



FSFN: feature separation and fusion network for single image super-resolution

Kai Zhu¹ · Zhenxue Chen^{1,2}  · Q. M. Jonathan Wu³ · Nannan Wang² · Jie Zhao^{1,4} · Gan Zhang¹

Received: 2 September 2020 / Revised: 5 January 2021 / Accepted: 3 June 2021 /
Published online: 16 July 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In recent years, image super-resolution (SR) based on deep learning technology has made significant progress. However, most methods are difficult to apply in real life because of their large parameters and heavy computation. Recently, residual learning has been widely applied to the problem of super-resolution. It can make the shallow features extracted from the input image act on each middle layer through long and short connection. Therefore, residual learning can be focused on processing high-frequency feature information, which significantly improves the SR performance of the network. However, with the improvement of network depth, the features that can be effectively utilized are still the shallow ones extracted from the input image. In this paper, we propose the feature separation and fusion network (FSFN). We further enrich the high-frequency feature information by separating and fusing the extracted and unextracted features in the internal shallow layer of each feature separation and fusion module. As the depth of the network increases, the shallow features extracted from the input image can be updated in a direction closer to those extracted from the real high-resolution image. A large number of experimental results show that this method has a strong performance compared with the existing SR algorithm with similar parameters and computation.

Keywords Super-resolution · Feature separation · Feature fusion · Residual learning · Deep learning · CNN

1 Introduction

Single image super-resolution (SISR) aims to reconstruct a low-resolution image into a high-resolution image, which is a low-level visual recovery task in computer vision [5, 13]. Since the reconstruction of a high-resolution image from a low-resolution image is not just a

Kai Zhu and Zhenxue Chen have contributed equally.

✉ Zhenxue Chen
chenzhenxue@sdu.edu.cn

Extended author information available on the last page of the article.

one-to-one mapping, single image super-resolution (SISR) is a seriously ill-posed problem. Various SISR methods have been proposed and achieved amazing results, among which the most notable ones are some methods based on deep learning [2, 7, 8, 20, 33, 38, 40].

With the development of a convolutional neural network, single image super-resolution has been granted significant attention by researchers, who have achieved state-of-the-art performance on various benchmarks of SR. Dong et al. [7] first applied the convolutional neural network to the super-resolution task. Then, they proposed the FSRCNN [8], which can directly learn the low-resolution inputs and then up-sample the features at the end of the network. In order to improve the performance of the model, Kim et al. proposed the VDSR [15] by increasing the depth of the network model. Inspired by the image classification problem [9], a lot of work has been done to apply residual learning to SR problems. As a result, many models with good performance have relatively large parameters. In order to reduce the parameters of the model, the recursive network model is applied to the SR problem. The recursive model decomposes the complex SR problem into a series of simple and easily solved problems by sharing parameters. Many researchers have taken a recursive network as their basic network architecture, such as DRCN [16], DRRN [28], and MemNet [29]. All of these models guarantee better performance with fewer parameters (Fig. 1).

With the improvement of computing performance, many super-resolution networks [20, 33, 38, 40] have large network parameters and large computing overhead. It is difficult to apply this process in real life. Currently, there are many ways to design lightweight SR networks. Recursive networks can achieve better SR performance with fewer model parameters, but they require huge computational overhead. The CARN network [2] adopts the local and global cascading modules, which make full use of the feature information at all levels but can't avoid information redundancy. Network structure search [6] can give full play to the performance of each module, although the model based on network structure search does not improve the performance of SR very much, as it is limited in terms of search space and search strategy. After learning that the residual network improves the performance of SR [20], we noticed that the residual block promoted the performance of SR by integrating the extracted features with the original features that are not extracted. Because the original features can reach all hierarchical structures of the network through short or long

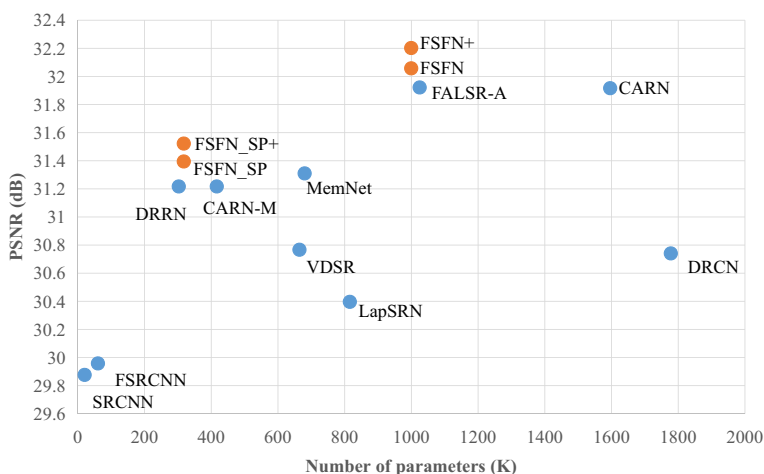


Fig. 1 Trade-off between performance and number of parameters on the Urban100×2 dataset. The orange circle represents the method we proposed, and the blue circle represents the other methods

connections, the incremental features extracted from the input image account for a large part of the original features that can be used repeatedly. This also limits the network’s SR performance. In order to make better use of the feature information, we adopt the feature separation technology to separate the extracted features and the unextracted features and then merge them to continuously update the original features information at different levels. Therefore, we propose a feature separation and fusion module, which reduces feature information redundancy. Our network is of great significance in the design of lightweight network models. We can learn from VDSR [15] that increasing the network depth can improve the SR performance of the network, so we adopt the idea of partitioning. As shown in Fig. 2, several feature separation and fusion modules and a convolutional layer constitute our feature residual learning module, and several feature residual learning modules and a convolutional layer constitute a global residual learning module. So our network can integrate and learn features at different levels to increase the richness of the features.

In this work, we propose a feature separation and fusion network for SISR. Compared to networks with similar model parameters and multi-adds, our network showed better SR performance. The contributions of this paper can be summarized as follows:

- We propose a feature separation and fusion module, which separates the original features into extracted features and unextracted features, and then extracts the unextracted features in the next step to increase the diversity of features. Finally, we use a 1x1 convolution to adaptively select fusion features. This makes the fusion feature more representative, which means the image reconstruction quality can be greatly improved. The experimental results show that the proposed feature separation and fusion module improves the performance of SR.
- We adopt the idea of partitioning, which increases the depth of the network and enables us to better integrate and learn the characteristics of other modules. This approach further improves the SR performance of our network.
- An extensive experimental evaluation of several publicly available datasets shows that the proposed FSFN model performs better than most existing methods.

