different directions and a global channel attention (GCA) mechanism for the global fusion of features extracted in different convolutional stages. Global residual learning is added to ease the training process. Finally, we use two 3×3 convolution layers and the pixel-shuffle operation to reconstruct the final HR image output I^{SR} .

B. Feature Extraction and Fusion

The existing CNN-based SISR models such as SRCNN [20], VDSR [6], DRRN [7], SRDenseNet [8], and MemNet [9] typically deploy square-shaped 3×3 or 5×5 convolutional kernels to extract feature maps for the following super-resolution reconstruction. Li et al. proposed to utilize convolution kernels of different sizes to construct scale-dependent image features for better restoration of both large-size structures and small-size details [23]. He et al. designed multi-receptive-model to extract features in different receptive fields from local to global [18]. In some other computer vision tasks, researchers attempted to deploy a number of kernels of different shapes to generate more comprehensive and distinctive features. For instance, Liao et al. introduced TextBoxes [37] which employed irregular 1×5 convolutional filters to yield rectangular receptive fields for text detection. In Google Inception-Net V2 [38], Ioffe et al. utilized different $1 \times n$ and $n \times 1$ rectangular kernels instead of $n \times n$ square kernels so as to decrease parameters. Li et al. introduced multi-scale feature extraction blocks which contain convolutional kernels of various shapes (e.g., 3×3 , 1×5 , 5×1 , 1×7 , 7×1 , and 1×1) to generate informative features for classification of eye defects [39]. In this paper, we present a novel feature extraction module which contains the mixture of 1D and 2D convolutional kernels (i.e., 5×1 , 1×5 , and 3×3) for computing orientation-aware features for the SISR task.

To perform high-accuracy HR image reconstruction, it is important to utilize the hierarchical features extracted in different convolutional stages. Many deep CNN models added dense skip connections to combine low-level features extracted in shallower layers with semantic features computed in deeper layers to generate more informative feature maps and tackle the problem of gradient vanishing [6], [7], [10], [25]. Huang et al. introduced dense skip connections into DenseNet models, reusing the feature maps of preceding layers to enhance the representation of features, and alleviate the problem of gradient vanishing [10]. Zhang et al. proposed to fuse hierarchical feature maps extracted in stacked residual dense

blocks, achieving better reconstruction results [26]. Tai et al. proposed densely concatenated memory blocks to reconstruct accurate details for the task of image restoration [9]. Given the channel-wise concatenated features, it is desirable to design an effective fusion scheme for selecting the most distinctive ones. The channel attention (CA) mechanism, which was initially proposed for image classification tasks [40], [41], has recently been adopted to solve the challenging SISR problem via recalibrating the feature responses towards the most informative and important channels of the feature maps [11], [15], [42]. In this paper, we design/optimize CA-based fusion schemes to adaptively combine features extracted in different directions and in hierarchically stacked convolutional stages.

III. APPROACH

In this section, we propose a CNN-based model for fast and accurate SISR via Channel Attention-based fusion of Orientation-Aware features (SISR-CA-OA). We first present the architecture of the proposed SISR-CA-OA model. Then we provide details of the key building blocks of the SISR-CA-OA model include **a.** the orientation-aware feature extraction modules and **b.** the channel attention-based multiple feature fusion schemes.

A. Network Architecture

As illustrated in Fig. 1, the SISR-CA-OA model consists of three major processing steps: (1) initial feature extraction on the input LR image I^{LR} , (2) orientation-aware feature extraction and fusion, (3) HR image reconstruction.

Given a LR input image I^{LR} ($H \times W$), a 3×3 convolutional layer is firstly deployed to extract low-level features $F_0 \in \mathbb{R}^{C \times H \times W}$ (C - channel number, H - image height, W - image width) as

$$F_0 = \text{Conv}_{3 \times 3}(I^{LR}), \quad (1)$$

where $\text{Conv}_{3 \times 3}$ denotes the convolution operation using a 3×3 kernel. Then, the extracted F_0 is fed to a number of stacked OAMs to compute orientation-aware features F_n

in different convolutional stages (more details of orientation-aware feature extraction are provided in Sec. III-B). Within each OAM, we design a LCA-based fusion scheme to perform the scene-specific fusion of multiple orientation-aware features. Moreover, we present a feature fusion technique based on the GCA mechanism to integrate low-level and high-level semantic features extracted in hierarchically stacked OAMs. Detail information of these channel attention-based feature fusion schemes is provided in Sec. III-C.

We adopt the global residual learning technique in the proposed SISR-CA-OA model by adding an identity branch from its initial input F_0 to the hierarchically fused feature F^{GCA} (green line in Fig. 1) as

$$F_{out} = F_0 + F^{GCA}, \quad (2)$$

where $+$ calculates the sum of feature maps F_0 and F^{GCA} at the same spatial locations and channels. The computed feature maps F_{out} is then fed to two convolutional layers and an up-sampling layer to reconstruct the HR image. For a $\times R$ upscaling SISR task, two 3×3 convolutional layers are utilized to convert the channel number of F_{out} from C to $R \times R$ and the up-sampling layer performs the pixel shuffle operation [35] to reconstruct the super-resolved output I^{SR} ($RH \times RW$).

The SISR-CA-OA model is optimized by minimizing the pixel-wise difference between the predicted super-resolved image I^{SR} and corresponding ground truth I^{GT} . In this paper, the training and testing of SISR models are performed on the Y channel (i.e., luminance) of transformed YCbCr space [5], [7], [9], [26]. We adopt the two-parameter weighted Huber loss function to drive the weights learning [18]. The weighted Huber loss function sets larger back-propagated derivatives to accelerate the training process when the training residuals are significant. In comparison, it linearly decreases the back-propagated derivative to zero when the residual value is approaching zero. As a result, the weighted Huber loss combines the advantages of $L1$ and $L2$ loss functions and fits the reconstruction error more effectively.

B. Orientation-aware Feature Extraction

The CNN-based SISR models typically deploy a set of convolutional kernels to extract semantic features for HR image reconstruction. The 3×3 convolutional kernel is the most widely used option in many state-of-the-art SISR models such as VDSR [6], DRCN [21], DRRN [7], SRDenseNet [8], EDSR [25], and TSCN [22]. More recently, some researchers adopted convolutional kernels of larger sizes (e.g., 5×5 or 7×7) to generate multi-scale features [18], [23], [43]. It is noted that the existing SISR models typically utilize square-shaped and orientation-independent convolutional kernels (e.g., 3×3 or 5×5) to extract feature maps for reconstructing image structures/textures in different directions. One possible solution to build more distinctive features for high-accuracy SISR is incorporating multiple convolutional kernels of various shapes for extracting orientation-aware features in a single feature extraction module.

