IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 22, 2019, accepted December 18, 2019, date of publication December 23, 2019,
date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961369

A Two-Branch Convolution Residual Network for Image Compressive Sensing

CHENQUAN GAN, XIAOQIN YAN, YUNFENG WU, AND ZUFAN ZHANG
School of Communication and Information Engineering, Chongqing University of Post and Telecommunications, Chongqing 400065, China

Corresponding author: Zufan Zhang (zhangzf@cqupt.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61702066 and Grant 61903056, in part by the Major Project of Science and Technology Research Program of Chongqing Education Commission of China under Grant KJZD-M201900601, in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2017jcyjAX0256, Grant cstc2019jcyj-msxmX0681, and Grant cstc2018jcyjAX0154, and in part by the Chongqing Municipal Key Laboratory of Institutions of Higher Education under Grant cqupt-mct-201901.

ABSTRACT Deep learning has made great progress in image compressive sensing (CS) tasks recently, and several CS models based on it have achieved superior performance. In practice, sensing the entire image requires huge memory and computational effort. Although the block-based CS method can effectively realize image sensing, it will cause block effects that severely decrease the reconstruction performance. To this end, this paper proposes a two-branch convolution residual network for image compressive sensing (denoted as TCR-CS), which mainly consists of a two-branch convolution autoencoder network and a residual network. Specifically, the two-branch convolution autoencoder network senses the entire image through multiple scale convolutional filters to obtain measurements. For better CS reconstruction, the image is preliminarily reconstructed by the deconvolution decoder network, and then the residual network is used to optimize the pre-reconstructed image. Through the end-to-end training, all networks can be jointly optimized. Finally, experimental results demonstrate that the proposed TCR-CS method is superior to existing state-of-the-art CS methods in terms of structural similarity, reconstruction performance and visual quality at different measurement rates.

INDEX TERMS Image compressive sensing, two-branch convolution, residual network, structural similarity, reconstruction performance, visual quality.

I. INTRODUCTION

Compared to the Nyquist Sampling method, compressive sensing (CS) [1], [2] is a more efficient transformative sampling technique, which directly senses signals at sub-Nyquist rates while retaining the necessary information and recovering signal with high probability. This theory has great potential in improving imaging speed and reducing energy consumption of the sensor. Several new imaging applications based on CS have been developed, such as radar imaging [3], single-pixel camera [4], high-speed video camera [5], spectrum sensing [6], and magnetic resonance imaging (MRI) [7]. However, the promise of CS is often offset by challenges associated with two limitations in practical applications. First, it is difficult to store a random sensing matrix of large images. Second, when a high-dimensional signal vector is multiplied by an arbitrary random matrix, the lack of any fast matrix multiplication algorithm will lead to a high computational complexity.

Usually, the actual natural images have a high dimension, thus the dimension of measurement matrix also becomes prohibitively high when CS is applied to the images. In addition, such large sensing matrix results a high computational cost in the process of CS reconstruction. To overcome this difficulty, Gan [8] proposed a block-based CS (BCS) framework, in which the image is segmented into many non-overlapping image blocks, then each image block is sensed and reconstructed independently. After reconstructing block by block, all reconstructed blocks are placed their locations and then are spliced into a full-size image stiffly. However, the blocks are reshaped into columns in this method, which damages the structure information and results in serious block effects especially in the low measurement rate. To improve the visual quality, the reconstructed image requires a certain post-processing to eliminate the blocking artifacts.

The associate editor coordinating the review of this manuscript and approving it for publication was Dalei Wu.

Different from traditional hand-based features and algorithms, deep learning is an emerging field that automatically extracts features from data to construct multiple levels of abstract representations. Recently, deep learning has been successfully applied in many fields, such as image processing [9], sentiment analysis [10], [11], Internet of Things [12], [13], and modulation recognition [14]. In the aspect of the task of CS [15]–[17], it is used to replace the optimization process of conventional methods for reducing the reconstruction time and computational complexity. By minimizing the error between the original image and reconstructed image, the deep neural network can adaptively learn a transform function of the measurement of the reconstructed image from the large dataset.

The first CS reconstruction method based on deep learning is proposed in [15], which uses stacked denoising autoencoders as an unsupervised feature learner to map a functional relationship between measurements and the original image. This greatly reduces the reconstruction time while the reconstruction quality is comparable to the existing advanced algorithms. Later, Dong *et al.* [16] demonstrated that the convolutional neural network (CNN) can learn a mapping from a low-resolution image to a high-resolution one by the end-to-end training manner. For further improving the reconstruction performance, the CNN is applied to recover signals from measurements in ReconNet [18], [19] and DeepInverse [20]. The DR^2-Net [21], inheriting ReconNet, utilized residual learning in the network, which can effectively avoid signal attenuation by transmitting the features directly to the later layers. Instead of learning to directly reconstruct the high-resolution image from the low-resolution one, DR^2-Net learns the residual information between the ground truth image and the preliminary reconstructed image. However, these methods only consider the optimization of the reconstruction process and neglect the measurement process, because the measurement is obtained by random measuring, not designed for image signal.

In response to this situation, DeepCodec [22] learned a transformation mapping from images to undersampled measurements, and then reconstructed original signals from them. Through effectively combining ReconNet with the fully connected layer, Xie *et al.* [23] proposed a new adaptive measurement network, in which measurements are designed through training and learning. Although these two methods can extract more effective information from the scene and further improve the reconstruction performance by jointly training the measurement and reconstruction stages, the input images of them are measured and recovered block by block, which will damage the structure information of image and result in serious block effects in the reconstructed image. This is also a problem with all of the above block-based methods.

For this issue, Xie *et al.* [24] firstly designed a measurement network based on the convolution method, in which the input is measured by the overlapped convolution operation. Different from the block-based approach, a convolution layer is used to sense measurements from the whole image, which retains the integral structure information of the original image and removes the block effect effectively. Additionally, a new CNN-based network is proposed to reduce the block effects of reconstructed image in [25]. In the measurement part, a group of measurements are obtained from the input image via adaptively measuring block by block. While in the reconstruction part, the full image is reconstructed at one time from the block-based measurements.

Unlike the existing methods, this paper proposes a two-branch convolution residual network for image compressive sensing (denoted as TCR-CS), which comprises of three parts: two-branch convolution sensing, pre-reconstruction and residual reconstruction. In the two-branch convolution sensing network, the whole image is sensed by two convolution networks with different scale filters. A large filter and a small filter sense respectively the whole images to obtain two measurement vectors from different perspectives, which can not only get more valid information, but also remove the block effect in the reconstruction process. Next, the final measurement vector is obtained through blending the features of the two branches. In the pre-reconstructed network, the fusion measurements are reconstructed by de-convolution network preliminarily. Then the pre-reconstructed images are optimized and reconstructed to high-quality images in the residual reconstruction network. By the end-to-end training, the proposed TCR-CS network is jointly trained from the measurement to the recovery part. Finally, experimental results show that the proposed TCR-CS method consistently outperforms existing state-of-the-art CS methods in terms of structural similarity, reconstruction performance and visual quality at different measurement rates.

