

## Visualizing and understanding of learned compressive sensing with residual network

Zhifu Zhao, Xuemei Xie\*, Chenye Wang, Wan Liu, Guangming Shi, Jiang Du

School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi 710071, PR China



### ARTICLE INFO

#### Article history:

Received 1 February 2019

Accepted 17 May 2019

Available online 25 May 2019

Communicated by Steven Hoi

#### Keywords:

Compressive sensing

Visualizing and understanding

Learned measurement matrix

Residual network

### ABSTRACT

Recent years a variety of CNN-based (convolutional neural network) approaches for compressive sensing (CS) have been proposed. They learn a transform to recover the original signals from the measurements obtained by measuring the scene at a sub-Nyquist sampling rate. Among them, the LMM-based ones (*learned measurement matrix*) exhibit better performance. In this paper, we visualize the LMM-based CS framework. This is the first time an insight look is taken into the CS network. It helps us understand how CS framework works. Taking the proposed LMM-based framework as an example, where reasonable residual blocks in the recovery part let it achieve excellent performance over the existing ones, we analyze the mechanism of CNN-based CS by the visualization. In the measurement part, intuitive representation of the measurement matrices is presented. As for the recovery procedure, an explanation of the preliminary recovery is given from the viewpoints of system and space. We analyze how the residual block adds the mainly high-frequency information. Through the comparison of the visualization of the typical methods, it is explored that the measurement and recovery part of the proposed method can promote each other, and the learned CS framework with residual network achieves the best performance. In addition, a set of experiments are conducted on a standard dataset to verify the better performance of our framework.

© 2019 Elsevier B.V. All rights reserved.

### 1. Introduction

Compressive sensing (CS) has been applied in various kinds of signal processing fields such as medical imaging [1,2], cameras [3,4] and radar imaging [5,6], because it can capture and represent compressible signals at a sampling rate significantly below the Nyquist rate [7,8]. In traditional compressive sensing theory [9–14], a set of under-sampled measurements are obtained from the scene, which can be considered as encoding procedure. And a recovery algorithm is used to recover the original signals from the measurements, which is considered as decoding procedure. Mathematically, the measurements are given by  $\mathbf{y} = \Phi\mathbf{x} + \mathbf{e}$ , where  $\mathbf{x} \in \mathbb{R}^N$  is the original signal,  $\mathbf{y} \in \mathbb{R}^M$  is the measurements, and  $\mathbf{e} \in \mathbb{R}^M$  is the measurement noise [15].  $\Phi \in \mathbb{R}^{M \times N}$  is the predefined measurement matrix. It is noted that recovering  $\mathbf{x}$  from  $\mathbf{y}$  when  $M < N$  is an ill-posed inverse problem.

In recent years, with the development of deep learning techniques, some CNN-based approaches for image compressive sensing have been proposed. In general, these methods can be divided

into two categories in terms of the definition of the measurement matrix: PDMM-based (predefined measurement matrix) CS [16–20] and LMM-based CS [21–24]. The methods of the first category only learn the inverse transformation from measurements to original signals using a deep neural network. Differently, the architectures of the second category jointly train the measurement matrix and the corresponding inverse transformation.

Although the existing CNN-based CS methods achieve encouraging performance, there is no clear understanding of how they perform so well. This is because of no visualization of CNN-based framework to our best knowledge. Especially for the LMM-based CS frameworks, there is no intuitive representation in the existing publications as we know. For diverse computer vision tasks, several papers aiming to understand the internal mechanism of convolutional neural network have been published. These papers span a wide spectrum and may be itemized as follows. In [25–29], the authors attempt to understand classified convolutional network by giving insight into the function of intermediate feature layers with different visualization techniques. Except for these, scholars try to find out the mechanism of other applications where CNN is used as the baseline, by visualization, such as face expressions [30], pattern classification [31], object detection [32] and action recognition [33]. While for compressive sensing

\* Corresponding author.

E-mail address: [xmxie@mail.xidian.edu.cn](mailto:xmxie@mail.xidian.edu.cn) (X. Xie).

field, the visualization and understanding of how the CNN-based CS methods work have lagged behind.

In this paper, we visualize the LMM-based CS framework. This is the first time an insight look is taken into the CS network. It helps us understand and design CS frameworks. In our work, we take a proposed framework as an example of visualization. This framework consists of fully connected layers for measurement and preliminary recovery, a residual network for enhanced recovery. Note that the measurement and recovery parts are jointly trained. Our visualization is mainly in the following three aspects. First, in the measurement part, intuitive representation of the measurement matrices is presented. Second, as for the recovery procedure, an explanation of the preliminary recovery is given from the viewpoints of system and space. Third, we analyze how the residual block adds the mainly high-frequency information. To verify the performance of the proposed network, we use peak signal to noise ratio (PSNR) and structural similarity index (SSIM) as the statistic standard to measure the recovery results. Experimental results show that our network outperforms the existing methods and achieves better performance than the state-of-the-art.

The organization of this paper is as follows: Section 2 briefly introduces related works with discussion. In Section 3, architecture and loss function of the proposed framework are described, and the internal mechanism of our framework is explained from different views. The experimental results are shown and discussed in Section 4. Finally, the conclusion of our works is given in Section 5.

## 2. Related work

### 2.1. Learned measurements

The data-driven CS approaches recover signals using a deep neural network from measurements. The methods can be summarized as two types according to measurement matrix. The *first type* uses predefined measurement matrix  $\Phi_p$ . The scene is divided into  $T$  small blocks. Then we apply  $\mathbf{y}_i = \Phi_p \mathbf{x}_i$  to obtain the measurements of the  $i$ th block  $\mathbf{x}_i$ . The networks are trained with the input-label pairs  $(\mathbf{y}_i, \mathbf{x}_i)$  by minimizing the following loss function

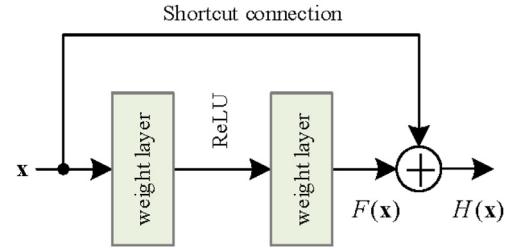
$$\mathcal{L}(\Theta) = \frac{1}{T} \sum_{i=1}^T \| f(\mathbf{y}_i, \Theta) - \mathbf{x}_i \|^2 \quad (1)$$

where  $\Theta$  means the parameters in the recovery network and  $f(\mathbf{y}_i, \Theta)$  is the output of the network for  $i$ th block. This kind of networks actually learn a mapping from measurements  $\mathbf{y}_i$  to signal  $\mathbf{x}_i$ .

Different from the first type, the *second type* learns the measurement matrix and the reconstruction algorithm jointly in a single network. These networks are trained with the input-label pairs  $(\mathbf{x}_i, \mathbf{x}_i)$  by minimizing the following loss function

$$\mathcal{L}(\Phi, \Theta) = \frac{1}{T} \sum_{i=1}^T \| f(\mathbf{x}_i, \{\Phi, \Theta\}) - \mathbf{x}_i \|^2 \quad (2)$$

where  $\Phi$  is parameters of the encoding layer and  $f(\mathbf{x}_i, \{\Phi, \Theta\})$  is the output of the network for the  $i$ th block. When it comes to the testing phase, the encoding layer parameters  $\Phi$  are equivalent to the measurement matrix. The difference between Eqs. (1) and (2) is that the output is decided by  $\mathbf{x}_i$  and  $\{\Phi, \Theta\}$  in (2), but  $\mathbf{y}_i$  and  $\{\Theta\}$  in (1). Based on this mechanism, Xie et al. [23] first adds a fully connected layer to ReconNet, which allows for the joint learning. Lohit et al. [22] propose two variants of ReconNet: ReconNet (Euc) trained with Euclidean loss and ReconNet (Euc + Adv) trained with a combination of Euclidean and adversarial loss. DeepCodec [21] develops a network which can learn to take under-sampled measurements and recover signals from it using convolutional and sub-pixel convolutional layer. These methods acquire high recovery



**Fig. 1.** Residual learning: a building block.

quality. However, they do not explain how the methods work well, e.g., the character of learned measurement matrix, the relationship between learned measurements and recovery network. We introduce those in details in the following Section 3.

Recently, the deep residual network (ResNet) [34] has achieved promising performance on several computer vision tasks. Compared with the traditional convolutional neural networks, the ResNet introduces identity shortcut connections that directly pass the data to later layers, which effectively avoids signal attenuation caused by multiple stacked non-linear transformations. As a consequence, a deeper network can be constructed with ResNet and faster training speed can be achieved. By deepening the network, ResNet generally gets better performance in comparison with other deep learning models. Residual learning block and shortcut connection as vital components of ResNet are shown in Fig. 1. A residual learning block can be expressed as  $\mathbf{y} = F(\mathbf{x}, W) + \mathbf{x}$ , where  $\mathbf{x}$  and  $\mathbf{y}$  are the input and output vectors of the residual learning block. The function  $F(\mathbf{x}, W)$  represents the residual mapping to be learned.  $W$  is parameters of the residual block.