In each OAM, we deploy a standard convolutional layer using the 3×3 square-shaped kernel and two additional

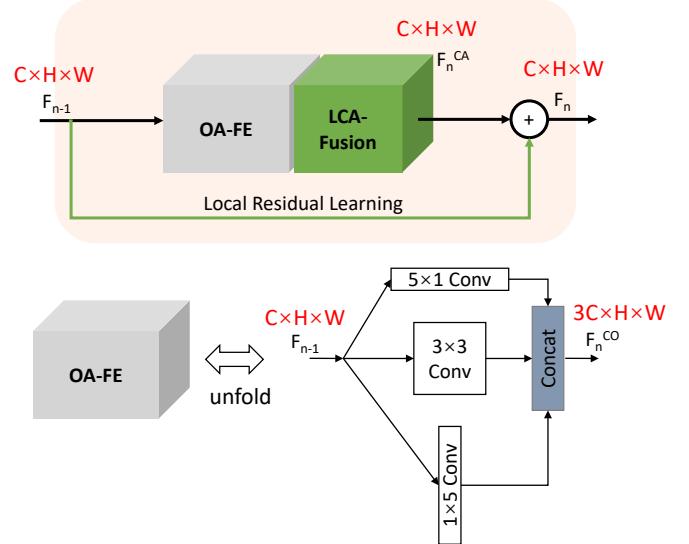


Fig. 2. The architecture of the backbone OAM for feature extraction. The input is firstly fed into three individual convolutional layers using kernels of different shapes (3×3 , 1×5 , and 5×1). Then the extracted orientation-aware feature maps are concatenated and then processed by a LCA-based fusion block to adaptively re-calibrate the channel-wise weights towards the most informative and important channels. The residual learning is also added to OAM to ease the training process.

convolutional layers using 1D kernels (i.e., 1×5 and 5×1) to extract features in different directions, as illustrated in Fig. 2. Let F_{n-1} denote the input feature maps of n -th OAM, the orientation-aware feature maps F_n^H , F_n^V , and F_n^D are computed as:

$$F_n^H = Conv_{5 \times 1}(F_{n-1}), \quad (3)$$

$$F_n^V = Conv_{1 \times 5}(F_{n-1}), \quad (4)$$

$$F_n^D = Conv_{3 \times 3}(F_{n-1}), \quad (5)$$

where $Conv_{5 \times 1}$, $Conv_{1 \times 5}$ and $Conv_{3 \times 3}$ represent the convolution operations using the 5×1 , 1×5 , and 3×3 kernels, respectively. The 1D 1×5 kernel only considers the information in the horizontal direction thus can better extract vertical features. On the other hand, the 5×1 kernel only covers vertical pixels thus is more suitable for the extraction of horizontal features. In this manner, we propose to firstly extract orientation-aware features in different directions (e.g., horizontal, vertical, and diagonal) and then perform scene-specific fusion to generate more informative features for SISR. In Sec. IV-C1, we will set up experiments to systematically evaluate the effectiveness of the proposed orientation-aware feature extraction technique.

C. Channel Attention-based Feature Fusion

Previous research works have proven the effectiveness of Channel attention (CA) mechanism [19], providing a way to re-calibrate channel-wise features via explicitly modeling interdependencies between channels in the task of super-resolution [15], [42], [44], [45]. In this paper, we design CA-based fusion schemes to integrate multiple features extracted in different directions and different convolutional stages. More

specifically, we incorporate a LCA mechanism within each OAM, performing the scene-specific fusion of orientation-aware features, as shown in Fig. 3 (a). Moreover, we present a GCA-based fusion scheme to integrate low-level and high-level features extracted in various convolutional stages, as illustrated in Fig. 3 (b).

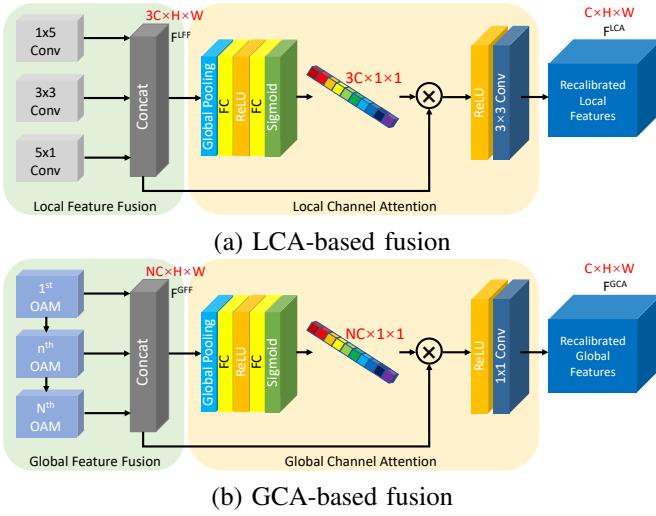


Fig. 3. CA-based fusion schemes for integrating multiple features extracted in different directions and different convolutional stages. (a) The LCA mechanism and (b) the GCA mechanism for the channel-wise weights recalibration of concatenated orientation-aware and hierarchical features.

1) *Fusion of Orientation-aware Features:* Within the n -th OAM, the computed orientation-aware feature maps F_n^H , F_n^V , and F_n^D are firstly combined through a simple concatenation operation as

$$F_n^{CO} = [F_n^H, F_n^V, F_n^D], \quad (6)$$

where $[\cdot]$ denotes the concatenation operation, and $F_n^{CO} \in \mathbb{R}^{3C \times H \times W}$ is the concatenated orientation-aware features. Given features extracted in different directions, we deploy the LCA mechanism to emphasize the informative features as well as to suppress redundant ones, performing an adaptive fusion of orientation-aware features. As illustrated in Fig. 3 (a), LCA firstly shrinks the concatenated orientation-aware features $F_n^{CO} \in \mathbb{R}^{3C \times H \times W}$ along the spatial dimensions $H \times W$ through a global average pooling operation. A channel-wise descriptor $z \in \mathbb{R}^{3C \times 1 \times 1}$ is computed and the c -th element of z is

$$z_c^{CO} = GP(F_{n,c}^{CO}) = \frac{1}{H \times W} \sum_{h=1}^H \sum_{w=1}^W F_{n,c}^{CO}(h, w), \quad (7)$$

where $GP(\cdot)$ denotes the global average pooling operation and $F_{n,c}^{CO}(h, w)$ is the value at coordinate position (h, w) of the c -th channel of F_n^{CO} . A gating mechanism [19] consisting of two fully connected (FC) layers and a ReLU activation function is then deployed to assign weights to different feature channels as

$$\alpha_c^{CO} = \sigma(FC(\delta(FC(z^{CO})))), \quad (8)$$

where $FC(\cdot)$ are the FC layers, $\delta(\cdot)$ represents the ReLU function. Note a sigmoid function $\sigma(\cdot)$ is utilized to adjust the channel attention weights to the range between 0 and 1.

The first FC layer reduces the channel dimension to $\frac{1}{s}$ and the second FC layer increases the channel dimension from $\frac{3C}{s}$ back to $3C$. The re-calculated output F_n^{CO-LCA} is obtained by rescaling the concatenated orientation-aware F_n^{CO} with the attention weights α_c^{CO} channel-wisely. More specifically, the c -th channel of F_n^{CO-LCA} can be calculated as

$$F_{n,c}^{CO-LCA} = \alpha_c^{CO} \cdot F_{n,c}^{CO}. \quad (9)$$

It is worth mentioning that the scene-specific channel weights α_c^{CO} are completely self-learned without supervision. As a result, LCA adaptively assigns higher weights for the informative features as well as to suppress redundant ones to perform the adaptive fusion of orientation-aware features.