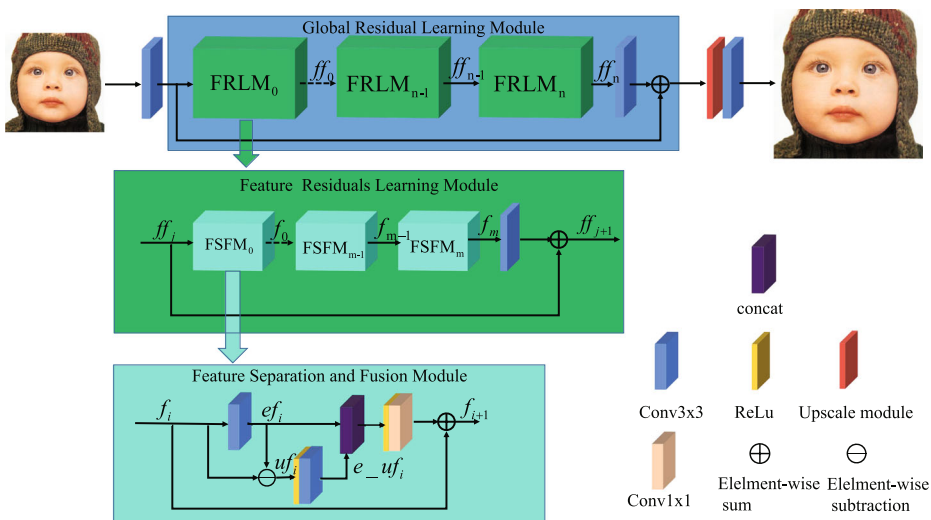


Fig. 2 Network architecture of our feature separation and fusion network (FSFN)

2 Related works

In the early stage, in order to solve the problem of image super-resolution, we mostly adopted interpolation technology based on sampling theory [3, 19, 42]. With the development of deep learning technology, the problem of image super-resolution is now mostly solved. Therefore, our main research focus is on the application of deep learning technology in image super-resolution.

2.1 Single image super-resolution

By using a variety of deep learning-based techniques, we have been able to find a solution to the SR problem experienced in the extensive literature on this topic [12, 14, 18, 21, 25, 26, 33–35, 41]. Dong et al. proposed SRCNN [7], which was the first successful attempt to use convolution to solve the problem of image super-resolution. Then, they came up with FSRCNN [8], which had better SR performance. In contrast to the shallow network architecture, Simonyan et al. proposed VDSR [15], which has a deeper hierarchical structure. Kim et al. proposed a deep recursive convolutional network named DRCN [16], which, by designing a repeatable convolution unit, enables the model to maintain better SR performance with fewer parameters. Inspired by the success of DenseNet [10] in the image classification architecture, Tong et al. proposed the SR-DenseNet [31], which achieved higher flexibility and richer feature representation through the densely connected CNN layer. Zhang et al. proposed the RDN [40] by introducing local and global residual connections. Then they proposed RCAN [38] by introducing a channel attention mechanism in each local residual block, which improved the performance of SR significantly. Ahn et al. proposed CARN [2], which allows the feature information of each residual block to flow between different levels through a large number of short connections. This model performs well in lightweight networks. Liu et al. proposed a residual feature aggregation network (RFANet) [22] consisting of an RFA framework and a powerful ESA block. The RFA framework groups several residual modules together and directly forwards the features on each local residual branch by adding skip connections. This work also effectively improves the SR performance of the network. Due to the uncertainty of image degradation, Zhang et al. [37] proposed an end-to-end trainable unfolding network which leverages both learning-based the methods and model-based methods. And it can handle the classical degradation model via a single model. This work expands the data processing scope of the network beyond the limitation of bicubic interpolation degradation.

2.2 Residual learning

Residual learning is now widely used in various computer vision tasks. It was originally proposed to avoid gradient disappearance and make it possible to design very deep networks. In the case of SR, residual learning mainly deals with the high-frequency information between the input and the ground truth. The processing of high-frequency features information will be an important factor affecting the SR performance. Lim et al. proposed EDSR [20] by modifying the ResNet architecture for image classification. They greatly improved the SR performance of the model by removing the batch normalization layer. In order to further improve SR performance, we will separate and fuse the high-frequency feature information in the residual block to improve the hierarchical and richness of the high-frequency features. Our network also showed better SR performance.

3 Proposed method

In this section, we describe our proposed feature separation and fusion network (FSFN) in detail.

3.1 Network structure

As shown in Fig. 2, our feature separation and fusion network is mainly composed of three parts: the shallow feature extraction module, the global residual learning module, and the up-sampling reconstruction module. Given an input LR image I_{LR} and its corresponding target HR image I_{HR} . super-resolution image I_{SR} can be obtained by

$$I_{SR} = F_{FSFN}(I_{LR}) \tag{1}$$

where $F_{FSFN}(\cdot)$ is our FSFN. Like most previous studies, our shallow feature extraction module only uses a convolution layer, described as

$$sf = F_{SFEM}(I_{LR}) \tag{2}$$

where $F_{SFEM}(\cdot)$ is represented as our shallow feature extraction module and sf represents the shallow features extracted from the input I_{LR} . Shallow features (sf) are used as inputs to the global residual learning module to generate more refined features (rf). As such,

$$rf = F_{GRLM}(sf) \tag{3}$$

where $F_{GRLM}(\cdot)$ denotes our proposed global residual learning module. Finally, rf is used as the input of the up-sampling reconstruction module (URM) to generate the super-resolution image I_{SR} . So

$$I_{SR} = F_{URM}(rf) \tag{4}$$

where $F_{URM}(\cdot)$ denotes our proposed up-sampling reconstruction module, which consists of the upscale module and a convolution layer. As narrative ESPCN [27] institutes, we choose to sub-pixel convolutional as our upscale module. This has been proved to be the most effective option.