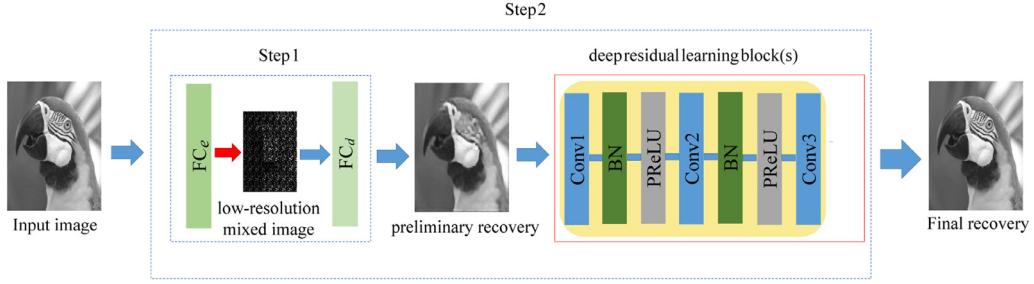
### 2.2. Deep residual network

We denote the desired underlying mapping performed by a few stacked layers in a neural network as  $H(\mathbf{x})$ . If multiple non-linear layers can asymptotically approximate complicated functions, they can asymptotically approximate the residual functions, i.e.,  $H(\mathbf{x}) - \mathbf{x}$ . So we explicitly make these layers approximate a residual function  $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$  instead of  $H(\mathbf{x})$ . The original function thus becomes  $F(\mathbf{x}) + \mathbf{x}$  and it can be realized by a feedforward neural network with shortcut connections.

ResNet is first applied in high-level computer vision tasks like image recognition and object detection [35–37]. It is also widely used in many other low-level computer vision tasks like image super-resolution [38], denoising [39], rain removal [40] and so on. For example, Ledig et al. [38] propose a super-resolution generative adversarial network in which they employ a deep residual network. Besides, Fu et al. [40] develop a rain removal framework called deep detail network which uses ResNet. Because of the good performance of ResNet acquired in image restoration field, we employ the ResNet as enhanced recovery part of the proposed framework.

## 3. Visualizing and understanding of the proposed framework

In this section, we begin our discussion by introducing the proposed network. And the network, which contains encoder and decoder, is considered as an equivalent CS system. The encoder actually is the measurement part. The decoder contains two stages: the preliminary recovery and the enhanced recovery. Subsequently, intuitive representations of learned measurements are given from two viewpoints. Finally, we visualize and understand the preliminary and enhanced recovery stages from different views.



**Fig. 2.** Structure of proposed CS framework. The two fully connected layers (encoder and decoder) is pre-trained to get preliminary parameters in step 1. Then the two fully connected layers and residual learning block(s) are fine-tuned together in step 2.

### 3.1. Proposed network

In this subsection, we introduce the architecture and loss function of the proposed network. As shown in Fig. 2, the network is composed of two fully connected (denoted as FC) layers and deep residual learning block(s). The first FC layer which has low-dimension output can be understood as measurement part. And the number of neurons of this layer is determined by the measurement rate. The second FC layer produces a preliminary reconstruction image. A deep residual learning block includes three convolutional layers, two batch normalization layers [41] and two PReLU layers [42]. This structure can be addressed in Fig. 2. As there is a trade-off between the reconstruction quality and computational cost, one can adjust the number of the residual blocks as needed.

In the training procedure, all the images in training set are  $33 \times 33$  patches which are extracted from original images. Different from the methods with predefined measurement matrix [17,18], our measurement matrix is trained jointly with recovery part. The input data of our network is also used as ground truth. The proposed network is trained by two steps: pre-training and fine-tuning shown as dashed rectangle in Fig. 2.

In the first step, the two FC layers with random Gaussian distribution initialization are pre-trained to get preliminary parameters. The loss function of pre-training procedure is given by

$$\mathcal{L}(W_e, W_d) = \frac{1}{T} \sum_{i=1}^T \| f(\mathbf{x}_i, \{W_e, W_d\}) - \mathbf{x}_i \|^2 \quad (3)$$

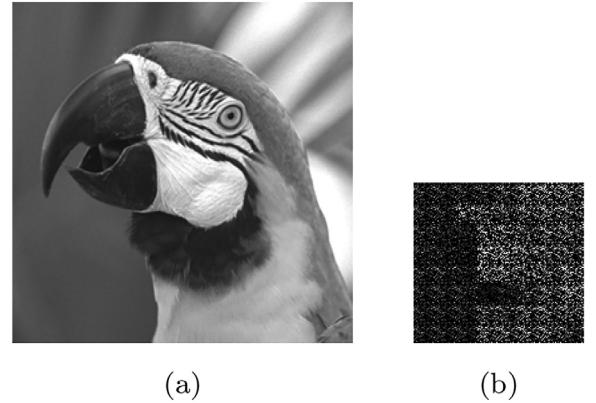
where  $f(\mathbf{x}_i, \{W_e, W_d\})$  is the  $i$ th reconstruction image block of preliminary procedure.  $\mathbf{x}_i$  is the  $i$ th original image block as well as the  $i$ th label.  $W_e$  and  $W_d$  denote the parameters of the encoding layer and decoding layers, respectively.  $T$  is the total number of image blocks. The loss function is minimized using backpropagation.

In the second step, the two fully connected layers and deep residual learning block(s) are fine-tuned together. Parameters of the fully connected layers are initialized with parameters of 2FC in the first step, while parameters of residual learning block are initialized with Gaussian distribution. The loss function is given by

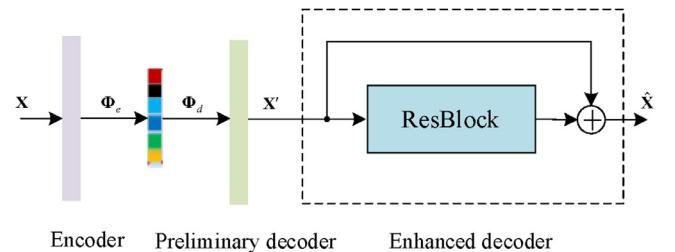
$$\mathcal{L}(W'_e, W'_d, K) = \frac{1}{T} \sum_{i=1}^T \| f(\mathbf{x}_i, \{W'_e, W'_d, K\}) - \mathbf{x}_i \|^2 \quad (4)$$

where  $f(\mathbf{x}_i, \{W'_e, W'_d, K\})$  is the  $i$ th reconstruction image block.  $\mathbf{x}_i$  is the  $i$ th original image block as well as the  $i$ th label.  $W'_e$  and  $W'_d$  denote parameters of encoding and decoding layers.  $K$  is all parameters in residual learning block(s).  $T$  is the total number of image blocks. The loss function is minimized by adjusting  $W'_e, W'_d, K$  using backpropagation.

When it comes to testing phase, the network can achieve two operators: measurement and recovery operators. For the measurement operator, the scene is measured block by block. The block



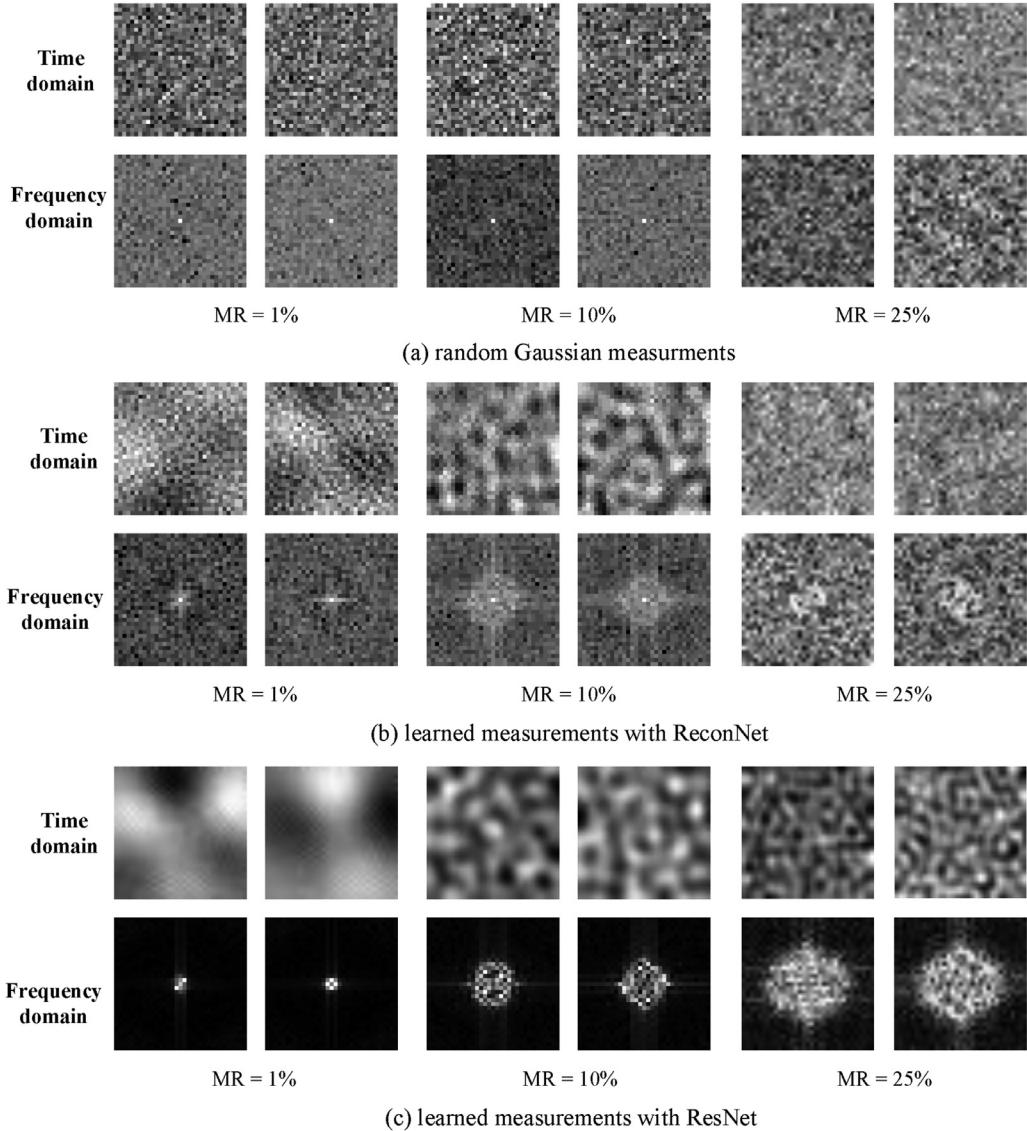
**Fig. 3.** (a) is the scene to be measured, and (b) is the corresponding measurements of (a) at MR=25%.