The computed features F_n^{CO-LCA} is then activated using a ReLU function and fed into a 3×3 convolutional layer, squeezing the channel number of F_n^{CO-LCA} from $3C$ to C as

$$F_n^{LCA} = Conv_{3 \times 3}(\delta(F_n^{CO-LCA})). \quad (10)$$

Local residual learning technique (green line in Fig. 2 (a)) is also deployed to alleviate the gradient vanishing/exploring problem [25], [26], thus the output of the n -th OAM is

$$F_n = F_{n-1} + F_n^{LCA}. \quad (11)$$

2) *Fusion of Hierarchical Features:* It is important to utilize the hierarchical features extracted in different convolutional stages for high-accuracy SISR [8], [21], [26], [46], [47]. As illustrated in Fig. 2 (b), we deploy a GCA mechanism to recalibrate the channel weights for the concatenated hierarchical features, adaptively combining semantic features extracted in deeper layers and low-level features extracted in shallower layers. Given the outputs of N OAMs (F_1, F_2, \dots, F_N), we compute the concatenated hierarchical features $F^{CH} \in \mathbb{R}^{NC \times H \times W}$ as

$$F^{CH} = [F_1, F_2, \dots, F_N]. \quad (12)$$

Similarly, the GCA mechanism calculates channel-wise attention weights for the concatenated hierarchical features F^{CH} as

$$\alpha^{CH} = \sigma(FC(\delta(FC(GP(F^{CH}))))). \quad (13)$$

The re-calibrated concatenated hierarchical features F^{CH-GCA} are calculated by rescaling F^{CH} with the attention weights α^{CH} channel-wisely as

$$F_c^{CH-GCA} = \alpha_c^{CH} \cdot F_c^{CH}. \quad (14)$$

Note the computed F^{CH-GCA} is also activated using a ReLU function to embed more nonlinear terms into the network. Moreover, a 1×1 convolutional layer is utilized to compress the channel number from $N \times C$ to C . The final output of GCA-based hierarchical feature fusion is

$$F^{GCA} = Conv_{1 \times 1}(\delta(F^{CH-GCA})). \quad (15)$$

In Sec. IV-C2, we set up systematical experiments to validate the effectiveness of the proposed LCA/GCA-based fusion schemes. Moreover, we investigate a number of design options for integrating the CA mechanism in our proposed SISR-CA-OA model to achieve better fusion of multiple features extracted in different orientations and different convolutional stages.

IV. EXPERIMENTAL RESULTS

In this section, we systematically evaluate the performance of our proposed SISR-CA-OA model and compare it with the state-of-the-art SISR methods quantitatively and qualitatively on a number of commonly used benchmark datasets.

A. Datasets and Metrics

Training: Following [6], [21], we train a light-weight version of SISR-CA-OA model consisting of 10 OAMs on RGB91 dataset from Yang et al. [2] and another 200 images from Berkeley Segmentation Dataset (BSD) [30]. Moreover, we make use of the DIVerse 2K resolution image dataset (i.e., DIV2K) [36] to train an enhanced SISR-CA-OA model (SISR-CA-OA*) which contains 64 stacked OAMs. Three commonly used data augmentation techniques are utilized to expand our training dataset, including 1. Rotation: rotate image by 90° , 180° and 270° . 2. Flipping: horizontally flip image. 3. Scaling: downscale image with the scale factors of 0.9, 0.8, 0.7, 0.6 and 0.5. After the augmentation, we randomly crop these images into a number of sub-images (48×48 for the RGB91 and BSD datasets and 96×96 for the DIV2K dataset). The LR images are obtained by down-sampling corresponding HR images using bicubic interpolation.

Testing: Five commonly used public benchmark datasets are utilized for evaluating the performance of our SISR-CA-OA model. Set5 [28] and Set14 [29] are widely used datasets in SISR tasks. B100 [30] contains 100 natural images collected from BSD, and Urban100 [31] consists of 100 real-world images which are rich of structures. Manga109 dataset, which consists of a variety of 109 Japanese comic books, is also employed [32].

Evaluation Metrics: Peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) [48] are used for SISR performance evaluation. The training and testing of SISR models are performed on the Y channel (i.e., luminance) of transformed YCbCr space [5], [7], [9], [26]. For a fair comparison, we crop pixels near image boundary according to [20].

B. Implementation Details

We implement our SISR-CA-OA model with Caffe [49] platform and train this model by optimizing Modified Huber Loss function on a single NVIDIA Quadro P6000 GPU with Cuda 9.0 and Cudnn 7.1 for 60 epochs. When training our model, we only consider the luminance channel (Y channel of YCbCr color space) in our experiments. Adam [50] solver is utilized to optimize the weights by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e^{-8}$. In each training batch, we randomly crop these augmented training images into 48×48 patches and the batch size is set to 64 for training our SISR-CA-OA. The initial learning rate is set to $1e^{-4}$ and halved after 50 epochs. Training of SISR-CA-OA models approximately takes two days. When training our SISR-CA-OA for scale factors $\times 3$ and $\times 4$, we initialize the weights with pre-trained $\times 2$ model and decrease the learning rate to $1e^{-5}$. The source codes will be made publicly available in the future.

C. Performance Analysis

In this section, we set up ablation experiments to evaluate the effectiveness of (1) orientation-aware feature extraction, and (2) channel attention based feature fusion.

1) Orientation-aware Feature Extraction: We evaluate the performance of three different designs of residual blocks including (a) a standard residual block which consists of two 3×3 convolutional layers and a ReLU activation layer [25], [42], (b) a residual block which utilizes three individual square-shaped (3×3) convolutional kernels to extract features, (c) our propose orientation-aware residual block which uses a standard 3×3 and two additional 1×5 and 5×1 convolutional kernels to extract features in different directions. For a fair comparison, three different residual blocks are implemented in the same EDSR the baseline model [25] without performing CA-based feature fusion. We set the number of residual blocks $N = 10$ and the channel number of each convolutional layer to 64. Tab. I summarizes the quantitative evaluation results (PSNR and SSIM) on Set5, Urban100, and Manga109 datasets with the scale factor $\times 2$. First of all, it is experimentally demonstrated that incorporating multiple convolutional kernels within a residual block (i.e., design (b) and (c)) can generally construct more distinctive features and achieve higher SISR accuracy. Moreover, the residual block incorporating a mixture of 1D and 2D convolutional kernels (3×3 , 1×5 , and 5×1) performs better than the one based on three square-shaped kernels (3×3), achieving higher PSNR and SSIM indexes with fewer parameters. The experimental results manifest the effectiveness of the orientation-aware design, utilizing convolutional kernels of various shapes to extract orientation-aware features for more accurate reconstruction of image structures/textures in different directions.

TABLE I
EXPERIMENTAL EVALUATION OF RESIDUAL BLOCKS OF THREE DIFFERENT DESIGNS. PSNR(DB) AND SSIM METRICS ARE CALCULATED ON SET5, URBAN100 AND MANGA109 DATASETS WITH SCALE FACTOR $\times 2$.

| | | Different Residual Blocks | | |
|----------|------|---------------------------|------------|---------------|
| | | Design (a) | Design (b) | Design (c) |
| Set5 | PSNR | 37.76 | 37.79 | 37.84 |
| | SSIM | 0.9596 | 0.9596 | 0.9600 |
| Urban100 | PSNR | 31.17 | 31.35 | 31.41 |
| | SSIM | 0.9187 | 0.9206 | 0.9216 |
| Manga109 | PSNR | 37.89 | 37.95 | 38.02 |
| | SSIM | 0.9747 | 0.9748 | 0.9749 |

2) CA-based Feature Fusion: In this section, we set up three ablation experiments to evaluate the effectiveness of the proposed CA-based feature fusion schemes, as illustrated in Fig. 5. In Experiment A, the concatenated orientation-aware and hierarchical features are directly fed to a ReLU activation layer and a convolution layer to compute the fused feature map without utilizing the LCA/GCA mechanisms for channel weights re-calibration. In Experiment B, we only perform the LCA-based fusion in individual OAMs to combine multiple outputs of orientation-dependent convolutional kernels. In Experiment C, we perform both LCA-based orientation-aware feature fusion and GCA-based hierarchical feature fusion.