The rest of this paper is organized as follows. Section II introduces the proposed method in detail. Experimental results and comparisons are analyzed in Section III. Finally, Section IV concludes this work.

II. THE PROPOSED METHOD

In this paper, a two-branch convolution compressed sensing network is formed by combining the two-branch convolution autoencoder network with the residual neural network. Similar to the traditional CS approaches including linear measurement sampling and non-linear reconstruction algorithm, the proposed TCR-CS network consists of two important parts: the full image convolution sensing module and the image reconstruction module (please refer to Figure 1).

The goal of the convolution sensing module is to realize the sensing of the whole image through the two-branch convolution autoencoder network. The image reconstruction module includes two networks: pre-reconstruction network and residual reconstruction network, which are used to realize pre-reconstruction and re-optimization reconstruction, respectively. Now, let us introduce these two parts of the TCR-CS network in detail.

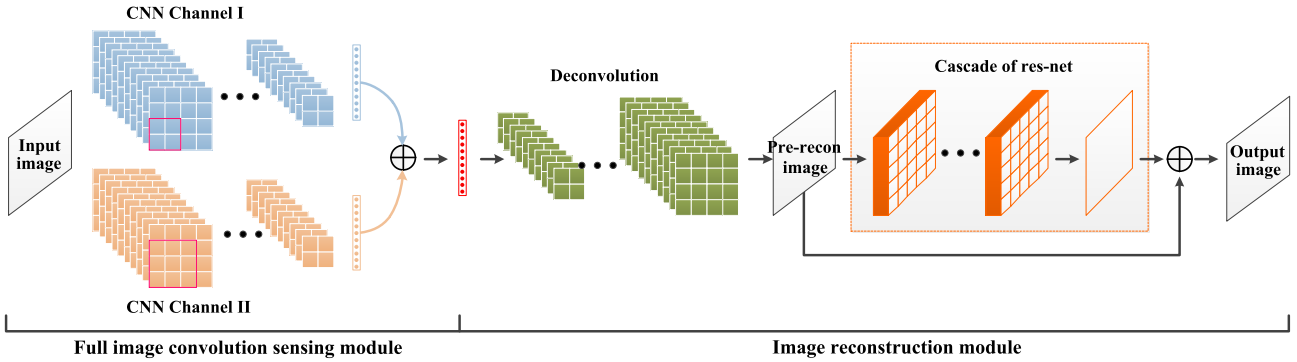


FIGURE 1. The framework of TCR-CS.

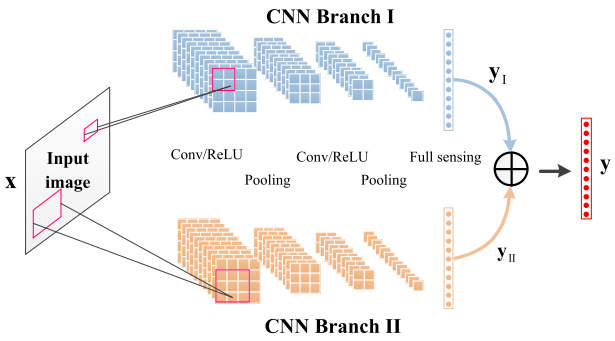


FIGURE 2. The full image convolution sensing module.

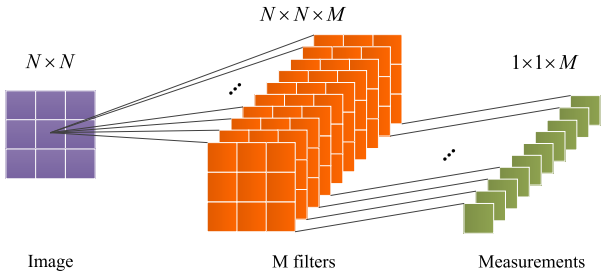


FIGURE 3. The full sensing layer.

A. THE FULL IMAGE CONVOLUTION SENSING MODULE

Due to the memory constraints, the existing CS methods based on deep learning usually adopt block-based pattern. However, the block effect comes accordingly. Unlike the existing BCS and convolutional CS, a full image convolution sensing method based on two-branch convolutional neural network is utilized for overcoming the block effect in this paper.

As shown in Figure 2, the full image convolution sensing module contains two CNN branches, which are identical in the network structure. In each branch, there are two convolution layers, two pooling layers, and a full sensing layer. The only difference is that the two branches adopt filters of different sizes.

The given image $\mathbf{x} \in \mathbb{R}^{N\times N}$ is fed into these two convolutional neural networks, respectively. In the CNN Branch I, a small filter with size $L_{\mathrm{I}}^{\mathrm{A}} \times L_{\mathrm{I}}^{\mathrm{A}}$ is used to sense the input image for getting the measurements $\mathbf{y_I}$. Likewise, a big filter with size $L_{\mathrm{I}}^{\mathrm{A}} \times L_{\mathrm{I}}^{\mathrm{A}}$ is utilized to sense the input image for obtaining the measurements $\mathbf{y_{II}}$ in the CNN Branch II. $\mathbf{y_I}$ and $\mathbf{y_{II}}$ respectively represent two measurement vectors sensed by different fields of view. To effectively integrate the features of two different fields of view, a linear addition method is applied to fuse $\mathbf{y_I}$ and $\mathbf{y_{II}}$, which is defined as:

$$\mathbf{y} = \alpha\mathbf{y}_{\mathrm{I}} + \beta\mathbf{y}_{\mathrm{II}}, \tag{1}$$

where $\mathbf{y}$ denotes the final measurements, α and β are two parameters. In the two-branch convolution sensing network, the measurements $\mathbf{y}$ also can be formulated as:

$$\mathbf{y} = \alpha f_{\mathrm{I}}^{\mathrm{A}}(\mathbf{x}) + \beta f_{\mathrm{II}}^{\mathrm{A}}(\mathbf{x}), \tag{2}$$

where $f_{\mathrm{I}}^{\mathrm{A}}(\cdot), f_{\mathrm{II}}^{\mathrm{A}}(\cdot)$ represent the CNN Branch I and Branch II of two-branch convolution sensing network, respectively.

