IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 August 2022, accepted 22 August 2022, date of publication 14 September 2022, date of current version 22 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206016

RESEARCH ARTICLE

Tree-Structured Dilated Convolutional Networks for Image Compressed Sensing

RUI LU AND KUNTAO YE
School of Science, Jiangxi University of Science and Technology, Ganzhou 341000, China
Corresponding author: Kuntao Ye (kuntaoye@126.com)

This work was supported in part by the Discipline Development Fund from School of Science, Jiangxi University of Science and Technology.

ABSTRACT To better recover a sparse image signal carrying redundant information from many fewer measurements than the Nyquist-Shannon sampling theorem suggested, convolutional neural networks (CNNs) can be used to emulate a compressed sensing (CS) process. However, the existing CS methods based on CNNs have the problems of high computational complexity and unsatisfactory reconstruction effect. This study aims to present a faster algorithm based on CNNs to obtain reconstructed images with finer texture details from CS measurements. A tree-structured dilated conventional network (TDCN) for image CS is proposed. To extract the image multi-scale features as much as possible for better image reconstruction, the TDCN combines tree-structured residual blocks made of three dilation convolution layers with different dilation factors; the output of each dilated convolution layer is directed to fusion layer to eliminate information loss due to the multiple cascading dilated convolutions. Moreover, L1 loss is employed as an objective optimization function instead of L2 loss to improve training results of the network and achieve better convergence. Extensive CS experiments in the study demonstrate that the proposed TDCN outperforms existing state-of-the-art methods in terms of both PSNR and SSIM at different sampling rates while maintaining a fast computational speed. Our code and the trained model are available at https://github.com/UHADS/TDCN.

INDEX TERMS Compressed sensing, deep learning, dilated convolution, image reconstruction.

I. INTRODUCTION

Compressed sensing (CS) theory [1], [2] shows that the reconstruction of a sparse signal $x \in \mathbb{R}^{N\times 1}$ can be accurately achieved from its compressed measurements $y \in \mathbb{R}^{M\times 1}$ by solving underdetermined equations. The compressed measurements $y = \Phi x$ are obtained through measurement matrix $\Phi \in \mathbb{R}^{M\times N}$. Compared with the Nyquist-Shannon sampling theorem [3], the CS theory suggests that a sparse signal can be recovered from many fewer measurements based on the sparsity of the signal.

In addition, the CS process can be seen as a process of random subsampling signals. Algorithm can eliminate the artifacts caused by random subsampling; thus, the original signal can be accurately recovered. As a result, the subsampling of CS reduces the demand for high transmission bandwidth and storage space. Due to its simultaneous sampling and compression at the same time, CS also offers low-cost on-sensor data compression [4]. CS has been applied in a variety of realities, including but not limited to radar image acquisition [5], [6], [7], novel imaging devices [8], [9], magnetic resonance imaging (MRI) [10], [11], and wireless telemonitoring [12].

It is well known that information carried by images is redundant and can be sparsely represented in sparse domains. Therefore, the image can be compressed and reconstructed accurately according to CS theory. The goal of image CS is to ensure that an image can be accurately reconstructed from very few measurements. There are two main challenges that need to be solved to meet the goal. These include selecting the sampling matrix and designing appropriate reconstruction methods.

Most studies [13], [14], [15], [16] select a random matrix, binary matrix, and structure matrix as the sampling matrix. However, these sampling matrices are image-independent

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

and ignore image features. To make full use of the features of the image and design a sampling matrix with high correlation with the image to achieve high-quality results, a convolutional layer has been proposed in CSNet [17] to simulate the CS sampling process and adaptively learn the sampling matrix from the training images.