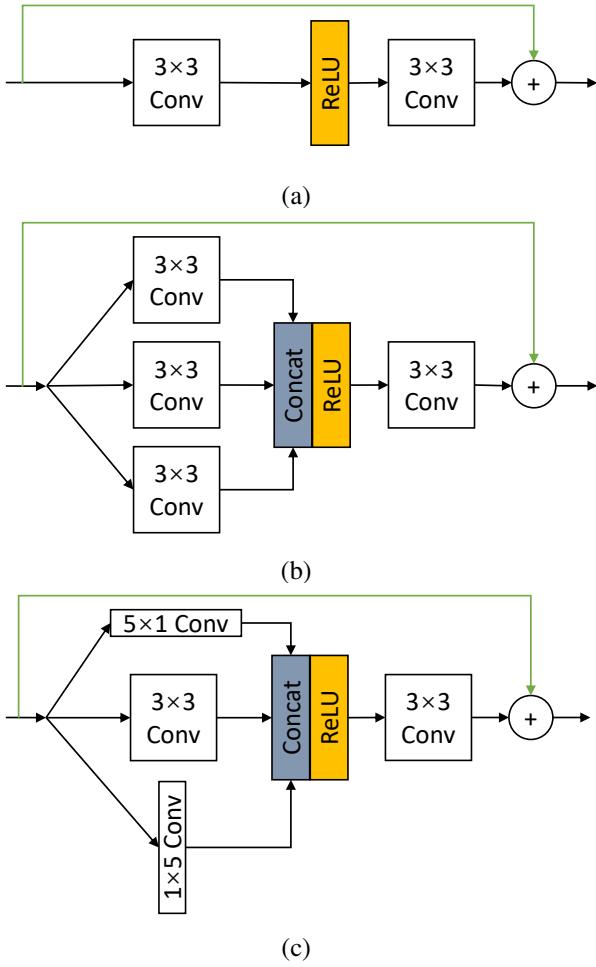


Fig. 4. Structures of three residual block designs: (a) A residual block utilized in many SISR models [25], [42], (b) A residual block which contains three square-shaped (3×3) convolutional kernels for feature extraction, (c) Our proposed orientation-aware residual block which incorporates a mixture of 1D and 2D convolutional (3×3 , 1×5 , and 5×1) kernels.

TABLE II
COMPARATIVE RESULTS OF THREE ABLATION EXPERIMENTS
WITH/WITHOUT PERFORMING THE CA-BASED CHANNEL WEIGHTS
RE-CALIBRATION. PSNR(DB) AND SSIM METRICS ARE CALCULATED ON
SET5, URBAN100 AND MANGA109 DATASETS WITH SCALE FACTOR $\times 2$.

| | Different Fusion Schemes | | | |
|----------|--------------------------|-----------------|-----------------|-------------------------|
| | Exp. A | Exp. B | Exp. C | |
| LCA | ✗ | ✓ | ✓ | |
| GCA | ✗ | ✗ | ✓ | |
| Set5 | PSNR SSIM | 37.86 0.9659 | 37.90 0.9600 | 37.97 0.9605 |
| Urban100 | PSNR SSIM | 31.45 0.9217 | 31.51 0.9220 | 31.57 0.9226 |
| Manga109 | PSNR SSIM | 38.03 0.9748 | 38.11 0.9750 | 38.38 0.9755 |

Tab. II shows the comparative results (PSNR and SSIM) on Set5, Urban100, and Manga109 datasets with the scale factor $\times 2$. It is experimentally observed that the CA mechanism provides a generally effective technique for the fusion of features extracted in different directions and at various convolutional

stages. For instance, the PSNR index increases from 38.03 dB to 38.11 dB on the Manga109 dataset when incorporating a LCA mechanism within each individual OAMs. The index is further boosted from 38.11 dB to 38.38 dB by utilizing the GCA mechanism to re-calculate channel-wise weights for the concatenated hierarchical features. The underlying principle is that LCA/GCA mechanisms can adaptively assign higher weights for the informative feature channels as well as to suppress redundant ones to generate more informative fused features and achieve higher SISR accuracy.

Moreover, we experimentally evaluate a number of design options in which the CA mechanisms are placed in different positions in a feature fusion module. In Design (a) (Fig. 6 (a)), we place the LCA/GCA mechanisms after a ReLU activation function and a convolutional layer which is the commonly adopted configuration in many SISR models [11], [15], [42], [44], [45]. In Design (b) (Fig. 6 (b)), we put the CA re-calibration functions between the ReLU and convolutional layers. In Design (c) (Fig. 6 (c)), we move the LCA/GCA mechanism to the position before the ReLU and convolutional layers. Note the ReLU activation layer is utilized to embed more nonlinear terms into the network, and the convolutional layer compresses the channel number of concatenated features. Tab. III shows the experimental results of different designs on Set5, Urban100, and Manga109 datasets with the scale factor $\times 2$. It is observed that Design (c) achieves higher PSNR and SSIM indexes on all testing datasets than other alternatives. The experimental results illustrate that it is better to immediately utilize the CA mechanism to re-calibrate channel-wise weights for concatenated feature maps before squeezing the channel number of features (the convolutional layer) or converting the negative inputs to zeros (the ReLU activation function).

TABLE III
COMPARATIVE EVALUATION OF THREE DESIGN OPTIONS IN WHICH THE
CA-BASED CHANNEL WEIGHTS RE-CALIBRATION IS PERFORMED IN
DIFFERENT POSITIONS IN A FEATURE FUSION MODULE. THE
EXPERIMENTAL METRICS (PSNR(DB) AND SSIM) ARE CALCULATED ON
SET5, URBAN100 AND MANGA109 DATASETS WITH SCALE FACTOR $\times 2$.

| Comparison | Different CA-based Fusions | | | |
|------------|----------------------------|-----------------|-----------------|-------------------------|
| | Design (a) | Design (b) | Design (c) | |
| Set5 | PSNR SSIM | 37.91 0.9603 | 37.93 0.9602 | 37.97 0.9605 |
| | PSNR SSIM | 31.46 0.9220 | 31.50 0.9221 | 31.57 0.9226 |
| Urban100 | PSNR SSIM | 38.03 0.9748 | 38.11 0.9750 | 38.38 0.9755 |
| | PSNR SSIM | 38.13 0.9751 | 38.21 0.9748 | 38.38 0.9755 |

D. Comparisons with State-of-the-art SISR Methods

Firstly, we compare our proposed light-weight SISR-CA-OA model (containing 10 OAMs) with a number of fast and accurate SISR methods which are also trained on the RGB91 [2] and BSD [30] datasets. More specifically, we consider Aplus [3], SelfExSR [31], SRCNN [20], VDSR [6], DRCN [21], ms-LapSRN [17], DRRN [7], MemNet [9], and TSCN [22]. Source codes or pre-trained models of these methods are publicly available.