**Fig. 4.** Equivalent structure of our framework.

scene images share a learned measurement matrix, which is well trained by the training data, to get the corresponding measurements. In this operator, the parameters  $W'_e$  of FC layer are considered as measurement matrix. The measurements vector is obtained by  $\mathbf{y}_i = W'_e \mathbf{x}_i$ . We reshape the measurement vector to a matrix for better visual effect and an example is given in Fig. 3. From the example, we can know that the under-sampled measurements, which are mixed by the learned measurement matrix, can reserve some structure information of the scene images. For recovery operator, the block measurements, which are obtained by the measurement operator, are recovered by the following sub-network. And then, the recovery blocks are arranged appropriately to form the final reconstruction of the scene. In this operator, the second FC layer acts as the first recovery part to recover the scene images from the measurements preliminarily. Then, the deep residual learning block(s) works as an enhanced part to obtain the final reconstruction results.

For better understanding the proposed network, we consider the network as an equivalent structure containing one encoder and two decoders as Fig. 4 shows. Encoding part marked as  $\Phi_e$  is



**Fig. 5.** The measurement matrix at different MRs. (a) is random Gaussian measurement matrix at measurement 1%, 10% and 25% in time and frequency domain, (b) is the learned measurement matrix with ReconNet at measurement rate 1%, 10% and 25% in time and frequency domain, (c) is the learned measurement matrix with ResNet at measurement rate 1%, 10% and 25% in time and frequency domain.

equivalent to a fully connected layer. It performs mixed sampling to obtain output measurements, which is a dimension reduction process. Quite the opposite, the two decoders can increase the dimension. The first decoder marked as  $\Phi_d$  increases the dimension of measurements to get the low-resolution reconstruction result and it could be considered as a preliminary recovery decoder. And the second one, i.e., residual network, generates the final high-resolution reconstruction result after adding high-frequency information to the low-resolution preliminary recovery result. For better understanding, we analyze the proposed network in three parts, i.e., learned measurements (encoder), preliminary decoder, and enhanced decoder. They are described in the following Sections 3.2, 3.3 and 3.4, respectively.

### 3.2. Intuitive representations of learned measurement matrix

This subsection focuses on encoding part, i.e., learned measurement matrix (LMM), of the proposed framework. To understand the LMM, intuitive representations of the LMM are given from two viewpoints: visualization and energy.

First, we show the representation of LMM in the viewpoint of visualization. In the measurement part, the scene images are firstly divided into blocks with size  $33 \times 33$ . Then the block is reshaped to a column vector before being measured by the learning measurement matrix. To further understand the character of LMM, we reshape different rows of the learned measurement matrix to size  $33 \times 33$  and obtain the corresponding frequency domain images. Some examples of the reshaped LMM generated by the proposed network as shown in Fig. 5(c). It is displayed in time and frequency domain at measurement rates 1%, 10% and 25%. From the examples, it is clear that each row of the LMM is actually a low-pass filter. Thus, the measurements obtained by LMM of the proposed method can preserve the low-frequency information of the scene image. Besides, comparing the reshaped row vectors at different measurement rates in frequency domain (bottom row of Fig. 5(c)), we can observe that with the measurement rate increasing, the ability of LMM to keep high-frequency information is enhanced.

Though most of the information kept during the measurement process is low-frequency, our learned measurement matrix

**Table 1**

The energy ratios of the proposed network at different measurement rates.

MR	1%	4%	10%	25%
Energy ratio	9.13%	19.49%	29.65%	49.64%

**Table 2**

The ERs of different networks at different measurement rates: ReconNet [17] contains predefined measurement matrix (predefined), Adp-Rec [23] is composed of LMM and the recovery part of ReconNet (2FC + ReconNet), and the proposed method is composed of LMM, deconvolutional layer and residual blocks (2FC + ResNet).

MR	ReconNet (predefined)	Adp-Rec (2FC+ReconNet)	Proposed (2FC+ResNet)
1%	8.80%	9.06%	9.13%
10%	29.12%	30.26%	31.77%
25%	46.39%	48.97%	49.64%

performs better for extracting low-frequency information than other predefined or learned ones [17,23]. In order to prove that, some reshaped rows of the random Gaussian matrix used in ReconNet [17] at measurement rate 1%, 10% and 25% in time and frequency domain are shown in Fig. 5(a). Similarly, some reshaped rows of the LMM used in [23] are shown in Fig. 5(b). In Fig. 5(a), the random Gaussian measurement matrix is obviously irregular. We cannot get any useful information. The reshaped rows of measurement matrix shown in Fig. 5(b) is a LMM based on [23]. While comparing the LMM of the proposed network with ReconNet, it is obvious that ours can extract the low-frequency information of scene more efficiently according to the frequency domain images. This is due to that the recovery sub-networks of the whole CS framework are different. A powerful recovery sub-network can improve the ability of the LMM for obtaining information from scene.

Second, the representation of LMM is given in the viewpoint of energy. We show the energy ratio (ER) between the measurements and the scene image. Different from visualization of the LMM above, we draw a qualitative analysis in this part. As we know, measurements are obtained in the measurement stage. We use the ratio of energy between measurements and scene image to evaluate the ability of measurement matrix for obtaining information.

We assume that  $\mathbf{x} \in R^N$  is the scene image.  $\Phi_e \in R^{M \times N}$  ( $M \ll N$ ) and  $\mathbf{y}$  are the learned measurement matrix and measurements, respectively. The measurements can be obtained by  $\mathbf{y} = \Phi_e \mathbf{x}$ . The energy ratio can be formulated as  $ER = \frac{\|\Phi_e \mathbf{x}\|}{\|\mathbf{x}\|}$ . The ability of the measurement matrix would be stronger with the increase of the ER.

Table 1 shows the ERs of the proposed network at different measurement rates. It is noted that with the measurement rate increasing, energy ratio grows accordingly. Besides, we make a comparison among three different methods in order to illustrate the ability of measurement matrix, as shown in Table 2. The table shows the ERs of ReconNet [17] with predefined measurement matrix, Adp-Rec [23] with LMM and the recovery part of ReconNet, and the proposed method with LMM and residual blocks. The ERs at measurement rates 1%, 10% and 25% are given as examples. From Table 2, we can see that the energy ratios of the proposed network are larger than others. It means that our learned measurement matrix is able to extract more energy at the same measurement rate.

### 3.3. Visualizing and understanding of the preliminary recovery

In this subsection, we illustrate the preliminary recovery procedure from the transform and space viewpoints. Visualization rep-

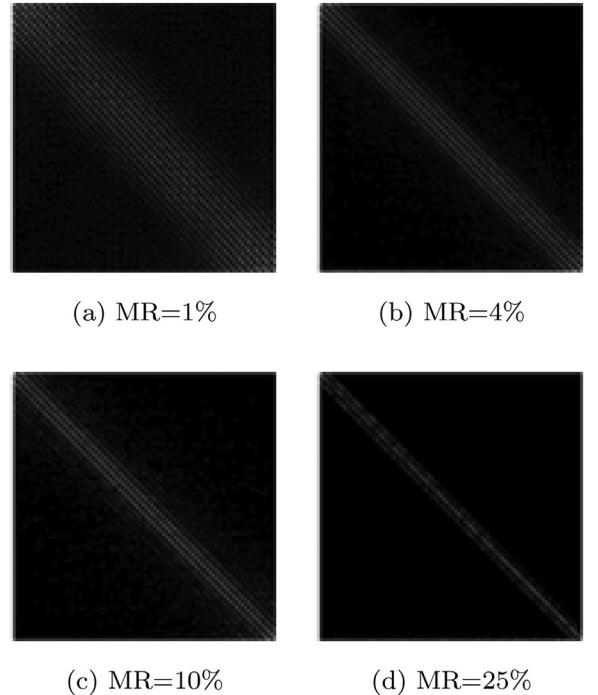


Fig. 6. Preliminary transform operation  $\mathbf{T}$  at measurement rates 1%, 4%, 10% and 25%.

resentation is used to help us understand the process of the preliminary recovery.