For the design of the reconstruction method, some studies [18], [19], [20], [21], [22] use iterative operations to achieve high-quality results, but these operations often take a long time to reconstruct an image. Therefore, research [23] adopts the block-by-block reconstruction method to recover images in blocks then joins the recovered blocks together to obtain a final restored image, which can significantly reduce the reconstruction time and memory storage requirements. Based on traditional block-by-block CS methods, convolutional neural networks (CNNs) have been developed to recover CS sampling data in blocks, which not only improves the quality of reconstructed images, but also saves time of recovery [24], [4]. However, these methods of using CNNs ignore the linkages between blocks and only rely on intra-block information to recover the image, and blocking artifacts will appear. Generally, post-processing methods are required to eliminate blocking artifacts in the block-based method. Such post-processing usually increases computational complexity and affects recovery efficiency.

To address these problems, researchers [17] have proposed a linear preliminary reconstruction network combined with a nonlinear deep reconstruction network, which reconstructs images from CS measurements by end-to-end learning without additional post-processing operations and can quickly and efficiently obtain high-quality recovered images.

To enhance the quality of the reconstructed images, MR-CSGAN [25] increases the depth of the recovery network and uses multi-scale residual blocks (MSRB) made of convolution kernels of different sizes to exploit the multi-scale structural features of the images in the generator network. Although MR-CSGAN exploits image structural features through MSRB to obtain more detailed reconstructed images, the high computational cost limits its practical application.

Exploring the deeper structural features of the original image while reducing the time of recovery has become an urgent challenge in image CS. Recently, the use of dilated convolution (DConv) rather than standard convolution to extract image features in CNN has become mainstream [26]. Compared with ordinary convolution, dilated convolution for image recovery tasks can significantly expand the receptive field under the same computing conditions [27].

Therefore, in the field of image segmentation, TKCN [28] has been proposed a tree-structured feature aggregation (TFA) module composed of a tree structure of dilated convolution and a batch normalization (BN) layer to better extract multi-scale features of images in complex scenes. However, research [29] shows that the BN layer can eliminate network range flexibility by standardizing features; therefore, the BN layer is harmful if used for image reconstruction.

Based on the above research results, we propose an image CS model made of a tree-structured dilated convolutional network (TDCN), in which a tree-structured residual block (TSRB) is newly designed to learn features of different scales of images. The TSRB module comes from removing the BN layer of the TFA module in the TKCN and selecting dilation factors that are better suited for the dilated convolution.

The TDCN consists of a sampling network and a reconstruction network. The sampling network adopts the same network structure as in CSNet [17], and can obtain CS measurements through a sampling matrix that is trained adaptively from training datasets. The reconstruction network, which is established to learn end-to-end mapping from CS measurements to reconstructed images, contains a linear preliminary reconstruction network and a nonlinear deep reconstruction network. The preliminary reconstruction network results in a preliminary recovery image through a deconvolutional layer, whereas the deep reconstruction network uses several TSRB modules to refine the preliminary reconstruction image further and obtain better recovery quality.

In addition, instead of the mean square error (MSE) or L2 loss, the mean absolute error (MAE) or L1 loss is used as a loss function in the image reconstruction network because the literature [29] suggests that L1 loss can potentially help achieve better training results on many occasions.

The experimental results show that the proposed TDCN can achieve higher PSNR and SSIM values than most existing methods in image CS because of the following contributions.

1) We propose a tree-structured convolutional network (TDCN) for image CS. TDCN uses multiple TSRB modules to learn multi-scale features and then combines the outputs of each TSRB module through a feature fusion layer to guarantee high-quality recovery images.
2) To quickly obtain recovery images from CS measurements, we introduce dilated convolution to the TSRB modules and dilated convolutions in TSRB made as a tree structure. Therefore, TSRB can easily obtain multi-scale features of images and ensure that the extracted shallow information is not lost in the deep network.
3) We use the L1 loss function in TDCN instead of the L2 loss function. Experiments show that L1 loss results in recovered images with more detail and better visual effects while achieving better convergence.

The remainder of this paper is organized as follows. Section II introduces the background of the model. Section III introduces the proposed TDCN method. In Section IV, the performance of TDCN is discussed and compared with that of some state-of-the-art methods through experiments, and we conclude the paper in Section V.