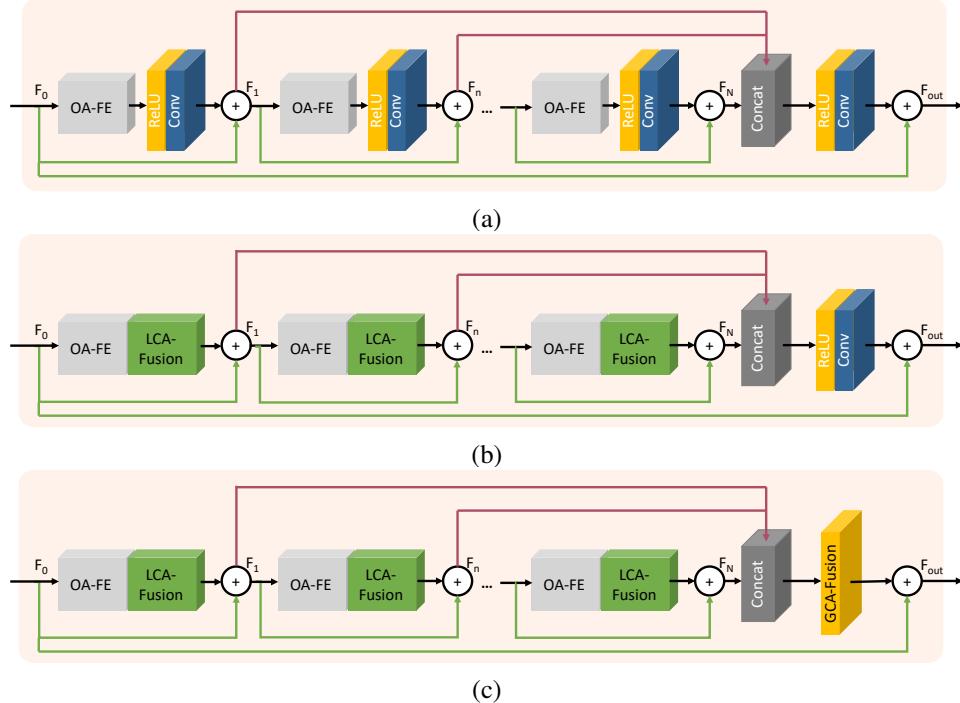


Fig. 5. Ablation experiments to evaluate the effectiveness of the proposed CA-based feature fusion schemes. (a) Feature fusion without utilizing the LCA/GCA mechanisms for channel weights re-calibration, (b) Feature fusion with LCA-based channel weights re-calibration only, (c) Feature fusion incorporating both LCA and GCA mechanisms.

Tab. IV (PSNR and SSIM indexes) show quantitative evaluation results on Set5, Set14, B100, Urban100, and Manga109 with the scale factors $\times 2$, $\times 3$, $\times 4$. Tab. V shows the averaged running time of different SISR methods to process 100 input images of three different resolutions including 480×360 , 640×480 and 1280×720 . The testing is conducted on a PC which is equipped with NVIDIA Quadro P6000 GPU (24 GB memory). It is observed that our proposed SISR-CA-OA model performs favorably against these SISR models in terms of both restoration accuracy and computational efficiency. It achieves higher PSNR and SSIM values than some very deep networks (e.g., DRCN [21], DRRN [7], MemNet [9]) and runs faster compared with some light-weight SISR models such as TSCN [22] and ms-LapSRN [17]. Some visual comparative results with state-of-the-art deep-learning-based SISR methods are shown in Fig. 7, 8, and 9. It is observed that our SISR-CA-OA model can achieve better image restoration results for three different scale factors ($\times 2$, $\times 3$, and $\times 4$). As shown in Fig. 7 and Fig. 9, the SISR-CA-OA model can restore sharper and clearer texture patterns in the highlighted regions. Moreover, it can effectively suppress undesired artifacts or distortions when reconstructing parallel edges/structures, as illustrated in Fig. 8.

Moreover, we compare the enhanced SISR-CA-OA* model (containing 64 OAMs) with the best-performing SISR models trained on the high-resolution DIV2K dataset including MSRN [23], D-DBPN [24], EDSR [25], and RDN [26]. The pre-trained models of these methods are publicly available. We calculate the average PSNR and SSIM values for scale factors $\times 2$, $\times 3$ and $\times 4$ on Set5, Set14, B100, Urban100 and Manga109 testing datasets. As illustrated in Tab. VI, the proposed SISR-CA-OA* model also achieves the highest PSNR and SSIM

values in most cases. Compared with other SISR models trained on the high-resolution DIV2K dataset, our SISR-CA-OA* can more accurately restore complex image details (Fig. 10) without incurring undesired artifacts (Fig. 11) in large scale factor ($\times 4$) SISR tasks.

V. CONCLUSION

In this paper, we proposed a novel CNN-based model for high-quality SISR via channel attention-based fusion of orientation-aware features. Instead of utilizing square-shaped convolutional kernels (e.g., 3×3 or 5×5) to extract features [5]–[9], we integrate multiple convolutional kernels of various shapes (i.e., 5×1 , 1×5 , and 3×3) in a single feature extraction module to extract orientation-aware features. Moreover, we adopt the channel attention mechanism for the local fusion of features extracted in different directions and the global fusion of features extracted in hierarchical stages. Extensive benchmark evaluations well demonstrate that our proposed SISR-CA-OA model achieves superiority over state-of-the-art SISR methods [6], [7], [9], [20]–[26] in terms of both restoration accuracy and computational efficiency.

REFERENCES

- [1] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” in *2008 IEEE conference on computer vision and pattern recognition*. Citeseer, 2008, pp. 1–8.
- [2] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [3] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *Asian conference on computer vision*. Springer, 2014, pp. 111–126.

TABLE IV

BENCHMARK RESULTS ON STATE-OF-THE-ART SISR METHODS. WE CALCULATE THE AVERAGE PSNR(DB)/SSIM VALUES ON SET5, SET14, B100, URBAN100 AND MANGA109 DATASETS WITH SCALE FACTORS $\times 2$, $\times 3$ AND $\times 4$. THE COLOR RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCE RESPECTIVELY. IT IS NOTED THAT THE METRICS ARE CALCULATED ON Y CHANNEL (ILLUMINATION CHANNEL OF YCbCr COLOR SPACE).

| Scale | Method | Set5 PSNR / SSIM | Set14 PSNR / SSIM | B100 PSNR / SSIM | Urban100 PSNR / SSIM | Manga109 PSNR / SSIM |
|------------|----------------|---------------------|----------------------|---------------------|-------------------------|-------------------------|
| $\times 2$ | Bicubic | 33.66 / 0.9299 | 30.24 / 0.8688 | 29.56 / 0.8431 | 26.88 / 0.8403 | 30.81 / 0.9341 |
| | Aplus [3] | 36.54 / 0.9544 | 32.28 / 0.9056 | 31.21 / 0.8863 | 29.20 / 0.8938 | 35.37 / 0.9680 |
| | SelfExSR [31] | 36.50 / 0.9536 | 32.22 / 0.9034 | 31.17 / 0.8853 | 29.52 / 0.8965 | 35.12 / 0.9660 |
| | SRCNN [20] | 36.66 / 0.9542 | 32.45 / 0.9067 | 31.36 / 0.8879 | 29.51 / 0.8964 | 35.60 / 0.9663 |
| | VDSR [6] | 37.53 / 0.9587 | 33.03 / 0.9124 | 31.90 / 0.8960 | 30.76 / 0.9140 | 37.15 / 0.9738 |
| | DRCN [21] | 37.63 / 0.9588 | 33.04 / 0.9118 | 31.85 / 0.8942 | 30.75 / 0.9133 | 37.63 / 0.9740 |
| | ms-LapSRN [17] | 37.70 / 0.9590 | 33.25 / 0.9138 | 32.02 / 0.8970 | 31.13 / 0.9180 | 37.71 / 0.9747 |
| | DRRN [7] | 37.74 / 0.9591 | 33.23 / 0.9136 | 32.05 / 0.8973 | 31.23 / 0.9188 | 37.88 / 0.9749 |
| | MemNet [9] | 37.78 / 0.9597 | 33.28 / 0.9142 | 32.08 / 0.8978 | 31.31 / 0.9195 | 38.03 / 0.9755 |
| | TSCN [22] | 37.88 / 0.9602 | 33.28 / 0.9147 | 32.09 / 0.8985 | 31.29 / 0.9198 | 38.07 / 0.9750 |
| $\times 3$ | SISR-CA-OA | 37.97 / 0.9605 | 33.42 / 0.9158 | 32.15 / 0.8993 | 31.57 / 0.9226 | 38.38 / 0.9755 |
| | Bicubic | 30.39 / 0.8682 | 27.55 / 0.7742 | 27.21 / 0.7385 | 24.46 / 0.7349 | 26.96 / 0.8546 |
| | Aplus [3] | 32.58 / 0.9088 | 29.13 / 0.8188 | 28.29 / 0.7835 | 26.03 / 0.7973 | 29.93 / 0.8120 |
| | SelfExSR [31] | 32.64 / 0.9097 | 29.15 / 0.8196 | 28.29 / 0.7840 | 26.46 / 0.8090 | 29.61 / 0.9050 |
| | SRCNN [20] | 32.75 / 0.9090 | 29.29 / 0.8215 | 28.41 / 0.7863 | 26.24 / 0.7991 | 30.48 / 0.9117 |
| | VDSR [6] | 33.66 / 0.9213 | 29.77 / 0.8314 | 28.82 / 0.7976 | 27.14 / 0.8279 | 32.00 / 0.9329 |
| | DRCN [21] | 33.82 / 0.9226 | 29.76 / 0.8311 | 28.80 / 0.7963 | 27.15 / 0.8276 | 32.31 / 0.9360 |
| | ms-LapSRN [17] | 34.06 / 0.9249 | 29.97 / 0.8353 | 28.92 / 0.8006 | 27.47 / 0.8369 | 32.68 / 0.9385 |
| | DRRN [7] | 34.03 / 0.9244 | 29.96 / 0.8349 | 28.95 / 0.8004 | 27.53 / 0.8378 | 32.71 / 0.9379 |
| | MemNet [9] | 34.09 / 0.9248 | 30.00 / 0.8350 | 28.96 / 0.8001 | 27.56 / 0.8376 | 32.79 / 0.9388 |
| $\times 4$ | TSCN [22] | 34.18 / 0.9256 | 29.99 / 0.8351 | 28.95 / 0.8012 | 27.46 / 0.8362 | 32.68 / 0.9381 |
| | SISR-CA-OA | 34.23 / 0.9261 | 30.05 / 0.8363 | 29.01 / 0.8023 | 27.67 / 0.8403 | 32.92 / 0.9391 |