Similar to the encoder  $\Phi_e$  in the measurement part, the preliminary recovery decoder  $\Phi_d$  is also employed with a fully connected layer as Fig. 4 shows. The parameters of the encoder and preliminary recovery decoder are equivalent to a fat matrix and a tall matrix, respectively. According to measurement process  $\mathbf{y} = \Phi_e \mathbf{x}$ , the preliminary recovery procedure can be represented as

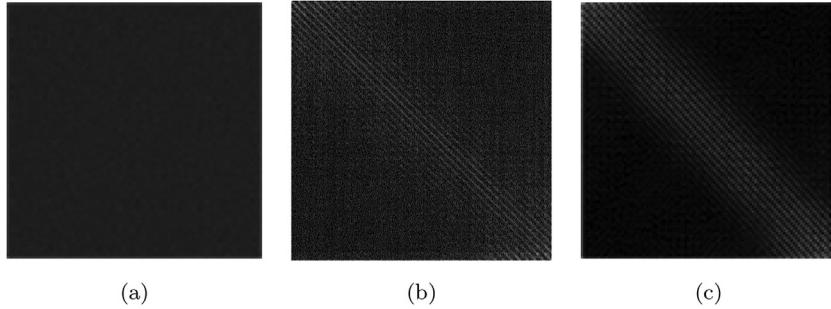
$$\mathbf{x}' = \Phi_d \mathbf{y} = \Phi_d \Phi_e \mathbf{x} \quad (5)$$

where  $\mathbf{x}$  is the scene image, and  $\mathbf{x}'$  is the preliminary reconstruction image. We assume  $\Phi_d \Phi_e$  as transform operation  $\mathbf{T}$ , then Eq. (5) can be written as

$$\mathbf{x}' = \mathbf{T} \mathbf{x}. \quad (6)$$

When the preliminary transform operation  $\mathbf{T}$  is closer to identity matrix  $\mathbf{I}$ , the preliminary reconstruction image  $\mathbf{x}'$  is closer to scene image  $\mathbf{x}$ . In the CS field, the recovery of the scene is an ill-posed inverse problem and the preliminary reconstruction result  $\mathbf{x}'$  is not totally equal to input signal  $\mathbf{x}$ . Therefore the preliminary transform operation  $\mathbf{T}$  is not an identity matrix. As an example, the preliminary transform operation  $\mathbf{T}$  of the proposed network at different measurement rates 1%, 4%, 10%, and 25% are shown in Fig. 6. From this figure, we can observe that with the measurement rate increasing, the preliminary transform operation approximates to identity matrix gradually.

At the same measurement rate, different methods will produce various preliminary transform operations. For example, ReconNet [17] uses fixed measurement, Adp-Rec [23] uses learned measurement with ReconNet, and we use learned measurement with ResNet. The preliminary transform operation  $\mathbf{T}$  at 1% measurement rate of these methods are shown in Fig. 7. Obviously, we can hardly observe the structure from the visualization result of ReconNet [17]. For the LMM-based methods, because the recovery network of the proposed method is better than ReconNet [17], the preliminary transform operation  $\mathbf{T}$  of the proposed method is closer to identity matrix than Adp-Rec [23].



**Fig. 7.** The preliminary transform operation  $\mathbf{T}$  at measurement rate 1%: (a) projection  $\mathbf{T}$  of ReconNet [17], (b) projection  $\mathbf{T}$  of Adp-Rec [23], (c) projection  $\mathbf{T}$  of the proposed method.

To better understand the preliminary recovery procedure, we explain it from the aspect of vector space through mathematic analysis. We further represent the measurement matrix  $\Phi_e \in R^{M \times N}$  as  $\Phi_e = (\varphi_{e_1} \dots \varphi_{e_j} \dots \varphi_{e_M})^T$ , where  $(\cdot)^T$  is the transpose of the matrix. The vector  $\varphi_{e_j} \in R^N$ ,  $j = 1, \dots, M$  denotes the  $j$ th column vector of  $\Phi_e^T$ .  $\varphi_{e_j}^i$ ,  $i = 1, \dots, N$  is the  $i$ th element of  $\varphi_{e_j}$ . Similarly, the preliminary recovery matrix  $\Phi_d \in R^{N \times M}$  is represented as  $\Phi_d = (\varphi_{d_1} \dots \varphi_{d_l} \dots \varphi_{d_M})^T$ .  $\varphi_{d_l} \in R^N$  is the  $l$ th column vector of  $\Phi_d$  and  $\varphi_{d_l}^k$  is the  $k$ th element of  $\varphi_{d_l}$ , where  $l = 1, \dots, M$  and  $k = 1, \dots, N$ . To express the measurement and preliminary recovery procedure more clearly, a toy example is given as follows.

Assuming  $N = 4$ ,  $M = 2$ , the signal  $\mathbf{x}$  and  $\Phi_e$  can be represented as  $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$  and  $\Phi_e = (\varphi_{e_1} \varphi_{e_2})^T$ , respectively. The measurements  $\mathbf{y}$  can be obtained by:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \varphi_{e_1}^1 & \varphi_{e_1}^2 & \varphi_{e_1}^3 & \varphi_{e_1}^4 \\ \varphi_{e_2}^1 & \varphi_{e_2}^2 & \varphi_{e_2}^3 & \varphi_{e_2}^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}. \quad (7)$$

The preliminary recovery matrix  $\Phi_d = (\varphi_{d_1} \varphi_{d_2})$  and preliminary recovery signal  $\mathbf{x}'$  can be expressed as:

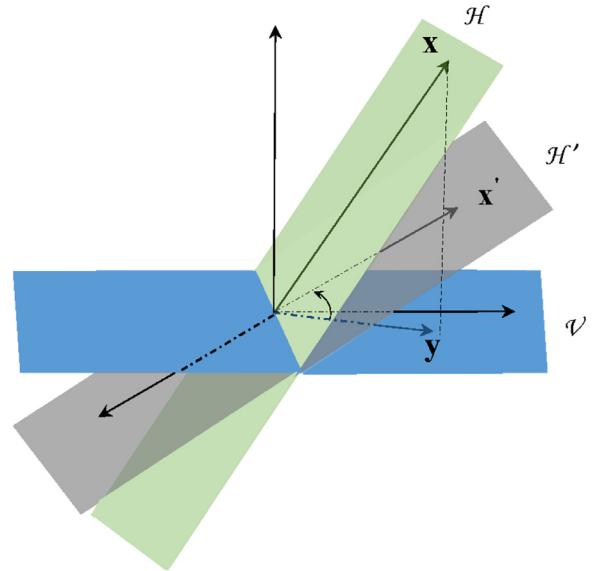
$$\mathbf{x}' = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{pmatrix} = \begin{pmatrix} \varphi_{d_1}^1 & \varphi_{d_2}^1 \\ \varphi_{d_1}^2 & \varphi_{d_2}^2 \\ \varphi_{d_1}^3 & \varphi_{d_2}^3 \\ \varphi_{d_1}^4 & \varphi_{d_2}^4 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \quad (8)$$

According to the above assumption, as shown in Fig. 8, the scene signal  $\mathbf{x}$  belongs to the space  $\mathcal{H}$  and the measurements  $\mathbf{y}$  belongs to the space  $\mathcal{V}$ . The measurement procedure can be abstracted as the process of dimension reduction. The preliminary recovery signal  $\mathbf{x}'$  belongs to the space  $\mathcal{H}'$  which is different from space  $\mathcal{H}$ , because the recovery of the signal is an ill-posed inverse problem. In other words, there is always a gap between the input signal  $\mathbf{x}$  and the preliminary recovery signal  $\mathbf{x}'$ .

From the analysis above, we can observe that the difference between the input scene image and the preliminary recovery image always exists. We need to reduce this gap by further optimizing our recovery network.

### 3.4. Understanding of residual network

As discussed above, the learned measurement matrix and preliminary decoder produce the low-resolution image  $\mathbf{x}'$ . To enhance the image quality of  $\mathbf{x}'$ , we apply the enhanced decoder to add high-frequency information to  $\mathbf{x}'$ . In this subsection, we introduce our enhanced decoder in details. Fig. 9 shows a detailed example of the two-stage reconstruction process. The preliminary recovery



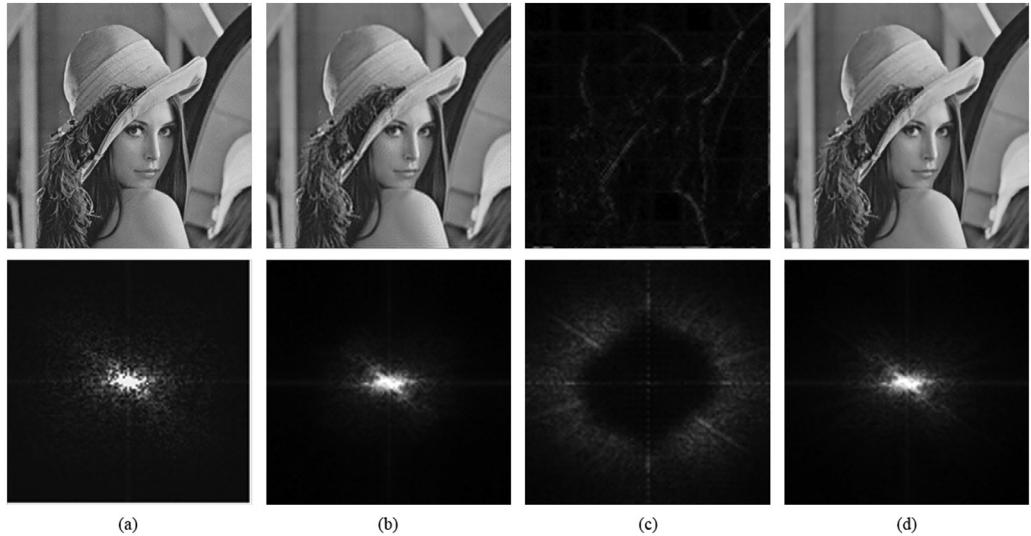
**Fig. 8.** The explanation of preliminary recovery procedure in the viewpoint of vector space. The signal  $\mathbf{x}$  is the scene signal which lies on the space  $\mathcal{H}$ . Signal  $\mathbf{y}$  is the measurements which lie on the space  $\mathcal{V}$ . And the signal  $\mathbf{x}'$  is the preliminary recovery signal which lies on the space  $\mathcal{H}'$ .

image mainly contains low-frequency components, then the high-frequency components are obtained by residual network. At last, the learned residual is added to the preliminary recovery image to get the final result.