Finally, FSFN is optimized using L1 loss function, just like in other networks [2, 20, 41]. Given a training set $\{I_{LR}^i, I_{HR}^i\}_{i=1}^N$ that has N LR-HR pairs, the loss function of our FSFN network can be expressed as follows:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \left\| F_{FSFN}(I_{LR}^i) - I_{HR}^i \right\|_1 \tag{5}$$

where θ represents the updated parameters in the training process of our model and $\|\cdot\|_1$ is the l_1 norm.

3.2 Global residual learning module(GRLM)

In this section, we will describe the core module of the network in detail, which is the global residual learning module referred to as the GRLM (see Fig. 2). The GRLM consists of N feature residual learning modules (FRLM), a convolutional layer, and a long connection. This was inspired by the EDSR [20], which allows the network to centralize the processing of the high frequency parts of the feature. Similar to the RCAN [38], our FRLM module is divided into M feature residual learning modules, a convolutional layer, and a short con-

nection. This enables the shallow feature (sf) extracted by the shallow feature module to be applied at a deeper level. The GRLM module can be described by the following formula:

$$\begin{aligned}
 rf &= F_{conv}(ff_n) + sf \\
 ff_j &= F_{FRLM_j}(ff_{j-1}) \quad (j = 1, 2, \dots, n) \\
 ff_0 &= F_{FRLM_0}(sf)
 \end{aligned}
 \tag{6}$$

In the formula, $F_{conv}(\cdot)$ represents a convolutional layer, rf represents a refined feature, and ff_j represents the output of the i th feature residual learning module(F_{FRLM_j}), which is defined as the fusion feature. The FRLM can be described as follows:

$$\begin{aligned}
 ff_{j+1} &= F_{conv}(f_m) + ff_j \\
 f_i &= F_{FSFM_i}(f_{i-1}) \quad (i = 1, 2, \dots, m) \\
 f_0 &= F_{FSFM_0}(ff_j)
 \end{aligned}
 \tag{7}$$

where f_i represents the input features of the $(i - 1)$ th feature fusion module($F_{FSFM_{i-1}}$) and the output features of the i th feature fusion module(F_{FSFM_i}). We will discuss the FSFM in more detail in the next section.

3.3 Feature separation and fusion module (FSFM)

We can learn from the residual block proposed by EDSR [20] that focusing on the processing of high-frequency features will lead to a strong SR performance. As we can see in Fig. 3, the processing of high-frequency characteristics only uses two convolutional layers. In view of the limited feature extraction capability of a single convolution, we propose a feature separation and fusion module. We will retain the feature information extracted by the first convolution and the feature information not yet extracted. Then, we re-extract the unextracted features. We define this module as a local feature extraction module (LFEM) which is shown in Fig. 4. This module can be described as follows:

$$euf_i = F_{LFEM}(uf_i)
 \tag{8}$$

In the formula, uf_i represents unextracted features and euf_i represents features of extracted uf_i . So, our feature separation module (FSM) can be expressed as

$$\begin{aligned}
 efi, eufi &= F_{FSM}(fi) \\
 &= F_{conv}(fi), F_{LFEM}(fi - F_{conv}(fi))
 \end{aligned}
 \tag{9}$$

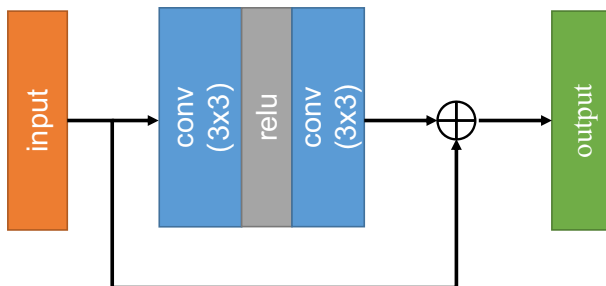


Fig. 3 The structure of the residual blocks

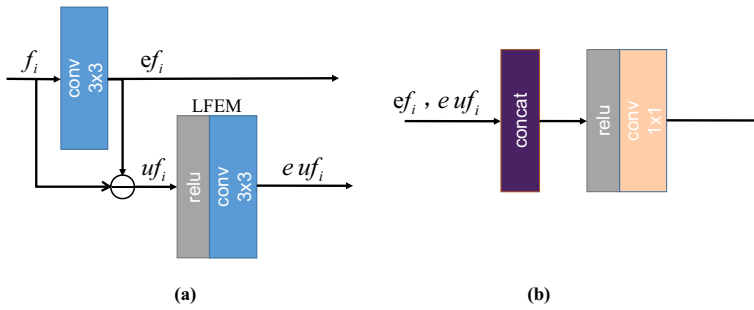


Fig. 4 Decomposition of feature separation and fusion module (FSFM); (a) represents the feature separation module (FSM), while (b) represents the feature fusion module (FFM)

where f_i represents the input feature of the i th FSFM. $e u f_i$ and $e f_i$ will be combined adaptively by the feature fusion module (FFM). Therefore, our FSFM can be expressed as

$$\begin{aligned}
 f_{i+1} &= F_{FSFM}(f_i) \\
 &= F_{FFM}(F_{FSM}(f_i)) + f_i
 \end{aligned}
 \tag{10}$$

where $F_{FSFM}(\cdot)$ denotes our proposed feature separation and fusion module. Compared with the residual block proposed by EDSR [20], our FSFM behaves like a residual block which means that the associated path is disconnected assuming that LFEM is not effective. We can understand that the residual block is a special case of FSFM. Since our FSM module uses only a 1×1 convolutional layer, the increase in network volume can be ignored. As shown in Figs. 5 and 6, the number of indistinctive feature maps accounts for more than half among the unextracted feature maps. This fully shows that our network can effectively reduce feature redundancy.

