Neurocomputing 423 (2021) 98–109

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Image super-resolution based on dense convolutional auto-encoder blocks

Yuan Zhou [a], Yeda Zhang [b], Xukai Xie [a,*], Sun-Yuan Kung [c]

[a] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
[b] KylinSoft, Changsha 410005, China
[c] Department of Electrical Engineering, Princeton University, Princeton, NJ 08540, USA

ARTICLE INFO

Article history:
Received 3 December 2019
Revised 4 August 2020
Accepted 26 September 2020
Available online 6 October 2020
Communicated by Steven Hoi

Keywords:
Super resolution
Dense connection
Convolutional auto-encoders

ABSTRACT

Deep convolutional neural networks (DCNNs) have recently boosted the performance of image super-resolution (SR) by learning deep non-linear mappings from low-resolution images to their high-resolution counterparts. In general, these methods learn the mapping relationship in image space of a single scale. In this paper, we consider that features across different scales can provide various types of information for SR. Thus, we propose a novel network that extracts features of different spatial resolutions for image super-resolution. We successfully build a model to learn non-linear mappings across feature spaces of various spatial resolutions. Specifically, we propose a dense convolutional auto-encoder block (DCAE), which includes several auto-encoder (AE) units and a squeeze unit, as the basic component of our model. The AE units exploit features of different resolutions through paired encoding and decoding layers. Further, we employ skip connections to combine features of the same spatial scale in one AE unit and dense connections across successive AE units in one DCAE block to establish a temporal feature reuse mechanism. The squeeze units combine features in a DCAE block and the previous DCAE block to achieve long-term temporal feature reuse. Furthermore, we extend our work by performing multi-scale supervised training to build a single framework for SR of all scale factors. Comprehensive experiments show that the proposed method outperforms state-of-the-art methods.

1. Introduction

Single image super-resolution (SISR) is a classical low-level computer vision task that aims to reconstruct a high-resolution (HR) image from a low-resolution (LR) counterpart. It has attracted considerable attention because of the growing demand for high-quality images in various applications [1–4]. Image SR is inherently ill-posed, as multiple HR solutions may exist for a single LR input. A number of SR methods have been developed to overcome this problem. These methods can be categorized into three main types: interpolation-based methods [5], reconstruction-based methods [6–8], and learning-based methods [9–24].

Recently, deep learning (DL)-based methods, especially deep convolutional neural networks (DCNNs), have achieved significant improvement in the field of super-resolution. DCNN-based SR methods reconstruct an HR image by predicting an end-to-end nonlinear mapping between the LR space and the HR space. The network can be mathematically regarded as a mapping function from low-dimensional space to high-dimensional space.

The mapping relationship is based on features extracted in the LR image space or interpolated LR (ILR) image space, according to which these methods can be primarily classified into two types.

Methods of the first type aim to learn features in the ILR image space (also called the HR space in some studies). The LR inputs are interpolated to HR and then fed to the network. SRCNN [25] is a pioneering technique that applies CNN to the SR problem. Three CNN layers are used for feature extraction, non-linear mapping, and reconstruction. Improved versions of SRCNN include very deep network for super-resolution (VDSR), which increases the network depth with a smaller filter size and residual learning [26], and deeply recursive convolutional network (DRCN), which uses recursive layers and multi-supervision [27]. Very deep CNN models developed using a block structure [28,29] based on residual units [30] uses features from different temporal levels for reconstruction. Although the aforementioned methods have significantly improved SR accuracy, the interpolated LR inputs increase the computational complexity and might introduce additional noise.

Methods of the second type learn features in the LR image space. Fast super-resolution CNN (FSRCNN) was the first method

* Corresponding author.
E-mail address: xkxie@tju.edu.cn (X. Xie).

https://doi.org/10.1016/j.neucom.2020.09.049
0925-2312/

proposed under this schema [31]. This method takes the original LR image as input, extracts features using convolutional layers, and then increases the spatial resolution using deconvolutional layers. Efficient sub-pixel CNN (ESPCNN) [32] employs sub-pixel convolutional layers to effectively upsample the LR feature maps and further reduce the computational complexity. SRResNet [33] introduces residual blocks in the non-linear mapping stage and uses skip connections to pass information between layers. Based on DenseNet [34], SRDenseNet [35] employs dense blocks to learn high-level features, and the outputs of the dense blocks are concatenated to generate the final output. Methods of this type mainly perform computation in the LR space; thus, they involve less computation in each layer compared to methods of the first type. However, it is difficult for these methods to learn multi-scale non-linear LR-HR mapping, as they use only one layer to learn the spatial upsampling process [36]. Hence, they need to retrain the model for each possible scale.

In general, these two types of methods learn the mapping relationship in image space of a single scale. They do not consider that features across different scales can provide various types of information for SR. Although some methods [36,21] gradually upsample the LR input in a Laplacian pyramid structure which generates multi-scale features, they only use the highest level features as final reconstructed outputs and ignore low-level features in SR process.

We propose a dense convolution auto-encoder (DCAE) block to extract multi-spatial scale features and establish multi-level feature reuse mechanism. Based on DCAE block, we have successfully trained the non-linear mapping between LR and HR images through features of various spatial resolutions and temporal ranges.

As shown in Fig. 1, the proposed DCAE block consists of serial convolutional auto-encoder (AE) units and a squeeze unit. A squeeze unit has inter-block skip (IBS) connections, while an auto-encoder unit has both intra-unit skip (IUS) connections and inter-unit dense (IUD) connections. Multi-scale features are extracted from each convolutional AE unit. The IUS and IUD connections aim to construct a short-term temporal reuse mechanism of multi-scale features in one DCAE block. The squeeze unit reuses features from the previous DCAE block with IBS connections, thereby establishing a long-term temporal reuse mechanism between DCAE blocks. At the end of the training process, a global squeeze unit is used to further fuse features of different temporal levels. Our experimental studies confirm that the proposed network can achieve the state-of-the-arts super-resolution performance for different scale factors using a single model.

The contributions of this study can be summarized as follows:

1. We design a novel framework based on a multi-spatial scale and multi-temporal term feature learning. Features are extracted from different temporal ranges and various spatial resolutions to perform non-linear mapping from the LR space to the HR space.
2. We propose three types of connections, which are used as short paths to ensure maximum information flow through the network. First, the IUS connections fuse features in one AE unit in the encoding stage and the corresponding decoding stage. Second, the IUD connections merge features of the same spatial scale and establish short-term temporal reuse. Finally, the IBS connections combine features from different DCAE blocks for long-term temporal reuse.
3. In contrast to previous methods that independently retrain a model for each possible scale, we take advantage of joint training. Our network is based on a multi-scale supervised architecture that establishes a single model for SR of all scales. Thus, the network parameters are reduced.

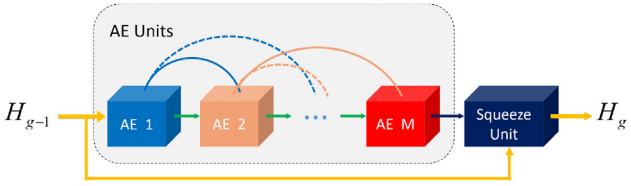

Fig. 1. DCAE block Architecture. One DCAE block contains several AE units and a squeeze unit. H_{g-1} and H_g are the input and output of the g-th DCAE block, respectively.

The remainder of the paper is organized as follows. Section 2 briefly reviews related studies. Section 3 describes the structure of a DCAE block in detail. Section 4 introduces the framework of the proposed network and multi-scale supervised training. Section 5 presents experimental results, including those of an ablation investigation of our model. Our method are compared with state-of-the-art methods. Finally, Section 6 concludes the paper.

2. Related work

2.1. DCNN-based super-resolution

Recently, DCNN-based methods have revolutionized the field of super-solution. Dong et al. [25] first introduced an end-to-end CNN model (i.e., SRCNN) to reconstruct ILR images from their HR counterparts. Their model employed three convolutional layers to equally perform three stages of sparse coding [11]. Subsequent studies further improved the performance by increasing the network depth. Inspired by VGG-net [37], Kim et al. [26] proposed a 20-layer CNN model (i.e., VDSR) with a small filter size and high learning rate. They adopted residual learning and gradient clipping to train the deep network effectively. They also proposed another deep CNN model (i.e., DRCN) [27] with parameter sharing and recursive supervision. Both their methods significantly improve the SR quality. Yang et al. [18] proposed a recurrent residual learning network with edge priors. They improved the edge-preserving capability of the model by jointly using the LR edge maps as input. Dong et al. [31] and Jin et al. [38] built models (FSRCNN and DCSCN, respectively) to extract LR image features and then upsample the features for output using deconvolutional layers or feature reshaping. Shi et al. [39] upsample the LR feature maps using one deconvolutional module in a content-adaptive way, and then fed the upsampled feature maps into two branched subnetworks for image reconstruction. Yang et al. [40] utilized transposed convolution instead of bicubic interpolation to upsample the LR input and adopt deep recurrence learning for a larger receptive field in their SR network. Shi et al. [32] proposed sub-pixel convolutional layers to effectively upsample LR feature maps. They reduced the computational complexity and achieved real-time speed for 1080p video SR. Lai et al. [36] proposed a deep network based on a Laplacian pyramid structure. They gradually upsampled the LR input in a feed-forward structure.