TABLE V

AVERAGE RUNNING TIME FOR SCALE FACTOR $\times 4$ ON THREE DIFFERENT RESOLUTION SETTINGS, INCLUDING 480×360 , 640×480 , 1280×720 . THE TESTING IS CONDUCTED ON A PC WHICH IS EQUIPPED WITH NVIDIA QUADRO P6000 GPU (24 GB MEMORY). THE COLOR RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCE RESPECTIVELY.

| Dataset | Resolution | VDSR [6] | DRCN [21] | DRRN [7] | MemNet [9] | ms-LapSRN [17] | TSCN [22] | SISR-CA-OA |
|---------|-------------------|----------|-----------|----------|------------|----------------|---------------|---------------|
| Time(s) | 480×360 | 0.0292 | 0.4895 | 5.9494 | 8.3749 | 0.0317 | 0.0235 | 0.0151 |
| | 640×480 | 0.0512 | 0.8687 | 9.8489 | 15.7984 | 0.0453 | 0.0379 | 0.0221 |
| | 1280×720 | 0.1553 | 2.6031 | 31.7105 | 47.8494 | 0.1368 | 0.1047 | 0.0507 |

- [4] S. Schulter, C. Leistner, and H. Bischof, “Fast and accurate image upscaling with super-resolution forests,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.
- [5] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *European conference on computer vision*. Springer, 2014, pp. 184–199.
- [6] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.
- [7] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, 2017, pp. 3147–3155.
- [8] T. Tong, G. Li, X. Liu, and Q. Gao, “Image super-resolution using dense skip connections,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4799–4807.
- [9] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: A persistent memory network for image restoration,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4539–4547.
- [10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [11] Y. Hu, J. Li, Y. Huang, and X. Gao, “Channel-wise and spatial feature modulation network for single image super-resolution,” *IEEE Transactions on Circuits and Systems for Video Technology*, p. In Press, 2019.
- [12] R. Timofte, E. Agustsson, L. V. Gool, M. H. Yang, and L. Zhang, “NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results,” in *CVPR workshop*, vol. 2017-July, 2017, pp. 1110–1121.
- [13] R. Timofte, S. Gu, J. Wu, L. V. Gool, and L. Zhang, “NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results,” in *CVPR workshop*, 2018, pp. 1–12.
- [14] J. Cai, S. Gu, R. Timofte, and L. Zhang, “Ntire 2019 challenge on real image super-resolution: Methods and results,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June

TABLE VI

BENCHMARK RESULTS OF SISR MODELS TRAINED ON THE HIGH-RESOLUTION DIV2K DATASET. AVERAGE PSNR AND SSIM VALUES ARE CALCULATED FOR SCALE FACTORS $\times 2$, $\times 3$ AND $\times 4$ ON SET5, SET14, B100, URBAN100 AND MANGA109 DATASETS. THE COLOR RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCE RESPECTIVELY.

| Dataset | Scale | MSRN [23] | D-DBPN [24] | EDSR [25] | RDN [26] | SISR-CA-OA* |
|----------|------------|----------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Set5 | $\times 2$ | 38.08 / 0.9605 | 38.09 / 0.9600 | 38.11 / 0.9601 | 38.24 / 0.9614 | 38.22 / 0.9613 |
| | $\times 3$ | 34.38 / 0.9262 | — | 34.65 / 0.9282 | 34.71 / 0.9296 | 34.73 / 0.9297 |
| | $\times 4$ | 32.07 / 0.8903 | 32.47 / 0.8980 | 32.46 / 0.8968 | 32.47 / 0.8990 | 32.55 / 0.8994 |
| Set14 | $\times 2$ | 33.74 / 0.9170 | 33.85 / 0.9190 | 33.92 / 0.9195 | 34.01 / 0.9212 | 33.91 / 0.9208 |
| | $\times 3$ | 30.34 / 0.8395 | — | 30.52 / 0.8462 | 30.57 / 0.8468 | 30.59 / 0.8470 |
| | $\times 4$ | 28.60 / 0.7751 | 28.82 / 0.7860 | 28.80 / 0.7876 | 28.81 / 0.7871 | 28.86 / 0.7882 |
| B100 | $\times 2$ | 32.23 / 0.9013 | 32.27 / 0.9000 | 32.32 / 0.9013 | 32.34 / 0.9017 | 32.35 / 0.9016 |
| | $\times 3$ | 29.08 / 0.8041 | — | 29.25 / 0.8093 | 29.26 / 0.8093 | 29.29 / 0.8099 |
| | $\times 4$ | 27.52 / 0.7273 | 27.72 / 0.7400 | 27.71 / 0.7420 | 27.72 / 0.7419 | 27.76 / 0.7424 |
| Urban100 | $\times 2$ | 32.22 / 0.9326 | 32.55 / 0.9324 | 32.93 / 0.9351 | 32.89 / 0.9353 | 33.03 / 0.9359 |
| | $\times 3$ | 28.08 / 0.8554 | — | 28.80 / 0.8653 | 28.80 / 0.8653 | 28.98 / 0.8680 |
| | $\times 4$ | 26.04 / 0.7896 | 26.38 / 0.7946 | 26.64 / 0.8033 | 26.61 / 0.8028 | 26.74 / 0.8060 |
| Manga109 | $\times 2$ | 38.82 / 0.9868 | 38.89 / 0.9775 | 39.10 / 0.9773 | 39.18 / 0.9780 | 39.24 / 0.9778 |
| | $\times 3$ | 33.44 / 0.9427 | — | 34.17 / 0.9476 | 34.13 / 0.9484 | 34.38 / 0.9493 |
| | $\times 4$ | 30.17 / 0.9034 | 30.91 / 0.9137 | 31.02 / 0.9148 | 31.00 / 0.9151 | 31.22 / 0.9168 |