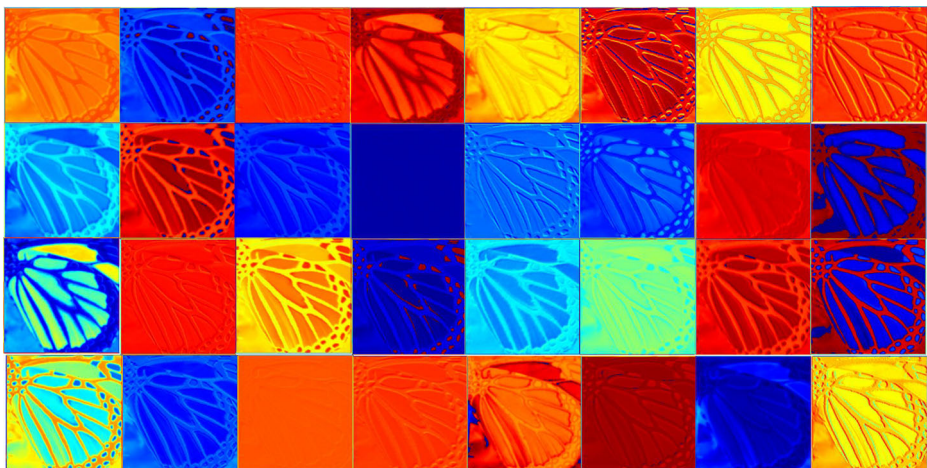


Fig. 5 extracted features in the first feature separation and fusion module

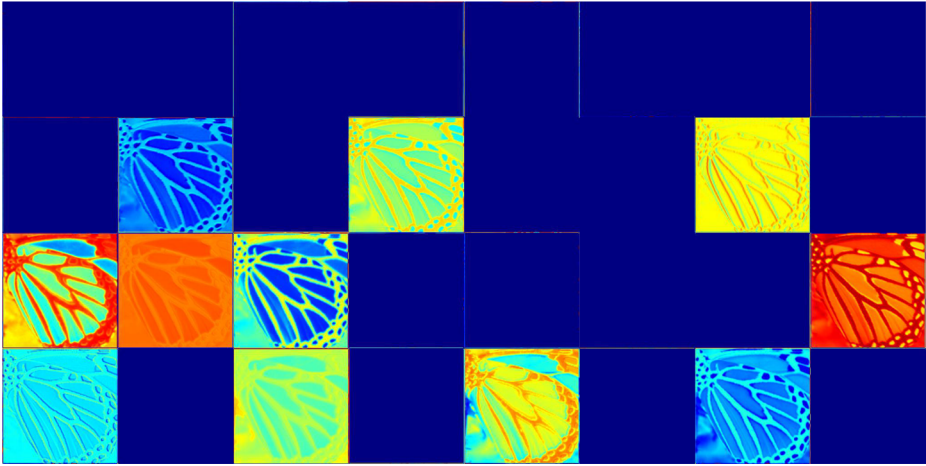


Fig. 6 Unextracted features in the first feature separation and fusion module

4 Experiments

4.1 Datasets and metrics

The DIV2K dataset [1] has been widely used for many image restoration tasks. Like these networks [2, 20, 38], we used DIV2K as our FSFN training dataset which contained 800 RGB images. Five commonly used datasets, Set5 [4], Set14 [36], BSD100 [23], Urban100 [11], and Manga109 [24], were used to evaluate network SR performance. To gauge the SR results, we applied two objective image quality assessment criteria: peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [32]. All criteria were calculated on the Y channel of transformed YCbCr space.

4.2 Implementation details

The HR image patch with a size of 192×192 was randomly cropped from the HR image in the DIV2K dataset as the input of our model. The LR image was obtained from the HR image by using the bicubic interpolation according to the scaling factor ($2 \times, 3 \times, 4 \times$). And the mini-batch size is set to 16. We set filter size $f = 3$. The number of filters for FSFN_SP is set to 32 and for FSFN to 48. We train our FSFN with an ADAM optimizer [17] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. The learning rate is initialized to 10^{-4} and halved at every 2×10^5 minibatch update. We use a PyTorch framework to implement our proposed FSFN network with a Titan Xp GPU.

4.3 Model analysis

In this section, we will delve into the model parameters and calculations, the effectiveness of FSFM, and the superiority of the idea of block processing (BP).

4.3.1 Model parameters and calculations

As we mentioned in Section 3.2, our core module GRLM is composed of N FRLM, and each FRLM is composed of M FSFM which is the smallest indivisible unit of our network. In our proposed FSFN network, we set N to be 3 and M to be 6. In contrast to most networks [2, 20, 39], in our proposed network, the number of feature maps is 48. This greatly reduces the parameters of the model. At the same time, we also proposed a small version of FSFN named FSFN_N2M4. In FSFN_N2M4, we set N to 2 and M to 4. Then, the number of feature maps is 32. In order to further enhance the performance of the small version of FSFN, we proposed FSFN_SP, which changed the number of output feature maps of convolution of LFEF in FSFM to 64. In this way, the SR performance of the model is further improved at the cost of a small parameter increase (Table 1). Table 2 shows the comparison of models under different amounts of FRLM and FSFM. It can be seen that all the models we selected can show better SR performance under the condition of limited number of parameters. In order to enhance the quality of the SR images, we adopt the self-ensemble strategy. This strategy can be summarized as follows. We flip and rotate the input image I_{LR} to generate augmented image $I_{LR}^{n,i} = T_i(I_{LR}^n)$ for each sample, where T_i represents eight geometric transformations, including identity. Then, we'll use augmented images as input to our network to generate super-resolved images $I_{SR}^{n,1}, \dots, I_{SR}^{n,8}$. We then apply inverse transform to those super-resolved images to get the original geometry $\tilde{I}_{SR}^{n,i} = T_i^{-1}(I_{SR}^{n,i})$. Finally, we average the transformed outputs to get the following self-ensemble results. $I_{SR}^n = \frac{1}{8} \sum_{i=1}^8 (\tilde{I}_{SR}^{n,i})$. FSFN_SP+ and FSFN+ are obtained by applying the self-ensemble strategy. As can be seen in Fig. 1, our model performs best against other state-of-the-art algorithms on the parameter scale of 500K and 1000K. We all know that recursive networks greatly increase the number of model parameters by reusing modules, but they also increase the computational complexity of the model. As we can see from Table 1, our FSFN_SP has a significantly reduced computation capacity but better SR performance when compared with DRCN [16], DRRN [28] and MemNet [30].