2.2. Skip connections

As CNN models become increasingly deep [30,37,41], the vanishing gradients problem emerges. Many researchers have created short paths between layers to address this problem [30,42–44]. He et al. [30] passed a signal from one layer to the next layer via residual skip connections, and effectively trained networks with more than 100 layers. Ledig et al. [33] proposed SRResNet using residual

blocks and skip connections for 4× scale image super-resolution. Tai et al. [28] proposed a recursive residual model and employed both local and global skip connections to accelerate the training process. In the DenseNet model, Huang et al. [34] proposed a more efficient method to pass information between layers. They established dense connections to connect each layer to every other layer. Thus, they enhanced feature propagation and feature reuse. Tong et al. [35] employed similar dense skip connections to combine features at both low and high levels for reconstructing rich information in image space. Tai et al. [29] established dense connections globally. They densely connected outputs of each memory block, which contains several residual blocks and a gate unit. All the aforementioned SR models improve performance by adopting skip connections to combine features from different temporal terms.

2.3. Multi-scale methods in other computer vision tasks

In addition to using features from different temporal levels via skip connections, extracting features of multi-spatial resolutions can be useful in numerous computer vision tasks, such as image semantic segmentation and image restoration. Ronneberger et al. [45] proposed a fully convolutional network called U-net for biomedical image segmentation. U-net includes a contracting path for downsampling the features and a symmetric expansive path for upsampling the features. Thus, it produces features of different spatial resolutions for segmentation. Badrinarayanan et al. [46] proposed a convolutional encoder-decoder model (SegNet) for image segmentation. Features at multiple resolutions containing both local and global contexts were extracted for localizing the class boundaries. Mao et al. [47] built a deep convolutional auto-encoder composed of multiple convolutional and deconvolutional layers. The convolutional layers extracted the image edges and eliminated corruptions, whereas the deconvolutional layers upsampled the features and reconstructed the image. Symmetric skip connections were used to link features of corresponding spatial resolutions in the encoding and decoding stages. Nah et al. [48] designed a multi-scale convolutional neural network for reconstructing blurred images. Their model took a blurry image pyramid as input and reconstructed sharp images in various spatial resolutions. The final output was recovered from the lowest resolution to the original resolution in a coarse-to-fine manner. Ren et al. [49] propose a multi-scale deep neural network to learn effective features from hazy images for the estimation of scene transmission map. The scene transmission map is first estimated by a coarse-scale network and then refined by a fine-scale network. All the aforementioned models have benefited from multi-scale design, mainly because they can use features of various spatial resolutions.

Recently, generative adversarial networks (GANs) [50] are proved its effectiveness for low-level image restoration [51–53]. The discriminator along with the adversarial loss makes the generator recover better results in terms of SSIM, so as to better improve the visual quality.

Inspired by these methods, we apply the multi-scale concept to DCNN-based image super-resolution. We propose a dense convolution auto-encoder (DCAE) block as the basic component of the network. The DCAE block can extract features from multi-spatial resolutions. Further, for multi-temporal terms, we use skip connections and dense connections in the DCAE block to build a temporal feature reuse mechanism.

3. Dense convolutional auto-encoder block

We present our DCAE block in this section. As the key component of our network, a DCAE block contains several auto-encoder (AE) units and a squeeze unit, as shown in Fig. 1. $\boldsymbol{H_{g-1}}$ and $\boldsymbol{H_g}$ are respectively the input and output of the g-th DCAE block. Details of the AE units and the squeeze unit are show in Figs. 2 and 3, respectively.

AE units are used to extract features of different spatial resolutions and to model a non-linear mapping function from LR to HR space. Suppose that there are M AE units in each DCAE block. Each AE unit has a pair of encoding and decoding stages, in which a series of downsample and upsample convolution operations are performed. For each downsample convolutional layer, we use 2 as the stride size to reduce the feature size by half. Similarly, we set the stride size as 2 in each upsample deconvolutional layer to increase the feature size. This enables us to extract features of different spatial resolutions. Let $\boldsymbol{s}\epsilon\{0,1,\ldots,S\}$ denote the index of different spatial resolutions from small to large size. Let the original input has a spatial resolution of $\boldsymbol{p}\times\boldsymbol{p}$; the spatial resolution of the s-th scale would be $\frac{\boldsymbol{p}}{2^{S-s}}\times\frac{\boldsymbol{p}}{2^{S-s}}$. Let $\boldsymbol{m}\epsilon\{1,2,\ldots,M\}$ indicate the m-th AE unit. Then, $\boldsymbol{C^g_{m,s}}$ denotes the output of each encoding convolutional layer in the m-th AE unit of the g-th DCAE block, with the s-th spatial scale. Similarly, $\boldsymbol{D^g_{m,s}}$ denotes the output of each decoding convolutional layer.

Intra-unit skip (IUS, shown by black arrows in Fig. 2) connections are used within an AE unit to concatenate features of the same spatial scale from paired downsample convolution and upsample convolutional layers. IUS connections allow the network to propagate information in the encoding and decoding stages. The output features after an IUS connection in the s-th scale can be represented as

$$IUS^g_{m,s}=[C^g_{m,s},D^g_{m,s}],s=1,2,\ldots,S-1, \tag{1}$$

where $[\ ]$ denotes the concatenation operation. Note that IUS connections are not performed in the smallest spatial scale $s=0$ and the largest spatial scale $s=S$. Inter-unit dense (IUD, shown by blue arrows in Fig. 2) connections are introduced to pass information of all preceding AE units to the current AE unit in one DCAE block. The output features after each IUS connection are densely connected before they are fed to the upsample convolutional layer. In Fig. 2, feature maps are indicated by several colors. For clarity, only a part of the IUD connections is shown in the figure, and other IUD connections are indicated by the corresponding color of the feature maps. The output features after an IUD connection of the s-th spatial scale in the m-th AE unit can be represented as

$$IUD^g_{m,s}=\begin{cases}[C^g_{1,s},C^g_{2,s},\ldots,C^g_{m,s}],s=0,\\ [C^g_{1,s},D^g_{1,s},D^g_{2,s},\ldots,D^g_{m,s}],s=S,\\ [IUS^g_{1,s},IUS^g_{2,s},\ldots,IUS^g_{m,s}],s=others,\end{cases} \tag{2}$$

Suppose that each convolutional layer produces $\boldsymbol{c_0}$ (also known as the growth rate) feature maps. Then, $\boldsymbol{IUD^g_{m,s}}$ would include $\boldsymbol{m}\times\boldsymbol{c_0}$ feature maps (if $\boldsymbol{s}=0$), $(\boldsymbol{m}+\boldsymbol{1})\times\boldsymbol{c_0}$ feature maps (if $\boldsymbol{s}=S$) or $\boldsymbol{2m}\times\boldsymbol{c_0}$ feature maps (if $\boldsymbol{s}=1,2,\ldots,S-1$).

The objective of IUS connections is to combine information of different spatial resolutions from the encoding and decoding stages in one AE unit. Further, the objective of IUD connections is to fuse features of different AE units within one DCAE block. These two types of skip connections allow an AE unit to gain direct access to outputs of all the layers of the previous AE units in the same block, thereby establishing a short-term temporal reuse mechanism. After IUS and IUD connections are established, the output feature map of each upsample layer in the decoding stage can be represented as

$$D^g_{m,s}=F_{dec,s}(IUD^g_{m,s-1}),\ s=1,\ldots,S, \tag{3}$$

where $\boldsymbol{F_{dec,s}}$ denotes the s-th upsample convolutional layer in the decoding stage, which is performed by stride deconvolution [47].