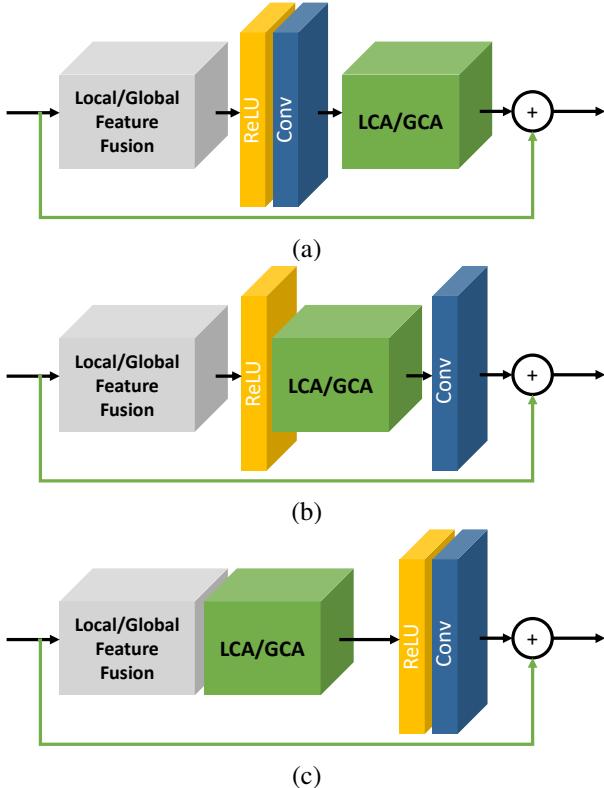


Fig. 6. Three design options in which the CA mechanisms are placed in different positions in a feature fusion module. (a) The CA-based re-calibration functions are applied after a ReLU activation function and a convolutional layer, (b) The CA-based re-calibration functions are placed between the ReLU and convolutional layers, (c) The LCA/GCA mechanisms are deployed before the ReLU and convolutional layers. Note Design (a) is the commonly adopted CA configuration in many SISR models [11], [15], [42], [44], [45].

2019.

- [15] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.
- [16] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the*

- European Conference on Computer Vision (ECCV)*, 2018, pp. 252–268.
- [17] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE transactions on pattern analysis and machine intelligence*, 2018.
 - [18] Z. He, Y. Cao, L. Du, B. Xu, J. Yang, Y. Cao, S. Tang, and Y. Zhuang, "Mrfn: Multi-receptive-field network for fast and accurate single image super-resolution," *IEEE Transactions on Multimedia*, 2019.
 - [19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
 - [20] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.
 - [21] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-Recursive Convolutional Network for Image Super-Resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.
 - [22] Z. Hui, X. Wang, and X. Gao, "Two-stage convolutional network for image super-resolution," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2670–2675.
 - [23] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 517–532.
 - [24] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664–1673.
 - [25] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.
 - [26] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.
 - [27] C. Du, H. Zewei, S. Anshun, Y. Jiangxin, C. Yanlong, C. Yanpeng, T. Siliang, and M. Ying Yang, "Orientation-aware deep neural network for real image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
 - [28] M. Bevilacqua, A. Roumy, C. Guillemot, and A. Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *British Machine Vision Conference (BMVC)*, 2012, pp. 1–10.
 - [29] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.
 - [30] D. Martin, C. Fowlkes, D. Tal, J. Malik *et al.*, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." *Iccv Vancouver*, 2001.
 - [31] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

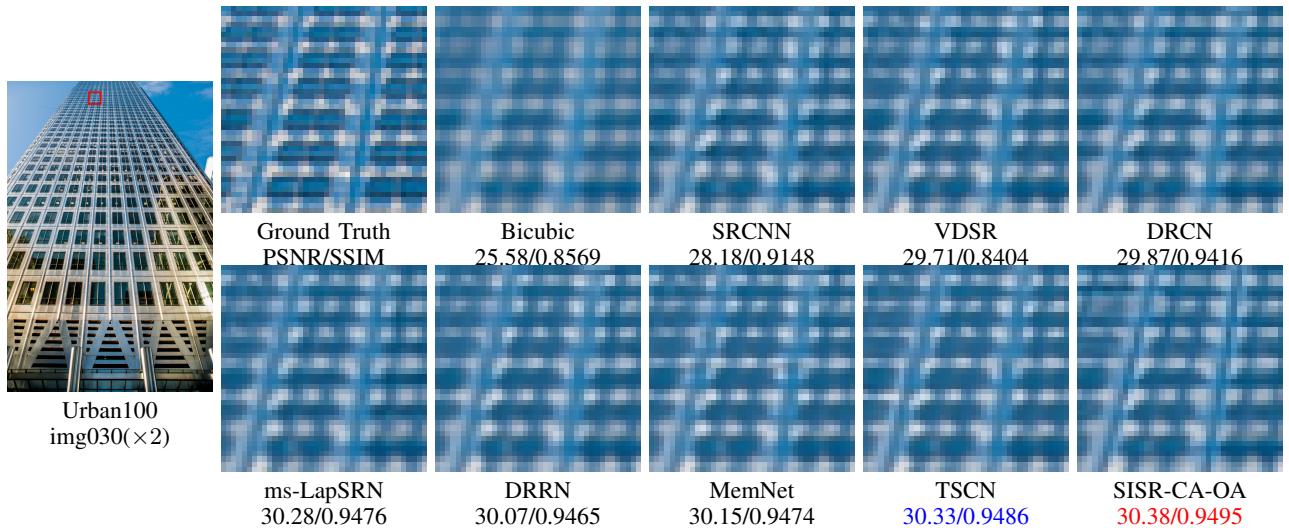


Fig. 7. Visual comparison of $\times 2$ SISR results for “img030” in the Urban100 dataset. Note all SISR models are trained on the RGB91 [2] and BSD [30] datasets.

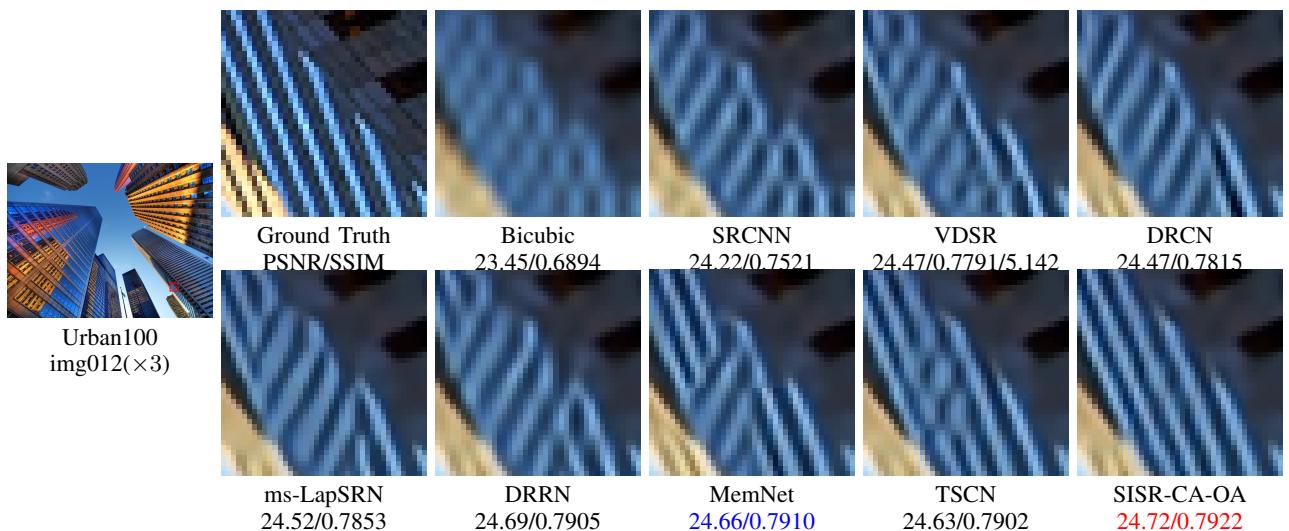


Fig. 8. Visual comparison of $\times 3$ SISR results for “img012” in the Urban100 dataset. Note all SISR models are trained on the RGB91 [2] and BSD [30] datasets.