In the full image convolution sensing module, the main task is that senses the whole image for generating the M-dimensional measurement vector. The full sensing layer is responsible for this critical work in the process of image sensing. Unlike existing BCS and convolutional CS, the image $\mathbf{x} \in \mathbb{R}^{N\times N}$ is convolved with M convolution filters whose size is the same as the input image to generate M measurements with size 1×1 in the proposed method (please refer to Figure 3). It can be expressed in mathematical formulas as:

$$\mathbf{y} \in \mathbb{R}^{1\times 1\times M} = \mathbf{x} \in \mathbb{R}^{N\times N} * F \in \mathbb{R}^{N\times N\times M}, \tag{3}$$

where $\mathbf{y} \in \mathbb{R}^{1\times 1\times M}$ denotes M measurements with size 1×1, $F \in \mathbb{R}^{N\times N\times M}$ stands for M convolution filters with size $N \times N$. However, one of the disadvantages of this method is that the parameters and computational complexity will increase dramatically when the image size $N \times N$ is large. To compensate for this shortcoming, two convolution layers and two pooling layers before the fully sensing layer are added.

The main function of convolution layer in the proposed TCR-CS network is to extract coding features and gradually reduce image size. For the output matrix $\mathbf{y}_{out} \in \mathbb{R}^{h_{out}\times w_{out}}$, the dimensions of the height h_{out} and width w_{out} are determined by the input matrix $\mathbf{x}_{in} \in \mathbb{R}^{h_{in}\times w_{in}}$, the size of filter

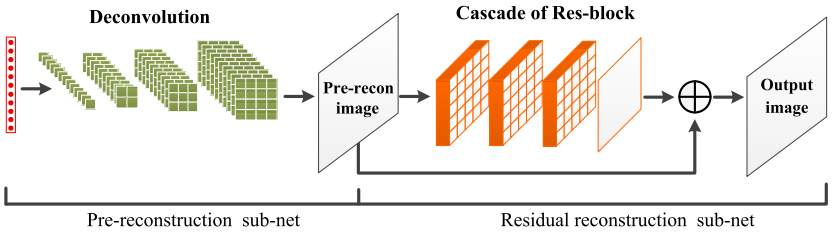


FIGURE 4. The image reconstruction module.

$h_{filter} \times w_{filter}$, padding p and stride s in the process of convolution. Then,

$$h_{out} = (h_{in} - h_{filter} + 2 * p)/s + 1, \quad (4)$$

$$w_{out} = (w_{in} - w_{filter} + 2 * p)/s + 1. \quad (5)$$

As can be seen from Eqs. (4) and (5), the sliding step of the filter is the main factor to change the size of the image, so stride s is set as a factor to adjust the compression ratio. When the image with size $N \times N$ is fed into the convolution layer, the dimension of the output image will reduce to $(N/s) \times (N/s)$. To get better performance, a nonlinear activation function is added before the convolution layer in the TCR-CS network. The Rectified Linear Unit (ReLU) can be viewed as a new activation function aiming sparse representation and preserving non-linearity.

The goal of the pooling layer is to compress the convolution layer again. It is able to extract the main features while reducing the computational complexity of the network. In the proposed TCR-CS network, the maximum pooling filter with stride 2 and size 2×2 is adopted, which reduces the height h_{out} and width w_{out} of the output image by half. Essentially, the pooling layer uses the downsampling technique, which is designed to reduce the dimension of features and preserve valid information while avoiding overfitting. Through the convolution and pooling operations, the dimension of the image is dramatically reduced. For example, the dimension of the output image will be reduced to $(N/4s^2) \times (N/4s^2)$ when the image with size $N \times N$ passes through two convolution layers with stride s and two pooling layers. Different from the traditional linear sensing, the multi-layer networks can preserve the key features of image while reducing gradually the dimension of image. In addition, one can adjust s to fit the different size of image.

The full image convolution sensing module encodes the image information and plays a similar role as the encoder of autoencoder. Importantly, the proposed TCR-CS network is clearly distinct from the existing CS methods in the following aspects. First, the proposed TCR-CS network is much easier to sense the whole large dimension image without the block effects. Second, the proposed TCR-CS network uses two CNNs with different filters to sense the image, and then combines these two features to obtain the CS measurements.

B. THE IMAGE RECONSTRUCTION MODULE

The goal of the image reconstruction module is to restore the original information from measurements. In recent years, CNN has been applied to many low-level image processing tasks and has achieved promising results, e.g., image super-resolution. Super-resolution image reconstruction aims at reconstructing a high-resolution image from a low-resolution image through restoring image detail information. CS image reconstruction is not exactly the same as the super-resolution problem, thus the existing super-resolution network cannot be directly applied to CS reconstruction task. Inspired by RED [26] and VDSR [27] networks, this paper proposes pre-reconstruction and residual reconstruction networks for CS reconstruction (please refer to Figure 4). In the pre-reconstructed network, the fused measurements are reconstructed by a de-convolution network preliminarily. Then the pre-reconstructed images are optimized and reconstructed to high-quality images in the residual reconstruction network. Next, we shall introduce these two networks in detail.

In essence, the image reconstruction is similar to the decoding process of autoencoder. In the left part of Figure 4, the pre-reconstructed network is a five-layer de-convolution neural network, which is completely symmetrical with a branch of two-branch convolution sensing network in the network structure. Unlike the existing sensing process, the convolution layer and the pooling layer are replaced by the de-convolution layer and unpooling layer, respectively. De-convolution is the inverse of convolution, and also known as transposition convolution. The convolution output is upsampling to the original resolution of image by de-convolution. Unpooling refers to the inverse process of maximum pooling. A set of conversion variables is used to record the position index of the maximum value during the pooling process. Then, the maximum value of the previous layer is placed in the original position according to the conversion variable, thereby protecting marginally the original structure. De-convolution is a learnable process, and its parameters can be optimized through training. However, unpooling process has no parameters to learn. The unpooling operation can only restore the position of maximum value as much as possible, thus the quality of the image will inevitably be affected. For the given measurement $\mathbf{y}$, one can obtain a preliminary reconstructed image $\mathbf{x}$ by the pre-reconstructed network. Mathematically,

the preliminary reconstructed process can be expressed as:

$$\mathbf{x}^{\mathrm{B}} = f^{\mathrm{B}}(\mathbf{y}, \mathbf{W}^{\mathrm{B}}), \tag{6}$$

where $f^{\mathrm{B}}(\cdot)$ denotes the pre-reconstructed network, $\mathbf{W}^{\mathrm{B}}$ represents the all parameters of pre-reconstructed network. The main contributions of pre-reconstructed network are reflected in two aspects: complete the preliminary reconstruction of the image; and restore the original dimension of image from M-dimension measurements.