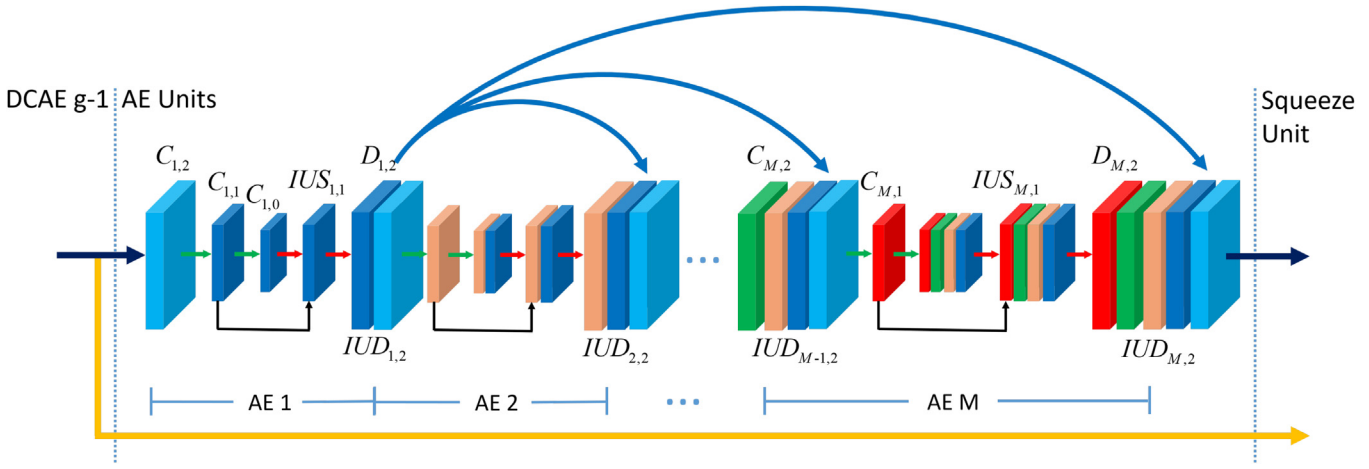

Fig. 2. Illustration of dense connected auto-encoder block architecture when S is set as $\{0, 1, 2\}$. Here, the green arrows indicate the downsample convolution, the red arrows indicate the upsample convolution, the black arrows indicate the intra-unit skip connections (IUS), the blue arrows indicate the inter-unit dense connections (IUD), and the yellow arrow indicates the inter-block skip connection (IBS). Further, $C_{m,s}$ denotes the feature map after the s-th downsample convolution in the m-th AE unit, $D_{m,s}$ denotes the feature map after the s-th upsample convolution in the m-th AE unit, and $IUS_{m,s}$ and $IUD_{m,s}$ denote the concatenated feature after IUS connections and IUD connections of the s-th spatial scale in the m-th AE unit.

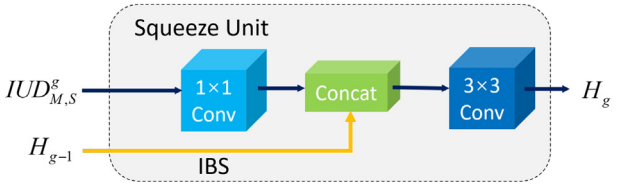

Fig. 3. Proposed Squeeze Unit. The yellow arrow indicates the inter-block skip (IBS) connection.

In the encoding stage of one AE unit, the output feature map of each downsample layer can be represented as

$$C^{g}_{m,S} = F_{enc,S}(C^{g}_{m,s+1}),\ s = S-1,\ldots,0, \tag{4}$$

where $\boldsymbol{F_{enc,S}}$ denotes the s-th downsample convolutional layer in the encoding stage, which is established by stride convolution [47]. As shown in Fig. 2, the input of the first AE unit in one DCAE block is actually the output of the previous DCAE block, while the input of other AE units is the output of the previous AE unit within the block. $\boldsymbol{C^{g}_{m,S}}$ denotes the input of the m-th AE unit. It is represented as

$$C^{g}_{m,S} = \begin{cases} H_{g-1}, & if \quad m = 1, \\ IUD^{g}_{m-1,S}, & if \quad m = 2,3,\ldots,M. \end{cases} \tag{5}$$

Squeeze unit is introduced to reduce the feature map dimension and combine long-term features. The details of the proposed squeeze unit are shown in Fig. 3. The squeeze unit contains a convolutional layer with filter size $\mathbf{1}\times\mathbf{1}$, a convolutional layer with filter size $\mathbf{3}\times\mathbf{3}$, and an inter-block skip (IBS) concatenation, leading to a long-term temporal reuse mechanism. $\boldsymbol{IUD^{g}_{M,S}}$ is the output of the M-th AE unit in the g-th DACE block, which contains $\boldsymbol{M}\times\boldsymbol{c_0}$ feature maps. $\boldsymbol{H_{g-1}}$ is the output of the (g-1)-th DCAE block, which contains $\mathbf{1}\times\boldsymbol{c_0}$ feature maps. Let $\boldsymbol{IUD^{g}_{M,S}}$ and $\boldsymbol{H_{g-1}}$ be the inputs of the squeeze unit. $\boldsymbol{IUD^{g}_{M,S}}$ is convoluted by the $\mathbf{1}\times\mathbf{1}$ convolutional layer to reduce the dimension from $\boldsymbol{M}\times\boldsymbol{c_0}$ to $\mathbf{1}\times\boldsymbol{c_0}$. The output of the g-th DCAE block, $\boldsymbol{H_g}$, can be formulated as

$$H_g = conv(3\times3, c_0) * [conv(1\times1, c_0) * IUD^{g}_{M,S}, H_{g-1}], \tag{6}$$

where $\boldsymbol{conv}(\mathbf{3}\times\mathbf{3}, \boldsymbol{c_0})$ and $\boldsymbol{conv}(\mathbf{1}\times\mathbf{1}, \boldsymbol{c_0})$ denote the convolution filters, and $*$ denotes the convolutional operation with stride 1.

4. SR network based on DCAE blocks

4.1. Network structure

The proposed network, as shown in Fig. 4, consists of three stages: feature initialization stage, multi-scale mapping stage, and reconstruction stage. Our network uses a filter size of 3×3 for all the convolutional layers except the dimension reduction layers in the squeeze unit, whose filter size is 1×1. All the convolutional layers are followed by a rectified linear unit (ReLU) except the final output layer.

1) The feature initialization stage consists of two convolutional layers. The first convolutional layer initially extracts features from the input and the second convolutional layer reduces the dimension of the features. The LR inputs are interpolated to HR and then fed to the network. Here, let I_{ILR} and I_{SR} denote the interpolated LR image and the output SR image, respectively. Let $\boldsymbol{conv}(\boldsymbol{s}\times\boldsymbol{s}, \boldsymbol{c})$ denote the convolution filter, where $\boldsymbol{s}\times\boldsymbol{s}$ indicates the filter size and $\boldsymbol{c}$ indicates the number of filters. The feature extracted from the convolutional layers can be expressed as

$$H_{-1} = conv(3\times3, c_{-1}) * I_{ILR}, \tag{7}$$

$$H_0 = conv(3\times3, c_0) * H_{-1}, \tag{8}$$

where $*$ denotes the convolutional operation with stride 1, $\boldsymbol{c_{-1}}$ and $\boldsymbol{c_0}$ indicates the number of filters of the two convolution layers, respectively. $\boldsymbol{H_{-1}}$ denotes the initial level feature extracted at the first convolutional layer, and $\boldsymbol{H_0}$ indicates the feature after the second convolutional layer.

2) The multi-scale mapping stage includes several DCAE blocks that have identical structures. $\boldsymbol{H_{g-1}}$ and $\boldsymbol{H_g}$ are the input and output of g-th DCAE block respectively. The output feature of g-th DCAE block can be expressed as

$$\begin{aligned} H_g &= F_{DCAE,g}(H_{g-1}) \\ &= F_{DCAE,g}(F_{DCAE,g-1}(\cdots F_{DCAE,1}(H_0)\cdots)), \end{aligned} \tag{9}$$

where $\boldsymbol{F_{DACE,g}}$ denotes the mapping of g-th DCAE block, which is a composite function of operations, including downsample and upsample convolution, skip connections, and ReLUs [55]. The detailed structure of the DCAE block has been described in Section 3.