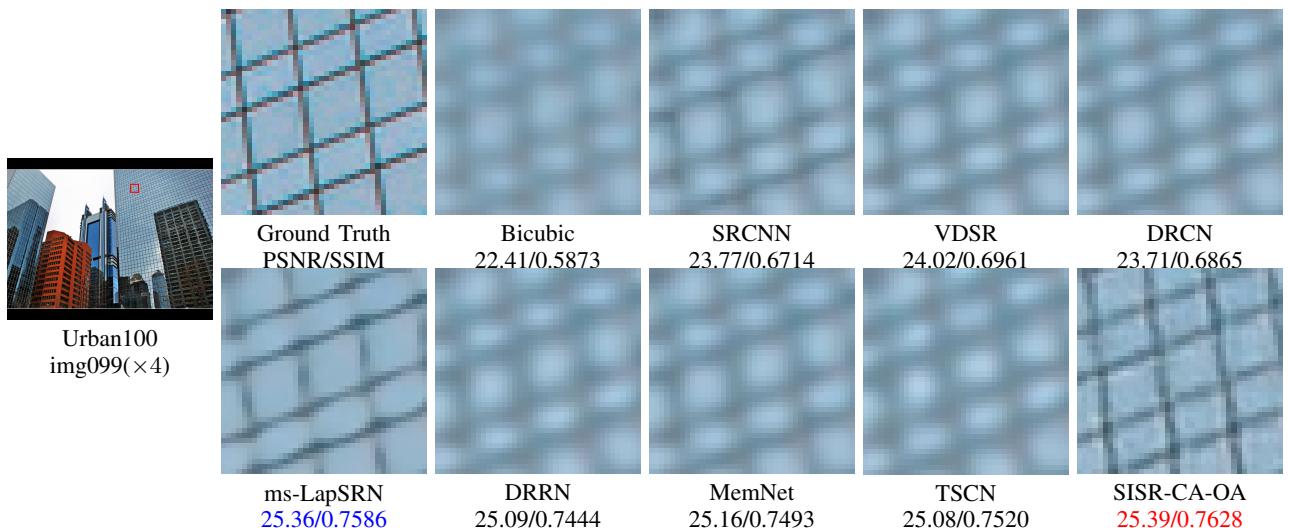
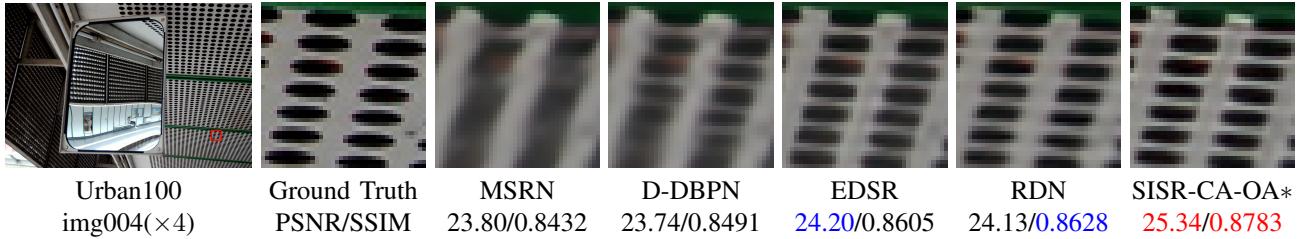
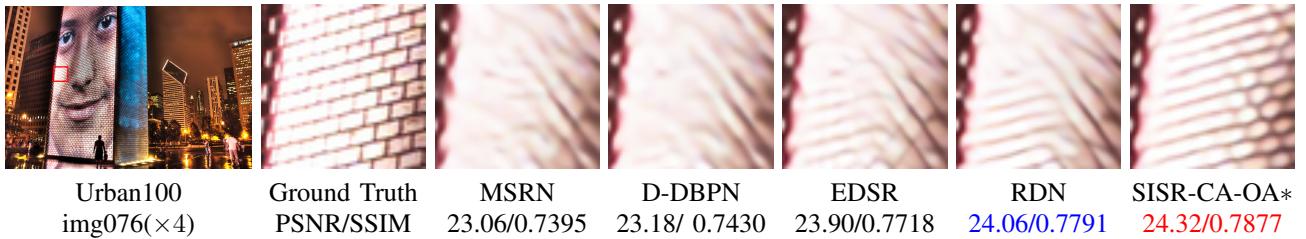


Fig. 9. Visual comparison of $\times 4$ SISR results for “img099” in the Urban100 dataset. Note all SISR models are trained on the RGB91 [2] and BSD [30] datasets.

Fig. 10. Visual comparison of $\times 4$ SISR results for “img004” in the Urban100 dataset. Note all SISR models are trained on the DIV2K dataset [36].Fig. 11. Visual comparison of $\times 4$ SISR results for “img076” in the Urban100 dataset. Note all SISR models are trained on the DIV2K dataset [36].

- [32] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, “Manga109 dataset and creation of metadata,” in *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*. ACM, 2016, p. 2.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [34] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” in *European conference on computer vision*. Springer, 2016, pp. 391–407.
- [35] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.
- [36] E. Agustsson and R. Timofte, “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017.
- [37] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, “Textboxes: A fast text detector with a single deep neural network,” in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [38] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [39] L. Li, M. Xu, H. Liu, Y. Li, X. Wang, L. Jiang, Z. Wang, X. Fan, and N. Wang, “A large-scale database and a cnn model for attention-based glaucoma detection,” *IEEE transactions on medical imaging*, 2019.
- [40] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, “The application of two-level attention models in deep convolutional neural network for fine-grained image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 842–850.
- [41] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.
- [42] T. Jiang, Y. Zhang, X. Wu, Y. Rao, and M. Zhou, “Single Image Super-Resolution via Squeeze and Excitation Network,” in *BMVC*, 2018, p. In Press.
- [43] Y. Hu, X. Gao, J. Li, Y. Huang, and H. Wang, “Single Image Super-Resolution via Cascaded Multi-scale Cross Network,” *arXiv preprint*, pp. 1–12, 2018.
- [44] Y. Lu, Y. Zhou, Z. Jiang, X. Guo, and Z. Yang, “Channel attention and multi-level features fusion for single image super-resolution,” *arXiv preprint arXiv:1810.06935*, 2018.
- [45] X. Cheng, X. Li, J. Yang, and Y. Tai, “Sesr: single image super resolution with recursive squeeze and excitation networks,” in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 147–152.
- [46] Y. Wang, J. Shen, and J. Zhang, “Deep bi-dense networks for image super-resolution,” in *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018, pp. 1–8.
- [47] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.
- [48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity.” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, apr 2004.
- [49] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.
- [50] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.