4.3.2 Block processing (BP) and feature separation and fusion module (FSFM)

As discussed in Section 2.2, residual learning enables our network to focus on high-frequency processing of features. EDSR [20] proposes a network architecture similar to that shown in Fig. 12a, and its smallest modular processing unit is called the residual block (Resblock). We call this structural framework the directly connected structure. The disadvantage of this structure is that, as the network hierarchy deepens, the shallow primitive features that can be utilized effectively will be ignored. To address this, Zhang et al. [39] put forward the Residual in Residual (RIR) module similar to Fig. 12b. We call this structural framework block processing (BP). Through short connection, the M smallest module processing unit is synthesized into a large module, and then N such large modules are synthesized into a larger module. This allows the original shallow features to be utilized effectively at different levels. Therefore, we think that block processing is a better structural framework. As we can see from Table 3, whether the smallest structural unit is Resblock or FSFM, when block processing is adopted, the SR performance of the network is improved. EDSR_S represents the structural framework with a direct connection structure, which contains eight residual blocks, and EDSR_N2M4 represents the structural framework with block processing. We found that EDSR_N2M4 is 0.023dB higher than EDSR_S in PSNR. By comparing EDSR_S and FSFN_S as well as FSFN_N2M4 and EDSR_N2M4, we found that, on the

Table 1 Average PSNR/SSIM for scales 2, 3, and 4 on datasets Set5, Set14, BSD100, Urban100, and Manga109

Method	Scale	Params	MultAdds	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM		
SRCNN [7]	×2	57K	52.7G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946	35.74/0.9661					
FSRCNN [8]		12K	6.0G	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9694					
VDSR [15]		665K	612.6G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9729					
DRCN [16]		1,774K	17,974G	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	37.63/0.9723					
LapSRN [18]		813K	29.9G	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100	37.27/0.9740					
DRRN [28]		297K	6,796.9G	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.92/0.9760					
MemNet [30]		677K	2,662.4G	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	–					
CARN-M [2]		412K	91.2G	37.53/0.9583	33.26/0.9141	31.92/0.8960	31.23/0.9193	–					
FALSR-B [6]		326k	74.7G	37.61/0.9585	33.29/0.9143	31.97/0.8967	31.28/0.9191	–					
FALSR-C [6]		408k	93.7G	37.66/0.9586	33.26/0.9140	31.96/0.8965	31.24/0.9187	–					
FSFN_SP(ours)		313K	61.61G	37.63/0.9611	33.20/0.9176	31.98/0.9007	31.38/0.9216	37.83/0.9758					
FSFN_SP+(ours)		313K	61.61G	37.76/0.9615	33.28/0.9184	32.04/0.9013	31.51/0.9228	38.14/0.9765					
FALSR-A [6]		1,021K	234.7G	37.82/0.9595	33.55/0.9168	32.12/0.8987	31.93/0.9256	–					
CARN [2]		1,592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	–					
FSFN(ours)	1,000K	195.8G	37.85/0.9619	33.43/0.9190	32.15/0.9025	32.05/0.9283	38.35/0.9770						
FSFN+(ours)	1,000K	195.8G	37.95/0.9621	33.55/0.9203	32.20/0.9031	32.20/0.9296	38.59/0.9775						
SRCNN [7]	×3	57K	52.7G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989	30.59/0.9107					
FSRCNN [8]		12K	5.0G	33.16/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080	30.98/0.9212					
VDSR [15]		665K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9310					
DRCN [16]		1,774K	17,974G	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276	32.31/0.9328					
DRRN [28]		297K	6,796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.74/0.9390					

Table 1 (continued)

Method	Scale	Params	MultAdd	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
MemNet [30]		677K	2,662.4G	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	–
CARN-M [2]		412K	46.1G	33.99/0.9236	30.08/0.8367	28.91/0.8000	27.55/0.8385	–
FSFN_SP(ours)		359K	31.7G	34.04/0.9274	29.97/0.8416	28.92/0.8072	27.56/0.8408	32.72/0.9395
FSFN_SP+(ours)		359K	31.7G	34.17/0.9285	30.06/0.8430	28.98/0.8084	27.67/0.8429	33.06/0.9417
CARN [2]		1,592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	–
FSFN(ours)		1,104K	96.5G	34.28/0.9294	30.13/0.8451	29.06/0.8106	28.07/0.8522	33.31/0.9440
FSFN+(ours)		1,104K	96.5G	34.40/0.9304	30.24/0.8467	29.12/0.8117	28.22/0.8546	33.65/0.9458
SRCNN [7]	×4	57K	52.7G	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221	27.66/0.8505
FSRCNN [8]		12K	4.6G	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280	27.90/0.8517
VDSR [15]		665K	612.6G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8809
DRCN [16]		1,774K	17,974G	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.98/0.8816
LapSRN [18]		813K	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560	29.09/0.8845
DRRN [28]		297K	6,796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.46/0.8960
MemNet [30]		677K	2,662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	–
CARN-M [2]		412K	32.5G	31.92/0.8903	28.42/0.7762	27.44/0.7304	25.62/0.7694	–
FSFN_SP(ours)		350K	23.1G	31.65/0.8928	28.22/0.7812	27.41/0.7392	25.54/0.7706	29.65/0.8990
FSFN_SP+(ours)		350K	23.1G	31.77/0.8947	28.31/0.7831	27.47/0.7406	25.64/0.7733	29.97/0.9025
CARN [2]		1,592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	–
FSFN(ours)		1,080K	65.8G	31.93/0.8966	28.41/0.7860	27.55/0.7436	25.99/0.7851	30.30/0.9073
FSFN+(ours)		1,080K	65.8G	32.05/0.8981	28.48/0.7878	27.61/0.7450	26.11/0.7881	30.61/0.9105

The best results was bolded

Table 2 PSNR results of the models with different numbers of FRLM and FSFM on Urban100 at 500 epochs (with scaling factor $\times 4$)

Method	FSFN_N2M3	FSFN_N2M4	FSFN_N4M4	FSFN_N3M6
Params	226K	268K	1010K	1080K
PSNR/dB	25.293	25.411	25.900	25.928

same structural framework, our FSFM improved SR performance more significantly than Resblock. By comparing EDSR_N2M4 with FSFN_S, we find that FSFM improves network performance to a greater degree than block processing and that FSFN_S achieves better performance with fewer parameters. As described in Section 4.3.1, we obtained FSFN_SP through the Extended Local Feature Extraction Module. FSFN_SP performs best on those networks with parameters below 500K. Although the average running time of FSFN_SP on the Urban100 dataset was slightly increased compared to our base network EDSR_S, we chose FSFN_SP as the smaller version of our FSFN network for better SR performance.