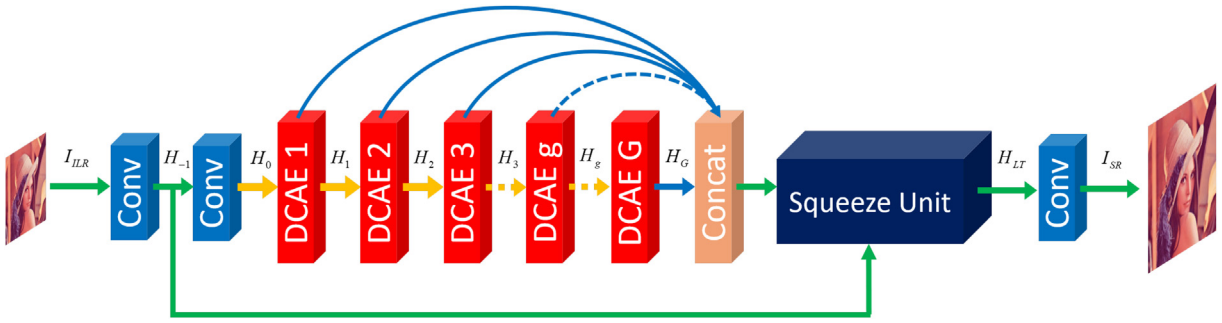

Fig. 4. Architecture of the proposed network.

3) The reconstruction stage aims to combine the features generated by the DCAE blocks and then reduce the dimension of the features for the final output. First, the output features are concatenated and then fed to the squeeze unit with the initial level feature $\boldsymbol{H_{-1}}$. The squeeze unit performs dimension reduction for all the outputs of the DCAE blocks, $\boldsymbol{H_g}$ $(g = 1, 2, \ldots, G)$, and combines the initial feature $\boldsymbol{H_{-1}}$ to establish long-term temporal reuse as follows

$$H_{LT} = F_{SQE}([H_{-1}, H_0, H_1, \ldots, H_D]), \tag{10}$$

where $\boldsymbol{F_{SQE}}$ denotes global squeeze operation, which is performed by several convolutional layers using $\mathbf{1} \times \mathbf{1}$ filters and $\mathbf{3} \times \mathbf{3}$ filters. The squeeze units are also used in each DCAE block. The details of the squeeze unit have been described in Section 3.

After fusing the features of different spatial resolutions and temporal terms, we use a single convolutional layer for image reconstruction. We only perform super-resolution in the luminance channel. The other two channels are then interpolated for the final reconstruction. Therefore, the number of convolution filters is set as one. The final output of our network can be represented as

$$I_{SR} = conv(3 \times 3, 1) * H_{LT}. \tag{11}$$

Given a training set images $\{\boldsymbol{I_{ILR}}, \boldsymbol{I_{GT}}\}$, where $\boldsymbol{I_{GT}}$ is the ground-truth image, we define the content loss function of our model as below:

$$L_c = \|\boldsymbol{I_{GT}} - \boldsymbol{I_{SR}}\|_2. \tag{12}$$

4.2. Adversarial loss

We follow the architecture introduced in [52], and build a discriminator to take the final output of our network or the ground-truth SR image as input. The adversarial loss is defined as follows:

$$\begin{aligned} L_{adv} = & \mathop{\mathbb{E}}_{\boldsymbol{J} \sim p_{\mathrm{HR}}(\boldsymbol{J})} [\log D(\boldsymbol{J})] \\ & + \mathop{\mathbb{E}}_{\boldsymbol{I} \sim p_{\mathrm{LR}}(\boldsymbol{I})} [\log (1 - D(G(\boldsymbol{I})))] \end{aligned} \tag{13}$$

where G is our SR network described in Fig. 4, and D is the discriminator. Finally, by combining the content loss and adversarial loss, our final loss function is

$$L_{total} = L_c + \alpha L_{adv} \tag{14}$$

The hyperparameter $\alpha = 0.002$ was set to trade off the two losses. The combination of the two loss is capable of making the network better improve the visual quality.

4.3. Multi-scale supervised training

Most existing SR algorithms are trained for a single scale factor and are thus supposed to work only with the specified scale. They treat super-resolution for different scale factors as independent problems. To achieve multi-scale SR, they need to construct individual single-scale SR systems for each scale of interest.

As the network becomes deeper, it is possible for us to train only one multi-scale model for all scale factors [26], considering the mutual relationships among the different enlargement scales in SR. Under this approach, the parameters are shared across networks for all predefined scale factors.

To train such a multi-scale model, training datasets for several specified scales are combined into a single total dataset. To prepare multi-scale data, we downscale the original image to LR with several specific scales and then reconstruct these LR images with the original size by upscaling. The reconstructed images are cropped into small patches that are used as training inputs. In the training process, we randomly shuffle the entire dataset and construct mini-batches, where sub-images from different scales can be in the same batch.

5. Experimental results

In this section, we first describe the experimental settings, including the datasets and training details. Second, we study the effect of each component in the proposed DCAE block. Third, the effects of the parameter settings of the DCAE blocks are analyzed. Fourth, we compare the proposed multi-scale supervised training with the traditional single-scale supervised training. It is proved that the proposed multi-scale supervised training effectively reduces the number of network parameters. Finally, we compare our model with state-of-the-art methods in both objective and subjective aspects.

5.1. Settings

Datesets. We used DIV2K [56], a publicly available benchmark dataset, for training. DIV2K is a high-quality (2 K resolution) dataset for image restoration tasks, which includes 800 images for training, 100 images for validation, and 100 images for testing. We used 800 images from DIV2K without data augmentation as a training set in this study. Following [25–27], we evaluated the proposed model on four popular benchmark datasets, namely Set5 [57], Set14 [54], BSD100 [58] and Urban100 [20], with scale factors of $2\times, 3\times$, and $4\times$. We evaluated the SR results in terms of the peak signal-to-noise ratio (PSNR) and structural similarity image measurement (SSIM) [59] on the Y channel (luminance) in the YCbCr image space.

Training. The LR images were obtained by downsampling the HR images using a bicubic kernel. Then, the ILR images were obtained by upsampling the LR images using the same kernel. Only the luminance channel of non-overlapped patches was fed to the network. The other two chrominance channels were directly

Table 1
Ablation study of IUD, IUS, and IBS connections in a DCAE block. We recorded the best performance (PSNR) in 300 epochs on Set14 [54] and Urban100 [20] for a scale factor of 2×.

Models	DCAE-baseline	DCAE-IUD-s2	DCAE-IUD-s0s1s2	DCAE-IUD-IUS	DCAE-IUD-IUS-IBS
IUD-s2	×	✓	✓	✓	✓
IUD-s0s1	×	×	✓	✓	✓
IUS	×	×	×	✓	✓
IBS	×	×	×	×	✓
Set14	32.50	32.97	33.15	33.26	**33.39**
Urban100	29.70	30.59	30.99	31.27	**31.52**

transformed from the ILR images for displaying the results. The size of image patches used in training is set as 48×48. We used TensorFlow [60] to implement our networks and employed Adam [61] as the optimizer with momentum of 0.9 and weight decay of 10^{-4}. The learning rate was set as 10^{-4} for all the layers. We trained the network for 300 epochs, comprising a total of 3×10^5 iterations. Training a basic DCAE network roughly required one day on a single GTX 1080 GPU.

5.2. Ablation study of DCAE block

Here, we analyze the effects of each component of our DCAE block. We gradually added IUD, IUS, and IBS connections in the DCAE block structure. The concatenation unit, global squeeze unit, and 1×1 layers in DCAE blocks were not removed, because they are required for training the network. According to the added components, five networks, namely DCAE-baseline, DCAE-IUD-s2, DCAE-IUD-s0s1s2, DCAE-IUD-IUS, and DCAE-IUD-IUS-IBS, were constructed, as shown in Table 1. These networks have the same number of DCAE blocks ($G = 6$), number of AE units ($M = 6$) per DCAE block, and growth rate ($c0 = 32$). The DCAE block without IUD, IUS, or IBS connections in the baseline model (denoted as DCAE-baseline) was combined with several convolutional auto-encoders and one 1×1 convolutional layer. DCAE-baseline showed the worst performance among the five networks, but its results were still comparable with those of SRCNN [25] (Set14 [54], 32.45 dB; Urban100 [20], 29.50 dB). We then gradually added IUD, IUS, and IBS connections to the baseline model and denoted these models as DCAE-IUD, DCAE-IUD-IUS, and DCAE-IUD-IUS-IBS, respectively, as shown in Table 1. To further prove the effectiveness of the multi-spatial scale, we added IUD connections in the largest spatial scale (denoted as DCAE-IUD-s2) and downsampled the spatial scale (DCAE-IUD-s0s1s2) step by step. Each com-