II. BACKGROUND

A. CNNS FOR IMAGE COMPRESSED SENSING RECONSTRUCTION

At present, many image CS models based on deep convolutional neural networks (DCNNs) have shown good performance on several benchmark test sets [24], [30], [31], [32]. For example, ReconNet [24] proposed by Kulkarni *et al.*, where a non-iterative reconstruction architecture based on a CNN is placed after a random Gaussian sampling matrix to achieve non-iterative image CS reconstruction, provides a good trade-off between computational complexity and reconstruction quality; ISTA-Net [4], which uses a deep network to replace the iterative threshold algorithm (ISTA [33]) in the reconstruction process, improves both the quality and speed of image reconstruction; CSNet [17] uses a convolution layer to complete the sampling and reconstruction processes simultaneously; MR-CSGAN [25] uses perceptual loss as a loss function and several MSRB modules to exploit multi-scale structural features of the images in the generator network, and then all the outputs of each MSRB are integrated through a fusion layer, as shown in FIGURE 1.

B. DILATED CONVOLUTION

To increase the receptive field, most semantic segmentation algorithms contain a pooling layer and convolutional layer [28]. Thus, the resolution of the feature maps is reduced, and up-sampling is required to restore the image resolution. Because of the downsampling and upsampling activities in the process, there is a loss of accuracy. Dilated convolution [26] has been proposed to solve this problem. With the size of the feature map unchanged, using a dilated convolution operation to replace the downsampling and upsampling operations can increase the receptive field.

Dilated convolution has been also introduced to CNN to solve super-resolution problems, e.g., DCBI [34] module shown in FIGURE 1 uses dilated convolutions with different dilation rates to extract image features simultaneously.

Unlike standard convolutions, dilated convolution introduces a hyper-parameter called the dilation factor m, which defines the spaces between values processed by the convolution kernel. k-dilated convolutions of size 3×3 with different dilation factors are shown in FIGURE 2. The gray areas in FIGURE 2 are receptive fields. It can be seen that dilated convolution appears in the form of a standard convolution when dilation factor is 1.

On the premise of occupying the same computing resources, increasing the dilation factor can obtain a larger receptive field of the network and detect multi-scale image features without losing resolution.

C. TREE-STRUCTURED RESIDUAL BLOCK (TSRB)

In TKCN [28], a tree-structured feature aggregation (TFA) module shown in FIGURE 1 has been proposed for image segmentation tasks. TFA is composed of several Kronecker convolution (KConv) layers and BN layers, where the