It is found from experiments that the quality of this pre-reconstructed image is far from the original image. Furthermore, inspired by the residual learning proposed by ResNet [28], the residual block (Res-block) is introduced to enhance the reconstruction performance. To further reduce the gap between the pre-reconstructed image $\mathbf{x}^{\mathrm{B}}$ and reconstructed image $\mathbf{x}^{\mathrm{C}}$, researchers proposed residual learning to estimate the error between two images. Most traditional algorithms directly optimize the underlying mapping between the pre-reconstructed image $\mathbf{x}^{\mathrm{B}}$ and reconstructed image $\mathbf{x}^{\mathrm{C}}$,

$$\mathbf{x}^{C} = h(\mathbf{x}^{B}, \mathbf{W}^{h}), \tag{7}$$

where $h(\cdot)$ denotes the underlying mapping between $\mathbf{x}^{\mathrm{B}}$ and $\mathbf{x}^{\mathrm{C}}$, $\mathbf{W}^{\mathrm{h}}$ stands for the parameters of the underlying mapping. However, optimizing directly $h(\cdot)$ as an identity mapping is a very challenging task in practice. In addition, the pre-reconstructed image $\mathbf{x}^{\mathrm{B}}$ is similar to the reconstructed image $\mathbf{x}^{\mathrm{C}}$, the residuals between $\mathbf{x}^{\mathrm{B}}$ and $\mathbf{x}^{\mathrm{C}}$ would be closed to zero. Compared with the underlying mapping $h(\cdot)$, the residual mapping $\mathbf{r}$ is easier to converge during the training. Thus, the residual mapping $\mathbf{r}$ can be expressed in mathematical formulas as:

$$\mathbf{r} = \mathbf{x}^{\mathrm{C}} - \mathbf{x}^{\mathrm{B}} = h(\mathbf{x}^{\mathrm{B}}, \mathbf{W}^{\mathrm{h}}) - \mathbf{x}^{\mathrm{B}}. \tag{8}$$

Particularly, the residual reconstructed network takes the pre-reconstructed image $\mathbf{x}^{\mathrm{B}}$ as the input and generates the estimated residual mapping $\mathbf{r}$. Then, the residual mapping $\mathbf{r}$ can be defined as:

$$\mathbf{r} = f^{\mathrm{C}}(\mathbf{x}^{\mathrm{B}}, \mathbf{W}^{\mathrm{C}}), \tag{9}$$

where $f^{\mathrm{C}}(\cdot)$ represents the residual reconstructed network, $\mathbf{W}^{\mathrm{C}}$ denotes the all parameters of the residual network. By replacing the $\mathbf{x}^{\mathrm{B}}$ and $\mathbf{r}$ with Eqs. (7) and (8), the reconstructed image $\mathbf{x}^{\mathrm{C}}$ also can be described as:

$$\mathbf{x}^{\mathrm{C}} = \mathbf{x}^{\mathrm{B}} + \mathbf{r}. \tag{10}$$

As illustrated in the left part of Figure 4, the final reconstruction result is added by the output of the residual learning and the linear maping. By replacing the $\mathbf{x}^{\mathrm{B}}$ and $\mathbf{r}$ with Eqs. (6) and (9), the final reconstructed image $\mathbf{x}^{\mathrm{C}}$ can be obtained by the pre-reconstructed network and reconstructed network. Hence,

$$\mathbf{x}^{\mathrm{C}} = f^{\mathrm{B}}(\mathbf{y}, \mathbf{W}^{\mathrm{B}}) + f^{\mathrm{C}}(f^{\mathrm{B}}(\mathbf{y}, \mathbf{W}^{\mathrm{B}}), \mathbf{W}^{\mathrm{C}}). \tag{11}$$

C. IMPLEMENTATION PROCESS

Given the input image $\mathbf{x} \in \mathbb{R}^{N \times N}$, the goal of the two-branch convolution sensing network is to obtain the highly compressed M-dimension measurements. For the image with size 256×256, M is various with measurement rates (MR), e.g., M=16384, 6553, 2621, and 655 corresponding to MR=0.25, 0.10, 0.04, and 0.01, respectively. In the CNN Branch I, the first and third convolution layers use filter with size $3 \times 3 \times 32$ and generate 32 feature maps. However, the first and third convolution layers use filter of size $7 \times 7 \times 32$ and generates 32 feature maps in the CNN Branch II. All pooling layers adopt the maximum pooling filter with stride 2 and size2 $\times$ 2, and the appropriate zero padding is used to keep the feature map of the two branches consistent the same size.

As depicted in Figure 4, one can get the preliminary reconstructed image through the pre-reconstruction network $f^{\mathrm{B}}(\cdot)$. Then the residual reconstruction network $f^{\mathrm{C}}(\cdot)$ takes the pre-reconstructed image as input and generates an image with size 256×256. The residual reconstruction network consists of three residual blocks, each of which contains three CNN layers. In each Res-block, the first CNN layer employs filter with size 3×3 and obtains 64 feature maps; the second CNN layer generates 32 feature maps with 7×7 filter; and the third layer produces the output by 3*3 filter. In addition, the ReLU activation function is added to all other convolution layers except the final convolution layer. For ensuring the same size of all feature maps, it needs to add the corresponding zero padding in each layer.

III. EXPERIMENTS

In this section, a series of comparative experiments are performed to evaluate the proposed TCR-CS method. We shall introduce the experimental setup, loss function, and performance evaluation metrics, and conduct a detailed comparative analysis from the aspects of reconstruction quality, structure and vision.

A. PARAMETER SETTINGS AND TRAINING DETAILS

In ImageNet Val dataset [29], the central 256×256 part of images is cropped for making a training set. The cropped blocks are turned into the grayscale and are augmented by operations such as vertical or horizontal flipping, rotating, etc. Finally, a dataset containing 50000 images is used to train. For benchmark, a set of images with the same size, which is widely used for benchmark in other work (e.g., [16], [18], [21]–[23]), is used for testing (please refer to Figure 5). However, it is worth noting that the training dataset and testing images are strictly separated during the training process.