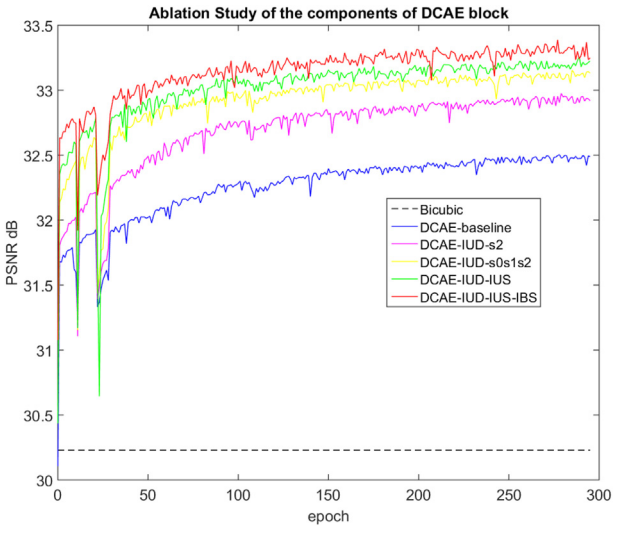

Fig. 5. Convergence curves of network models with and without IUD, IUS, and IBS. The curves for each model are based on the PSNR in 300 epochs of Set14 [54], with a scale factor of 2×. Bicubic results are used as a reference.

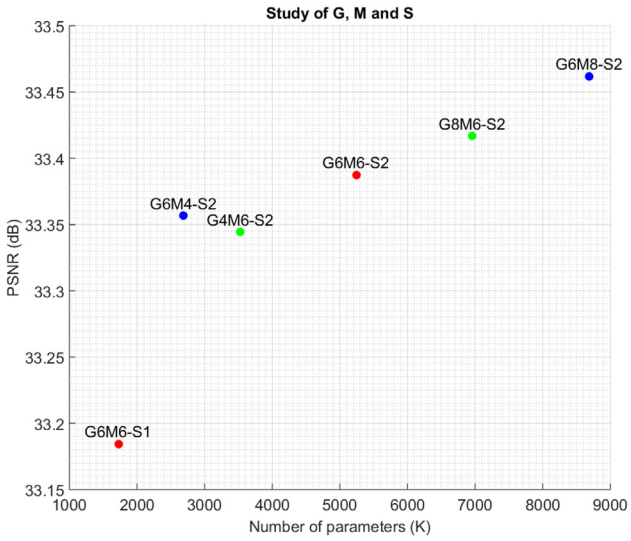

Fig. 6. The depth analysis of our network with different G, M, and S settings on Set14 [54] for 2× enlargement.

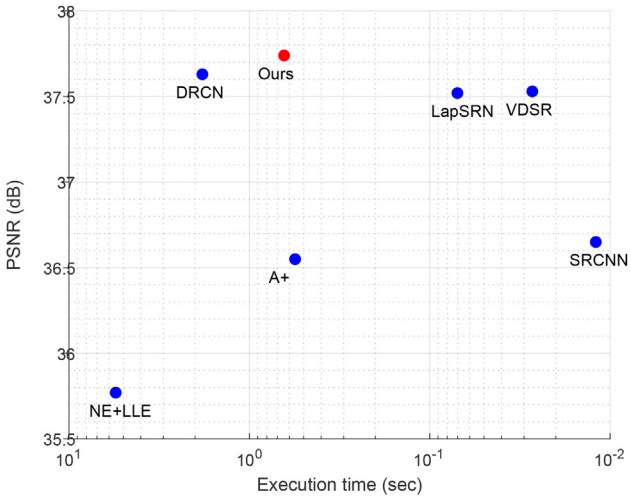

Fig. 7. Performance and execution time on Set5.

Table 2
Quantitative comparison (PSNR) of the multi-supervised training model and the single-supervised training model on BSD100 [58]. The red entries indicate the best performance and the blue entries indicate the second-best performance.

Test/Train	Bicubic	2×	3×	4×	2×, 3×, 4×
2×	29.56	32.11	27.90	25.20	32.05
3×	27.21	27.31	28.91	27.40	28.94
4×	25.96	26.01	26.18	27.41	27.44

ponent added in the DCAE structure can obviously improve the reconstruction performance. We achieved the best performance when using all the components in a DCAE block. This is because these components utilize information in multi-spatial resolutions, encoding/decoding stages, and different temporal terms, which are all able to contribute to the gradient flow.

To demonstrate the convergence of these five models, we plot the PSNR curves in Fig. 5. Bicubic results are used as a reference. All the five models have a stable training process without obvious performance degradation. We can observe from Fig. 5 that the three components, namely IUD, IUS, and IBS, can not only accelerate the convergence but also significantly improve the model performance.

5.3. Depth analysis of our network

In this section, we investigate the depth of our network. The depth of the network is related to three basic parameters: (1) the number of DCAE blocks, denoted as G; (2) the number of AE units contained in each DCAE block, denoted as M; and (3) the index of spatial resolutions in each AE units, denoted as S. We set G6M6-S2 with $G = 6, M = 6, S2 = \{0, 1, 2\}$ as the baseline model. By changing

Table 3
Qualitative results of state-of-the-art methods: average PSNR/SSIM for scale factors of 2×, 3×, and 4×.

Scale	Methods	Set5		Set14		BSD100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2×	Bicubic	33.68	0.9304	30.24	0.8691	29.56	0.8440	26.88	0.8410
	A+ [14]	36.54	0.9544	32.40	0.9056	31.22	0.8863	29.20	0.8938
	SRCNN [25]	36.65	0.9536	32.45	0.9067	31.36	0.8879	29.52	0.8965
	VDSR [26]	37.53	0.9587	33.05	0.9127	31.90	0.8960	30.77	0.9141
	DRCN [27]	37.63	0.9588	33.06	0.9121	31.85	0.8942	30.76	0.9133
	LapSRN [36]	37.52	0.9591	32.99	0.9124	31.80	0.8949	30.41	0.9101
	DRRN [28]	37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188
	MemNet [29]	37.78	0.9597	33.28	0.9142	32.08	0.8984	31.31	09195
	DRFN [40]	37.71	0.9595	33.29	0.9142	32.02	0.8979	31.08	0.9123
	EDSR [62]	38.11	0.9601	33.92	0.9195	32.32	0.9013	32.93	0.9351
	RDN [63]	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353
	RCAN [64]	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384
	MFFRnet [65]	37.68	0.9590	32.99	0.9134	32.03	0.8975	30.89	0.9156
	SCRSR [66]	37.78	0.9593	33.17	0.9133	32.06	0.8998	31.08	0.9194
	HCN [67]	37.62	0.9594	33.03	0.9127	31.91	0.8965	–	–
	MZSR [68]	36.64	0.9498	–	–	31.25	0.8818	29.83	0.8965
	IKC [69]	36.62	0.9658	32.82	0.8999	31.36	0.9097	30.36	0.8949
	Ours-Single	37.82	0.9603	33.35	0.9156	32.10	0.8992	31.53	0.9216
	Ours-Multi	37.74	0.9594	33.29	0.9144	32.05	0.8977	31.44	0.9207
3×	Bicubic	30.40	0.8686	27.54	0.7741	27.21	0.7389	24.46	0.7349
	VDSR [26]	33.66	0.9213	29.78	0.8318	28.83	0.7976	27.14	0.8279
	DRCN [27]	33.82	0.9226	29.77	0.8314	28.80	0.7963	27.15	0.8277
	LapSRN [36]	33.82	0.9227	29.79	0.8320	28.82	0.7973	27.07	0.8271
	DRRN [28]	34.03	0.9244	29.96	0.8349	28.95	0.8004	27.53	0.8377
	MemNet [29]	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376
	DRFN [40]	34.01	0.9234	30.06	0.8366	28.93	0.8010	27.43	0.8359
	EDSR [62]	34.65	0.9282	30.52	0.8462	29.25	0.8093	28.80	0.8653
	RDN [63]	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653
	RCAN [64]	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702
	MFFRnet [65]	33.91	0.9226	29.60	0.8327	28.87	0.7984	27.16	0.8288
	SCRSR [66]	33.95	0.9233	29.93	0.8334	28.89	0.8017	27.26	0.8367
	HCN [67]	33.77	0.9230	29.79	0.8318	28.84	0.7985	-	-
	IKC [69]	32.16	0.9420	29.46	0.8229	28.56	0.8493	25.94	0.8165
	Ours-Single	33.91	0.9240	29.95	0.8348	28.98	0.8016	27.59	0.8386
	Ours-Multi	34.12	0.9251	30.02	0.8353	29.00	0.8018	27.63	0.8394
4×	Bicubic	28.42	0.8109	26.10	0.7023	25.96	0.6678	23.15	0.6574
	DRCN [27]	31.53	0.8854	28.04	0.7673	27.24	0.7233	25.14	0.7511
	LapSRN [36]	31.54	0.8866	28.19	0.7694	27.32	0.7264	25.21	0.7553
	DRRN [28]	31.68	0.8888	28.21	0.7720	27.38	0.7284	25.44	0.7638
	MemNet [29]	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630
	DRFN [40]	31.55	0.8861	28.30	0.7737	27.39	0.7293	25.45	0.7629
	EDSR [62]	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033
	RDN [63]	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028
	RCAN [64]	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087
	MFFRnet [65]	31.39	0.8843	27.85	0.7683	27.33	0.7260	25.20	0.7541
	SCRSR [66]	31.56	0.8844	28.18	0.7672	27.30	0.7233	25.22	0.7620
	HCN [67]	31.39	0.8849	28.04	0.7674	27.29	0.7260	-	-
	IKC [69]	31.52	0.9278	28.26	0.7688	27.29	0.8014	25.33	0.7760
	TPSR-NOGAN [70]	31.10	0.8779	27.95	0.7663	27.15	0.7214	24.97	0.7456
	Ours-Single	31.72	0.8884	28.27	0.7733	27.40	0.7288	25.55	0.7660
	Ours-Multi	31.79	0.8895	28.31	0.7741	27.42	0.7291	25.66	0.7695