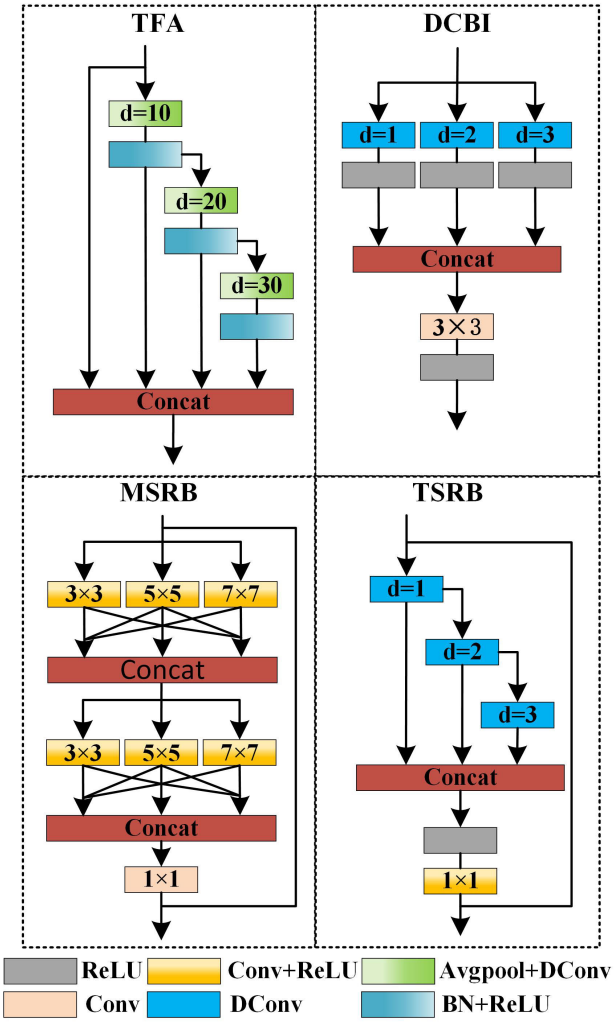

FIGURE 1. Tree-structured Feature Aggregation module (TFA), Dilated Convolution Based Inception module (DCBI), Multi-Scale Residual Blocks (MSRB) and Tree-Structured Residual Block (TSRB).

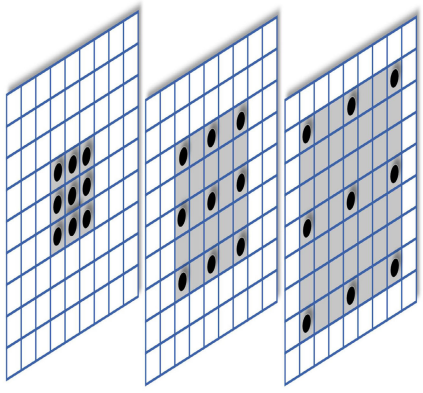

FIGURE 2. Dilation convolutions with different dilation factor k. The values of m from left to right are 1, 2, 3.

Kronecker convolution layer contains a dilated convolution layer and a layer to capture local contextual information

ignored by the dilated convolution, dubbed AvgPool. More detailed structural information is extracted using TFA.

This study presents a tree-structured residual block (TSRB) module shown in FIGURE 1. Similar to the TFA module, the TSRB module adopts a tree structure with three layers of dilated convolutions of size 3×3 of different dilation factors as shown in FIGURE 1. To compensate for the checkerboard effect caused by multiple cascade dilated convolutions and capture local contextual information ignored by dilated convolutions, the output of each dilated convolution layer is replicated into two branches: one branch retains the current scale features, while the other aggregates spatial dependencies over a larger range; therefore, in complicated circumstances, it can easily learn representations of multi-scale objects and encode hierarchical contextual information.

Meanwhile, the output feature maps of the current step are stacked with the previous feature maps in tandem, and then a convolution of size 1×1 is input for multi-scale feature fusion. Finally, the input and output are added to form a local residual block to reduce the network's loss of local contextual information; thus, AvgPool is not required for the TSRB.

The TSRB process can be described by Equations (1) and (2).

$$O_k = W_k *_k O_{k-1} + B_k, \quad k = 1, 2, 3 \tag{1}$$

$$H_j = W_c * \begin{bmatrix} O_1 \\ \vdots \\ O_k \end{bmatrix} + H_{j-1}. \tag{2}$$

where $*$ represents the convolution operation or an 1-dilated convolution operation, $*_k$ represents k-dilated convolution operation, O_k is the output of each convolutional layer in TSRB, W_k and B_k are its convolutional kernel and biases, respectively. H_j is the output of the j-th TSRB module and W_c is a convolution kernel of 1×1. During this process, $O_0 = H_0$.

Compared to MSRB shown in FIGURE 1, TSRB modules will have same size of receptive field as the traditional convolution kernels of size 5×5 or 7×7 if using 3×3 dilated convolution kernels with dilation rates of 2 or 3. Thus 25 or 49 parameters of traditional convolution kernels are replaced by 9 parameters of dilated convolution kernels, it brings down the computational complexity of TSRB compared with MSRB when receptive field kept same.

TSRB uses tree-structured dilated convolutions to extract different scale features, which increases the network depth, while DCBI applied dilated convolutions with different dilation rates to extract image features simultaneously [34]. Feature fusion is performed through 3×3 convolution kernels in DCBI while 1×1 convolution kernels are used in TSRB architecture to prevent the loss of information caused by the increase of the network depth.