The full image convolution sensing module and the image reconstruction module are jointly trained to strengthen the connection. Two-branch convolution sensing, deconvolution pre-reconstruction and residual reconstruction networks form an end-to-end network $f(\cdot)$, thus the parameters $\mathbf{W}^{\mathrm{A}}$, $\mathbf{W}^{\mathrm{B}}$ and $\mathbf{W}^{\mathrm{C}}$ can be jointly optimized, which do not care about

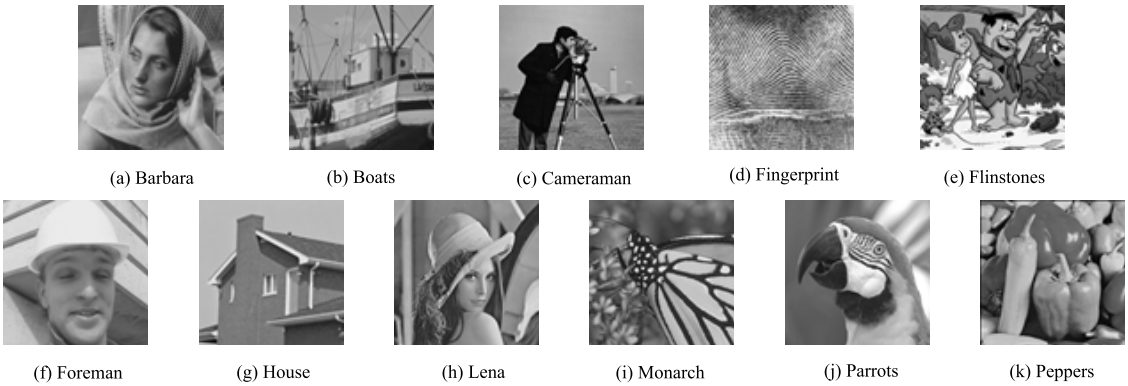


FIGURE 5. The testing images.

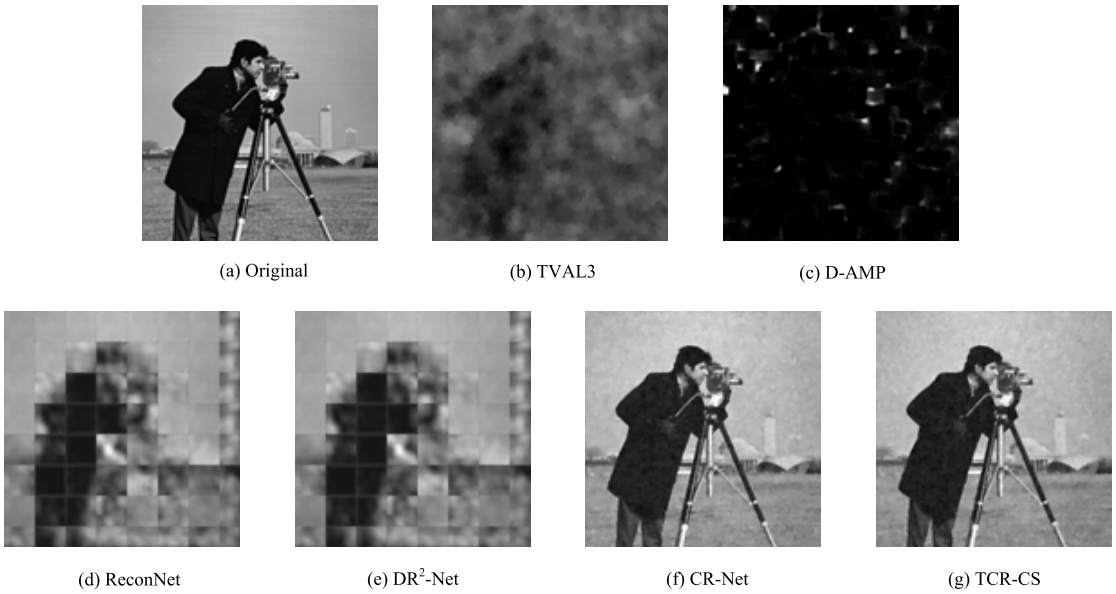


FIGURE 6. The reconstruction results of cameraman at MR=0.01.

the compression measurements $\mathbf{y}$. Like most of deep learning based image restoration methods, the mean square error is also adopted as the cost function of our network. The loss function is described as:

$$L(\Theta) = \frac{1}{T}\sum_{i}^{T} ||f(\mathbf{x}_i, \Theta) - \mathbf{x}_i||_2^2, \tag{12}$$

where Θ denotes all parameters of $f(\cdot)$, T represents the number of a batch of images, $\mathbf{x}_i$ stands for the original image. The image sensing module and image reconstruction module are trained together, but they can be run separately. The adaptive moment estimation (Adam) [30] is used to minimize and update parameters during training. Parameters like learning rate, stride and batch size are set as 0.0001, 3 and 128, respectively. The proposed network is trained with measurement rate 0.01, 0.04, 0.1 and 0.25, corresponding to M=655, M=2621, M=6553 and M=16384, respectively.

B. PERFORMANCE INDICATORS

Peak signal to noise ratio (PSNR) is an objective evaluation standard of images based on the pixel point error. Then, the PSNR is defined as:

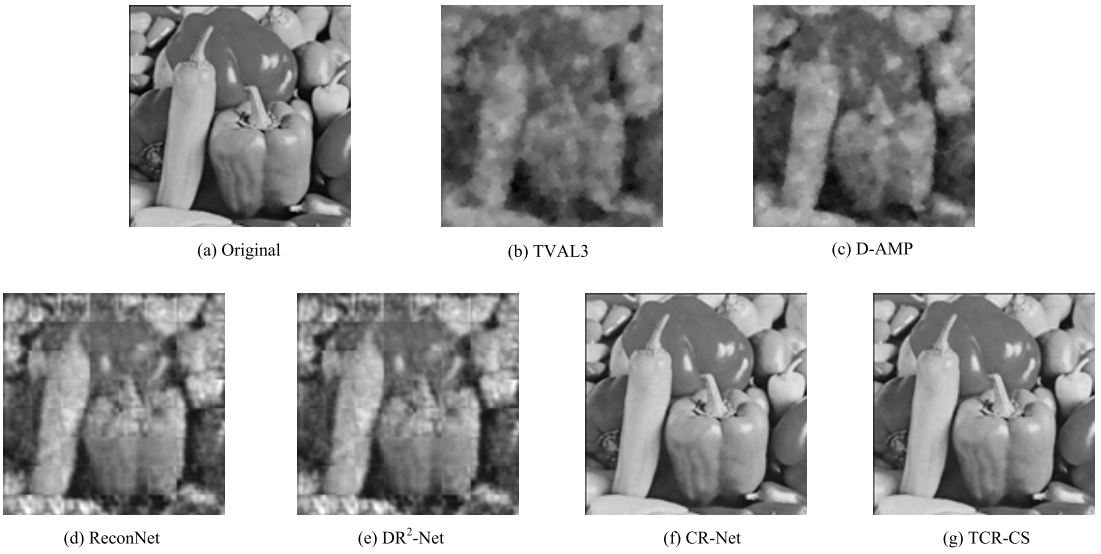
$$\text{PSNR (dB)}=10\log_{10}\frac{peak^2}{MSE}, \tag{13}$$

where *peak* is the maximum pixel value of the image, $MSE = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$ denotes the mean square error between the original image and reconstructed image.

Since people are more sensitive to the structural information of the image, the contrast information of the structure can be used to evaluate the quality of the reconstructed image. Structural similarity index (SSIM) is introduced to evaluate the structural performance of image reconstruction in the experiments. The SSIM, as an indicator to evaluate the structural similarity of two images, calculates the similarity from three aspects: brightness, contrast, and structure.