the three basic parameters setting, we get five other models, namely G4M6-S2, G8M6-S2, G6M4-S2, G6M8-S2 and G6M6-S1. We show the test results on Set14 [54] of different models in Fig. 6.

A comparison of the test results on Set14 [54] of G4M6-S2, G6M6-S2, and G8M6-S2 shown in Fig. 6 indicates that the network achieves higher performance as G increases. It is also shown in Fig. 6 that a larger M leads to a better result, and G6M8-S2 shows the best performance among the six models. Thus, the performance improves as the network becomes deeper. Although our DCAE block based networks may suffer from some performance degradation with smaller G, M or S, it consistently outperforms SRCNN [25] (32.45 dB).

We also changed the network depth by setting different spatial resolution indices (S) in each AE unit. The baseline model G6M6-S2 having AE units with spatial resolution index $S2 = \{0, 1, 2\}$ and model G6M6-S1 having AE units with spatial resolution index $S1 = \{0, 1\}$. The G6M6-S2 model can provide features of $\frac{\boldsymbol{p}}{2^i} \times \frac{\boldsymbol{p}}{2^i}, \boldsymbol{i} = \mathbf{0}, \mathbf{1}, \mathbf{2}$, whereas G6M6-S1 can provide features of $\frac{\boldsymbol{p}}{2^i} \times \frac{\boldsymbol{p}}{2^i}, \boldsymbol{i} = \mathbf{0}, \mathbf{1}$; here, $\boldsymbol{p}$ is the spatial size of the input image patch. A comparison of the test results of G6M6-S1 and G6M6-S2 in Fig. 6 shows that our G6M6-S2 model outperforms the G6M6-S1 model. This is mainly because our G6M6-S2 model can provide features of more spatial resolutions, which contributes considerable global and local context information to the SR process.

We also illustrate comparisons about execution time and performance in Fig. 7. The proposed method takes only 0.64 s to process an image, which is superior to a lot of comparative state-of-the-art methods. Noticeably, our method achieves a high performance with a relatively low execution time, having a better trade-off between processing time and performance.

5.4. Multi-scale supervised training vs. single-scale supervised training

We conducted experiments to compare the proposed multi-scale supervised training model with the traditional single-scale supervised training model. To train a multi-scale model, training datasets for several specified scales were combined into one total dataset. By contrast, the single-scale supervised models were trained with datasets of one specific scale. Each model was tested on all predefined scale factors. When building the models, we set $G = 8, M = 6, S = \{0, 1, 2\}$, and $c0 = 32$. The test results for the BSD100 [58] dataset are summarized in Table 2.

As shown in Table 2, the models trained on one scale achieved good performance when tested on the dataset of the same scale; however, when tested on other scales, the performance suffered obvious degradation. For example, in the 2× test, models trained with scale factors of 3× and 4× yielded worse results than bicubic interpolation. The SR model trained by a one-scale dataset is not

Fig. 8. Qualitative comparison of our methods with other methods on Set14 [54] for a scale factor of 2×. Our methods recover sharp edges of letters such as "W" in the image.

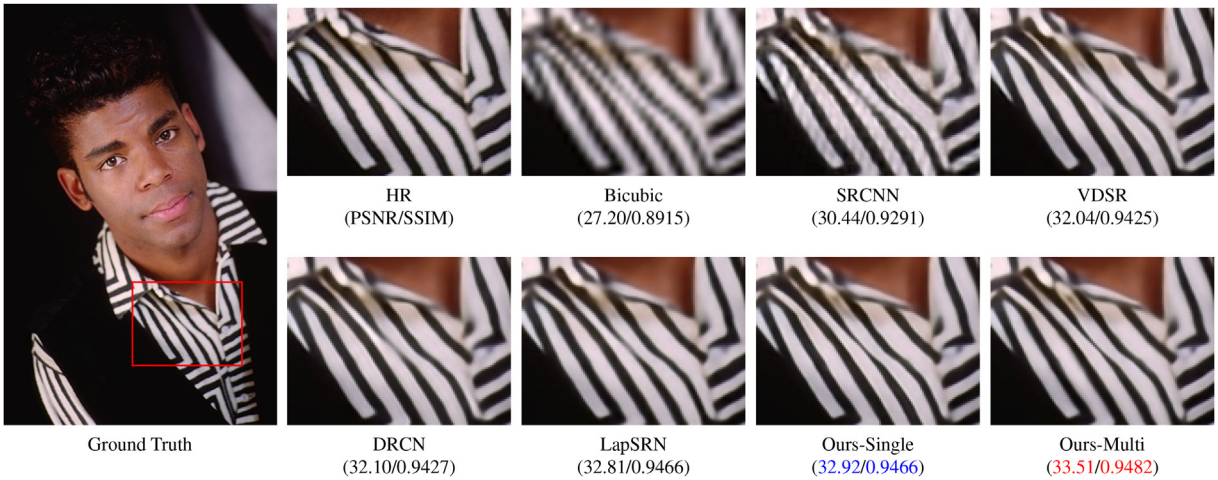

Fig. 9. Qualitative comparison of our methods with other methods on BSD100 [58] for a scale factor of 4×. Our methods recover the clearest stripe pattern in the image.

capable of effectively reconstructing an image of other scales [26], whereas the model trained by the multi-scale dataset shows comparable performance ($2\times$, 32.05 dB vs 32.08 dB) or better performance ($3\times$, 28.94 dB vs. 28.91 dB, $4\times$ 27.44 dB vs. 27.41 dB) compared with the models trained on one scale. Thus, we observe that training on multiple scales boosts the performance for large scales.

5.5. Comparison with the-state-of-the-arts

We conducted quantitative and qualitative comparisons through experiments on four datasets, namely Set5 [57], Set14 [54], BSD100 [58] and Urban 100 [20]. In the experiments, we set $G = 10, M = 6, S = \{0, 1, 2\}$, and $c0 = 32$. Considering both the training time and the storage complexity, we set the batch size as 16. Eight state-of-the-art methods, namely VDSR [26], MemNet [29], EDSR [62], RDN [63], RCAN [64], MFFRnet [65], SCRSR [66] and HCN [67] were compared in the quantitative evaluation. For fair comparison, we conducted all the methods only on the luminance channel for all scale factors. Following [14,31,62], several pixels near the image boundary were cropped before calculating PSNR and SSIM. The quantitative results on the four benchmark datasets for three scale factors ($2\times, 3\times$, and $4\times$) are summarized in Table 3.

As indicated in Table 3, our model without adversarial loss significantly outperformed all the state-of-the-art methods on the four datasets for all the upscaling factors in terms of PSNR and SSIM. In particular, on the Urban100 [20] dataset, our single-scale supervised model outperformed LapSRN [36] by a PSNR gain of 1.12 dB for the $2\times$ scale. On the BSD100 [58] dataset, our single-scale supervised model achieved a PSNR gain of only 0.30 dB compared with LapSRN [36]. Similar cases are shown in $3\times$ and $4\times$ tests. These results indicate that our network performs better on structured images with similar geometric patterns across various spatial resolutions, such as urban scenes. This performance improvement is mainly because our DCAE block can fully utilize features of different spatial sizes.