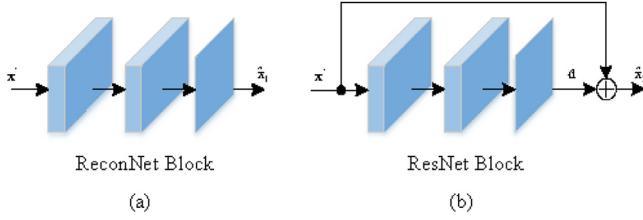
In order to add more high-frequency information to  $\mathbf{x}'$ , we utilize different reconstruction networks, i.e. ReconNet, ResNet. They are compared from two aspects and their structures are shown in Fig. 10(a) and (b), respectively.

From the standpoint of mapping, these two frameworks will learn different mappings. The mapping of ReconNet block can be denoted as  $\hat{\mathbf{x}}_1 = f_1(w_1, \mathbf{x}')$  where  $w_1$  denotes the parameters of ReconNet. It learns the direct mapping from the low-resolution image  $\mathbf{x}'$  to the high-resolution image  $\hat{\mathbf{x}}_1$ , which is hard to achieve. In contrast, the ResNet block only learns the residual  $d$  between LR and HR image. The mapping can be denoted as  $\hat{\mathbf{x}}_2 = \mathbf{x}' + \mathbf{d}$ , where  $\mathbf{d} = f_2(w_2, \mathbf{x}')$ ,  $w_2$  means the parameters of ResNet. It is obvious that the mapping of ResNet could be easier and more efficient.

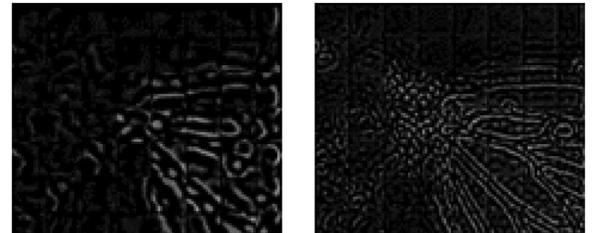
Then we will illustrate the difference between ReconNet and ResNet from the standpoint of the dictionary. For these two frameworks, the input signal  $\mathbf{x}'$  is first projected on the learned low-resolution dictionary to produce low-resolution coefficients. Second, the low-resolution coefficients are mapped to total-frequency coefficients and high-frequency coefficients in ReconNet



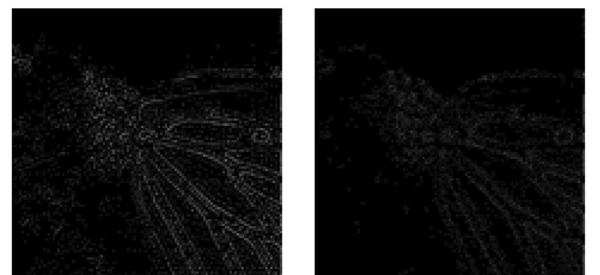
**Fig. 9.** Different images in time and frequency domain at measurement rate 25%. The first row shows images in time domain. The second row is in frequency domain. (a) Original image in time and frequency domain, (b) low-resolution image in time and frequency domain, (c) residual image in time and frequency domain, (d) reconstruction image in time and frequency domain.



**Fig. 10.** Two different enhanced blocks. (a) Diagram of ReconNet block, (b) diagram of ResNet block.



(a) MR=1% (b) MR=4%



(c) MR=10% (d) MR=25%

**Fig. 11.** The residual images at different measurement rates.

and ResNet, respectively. Finally, the coefficients, along with the corresponding dictionaries, synthesize the final signal  $\hat{\mathbf{x}}_1$  of ReconNet and the residual signal  $\mathbf{d}$  of ResNet. In ResNet, the residual  $\mathbf{d}$  and the input signal  $\mathbf{x}'$  are added to obtain the final result  $\hat{\mathbf{x}}_2$ .

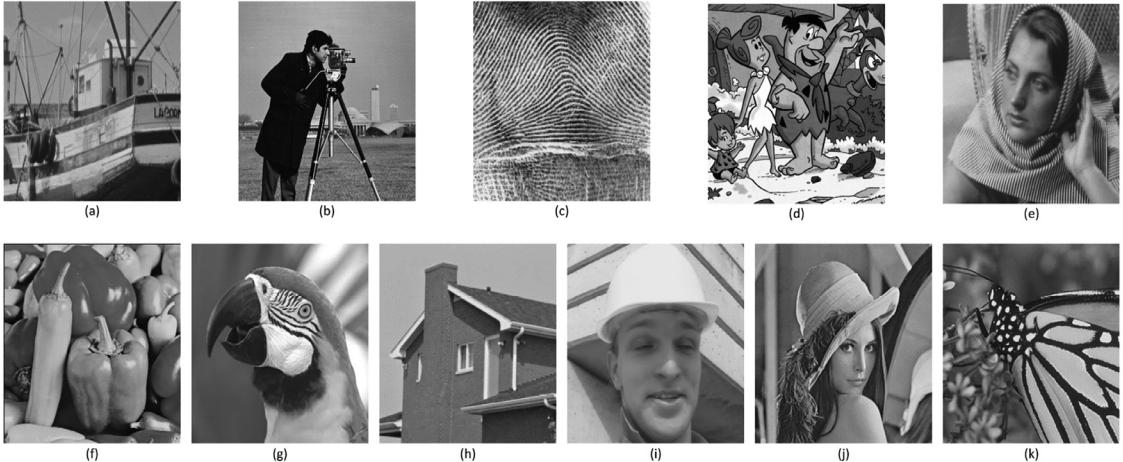
As we know, the information captured in the measurement process is mainly low-frequency and the preliminary recovery process mainly recovers the low-frequency information. Based on the preliminary recovery results, ResNet mainly supplements the high-frequency information, while ReconNet directly recovers the full frequency information of the signal. Due to ReconNet is hard to optimize, it is difficult to get the optimal results. ResNet, which is easy to learn the high-frequency residual information, matches the measurement and preliminary recovery processes better. From the comparison above, ResNet is more suitable for our task than ReconNet.

From the results of the proposed network, we observe that the learned residual image would be different with the change of measurement rate. The network is trained at four different measurement rates 1%, 4%, 10%, 25%. Then we input the same test image into these four networks. Their corresponding residual images, which are boosted for better visual effect, are shown in Fig. 11. We can observe from this figure that, with the measurement rate rising, the residual information with higher frequency. It is the reason that the final reconstruction image has higher frequency information. The added high-frequency information is based on the frequency information that the preliminary reconstruction image already has.

In conclusion, ResNet works better than ReconNet in the consideration of the overall network. The reason lies in that the ResNet and the measurement part can work in a cooperative manner.

#### 4. Experiments with discussion

In this section, we introduce training data and training strategy used for training the proposed network in detail. The proposed network composed of two decoders has been discussed above, here we compare the preliminary decoder alone with the overall network in terms of test loss and the average PSNR. At last we



**Fig. 12.** The 11 test images. From left to right and from up to bottom, the images are named as Boat, Cameraman, Fingerprint, Flintstones, Barbara, Peppers, Parrots, House, Foreman, Lena, Monarch.

**Table 3**

Comparison of training losses at different measurement rates. The 2FC\\_loss and 2FC\\_1Res correspond to the training loss of stage 1 and stage 2, respectively.

MR	1%	4%	10%	25%
2FC-loss	2.97	1.34	0.47	0.18
2FC-1Res	2.81	0.83	0.33	0.13

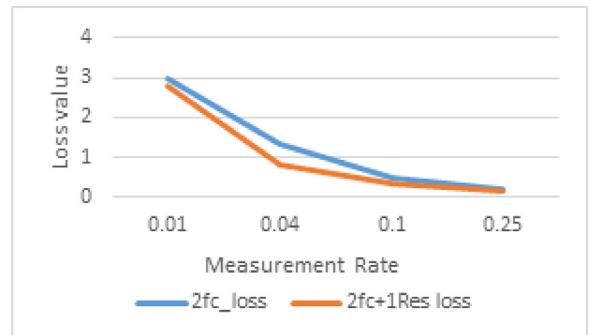
compare our proposed network with five existing methods without denoising.

*Training data:* The measurement process that we apply is block-wise, so the training data should be patches with fixed size. The dataset contains 21,760 patches of size  $33 \times 33$ . These patches are cropped with 14 stride from 91 images in [17]. These training patches are used as labels at the same time. Thus the input-output pair in training set is  $(x, x)$ .

*Training strategy and parameters:* The proposed network is trained at four different measurement rates 1%, 4%, 10%, 25%, and implemented on Caffe framework. The training process contains two stages. First, the two fully connected layers with random Gaussian initialization is trained to get initial parameters, which are called stage 1. In stage 2, the pretrained two fully connected layers are jointly trained with the ResNet recovery network, and the parameters of ResNet are initialized with random Gaussian. Max iterations of stage 1 are set as  $10^6$ . Stage 2 trains the whole network up to  $10^5$  iterations.

*Evaluation on linear mapping and residual learning:* We demonstrate that the two decoders can work in a cooperative way through two experiments. One aspect is about loss function, and the other is about PSNR. The loss functions of pretraining and fine-tuning procedures are already given in Section 3.1. Training loss of stage 1 and stage 2 are named as 2FC-loss and 2FC-1Res, respectively in Table 3. Based on the two fully connected layers, the enhanced part obviously improves the performance. That is why the proposed network achieves the lowest loss at different measurement rates. The 2FC-loss (blue curve) always outperforms 2FC-1Res (orange curve), as shown in Fig. 13.