Furthermore, every convolution in DCBI is followed by an activation layer, but there is only one activation layer after the feature fusion operation to effectively prevent network gradient explosion. So, the computational complexity of TSRB is much smaller than DCBI while keeping good performances.

D. MEAN ABSOLUTE ERROR LOSS FUNCTION

In a CNN-based image CS, the choice of loss function is also essential, and an appropriate loss function can help the model achieve the best and fastest convergence.

L2 loss is the most widely used loss function in image recovery and is also the main performance measure (PSNR) for these problems. However, research [29] reported that L2 loss training does not guarantee better performance in terms of PSNR and SSIM. In their experiments, L1 loss was used as a loss function, and the experiments showed that the network trained by L1 loss had better performance than that trained by L2 loss.

L2 loss function is the mean squared error (MSE) between the predicted value $f\ (x_i)$ and the target value x_i, which is defined in Equation (3):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (x_i - f\ (x_i))^2. \tag{3}$$

L1 loss function is the mean absolute error (MAE) between the predicted value $f\ (x_i)$ and the target value x_i, which is defined in Equation (4):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |x_i - f\ (x_i)|. \tag{4}$$

where N the total number of images.

III. TREE-STRUCTURED DILATED CONVOLUTIONAL NETWORKS (TDCN)

The TDCN proposed in this study imitates the image CS process, as shown in FIGURE 3. Similar to block-based compressive sensing (BCS), TDCN uses a CNN to complete three operations: compression sampling, preliminary reconstruction, and deep reconstruction. TDCN has a sampling network and a reconstruction network, where the sampling network is used to obtain CS measurements through a learning sampling matrix, and the reconstruction network is used to obtain the reconstructed images from the CS measurements. Normally, a reconstruction network consists of a preliminary reconstruction network and a deep reconstruction network. The preliminary reconstruction network is a linear operation that reconstructs images from the CS measurements initially, whereas the deep reconstruction network is a nonlinear operation that can further improve the quality of the preliminary reconstructed images.

A. SAMPLING NETWORK AND PRELIMINARY RECONSTRUCTION NETWORK

Traditional sampling matrices mostly use random matrices, such as a Gaussian matrix or Bernoulli matrix; however, the sampling matrix has no relevance to the signal. The large sampling matrix adds computational complexity to the image CS and requires a large memory space to store. An efficient block-by-block sampling network has been proposed in CSNet [17], in which the sampling network adaptively

learns the sampling matrix from the training datasets during training.

The memory required and computational complexity are both decreased because each image block-based sampling matrix is same and has a reduced dimension. And the sampling matrix resulting from training has a high correlation with the image, so a better image quality of reconstruction can be obtained from the image CS process.

In this study, the same sampling network as in CSNet [17] is used, where the input image is initially divided into non-overlapping blocks of size $B \times B \times l$; l is the number of channels and $B \times B$ represents the size of each channel of block. CS measurements are obtained through a sampling matrix Φ_{B} of size $n_B \times lB^2$, where $n_B = slB^2$ if the sampling ratio is set as s.

This process can be expressed as $y_j = \Phi_{\mathrm{B}} x_j$, where Φ_{B} can also be seen as a sampling convolution layer if each row of Φ_B is treated as a filter. Thus, the size of each filter in the sampling network is $B \times B \times l$, which is the same as the size of each image block. The stride of this convolution layer is $B \times B$ to guarantee nonoverlapping sampling. Moreover, the bias in each filter is zero.

The sampling convolution layer can be described as

$$y_{samp} = f_{samp}(x) = W_s * x. \tag{5}$$

where $*$ represents the convolution operation, y_{samp} is a $1 \times 1 \times slB^2$ matrix of the CS measurement for each image block, W_s corresponds to n_B filters of support $B \times B \times l$, x is the input image. In this process, there are all linear operations without a bias or an activation function, and each column of the output corresponds to the measurement of an image block.