It should be noted that in the $2\times$ scale, our single-scale supervised model outperformed our multi-scale supervised model, while in the $3\times$ and $4\times$ scales, our multi-scale supervised model achieved better performance than our single-scale supervised model. Similar experimental results have been presented in last subsection and in the case of VDSR [26]. This indicates that large-scale ($3\times, 4\times$) SR networks can benefit from multi-scale supervised training.

We performed visual comparisons with several state-of-the-art methods, as shown in Figs. 8–11. It can be seen that our method accurately recovers the edges and structure of the image content. As shown in Figs. 8 and 9, both our models reconstruct clearer letters and strip patterns compared to the other methods. As shown in Figs. 10 and 11, i.e., reconstruction results of Urban100 [20] in $3\times$ and $4\times$ scale factors, only our models are able to recover the

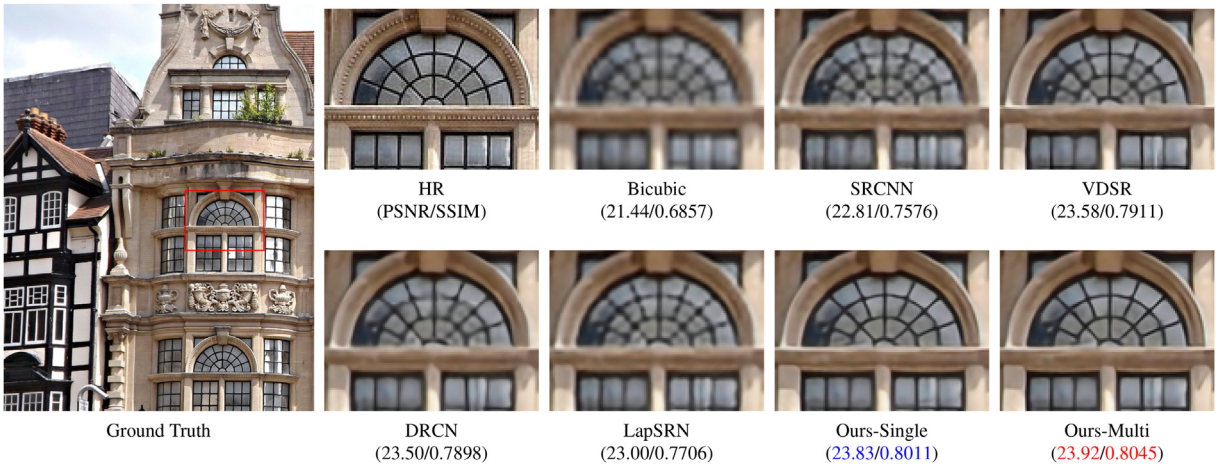

Fig. 10. Qualitative comparison of our methods with other methods on Urban100 [20] for a scale factor of $3\times$. Only our methods recover the clear pattern of the window.

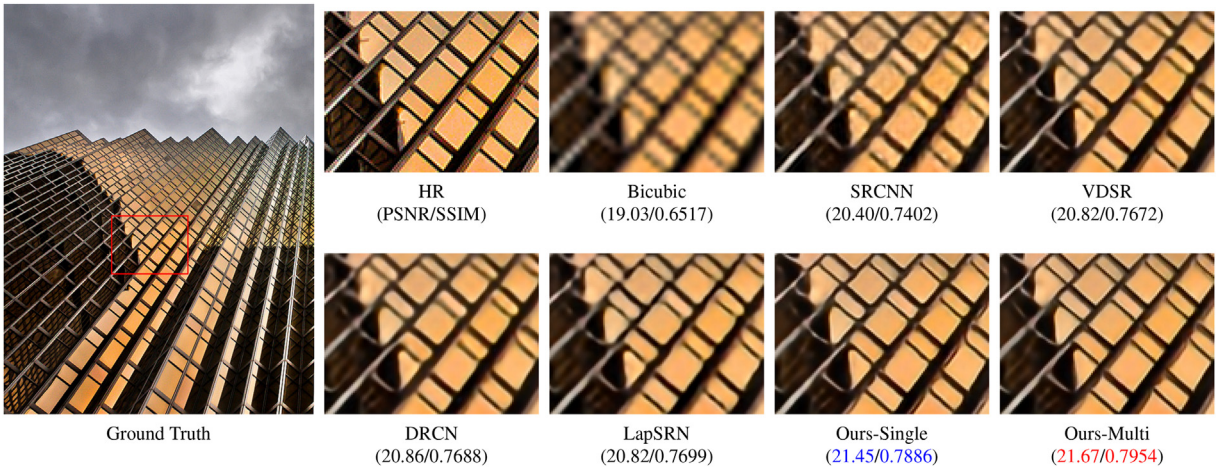

Fig. 11. Qualitative comparison of our methods with other methods on Urban100 [20] for a scale factor of $4\times$. Only our methods recover the parallel line structure clearly.

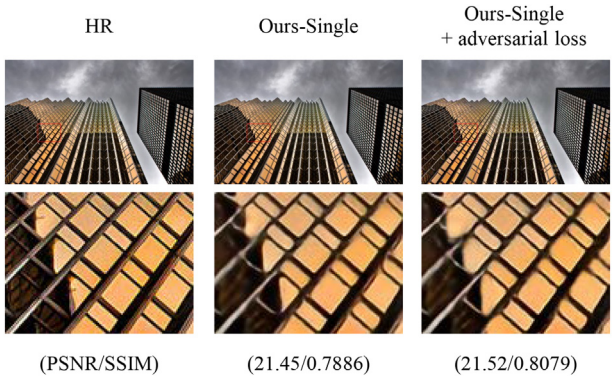

Fig. 12. Qualitative comparison of our methods with and without the adversarial loss on Urban100 for a scale factor of 4×.

building structure clearly, whereas the results of other methods have noticeable distortions. As shown in Fig. 10, our models accurately and clearly reconstruct the window grids whereas the other methods generate blurry results. In Fig. 11, our models recover the parallel line structure clearly, whereas the other results have obvious distortions. Thus, our model outperforms all the other methods both quantitatively and qualitatively. Furthermore, as have mentioned in Section 4.2, the discriminator along with the adversarial loss drives the generator to recover images with higher visual quality. As shown in Fig. 12, if we add a discriminator to our model with an additional adversarial loss, we can recover the parallel line structure more clearly and get better results.

6. Conclusion

We proposed a deep convolutional network for image super-resolution based on a dense convolutional auto-encoder block (DCAE). The AE units in each DCAE block extract features of various spatial resolutions through encoding and decoding convolutional layers. In addition, three types of skip connections were incorporated into each DCAE block for short-term and long-term temporal feature reuse. Specifically, IUS connections reuse features of paired encoding and decoding stages in one AE unit, IUD connections concatenate features across successive AE units in one DCAE block, and IBS connections combine features of the current DCAE block and previous DCAE block. Furthermore, a multi-scale supervised method was used for network training, and it effectively reduced the network parameters compared with single-scale supervised training. Benchmark evaluations showed that our method outperforms state-of-the-art methods especially on structured images.