We evaluate the performance with the different number of residual learning blocks. The mean PSNR values averaged on 11 test images in Fig. 12 are summarized in Table 4. As shown in Table 4, we can see that the reconstruction performance of 2FC+2Res is not as good as 2FC+3Res at MR=1%. However, in the case of MR=4%, 10%, and 25%, the result of 2FC+2Res is significantly better than 2FC+3Res.



**Fig. 13.** Loss of two architectures.

**Table 4**

Mean PSNR (dB) of raw encoder-decoder structure (2FC) and 2FC plus different number of ResBlock evaluated on test dataset in [17].

Model	MR			
	1%	4%	10%	25%
2FC	19.80	23.06	26.32	30.44
2FC+1Res	20.46	24.41	27.80	32.20
2FC+2Res	20.47	<b>24.50</b>	<b>27.93</b>	<b>32.36</b>
2FC+3Res	<b>20.49</b>	24.47	27.92	32.30

When the measurement rate is 1%, the measurements contain less information of the scene signal, and it is difficult to recover the original signal from the measurements. The bigger difficulty of reconstruction, the greater network capacity required. Through deepening the network, the capacity of the recovery network is stronger and the reconstruction performance is better. Therefore, the recovery performance of 2FC+3Res is better than 2FC+2Res.

When the measurement rates are 4%, 10% and 25%, the measurements contain more information of the scene. The difficulty of reconstruction and the required network capacity is less than MR=1%. In this situation, using a deep network, such as 2FC+3Res, easily causes overfitting, and the network has worse generalization ability. Therefore, the capacity of the 2FC+2Res network is more suitable for our task than 2FC+3Res when MR=4%, 10%, 25%.

**Table 5**

Comparison of our proposed method and classical methods in reconstruction time (in milliseconds) for a single  $256 \times 256$  image.

Method	MR			
	1%	4%	10%	25%
ReconNet	28.5	27.9	27.14	28.9
DR <sup>2</sup> -Net(Res1)	57.5	56.1	54.4	57.9
Adp-Rec	28.1	27.8	28.9	28.1
Proposed(Res1)	16.7	17.2	16.6	17.0
Proposed(Res2)	27.9	27.6	29.1	27.8
Proposed(Res3)	37.2	37.7	37.1	37.9

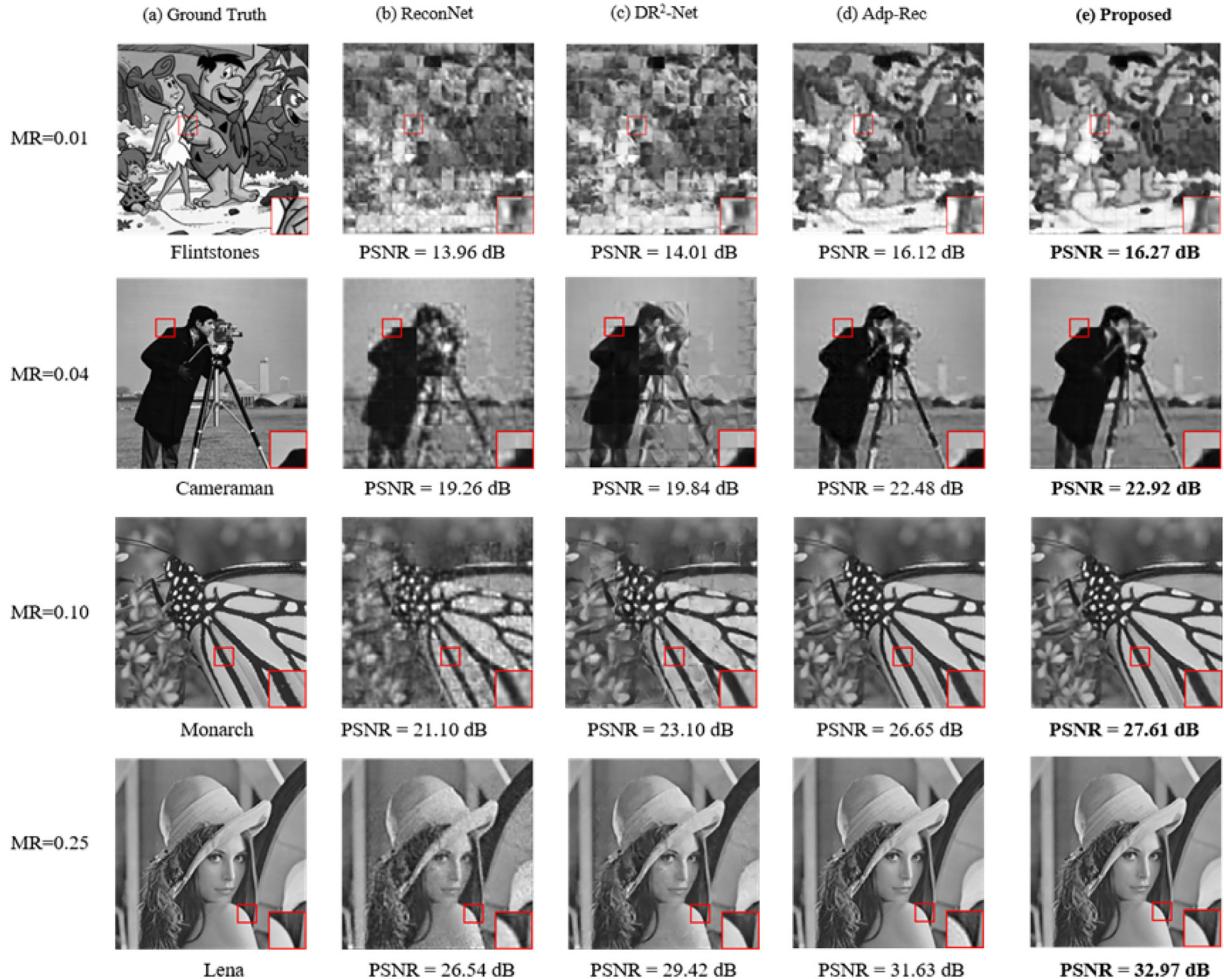
**Comparison with other methods:** We compare our algorithm with other three CS image reconstruction algorithms, which are ReconNet [17], DR<sup>2</sup>-Net [18] and Adp-Rec [23]. Test images in Fig. 12 are used in different methods, and the image size is  $256 \times 256$  except Fingerprint and Flintstones whose sizes are  $512 \times 512$ . The mean PSNR is shown in Table 6 and Fig. 14 displays the reconstruction images by different methods at different measurement rates.

As shown in Table 6, it is obvious that our algorithm outperforms the others in terms of PSNR and SSIM. However, the gap of reconstruction performance between the proposed and other methods is not obvious when the measurement rate is not large. This can be explained from the viewpoints of measurement rate and complexity. The aliasing signals obtained from the scene

**Table 6**

PSNR (dB) and SSIM with different methods at different measurement rates.