4.4 Comparison with the state-of-the-arts

In this section, several state-of-the-art methods will be compared with our proposed FSFN, including SRCNN [7], FSRCNN [8], VDSR [15], DRCN [16], LapSRN [18], DRRN [28], MemNet [30], FALSr [6], and CARN [2]. We will compare our FSFN with the above methods both mathematically and visually. The above methods are divided into two categories according to the number of parameters: those that use about 500K, and those that use about 1000K. As can be seen in Table 1, when the number of parameters of the model is around 500K, the overall SR performance of our proposed FSFN_SP+ network is better than that of all other networks, especially at the scaling factor of $\times 2$. When the number of parameters is around 1000K, our FSFN+ network also performs better than its peer network models. When the comparison criterion is structural similarity (SSIM), the SR performance of our FSFN_SP, FSFN network is superior to all other networks, even without a self-integration strategy.

4.4.1 Visual comparison

Since our proposed FSFN can separate and fuse the shallow features of a single module, we can imagine that the SR images generated by our FSFN network will have better detailed

Table 3 Investigations of FSFM, Block Processing (BP), and Extended Local Feature Extraction Module (ELFEM)

Method	EDSR_S	EDSR_N2M4	FSFN_S	FSFN_N2M4	FSFN_SP
FSFM	×	×	✓	✓	✓
BP	×	✓	×	✓	✓
ELFEM	×	×	×	×	✓
PSNR/dB	31.166	31.189	31.230	31.256	31.382
Times/s	0.0161	0.0165	0.0211	0.0223	0.0279
Paras/K	195.9	214.4	212.6	231.1	313.3

We observe the best PSNR (dB) values on Urban100 ($2\times$). Time is the average inference time per image in Urban100 as measured on a Titan Xp GPU. The best results was bolded



Fig. 7 Visual comparison for 4×SR on the Set14

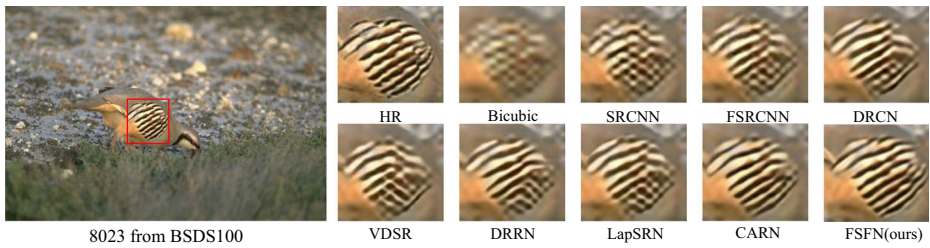


Fig. 8 Visual comparison for 4×SR on the BSDS100

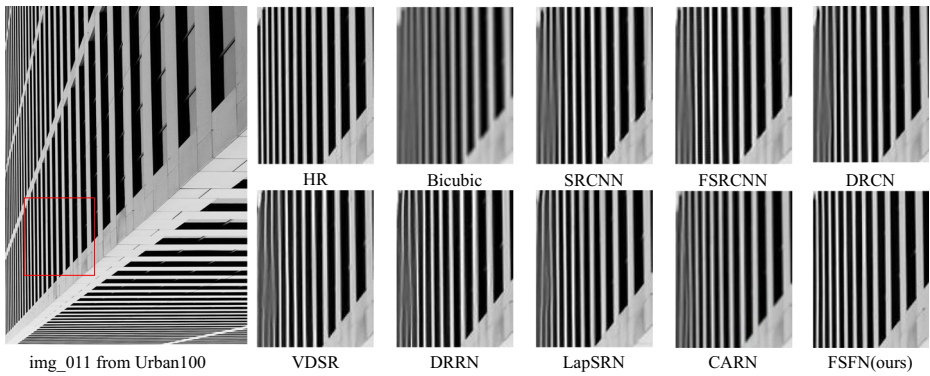


Fig. 9 Visual comparison for 4×SR on the Urban100

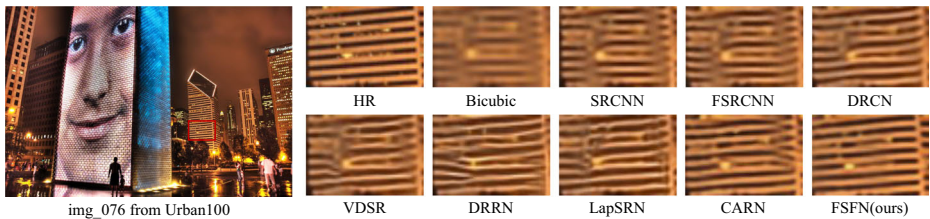


Fig. 10 Visual comparison for 4×SR on the Urban100

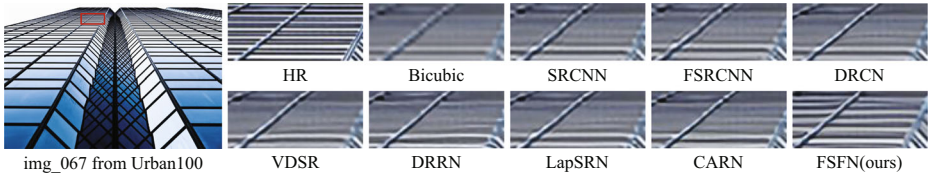
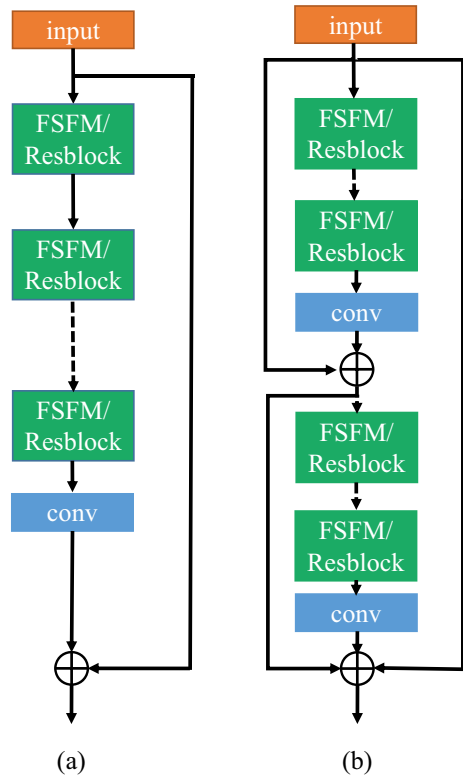