CRediT authorship contribution statement

Yuan Zhou: Conceptualization, Software, Visualization, Investigation. **Yeda Zhang:** Data curation, Methodology, Writing - original draft, Supervision. **Xukai Xie:** Software, Validation. **Sun-Yuan Kung:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Z. Wang, J. Chen, S.C.H. Hoi, Deep learning for image super-resolution: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).
[2] W.W. Zou, P.C. Yuen, Very low resolution face recognition problem, IEEE Transactions on Image Processing 21 (2012) 327–340.
[3] S. Huo, Y. Zhou, J. Lei, N. Ling, C. Hou, Iterative feedback control-based salient object segmentation, IEEE Transactions on Multimedia 20 (2018) 1350–1364.
[4] Y. Zhou, A. Mao, S. Huo, J. Lei, S. Kung, Salient object detection via fuzzy theory and object-level enhancement, IEEE Transactions on Multimedia 21 (2019) 74–85.
[5] L. Zhang, X. Wu, An edge-guided image interpolation algorithm via directional filtering and data fusion, IEEE Transactions on Image Processing 15 (2006) 2226–2238.
[6] Y. Tai, S. Liu, M. S. Brown, S. Lin, Super resolution using edge prior and single image detail synthesis, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2400–2407.
[7] J. Sun, Z. Xu, H.-Y. Shum, Image super-resolution using gradient profile prior, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[8] H. Zhang, J. Yang, Y. Zhang, T.S. Huang, Non-local kernel regression for image and video restoration, in: European Conference on Computer Vision, Springer, 2010, pp. 566–579.
[9] Z. Zhu, F. Guo, H. Yu, C. Chen, Fast single image super-resolution via self-example learning and sparse representation, IEEE Transactions on Multimedia 16 (2014) 2178–2190.
[10] L.-W. Kang, C.-C. Hsu, B. Zhuang, C.-W. Lin, C.-H. Yeh, Learning-based joint super-resolution and deblocking for a highly compressed image, IEEE Transactions on Multimedia 17 (2015) 921–934.
[11] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Transactions on Image Processing 19 (2010) 2861–2873.
[12] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 1, IEEE, 2004, pp. I–I.
[13] R. Timofte, V. De Smet, L. Van Gool, Anchored neighborhood regression for fast example-based super-resolution, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1920–1927.
[14] R. Timofte, V. De Smet, L. Van Gool, A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conference on Computer Vision, Springer, 2014, pp. 111–126.
[15] Y. Zhang, Y. Zhang, J. Zhang, Q. Dai, Ccr: Clustering and collaborative representation for fast single image super-resolution, IEEE Transactions on Multimedia 18 (2016) 405–417.
[16] N. Kumar, A. Sethi, Fast learning-based single image super-resolution, IEEE Transactions on Multimedia 18 (2016) 1504–1515.
[17] R. Timofte, R. Rothe, L. Van Gool, Seven ways to improve example-based single image super resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1865–1873.
[18] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, S. Yan, Deep edge guided recurrent residual learning for image super-resolution, IEEE Transactions on Image Processing 26 (2017) 5895–5907.
[19] W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration, IEEE Transactions on Image Processing 22 (2013) 1620–1630.
[20] J.-B. Huang, A. Singh, 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.
[21] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 370–378.
[22] J. Liu, W. Yang, X. Zhang, Z. Guo, Retrieval compensated group structured sparsity for image super-resolution, IEEE Transactions on Multimedia 19 (2017) 302–316.
[23] M. Haris, G. Shakhnarovich, 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.
[24] K. Zeng, J. Yu, R. Wang, C. Li, D. Tao, Coupled deep auto-encoder for single image super-resolution, IEEE Transactions on Cybernetics 47 (2017) 27–37.
[25] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2016) 295–307.
[26] J. Kim, J. Kwon Lee, 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.
[27] J. Kim, J. Kwon Lee, K. Mu Lee, Deeply-recursive convolutional network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1637–1645.
[28] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2017, p. 5.

[29] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4539–4547.
[30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[31] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: European Conference on Computer Vision, Springer, 2016, pp. 391–407.
[32] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, 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.
[33] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4681–4690.
[34] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[35] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4799–4807.
[36] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep laplacian pyramid networks for fast and accurate super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 624–632.
[37] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[38] J. Yamanaka, S. Kuwashima, T. Kurita, Fast and accurate image super resolution by deep cnn with skip connection and network in network, in: Neural Information Processing, Springer, 2017, pp. 217–225.
[39] Y. Shi, K. Wang, C. Chen, L. Xu, L. Lin, Structure-preserving image super-resolution via contextualized multi-task learning, IEEE Transactions on Multimedia 19 (2017) 2804–2815.
[40] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, X. Wei, Drfn: Deep recurrent fusion network for single-image super-resolution with large factors, IEEE Transactions on Multimedia 21 (2018) 328–337.
[41] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.
[42] G. Huang, Y. Sun, Z. Liu, D. Sedra, K.Q. Weinberger, Deep networks with stochastic depth, in: European Conference on Computer Vision, Springer, 2016, pp. 646–661.
[43] G. Larsson, M. Maire, G. Shakhnarovich, Fractalnet: Ultra-deep neural networks without residuals, arXiv preprint arXiv:1605.07648 (2016).
[44] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in: Advances in neural information processing systems, 2015, pp. 2377–2385.
[45] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[46] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 2481–2495.
[47] X.-J. Mao, C. Shen, Y.-B. Yang, Image restoration using convolutional auto-encoders with symmetric skip connections, arXiv preprint arXiv:1606.08921 (2016).
[48] S. Nah, T. Hyun Kim, K. Mu Lee, Deep multi-scale convolutional neural network for dynamic scene deblurring, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3883–3891.
[49] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, M.-H. Yang, Single image dehazing via multi-scale convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 154–169.
[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[51] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, H. Li, Adversarial spatio-temporal learning for video deblurring, IEEE Transactions on Image Processing 28 (2018) 291–301.
[52] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, M.-H. Yang, Gated fusion network for single image dehazing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3253–3261.
[53] M. Cheon, J.-H. Kim, J.-H. Choi, J.-S. Lee, Generative adversarial network-based image super-resolution using perceptual content losses, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 0–0.
[54] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: International Conference on Curves and Surfaces, Springer, 2010, pp. 711–730.
[55] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.
[56] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, et al., Ntire 2017 challenge on single image super-resolution: Methods and results, in: Computer Vision and Pattern Recognition Workshops, IEEE, 2017, pp. 1110–1121.
[57] M. Bevilacqua, A. Roumy, C. Guillemot, M.-L. A. Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: British Machine Vision Conference, 2012.
[58] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 2, IEEE, 2001, pp. 416–423.
[59] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (2004) 600–612.
[60] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: large-scale machine learning on heterogeneous systems. software available from tensorflow. org. 2015, https://www. tensorflow. org (2015).
[61] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[62] B. Lim, S. Son, H. Kim, S. Nah, 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.
[63] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, 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.
[64] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 286–301.
[65] X. Jin, Q. Xiong, C. Xiong, Z. Li, Z. Gao, Single image super-resolution with multi-level feature fusion recursive network, Neurocomputing 370 (2019) 166–173.
[66] D. Lin, G. Xu, W. Xu, Y. Wang, X. Sun, K. Fu, Scrsr: An efficient recursive convolutional neural network for fast and accurate image super-resolution, Neurocomputing 398 (2020) 399–407.
[67] B. Liu, D. Ait-Boudaoud, Effective image super resolution via hierarchical convolutional neural network, Neurocomputing 374 (2020) 109–116.
[68] J.W. Soh, S. Cho, N.I. Cho, Meta-transfer learning for zero-shot super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 3516–3525.
[69] J. Gu, H. Lu, W. Zuo, C. Dong, Blind super-resolution with iterative kernel correction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1604–1613.
[70] R. Lee, Ł. Dudziak, M. Abdelfattah, S. I. Venieris, H. Kim, H. Wen, N. D. Lane, Journey towards tiny perceptual super-resolution, arXiv preprint arXiv:2007.04356 (2020).

Yuan Zhou received the B.Eng., M.Eng., and Ph.D. degrees from Tianjin University, Tianjin, China, in 2006, 2008, and 2011, respectively, all in electronic engineering and communication engineering. Since 2011, she has been a Faculty Member with the School of Electronic Information Engineering, Tianjin University, Tianjin, China, where she is currently an Associate Professor. From 2013 to 2014, she was a Visiting Scholar with the School of Mechanical and Electrical Engineering, University of Southern Queensland, Toowoomba, QLD, Australia. From 2016 to 2017, she was a visiting scholar with the Department of Electrical Engineering, Princeton University, USA. Her current research interests include computer vision and image/video communications.

Yeda Zhang received the B.Eng. degree from Hebei University of Technology, Tianjin, China, in 2016. He received the M.Eng. degree from Tianjin University, Tianjin, China, in 2019. His research interest is in image super resolution.

Xukai Xie received the B.Eng. degree from Hebei University of Technology, Tianjin, China, in 2018. He is currently a graduate student majoring in information and communication engineering from Tianjin University, Tianjin, China. His research interest is in image super resolution and neural architecture search.

Sun-Yuan Kung is currently a Professor with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. He has authored or co-authored over 500 technical publications and numerous textbooks including the VLSI Array Processors (Prentice-Hall, 1988), the Digital Neural Networks (Prentice-Hall, 1993), the Principal Component Neural Networks Wiley, 1996, the Biometric Authentication: A Machine Learning Approach (Prentice-Hall, 2004), and the Kernel Methods and Machine Learning (Cambridge University Press, 2014). His current research interests include machine learning, data mining and privacy, statistical estimation, system identification, wireless communication, VLSI array processors, signal processing, and multimedia information processing.