Image name	Methods	MR = 1% w/o BM3D	MR = 4% w/o BM3D	MR = 10% w/o BM3D	MR = 25% w/o BM3D
Monarch	ReconNet	15.39/0.38	18.19/0.52	21.10/0.65	24.31/0.77
	DR <sup>2</sup> -Net	15.33/0.39	18.93/0.57	23.10/0.73	27.95/0.86
	Adp-Rec	17.70/0.48	22.32/0.72	26.65/0.86	29.25/0.92
	Ours(2FC2Res)	<b>17.85/0.50</b>	<b>23.55/0.77</b>	<b>27.60/0.89</b>	<b>32.46/0.96</b>
Fingerprint	ReconNet	14.82/0.15	16.91/0.42	20.75/0.69	25.57/0.87
	DR <sup>2</sup> -Net	14.73/0.17	17.40/0.46	22.03/0.75	27.65/0.91
	Adp-Rec	16.22/0.16	20.80/0.63	<b>26.55/0.88</b>	<b>32.31/0.97</b>
	Ours(2FC2Res)	<b>16.22/0.16</b>	20.60/0.65	25.92/0.87	32.17/0.97
Flintstones	ReconNet	13.96/0.25	16.30/0.36	18.92/0.50	22.45/0.67
	DR <sup>2</sup> -Net	14.01/0.27	16.93/0.42	21.09/0.63	26.19/0.80
	Adp-Rec	16.12/0.32	19.48/0.54	23.83/0.75	27.94/0.86
	Ours(2FC2Res)	<b>16.27/0.35</b>	<b>20.10/0.59</b>	<b>24.94/0.79</b>	<b>29.72/0.89</b>
House	ReconNet	19.31/0.53	22.58/0.61	26.69/0.68	28.46/0.77
	DR <sup>2</sup> -Net	19.61/0.55	23.92/0.67	27.53/0.75	31.83/0.84
	Adp-Rec	<b>22.93/0.64</b>	27.61/0.77	31.43/0.85	34.38/0.91
	Ours(2FC2Res)	22.92/0.64	<b>27.95/0.78</b>	<b>31.45/0.86</b>	<b>35.35/0.91</b>
Boats	ReconNet	18.49/0.41	21.36/0.53	24.15/0.64	27.30/0.78
	DR <sup>2</sup> -Net	18.67/0.43	22.11/0.59	25.58/0.72	30.09/0.86
	Adp-Rec	21.09/0.49	25.24/0.69	28.80/0.83	32.47/0.92
	Ours(2FC2Res)	<b>21.29/0.51</b>	<b>25.59/0.71</b>	<b>29.12/0.85</b>	<b>33.59/0.93</b>
Cameraman	ReconNet	17.11/0.45	19.26/0.53	21.28/0.61	23.15/0.71
	DR <sup>2</sup> -Net	17.08/0.48	19.84/0.58	22.46/0.68	25.62/0.79
	Adp-Rec	19.74/0.56	22.48/0.68	24.97/0.79	27.11/0.87
	Ours(2FC2Res)	<b>19.98/0.59</b>	<b>22.92/0.71</b>	<b>25.07/0.81</b>	<b>28.84/0.90</b>
Lena	ReconNet	17.87/0.44	21.28/0.56	23.83/0.65	26.54/0.77
	DR <sup>2</sup> -Net	17.97/0.46	22.13/0.62	25.39/0.73	29.42/0.85
	Adp-Rec	21.49/0.56	25.61/0.74	28.50/0.85	31.63/0.93
	Ours(2FC2Res)	<b>21.57/0.57</b>	<b>26.09/0.76</b>	<b>28.86/0.86</b>	<b>32.97/0.94</b>
Barbara	ReconNet	18.61/0.37	20.38/0.48	21.89/0.57	23.25/0.68
	DR <sup>2</sup> -Net	18.65/0.38	20.70/0.51	22.69/0.63	25.77/0.80
	Adp-Rec	21.36/0.46	23.38/0.62	<b>24.28/0.72</b>	27.40/0.87
	Ours(2FC2Res)	<b>21.48/0.48</b>	<b>23.44/0.63</b>	24.27/0.72	<b>27.92/0.87</b>
Parrot	ReconNet	17.63/0.53	20.27/0.63	22.63/0.70	25.59/0.79
	DR <sup>2</sup> -Net	18.01/0.56	21.16/0.67	23.94/0.76	28.73/0.86
	Adp-Rec	21.67/0.67	24.14/0.79	27.59/0.88	30.51/0.93
	Ours(2FC2Res)	<b>21.77/0.69</b>	<b>24.32/0.80</b>	<b>27.93/0.89</b>	<b>31.89/0.94</b>
Foreman	ReconNet	20.04/0.56	23.72/0.66	27.09/0.75	29.47/0.82
	DR <sup>2</sup> -Net	20.59/0.61	25.34/0.73	29.20/0.82	33.53/0.89
	Adp-Rec	25.53/0.69	30.14/0.83	33.51/0.90	36.18/0.95
	Ours(2FC2Res)	<b>25.77/0.71</b>	<b>30.75/0.85</b>	<b>34.29/0.91</b>	<b>38.25/0.96</b>
Peppers	ReconNet	16.82/0.40	19.56/0.51	22.15/0.61	24.77/0.72
	DR <sup>2</sup> -Net	16.90/0.41	20.32/0.56	23.73/0.68	28.49/0.82
	Adp-Rec	19.75/0.51	23.63/0.71	26.67/0.84	29.65/0.90
	Ours(2FC2Res)	<b>20.05/0.54</b>	<b>24.19/0.75</b>	<b>27.52/0.86</b>	<b>32.82/0.93</b>
Mean PSNR/SSIM	ReconNet	17.28/0.41	19.98/0.52	22.77/0.64	25.53/0.76
	DR <sup>2</sup> -Net	17.41/0.43	20.80/0.58	24.25/0.72	28.66/0.84
	Adp-Rec	20.33/0.50	24.08/0.70	27.53/0.83	30.80/0.91
	Ours(2FC2Res)	<b>20.47/0.52</b>	<b>24.50/0.73</b>	<b>27.91/0.85</b>	<b>32.36/0.93</b>



**Fig. 14.** The reconstruction images by different methods at different measurement rates with PSNR and SSIM. (a) Original image, (b) image reconstructed by ReconNet [17], (c) image reconstructed by DR<sup>2</sup>-Net [18], (d) image reconstructed by ReconNet with fully connected layer [23], (e) image reconstructed by the proposed network.

would lose information. Therefore, when the measurement rate is low, the measurements include less information of the scene. And it is very difficult to recover the scene from less information. Thus the gap between the reconstruction effects among different methods is hard to be widened. Besides, we adopt the same network structure at different measurement rate. The low measurement rate retains less information of the scene, which makes it difficult to optimize the reconstruction network with high parameter complexity. From Fig. 14, we can see that our proposed method outperform the other methods in the quality of reconstruction.

**Time complexity:** Time complexity is a key factor for evaluating image compressive algorithm and should be analyzed comprehensively. The methods for comparison are divided into two categories. ReconNet [17], DR<sup>2</sup>-Net [18] and Adp-Rec [23] constitute the category of comparable methods. The other category is the proposed method with different residual block numbers. We conduct different methods on the same Intel(R) Core(TM) i5 – 6400 @ 2.70 GHz and Nvidia GTX 980. Since DNN-based methods are orders of magnitude faster than traditional iterative methods, we only utilize some data-driven methods to compare with ours. And we summarize their reconstruction time in Table 5.

From Table 5, we could observe that the time complexity is linearity correlated with the residual block number in our network at different MRs. For different residual blocks of network lay on the same level of reconstruction time, we choose 2FC+2Res which

outperforms the other settings in Table 4 as our optimal network structure.

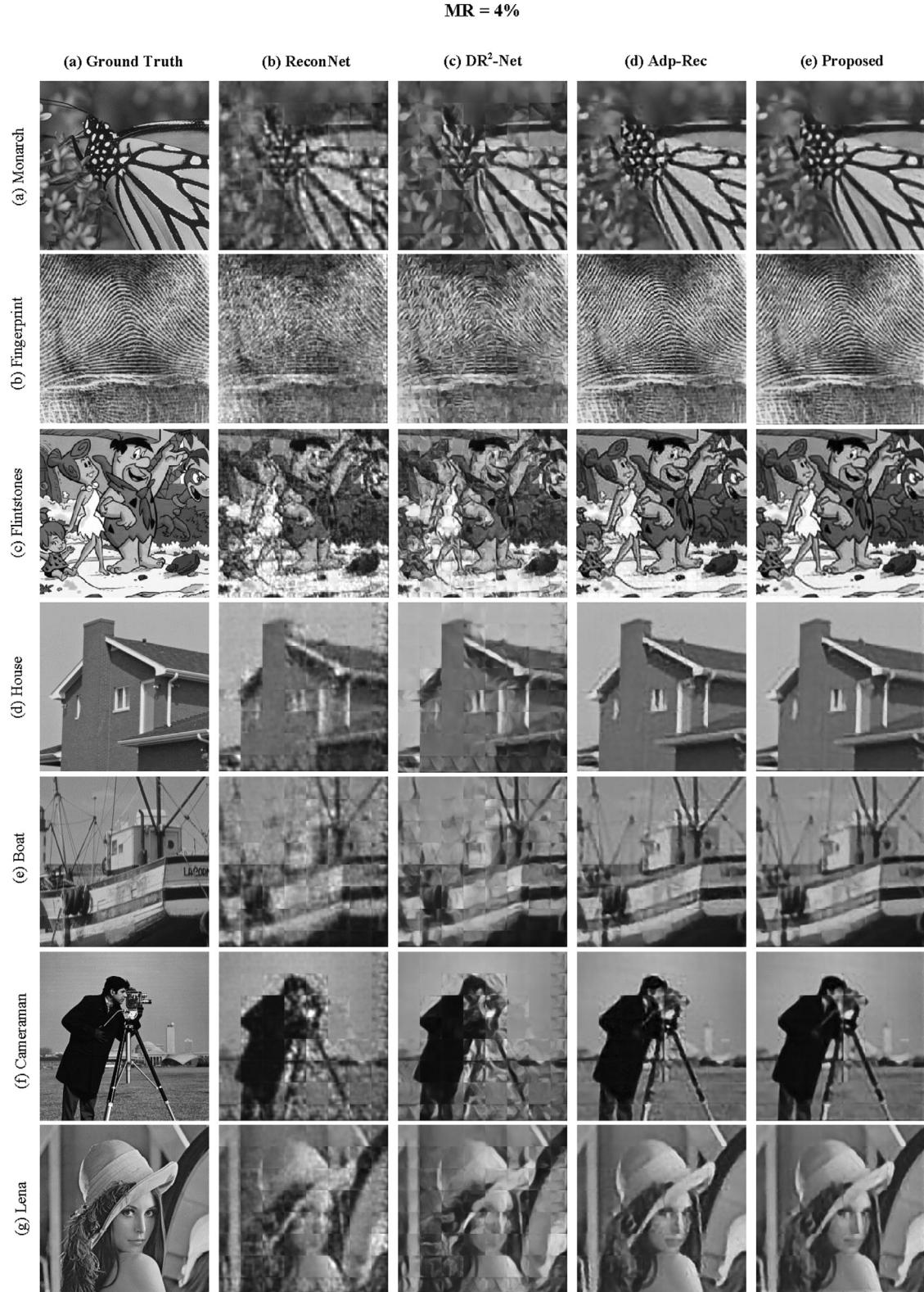
From results of the comparison of different methods in Table 5, we could see that our proposed method is faster than DR<sup>2</sup>-Net [18] significantly. Although our method has comparable reconstruction time with ReconNet [17] and Adp-Rec [23], it performs better than them in Table 6. In summary, our optimal proposed method outperforms the other methods considering both speed and performance.

## 5. Conclusion

In this paper, we visualize the LMM-based CS framework. Our work is the first time to take an insight look into the CS network. It helps to understand how CS framework works. We propose a LMM-based CS framework as an example to analyze the mechanism of CNN-based CS by the visualization. We conduct a set of experiments on a standard dataset. In the future, we would like to research the hardware implementation of LMM-based CS and apply it to real applications.