For a reconstructed image Y and an input image X, the SSIM is defined as:

$$\text{SSIM} = \frac{(2u_X u_Y + c_1)(2\sigma_X\sigma_Y + c_2)}{(u_X^2 + u_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}, \tag{14}$$



(a) Original (b) TVAL3 (c) D-AMP

(d) ReconNet (e) DR^2-Net (f) CR-Net (g) TCR-CS

FIGURE 7. The reconstruction results of Peppers at MR=0.04.

where u_X, u_Y denote the mean of the images X and Y, respectively; σ_X, σ_Y represent the standard deviation of the images X and Y, respectively; σ_{XY} stands for the covariance between the images X and Y; c_1, c_2 are constants.

C. COMPARISON WITH STATE-OF-THE-ART METHODS

The proposed TCR-CS method is compared with five existing methods, i.e., CR-Net [16], ReconNet [18], DR^2-Net [21], TVAL3 [31], and D-AMP [32]. TVAL3 and D-AMP are classified as the iterative-based methods, and the other three methods are deep learning-based methods. CR-Net is a CNN-based CS approach, which measures the input image block by block and reconstructs the full image. TVAL3 and D-AMP methods use very large measurement matrices for sensing the whole image. For an image with size 256×256, the dimensionality of measurement matrix is 16384×65536 when measurement rate at 0.25, a lot of memory is required to store it. The use of such large measurement matrices can greatly improve the quality of reconstruction in the algorithm, but it also will lead to high computational complexity in the iterative process. In addition, it should also be noted that the well-known BM3D denoising method [33] is used in the iterative process of D-AMP method.

To fully understand the performance of TCR-CS, the testing images are reconstructed at measurement rate 0.01, 0.04, 0.1, and 0.25, respectively. Table 1 lists the the PSNR results on the testing images under TVAL3, D-AMP, ReconNet, DR2-Net, CR-Net and the proposed TCR-CS method. When compared with other competing methods, it is obvious that the proposed TCR-CS method gets the better PSNR values at each measurement rate, which demonstrates the effectiveness of TCR-CS. Compared to the iterative-based methods, deep learning based methods achieve excellent reconstruction performance for the CS measurement at the low measurement rates. However, the performance of existing deep learning methods is worse than that of TCR-CS at all measurement rates. The proposed TCR-CS method is 0.3 to 1dB higher than the CR-Net method for part sample images. At MR=0.01, the TCR-CS outperforms ReconNet, DR2-Net and CR- Net by 4.96, 4.83 and 0.61dB, respectively.

TABLE 1. The PSNR results on the testing images at different measurement rates (dB).

MR	Image Name	TVAL3	D-AMP	ReconNet	DR2-Net	CR-Net	TCR-CS
0.01	Monarch	10.36	4.56	15.39	15.33	18.97	**19.76**
	Parrots	11.69	5.83	17.63	18.01	22.68	**23.05**
	Barbara	11.94	6.04	18.61	18.65	22.13	**22.76**
	Cameraman	11.56	5.24	17.11	17.08	20.92	**21.81**
	House	12.35	6.61	19.31	19.61	24.71	**25.13**
	Lena	11.87	5.96	17.87	17.97	23.19	**23.62**
	Mean PSNR	11.68	5.85	17.77	17.9	22.12	**22.73**
0.04	Monarch	17.26	14.49	18.19	18.93	24.69	**25.13**
	Parrots	18.04	15.93	20.27	21.16	25.5	**26.23**
	Barbara	18.98	16.37	20.38	20.7	23.76	**24.58**
	Cameraman	18.58	15.86	19.26	19.84	23.78	**24.73**
	House	19.78	17.23	22.58	23.92	30.29	**30.76**
	Lena	19.46	16.82	21.28	22.13	26.67	**27.15**
	Mean PSNR	18.69	15.97	20.47	21.26	25.97	**26.67**
0.1	Monarch	22.86	20.23	21.1	23.1	29.24	**29.84**
	Parrots	23.1	21.71	22.63	23.94	28.75	**29.44**
	Barbara	22.88	20.68	21.89	22.69	24.5	**25.68**
	Cameraman	23.12	20.84	21.28	22.46	26.42	**27.52**
	House	24.86	23.58	26.69	27.53	33.42	**33.84**
	Lena	24.16	22.47	23.83	25.39	29.71	**30.51**
	Mean PSNR	23.78	21.82	23.08	24.38	28.94	**29.73**
0.25	Monarch	26.33	26.13	24.31	27.95	34.36	**34.63**
	Parrots	27.91	27.51	25.59	28.73	34.01	**34.54**
	Barbara	26.54	25.96	23.25	25.77	29.16	**30.13**
	Cameraman	25.63	24.59	23.15	25.62	30.4	**30.94**
	House	30.12	29.53	28.46	31.83	37.45	**37.63**
	Lena	28.67	27.63	26.54	29.42	34.42	**34.83**
	Mean PSNR	27.86	27.26	25.51	28.49	33.57	**34.06**

To verify the rationality of two-branch structure, two variants of TCR-CS (denoted as TCR-CS-I and TCR-CS-II) are

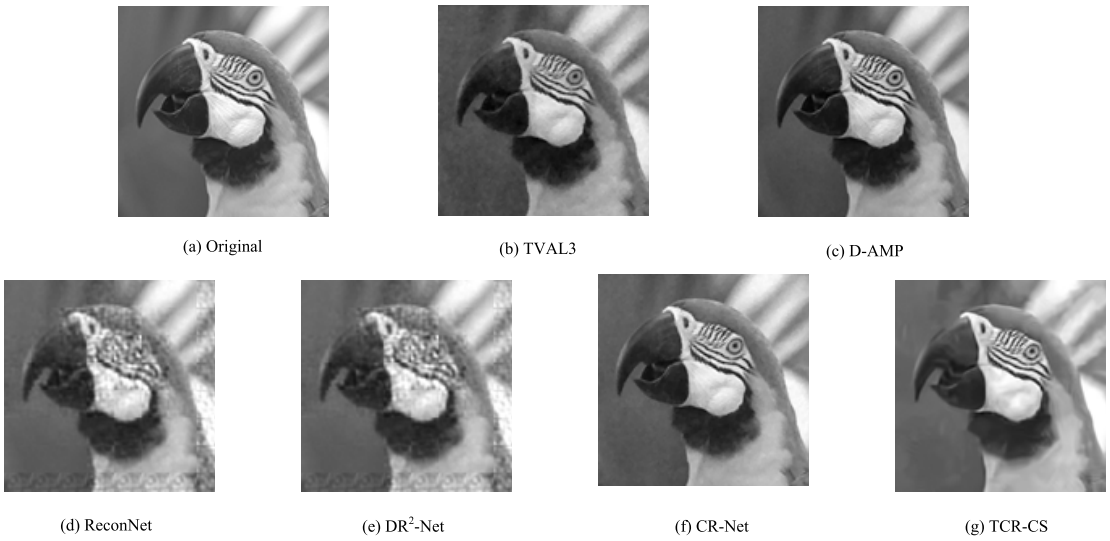


FIGURE 8. The reconstruction results of Peppers at MR=0.1.