According to the CS theory [2], the image can only be reconstructed from measurements under sparse conditions. We design a preliminary reconstruction network for the preliminary reconstruction from the output of the sampling layer as in MR-CSGAN. The preliminary reconstruction network consists of a deconvolution layer, described as:

$$y_{int} = f_{int}(x) = W_{int} *' y_{samp}. \tag{6}$$

where $*'$ represents the deconvolution operation and y_{int} is the preliminary reconstructed result. W_{int} corresponds to the deconvolution kernel of the support $B \times B \times l$, x is the input image. Similar to the sampling layer, the preliminary reconstruction network is a linear operation, without bias and activation functions.

B. DEEP RECONSTRUCTION NETWORK

Because the entire sampling recovery process is a linear transform, the quality of the preliminary reconstructed images is relatively poor. To improve the reconstruction quality, we add a deep reconstruction network composed of multiple residual blocks, each containing a ReLU layer, to prevent the gradient from vanishing and increase the nonlinearity of the network.

In the deep reconstruction network, we cascade multiple TSRB modules to increase the non-linearity of the network. To avoid losses of contextual information learned by TSRB, we extract the feature maps from each TSRB for fusion at layer ''Concat''. To reduce the memory cost and increase the running speed, we add two convolutional layers to reduce the output dimensions of the feature fusion layer. At the output of these two convolutional layers, the output of the first convolution layer of the deep reconstructed network is added to form a global residual network module. A feature aggregation operation is used to obtain the final output images.

The above process can be expressed as:

$$y_{out} = W_{out} * (y_{TSRB} + y_{int}) + B_{out}. \tag{7}$$

where y_{out} is the final recovered high-quality image, y_{int} is the output low-quality image of the preliminary reconstruction network, W_{out} and B_{out} correspond to the feature aggregation operation kernel and biases respectively, y_{TSRB} is the residual between quality images y_{int} and high-quality images y_{out}. The final TDCN is shown in FIGURE 3.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. IMPLEMENTATION DETAILS

For the purpose of comparison, the network parameters of TDCN are set as follows: the block size in the sampling process is the same as that of CSNet, that is, $B = 32$ and $l = 1$. We initialize the weights using the method described in [35], which is a reasonable and effective method for networks with the ReLU activation function. Training is performed by optimizing equation (4) using adaptive moment estimation (Adam) [36], and we use the default settings to initialize the other parameters of Adam.

All our experiments are conducted by training the network with a common image super-resolution dataset, DIV2K [37]. The DIV2K dataset includes 800 training images, 100 validation images, which are saved as ''. png'' file. Similar to CSNet, data augmentation technology has been applied to increase the training dataset [17]. We crop the training images with a stride of 32 to obtain a sub-image size of 96×96 pixels. We then randomly choose 96000 sub-images for network training. A total of 100 epochs are trained, and each epoch has 3,000 iterations, with a batch size of 32. We set the initial learning rate to 0.0004 and decay it to half per 10 epochs. Different sampling rates are used to measure the image. We use Set5 [30], Set11 [24], Set14 [31], and BSD100 [32] as test datasets. All experiments are performed on a platform with an i9-9900k CPU and NVIDIA RTX2080Ti GPU.

B. PERFORMANCE COMPARISON FOR DIFFERENT NUMBERS OF TSRBS

Experiments with the same settings are used to investigate the effect of varying numbers of TSRBs on the reconstructed image at a sampling rate of 0.1 on dataset Set5.

FIGURE 4 shows the effect of different numbers of TSRBs on the image reconstruction results at a sampling rate of 0.1. The horizontal axis represents the epoch number in training, and the vertical axis represents the average PSNR of all reconstructed images on dataset Set5. It is clear that the