## Appendix

Part of reconstruction results is presented before. Here, the results of the remaining test images for various measurement rates using our proposed algorithm are shown in Figs. 15 and 16.



**Fig. 15.** The results of the remaining test images for 4% measurement rates using the proposed algorithm.

**MR = 25%**



**Fig. 16.** The results of the remaining test images for 25% measurement rate using the proposed algorithm.

## References

- [1] M. Lustig, D. Donoho, J.M. Pauly, Sparse MRI: the application of compressed sensing for rapid mr imaging, *Magn. Reson. Med.* 58 (6) (2007) 1182–1195.
- [2] D.S. Smith, J.C. Gore, T.E. Yankelev, E.B. Welch, Real-time compressive sensing MRI reconstruction using GPU computing and split Bregman methods, *Int. J. Biomed. Imaging* 2012 (2012) 1–6.
- [3] M.F. Duarte, M.A. Davenport, D. Takhar, J.N. Laska, T. Sun, K.F. Kelly, R.G. Baraniuk, Single-pixel imaging via compressive sampling, *IEEE Signal. Proc. Mag.* 25 (2) (2008) 83–91.
- [4] G. Huang, H. Jiang, K. Matthews, P. Wilford, Lensless imaging by compressive sensing, in: *Proceeding of the IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 2101–2105.

- [5] M.A. Herman, T. Strohmer, High-resolution radar via compressed sensing, *IEEE Trans. Signal. Process.* 57 (6) (2009) 2275–2284.
- [6] J. Fang, Z. Xu, B. Zhang, W. Hong, Y. Wu, Fast compressed sensing SAR imaging based on approximated observation, *IEEE J-STARS* 7 (1) (2014) 352–363.
- [7] D.L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory* 52 (4) (2006) 1289–1306.
- [8] E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Inform. Theory* 52 (2) (2006) 489–509.
- [9] W. Dong, G. Shi, X. Li, Y. Ma, F. Huang, Compressive sensing via nonlocal low-rank regularization, *IEEE Trans. Image. Process.* 23 (8) (2014) 3618–3632.
- [10] M.F. Duarte, M.B. Wakin, R.G. Baraniuk, Wavelet-domain compressive signal reconstruction using a hidden Markov tree model, in: Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2008, pp. 5137–5140.
- [11] C. Li, W. Yin, H. Jiang, Y. Zhang, An efficient augmented lagrangian method with applications to total variation minimization, *Comput. Optim. Appl.* 56 (3) (2013) 507–530.
- [12] C.A. Metzler, A. Maleki, R.G. Baraniuk, From denoising to compressed sensing, *IEEE Trans. Inform. Theory* 62 (9) (2016) 5117–5144.
- [13] S. Som, P. Schniter, Compressive imaging using approximate message passing and a Markov-tree prior, *IEEE Trans. Image. Process.* 60 (7) (2012) 3439–3448.
- [14] J. Zhang, S. Liu, R. Xiong, S. Ma, D. Zhao, Improved total variation based image compressive sensing recovery by nonlocal regularization, in: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), 2013, pp. 2836–2839.
- [15] E. Arias-Castro, Y.C. Eldar, Noise folding in compressed sensing, *IEEE Signal. Process. Lett.* 18 (8) (2011) 478–481.
- [16] A. Mousavi, A.B. Patel, R.G. Baraniuk, A deep learning approach to structured signal recovery, in: Proceedings of the Fifty-Third Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2015, pp. 1336–1343.
- [17] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, A. Ashok, ReconNet: non-iterative reconstruction of images from compressively sensed measurements, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 449–458.
- [18] H. Yao, F. Dai, D. Zhang, Y. Ma, S. Zhang, Y. Zhang, in: DR<sup>2</sup>-Net: deep residual reconstruction network for image Compressive sensing, 2017. arXiv: 1702.05743.
- [19] A. Mousavi, R.G. Baraniuk, Learning to invert: signal recovery via deep convolutional networks, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 2272–2276.
- [20] S. Lohit, K. Kulkarni, P. Turaga, Direct inference on compressive measurements using convolutional neural networks, in: Proceedings of the International Conference on Image Processing (ICIP), 2016, pp. 1913–1917.
- [21] A. Mousavi, G. Dasarathy, R.G. Baraniuk, DeepCodec: adaptive sensing and recovery via deep convolutional neural networks, in: Proceedings of the Fifty-Fifth Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2017, pp. 744–745.
- [22] S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, A. Ashok, Convolutional neural networks for noniterative reconstruction of compressively sensed images, *IEEE Trans. Comput. Imaging* 4 (3) (2018) 326–340.
- [23] X. Xie, Y. Wang, G. Shi, C. Wang, J. Du, X. Han, Adaptive measurement network for CS image reconstruction, in: Proceedings of the Chinese Conference on Computer Vision (CCCV), 2017, pp. 407–417.
- [24] J. Du, X. Xie, C. Wang, G. Shi, X. Xu, Y. Wang, Fully convolutional measurement network for compressive sensing image reconstruction, *Neurocomputing* (2018) 105–112.
- [25] I. Rafegas, M. Vanrell, L.A. Alexandre, in: Understanding trained CNNs by indexing neuron selectivity, 2017. arXiv: 1702.00382.
- [26] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 818–833.
- [27] W. Yu, K. Yang, Y. Bai, H. Yao, Y. Rui, in: Visualizing and comparing convolutional neural networks, 2014. arXiv: 1412.6631.
- [28] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, in: Proceedings of the International Conference on Machine Learning Workshop on Deep Learning (ICMLW), 2015, pp. 1–12.
- [29] C.C.J. Kuo, Understanding convolutional neural networks with a mathematical model, *J. Vis. Commun. Image Represent.* 41 (2016) 406–413.
- [30] N. Mousavi, H. Siqueira, P. Barros, B. Fernandes, S. Wermtter, Understanding how deep neural networks learn face expressions, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2016, pp. 227–234.
- [31] P.E. Rauber, S.G. Fadel, A.X. Falcao, A.C. Telea, Visualizing the hidden activity of artificial neural networks, *IEEE Trans. Vis. Comput. Graph* 23 (1) (2016) 101–110.
- [32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene CNNs, in: Proceedings of the International Conference on Machine Learning (ICLR), 2015, pp. 15–23.
- [33] C. Feichtenhofer, A. Pinz, R.P. Wildes, A. Zisserman, What have we learned from deep representations for action recognition? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 19–29.
- [34] 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 (CVPR), 2016, pp. 770–778.
- [35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, *IEEE Trans. Pattern. Anal. Mach. Intell.* 40 (4) (2018) 834–848.
- [36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016, pp. 21–37.
- [37] S. Ren, R. Girshick, R. Girshick, Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Trans. Pattern. Anal. Mach. Intell.* 39 (6) (2017) 1137–1149.
- [38] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 105–114.
- [39] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising, *IEEE Trans. Image. Process.* 26 (7) (2017) 3142–3155.
- [40] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, J. Paisley, Removing rain from single images via a deep detail network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1715–1723.
- [41] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning (ICML), 2015, pp. 448–456.
- [42] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026–1034.

**Zhipu Zhao** is a Ph.D. student in the School of Artificial Intelligence at Xidian University, Xi'an, China. He received his B.S. degree in School of Information Engineering from Zhengzhou University in 2013. His research interests are deep learning, computer vision, image processing, neural networks and compressive sensing.



**Xuemei Xie** is a Professor in the School of Artificial Intelligence at Xidian University, Xi'an, China. She is an IEEE member. She received her M.S. degree in Electronic Engineering from Xidian University in 1994, and Ph.D. degree in Electrical & Electronic Engineering from the University of Hong Kong in 2004. She has published over 50 academic papers in international and national journals, and international conferences. Her research interests are artificial intelligence, compressive sensing, deep learning, image and video processing, and filter banks.



**Chenyue Wang** is a master student in the School of Artificial Intelligence at Xidian University, Xi'an, China. He received his B.S. degree in School of Physics and Optoelectronic Engineering from Xidian University in 2016. His research interests are compressive sensing, deep learning, image and video processing.



**Wan Liu** is a master student in the School of Artificial Intelligence at Xidian University, Xi'an, China. She received her B.S. degree in School of Physics and Optoelectronic Engineering from Xidian University in 2017. Her research interests are compressive sensing, neural networks, optimization algorithms, deep learning and image processing.





**Guangming Shi** received the B.S. degree in Automatic Control in 1985, the M.S. degree in Computer Control and Ph.D. degree in Electronic Information Technology, all from Xidian University in 1988 and 2002, respectively. He joined the School of Electronic Engineering, Xidian University, in 1988. From 1994 to 1996, as a Research Assistant, he cooperated with the Department of Electronic Engineering at the University of Hong Kong. Since 2003, he has been a Professor in the School of Electronic Engineering at Xidian University, and in 2004 the head of National Instruction Base of Electrician & Electronic (NIBEE). From June to December in 2004, he had studied in the Department of Electronic Engineering at University of Illinois at Urbana-Champaign (UIUC). Presently, he is the Deputy Director of the School of Electronic Engineering, Xidian University, and the academic leader in the subject of Circuits & Systems. His research interests include compressed sensing, theory and design of multirate filter banks, image denoising, low-bit-rate image/video coding and implementation of algorithms for intelligent signal processing (using DSP&FPGA).



**Jiang Du** is a Ph.D. student in the School of Artificial Intelligence at Xidian University, Xi'an, China. He is an IEEE student member. He received his B.S. degree in School of Electronic Engineering from Xidian University in 2016. His research interests are artificial intelligence, deep learning, computer vision, image and video processing, neural networks and compressive sensing.