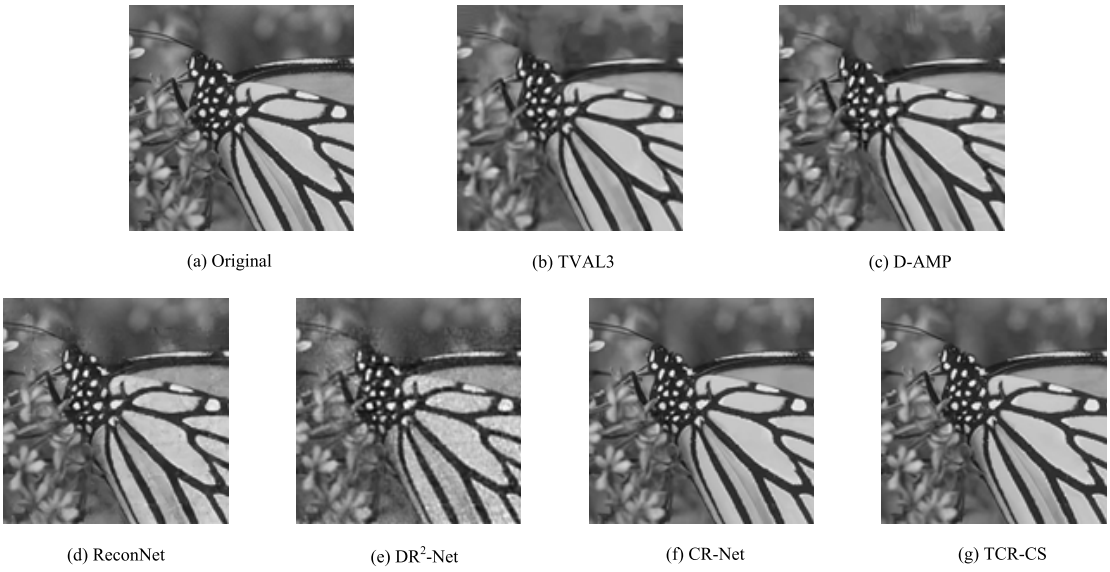


FIGURE 9. The reconstruction results of Peppers at MR=0.25.

also implemented. TCR-CS-I only uses the first branch of the sensing network in Figure 2 to obtain measurements, and TCR-CS-II only uses the second branch. Table 2 shows the comparison of TCR-CS and its variants at different measurement rates. CR-Net is the method to obtain the best reconstruction quality among all the contrast algorithms in this paper. Compared with the CR-Net method, both TCR-CS-I and TCR-CS-II achieve the higher PSNR values at all measurement rates, which reveals that our proposed method can improve the reconstruction performance of CS. However, TCR-CS based on two-branch sensing network is significantly better than TCR-CS-I and TCR-CS-II, which strongly proves the effectiveness of two-branch structure.

In addition, structural similarity index is used to evaluate structural differences between our method and others. Here, take the case of MR = 0.01 as an example in Table 3. Figures 6-9 show the reconstruction results of testing images at measurement rates 0.01, 0.04, 0.1 and 0.25, respectively. Clearly, the visual quality of the images reconstructed by

TABLE 2. The PSNR results of TCR-CS and its variants at different measurement rates (dB).

MR	Image Name	Cameraman	Monarch	Parrots	Barbara	House	Lena	Mean PSNR
0.01	CR-Net	20.92	18.97	22.68	22.13	24.71	23.19	22.12
	TCR-CS-I	21.35	19.46	22.88	22.48	24.89	23.43	22.48
	TCR-CS-II	21.16	19.18	22.71	22.34	24.76	23.29	22.31
	TCR-CS	**21.68**	**19.76**	**23.05**	**22.76**	**25.13**	**23.62**	**22.73**
0.04	CR-Net	23.78	24.69	25.5	23.76	30.29	26.67	25.97
	TCR-CS-I	24.31	24.93	25.95	24.13	30.52	26.91	26.38
	TCR-CS-II	24.04	24.81	25.73	23.94	30.34	26.81	26.16
	TCR-CS	**24.73**	**25.13**	**26.23**	**24.58**	**30.76**	**27.15**	**26.67**
0.1	CR-Net	26.42	29.24	28.75	24.5	33.42	29.71	28.94
	TCR-CS-I	27.08	29.61	29.24	25.12	33.71	30.23	29.45
	TCR-CS-II	26.86	29.45	28.97	24.86	33.58	30.03	29.23
	TCR-CS	**27.52**	**29.84**	**29.41**	**25.68**	**33.84**	**30.51**	**29.73**
0.01	CR-Net	30.4	34.36	34.01	29.16	37.45	34.42	33.57
	TCR-CS-I	30.86	34.56	34.31	29.86	37.56	34.72	33.94
	TCR-CS-II	30.56	34.38	34.13	29.54	37.48	34.51	33.72
	TCR-CS	**30.94**	**34.69**	**34.54**	**30.13**	**37.63**	**34.83**	**34.06**

TABLE 3. The SSIM results at measurement rate 0.01 (dB).

Samples	Original	TVAL3	D-AMP	ReconNet	DR2-Net	CR-Net	TCR-CS
Monarch	1	0.2971	0.0262	0.3801	0.3931	0.5516	0.6053
Parrots	1	0.4535	0.0329	0.5328	0.5617	0.673	0.6901
Barbara	1	0.3368	0.0252	0.3729	0.3847	0.5376	0.7475
Cameraman	1	0.3957	0.0366	0.4516	0.4783	0.7188	0.7332
House	1	0.4804	0.0268	0.5278	0.5526	0.9517	0.952
Lena	1	0.4012	0.0315	0.4418	0.4552	0.9683	0.9705
Mean	1	0.3735	0.0286	0.4083	0.4291	0.7023	0.7535

TCR-CS is significantly better than TVAL3, D-AMP, ReconNet and DR2-Net. ReconNet and DR2-Net have serious block effects at the low measurement rate, and the reconstructed image is even hard to understand the content. From Figure 6, it is difficult to distinguish the difference between the proposed TCR-CS method and CR-Net in the visual quality. But Table 3 displays that the proposed TCR-CS method has a higher SSIM value than CR-Net, which indicates that the structure information of the reconstructed images are better preserved.