Fig. 11 Visual comparison for $3\times SR$ on the Urban100

image features. Below we will show the image recovery effects of our network on each dataset. We randomly select an image from the Set14 dataset, named PPT3 as shown in Fig. 7. We can see that our network can show the words in the original image more clearly. Figure 8 shows image 8023 from the BSDS100 dataset. We can see from this image that our FSFN network is able to restore the detail of the bird’s wing with a scaling factor of 4. Figure 9 shows another image from the BSDS100 dataset named img_011. The hardest detail to recover from this image is the vertical line at the left of the locally enlarged image block. We can see that, unlike SR images generated by our FSFN network, SR images generated by other methods experience difficulty in showing the gap between vertical lines. Since the BSDS100 dataset contains a large number of high-resolution images, we have introduced a visual comparison of Figs. 10 and 11, and both images show better SR performance in our network when the scaling factor is 4 and 3. All of the above visual comparisons reflect the superior performance of our FSFN network framework (Fig. 12).

Fig. 12 Overall network framework selection; (a) represents a directly connected structure, while (b) represents a block processing structure



5 Conclusion

In this paper, we propose a lightweight feature separation and fusion network for single image super-resolution. We adopt the structural framework of block processing so that our network can enrich the features of different levels while making full use of the features of different levels. We also propose a feature separation and fusion module as our smallest module unit, which enhances the ability of our network to extract high-frequency features by separating and fusing the shallow features extracted from the interior of the smallest module unit. In this way, the SR performance of our network is improved significantly, especially in the restoration of texture details of some images. Experiments show that our network model offers better SR performance compared to other lightweight network models.

Acknowledgment This work was supported in part by the National Natural Science Foundation of China (61876099), in part by the National Key R&D Program of China (2019YFB1311001), in part by the National Natural Science Foundation of China (U1806202), in part by the National Natural Science Foundation of China (61533011), in part by the Scientific and Technological Development Project of Shandong Province (2019GSF111002), in part by the Shenzhen Science and Technology Research and Development Funds (JCYJ20180305164401921), in part by the Foundation of Ministry of Education Key Laboratory of System Control and Information Processing (Scip201801), and in part by the Foundation of State Key Laboratory of Integrated Services Networks (ISN20-06). Kai Zhu and Zhenxue Chen contributed equally to this work and should be considered as the co-first authors.

References

1. Agustsson E, Timofte R (2017) NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: CVPR, pp 1122–1131
2. Ahn N, Kang B, Sohn K (2018) Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV, pp 256–272
3. Allebach JP, Wong PW (1996) Edge-directed interpolation. In: Proceedings 1996 International conference on image processing, Lausanne, Switzerland, September 16–19, 1996, pp 707–710
4. Bevilacqua M, Roumy A, Guillemot C, Alberi-Morel M (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: British machine vision conference, BMVC 2012, Surrey, UK, September 3–7, 2012, pp 1–10
5. Chang H, Yeung D, Xiong Y (2004) Super-resolution through neighbor embedding. In: CVPR, pp 275–282
6. Chu X, Zhang B, Ma H, Xu R, Li J, Li Q (2019) Fast, accurate and lightweight super-resolution with neural architecture search. arXiv:1901.07261
7. Dong C, Loy CC, He K, Tang X (2016) Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. 38(2):295–307
8. Dong C, Loy CC, Tang X (2016) Accelerating the super-resolution convolutional neural network. In: ECCV, pp 391–407
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR, pp 770–778
10. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: CVPR, pp 2261–2269
11. Huang J, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. In: CVPR, pp 5197–5206
12. Hui Z, Gao X, Yang Y, Wang X (2019) Lightweight image super-resolution with information multi-distillation network. In: Proceedings of the 27th ACM International conference on multimedia, MM 2019, Nice, France, October 21–25, 2019, pp 2024–2032
13. Irani M, Peleg S (1991) Improving resolution by image registration. CVGIP Graph Model. Image Process. 53(3):231–239
14. Jin X, Xiong Q, Xiong C, Li Z, Gao Z (2019) Single image super-resolution with multi-level feature fusion recursive network. Neurocomputing 370:166–173

15. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: CVPR, pp 1646–1654
16. Kim J, Lee JK, Lee KM (2016) Deeply-recursive convolutional network for image super-resolution. In: CVPR, pp 1637–1645
17. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings
18. Lai W, Huang J, Ahuja N, Yang M (2019) Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 41(11):2599–2613
19. Li X, Orchard MT (2000) New edge directed interpolation. In: Proceedings of the 2000 international conference on image processing, ICIP 2000, Vancouver, BC, Canada, September 10-13, 2000, pp 311–314
20. Lim B, Son S, Kim H, Nah S, M Lee K (2017) Enhanced deep residual networks for single image super-resolution. In: CVPR, pp 1132–1140
21. Liu B, Ait-Boudaoud D (2020) Effective image super resolution via hierarchical convolutional neural network. *Neurocomputing* 374:109–116
22. Liu J, Zhang W, Tang Y, Tang J, Wu G (2020) Residual feature aggregation network for image super-resolution. In: CVPR, pp 2356–2365
23. Martin DR, Fowlkes CC, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, pp 416–425
24. Matsui Y, Ito K, Aramaki Y, Fujimoto A, Ogawa T, Yamasaki T, Aizawa K (2017) Sketch-based manga retrieval using manga109 dataset. *Multim. Tools Appl.* 76(20):21811–21838
25. Qiu Y, Wang R, Tao D, Cheng J (2019) Embedded block residual network: A recursive restoration model for single-image super-resolution. In: ICCV, pp 4179–4188
26. Ren C, He X, Pu Y, Nguyen TQ (2019) Enhanced non-local total variation model and multi-directional feature prediction prior for single image super resolution. *IEEE Trans. Image Process.* 28(8):3778–3793
27. Shi W, Caballero J, Huszar F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR, pp 1874–1883
28. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: CVPR, pp 2790–2798
29. Tai Y, Yang J, Liu X, Xu C (2017) Memnet: A persistent memory network for image restoration. In: ICCV, pp 4549–4557
30. Tai Y, Yang J, Liu X, Xu C (2017) Memnet: A persistent memory network for image restoration. In: ICCV, pp 4549–4557
31. Tong T, Li G, Liu X, Gao Q (2017) Image super-resolution using dense skip connections. In: ICCV, pp 4809–4817
32. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13(4):600–612
33. Wang X, Gu Y, Gao X, Hui Z (2019) Dual residual attention module network for single image super resolution. *Neurocomputing* 364:269–279
34. Yang X, Mei H, Zhang J, Xu K, Yin B, Zhang Q, Wei X (2019) DRFN: deep recurrent fusion network for single-image super-resolution with large factors. *IEEE Trans. Multimed* 21(2):328–337
35. Yang W, Wang W, Zhang X, Sun S, Liao Q (2019) Lightweight feature fusion network for single image super-resolution. *IEEE Signal Process. Lett.* 26(4):538–542
36. Zeyde R, Elad M, Protter M (2010) On single image scale-up using sparse-representations. In: Curves and surfaces - 7th international conference, Avignon, France, June 24-30, 2010, Revised Selected Papers, pp 711–730
37. Zhang K, Gool LV, Timofte R (2020) Deep unfolding network for image super-resolution. In: CVPR, pp 3214–3223
38. Zhang Y, Li K, Li K, Wang L, Zhong B, Fu Y (2018) Image super-resolution using very deep residual channel attention networks. In: ECCV, pp 294–310
39. Zhang Y, Li K, Li K, Wang L, Zhong B, Fu Y (2018) Image super-resolution using very deep residual channel attention networks. In: ECCV, pp 294–310
40. Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y (2018) Residual dense network for image super-resolution. In: CVPR, pp 2472–2481
41. Zhang Z, Wang X, Jung C (2019) DCSR: dilated convolutions for single image super-resolution. *IEEE Trans. Image Process.* 28(4):1625–1635

42. Zhang L, Wu X (2006) An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Process.* 15(8):2226–2238

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Kai Zhu received the B.S. degree in School of Automation from Huazhong University of Science and Technology, Wuhan, China, in 2018. He is pursuing the M.S. degree in Control Engineering at the School of Control Science and Engineering, Shandong University, Jinan, China. His current research interests include image super-resolution and image enhancement.



Zhenxue Chen was born in Shandong, China, in 1977. He received the B.S. degree in automatic from School of Electrical Engineering and Automation at Shandong Institute of Light Industry, Jinan, China, in 2000, the M.S. degree in computer science from School of Information Science and Engineering at Wuhan University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in pattern recognition and intelligent systems from Institute of Image Recognition and Artificial Intelligence at Huazhong University of Science and Technology, Wuhan, China, in 2007. Since 2007, Dr. Chen has been an associate professor with the School of Control Science and Engineering, Shandong University. From 2012 to 2013, he was a visiting scholar with the Michigan State University, East Lansing, Michigan, USA. Now, his main areas of interest include image processing, pattern recognition, and computer vision, with applications to face recognition. Dr. Chen has published over 100 papers in refereed international leading journals/conferences such as *IEEE T-II*, *IEEE T-CSVT*, *IEEE T-VT*, *IEEE T-ITS*, *Information Sciences*, *Neurocomputing*, *Neural Computing and Applications* and *SP-IC*, etc.



Q. M. Jonathan Wu (M'92-SM'09) received the Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990. He was with the National Research Council of Canada for ten years from 1995, where he became a Senior Research Officer and a Group Leader. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. He has published more than 300 peer-reviewed papers in computer vision, image processing, intelligent systems, robotics, and integrated microsystems. His current research interests include machine learning, 3-D computer vision, video content analysis, interactive multimedia, sensor analysis and fusion, and visual sensor networks. Dr. Wu holds the Tier 1 Canada Research Chair in Automotive Sensors and Information Systems. He was Associate Editor for IEEE Transactions on Systems, Man, and Cybernetics Part A, and the International Journal of Robotics and Automation. Currently, he is an Associate Editor for the IEEE Transaction on Neural Networks and Learning Systems and the journal of Cognitive Computation. He has served on technical program committees and international advisory committees for many prestigious conferences.



Nannan Wang (M'16) received the B.Sc. degree in information and computation science from the Xi'an University of Posts and Telecommunications, in 2009, and the Ph.D. degree in information and telecommunications engineering, in 2015. From 2011 to 2013, he was a visiting Ph.D. student with the University of Technology, Sydney, NSW, Australia. He is currently with the State Key Laboratory of Integrated Services Networks, Xidian University. His current research interests include computer vision, pattern recognition, and machine learning. He has published over 50 papers in refereed journals and proceedings, including the IEEE T-PAMI, IJCV, AAAI, and IJCAI.



Jie Zhao was born in Shandong, China, in 1997. She received the B.S. degree in School of Mathematics and Applied Mathematics from Shandong Normal University, Jian, China, in 2019. She is pursuing the M.S. degree in Control Science and Engineering at the School of Control Science and Engineering, Shandong University, Jinan, China. Her current research interests include machine learning, deep learning and Single Image Super-Resolution.



Gan Zhang received the B.S. degree in School of Automation from Qufu Normal University, Rizhao, China, in 2018. She is pursuing the M.S. degree in Biomedical Engineering at the School of Control Science and Engineering, Shandong University, Jinan, China. Her current research interests include machine learning and data mining of marine algae.

Affiliations

Kai Zhu¹ · Zhenxue Chen^{1,2}  · Q. M. Jonathan Wu³ · Nannan Wang² · Jie Zhao^{1,4} · Gan Zhang¹

Kai Zhu
zhukaiqqa@mail.sdu.edu.cn

Q. M. Jonathan Wu
jwu@uwindsor.ca

Nannan Wang
nnwang@xidian.edu.cn

Jie Zhao
zhaojiecode@gmail.com

Gan Zhang
201834626@mail.sdu.edu.cn

¹ School of Control Science and Engineering, Shandong University, Jinan 250061, China

² State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an 710071, China

³ Department of Electrical and Computer Engineering, University of Windsor, Windsor N9B 3P4, Canada

⁴ Shenzhen Research Institute of Shandong University, Shandong University, Shenzhen 518057, China