IV. CONCLUSION

In this paper, a two-branch convolution residual network for image compressive sensing (denoted as TCR-CS) has been proposed. The proposed TCR-CS network consists of two-branch convolution sensing, pre-reconstruction and residual reconstruction. Unlike existing BCS and convolutional CS, the whole image is sensed by two convolution networks with different scale filters in two-branch convolution sensing network. In the de-convolution decoder network, the measurements are preliminarily reconstructed to obtain pre-reconstructed images. Then the pre-reconstructed images are optimized and reconstructed to high-quality images in the residual reconstruction network. Through end-to-end training, all networks can be jointly optimized. Extensive experiments have shown that TCR-CS is superior to the existing iterative-based and deep learning-based methods in the aspects of structural similarity, reconstruction performance and visual quality at different measurement rates.

REFERENCES

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. J. Candès, "Compressive sampling," in *Proc. Int. Congr. Math.*, Madrid, Spain, Aug. 2006, pp. 1433–1452.

[3] P. Chen, C. Qi, L. Wu, and X. Wang, "Estimation of extended targets based on compressed sensing in cognitive radar system," *IEEE Trans. Veh. Technol*, vol. 66, no. 2, pp. 941–951, Feb. 2017.

[4] F. Magalhães, F. M. Araújo, M. V. Correia, M. Abolbashari, and F. Farahi, "Active illumination single-pixel camera based on compressive sensing," *Appl. Opt.*, vol. 50, no. 4, pp. 405–414, Feb. 2011.

[5] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded strobing photography: Compressive sensing of high speed periodic videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 671–686, Apr. 2011.

[6] S. Dikmese, Z. Ilyas, P. C. Sofotasios, M. Renfors, and M. Valkama, "Sparse frequency domain spectrum sensing and sharing based on cyclic prefix autocorrelation," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 1, pp. 159–172, Jan. 2017.

[7] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018.

[8] L. Gan, "Block compressed sensing of natural images," in *Proc. Int. Conf. Digit. Signal Process.*, Cardiff, U.K., Jul. 2007, pp. 403–406.

[9] T. Song, H. Li, F. Meng, Q. Wu, and J. Cai, "LETRIST: Locally encoded transform feature histogram for rotation-invariant texture classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 7, pp. 1565–1579, Jul. 2018.

[10] Z. Zhang, Y. Zou, and C. Gan, "Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression," *Neurocomputing*, vol. 275, pp. 1407–1415, Jan. 2018.

[11] C. Gan, L. Wang, Z. Zhang, and Z. Wang, "Sparse attention based separable dilated convolutional neural network for targeted sentiment analysis," *Knowl.-Based Syst.*, to be published, doi: 10.1016/j.knosys.2019.06.035.

[12] P. Zhang, X. Kang, D. Wu, and R. Wang, "High-accuracy entity state prediction method based on deep belief network toward IoT search," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 492–495, Apr. 2019.

[13] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A feature-based learning system for Internet of Things applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928–1937, Apr. 2019.

[14] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 469–478, Sep. 2019.

[15] A. Mousavi, A. Patel, and R. G. Baraniuk, "A deep Learning approach to structured signal recovery," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Aug. 2015, pp. 1336–1343.

[16] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[17] Z. Zhang, Y. Wu, C. Gan, and Q. Zhu, "The optimally designed autoencoder network for compressed sensing," *EURASIP J. Image Vide.*, vol. 56, no. 1, pp. 1–12, Apr. 2019.

[18] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed random measurements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 449–458.

[19] S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, and A. Ashok, "Convolutional neural networks for non-iterative reconstruction of compressively sensed images," *IEEE Trans. Comput. Imag.*, vol. 4, no. 3, pp. 34–326, Sep. 2018.

[20] A. Mousavi and R. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, USA, Mar. 2017, pp. 2272–2276.

[21] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, "DR^2-Net: Deep residual reconstruction network for image compressive sensing," *Neurocomputing*, vol. 359, no. 1, pp. 483–493, Sep. 2019.

[22] A. Mousavi, G. Dasarathy, and R. G. Baraniuk, "DeepCodec: Adaptive sensing and recovery via deep convolutional neural networks," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2017, pp. 1–17.

[23] X. Xie, Y. Wang, G. Shi, C. Wang, and X. Han, "Adaptive measurement network for cs image reconstruction," in *Proc. CCF Chin. Conf. Comput. Vis.*, Tianjin, China, Oct. 2017, pp. 407–417.

[24] X. Xie, J. Du, C. Wang, G. Shi, X. Xu, and Y. Wang, "Fully-convolutional measurement network for compressive sensing image reconstruction," *Neurocomputing*, vol. 328, no. 1, pp. 105–112, Feb. 2019.

[25] X. Xie, C. Wang, J. Du, and G. Shi, "Full image recover for block-based compressive sensing," in *Proc. IEEE Int. Conf. Multimedia Expo*, San Diego, CA, USA, Jul. 2018, pp. 1–6.

[26] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," 2016, *arXiv:1606.08921*. [Online]. Available: https://arxiv.org/abs/1606.08921

[27] J. Kim, J. Lee, and K. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 1646–1654.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 20–25.

[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–9.

[31] C. Li, W. Win, H. Jing, and Y. Zhang, "An efficient augmented Lagrangian method with applications to total variation minimization," *Comput. Optim. Appl.*, vol. 56, no. 3, pp. 507–530, Dec. 2013.
[32] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.
[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

CHENQUAN GAN received the M.Sc. and Ph.D. degrees from the Department of Computer Science, Chongqing University, in 2012 and 2015, respectively. He is currently an Associate Professor with the School of Communication and Information Engineering, Chongqing University of Post and Telecommunications (CQUPT), Chongqing, China. His research interests include difference equations, computer virus propagation dynamics, and deep learning.

XIAOQIN YAN received the B.Sc. degree from Nanjing Forestry University, in 2017. She is currently pursuing the master's degree in electronics and communication engineering with the Chongqing University of Post and Telecommunications, Chongqing, China. Her research interests include signal detection and deep learning techniques.

YUNFENG WU received the B.Sc. and M.Sc. degrees from the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, in 2016 and 2019, respectively. His research interests include signal processing and deep learning techniques.

ZUFAN ZHANG received the B.Eng. and M.Eng. degrees from CQUPT, in 1995 and 2000, respectively, and the Ph.D. degree in communications and information systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2007. He was a Visiting Professor with the Centre for Wireless Communications (CWC), Oulu of University, Finland, from February 2011 to January 2012. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Post and Telecommunications (CQUPT), Chongqing, China. His current main research interest concerns wireless and mobile communication networks.