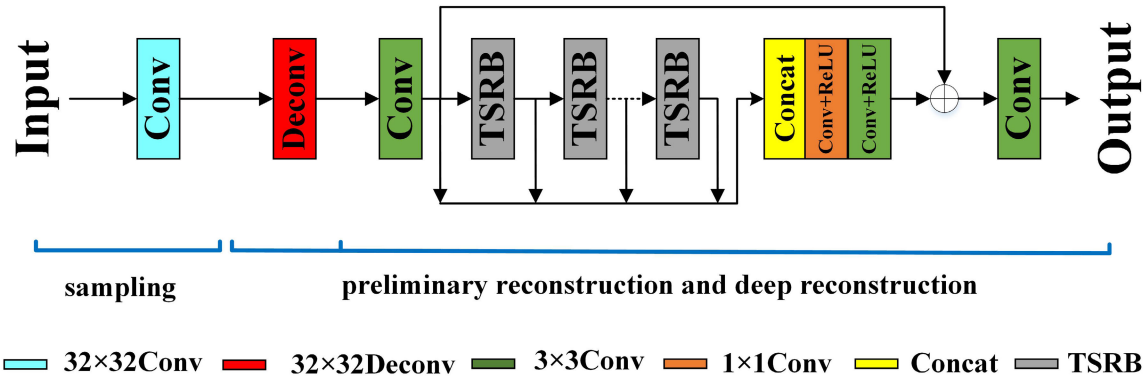

FIGURE 3. Tree-structured dilated convolutional networks.

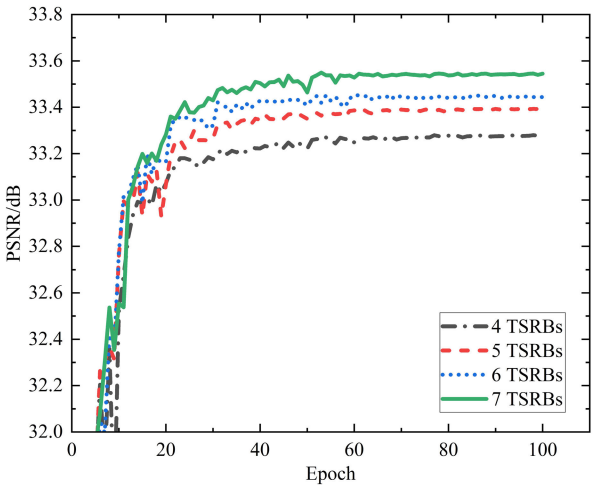

FIGURE 4. PSNR of reconstructed image with different number of Tree-Structured Residual Block (TSRB) on dataset Set 5. The sampling rate is 0.1.

reconstruction performance of the network improves as the number of TSRBs increases.

Considering the depth and running speed of the network, we select seven TSRB modules for the subsequent experimental studies.

C. COMPARISON WITH ALGORITHMS FOR IMAGE CS

In this section, the reconstructed image quality and running speed of TDCN are investigated.

TDCN is compared with four traditional algorithms and four deep learning-based algorithms. Then, the running speeds of the different algorithms are compared. The experiments are run in MATLAB 2020b and the Pytorch framework on a Windows 10 system. Some results for comparison are obtained from the published literature.

Four traditional algorithms for image CS compared are total variation (TV) [38], multi-hypothesis (MH) [39], group sparse representation (GSR) [21], and denoising-based approximate message passing (D-AMP) [22]. The experimental codes of the compared algorithms are obtained from the authors' websites, and all experiments use the default parameters. The test on these algorithms is performed on dataset Set11. It is noted that the four traditional algorithms use random matrix as sampling matrix but the proposed TDCN uses convolution layer.

As shown in TABLE 1, TDCN consistently performs better than all the compared algorithms at different sampling rates on dataset Set11. In terms of PNSR, on average, our proposed TDCN wins TV, MH, GSR, and D-AMP over 5.72 dB, 3.03 dB, 1.43 dB, and 6.54 dB, respectively.

TABLE 1. Average PSNR of different image CS algorithms on Set11.

SR	TV [38]	MH [39]	GSR [21]	D-AMP [22]	TDCN
0.01	16.43	**17.65**	16.79	5.21	**21.62**
0.04	18.75	**21.64**	21.63	18.40	**25.76**
0.1	22.99	26.95	**27.93**	22.64	**29.41**
0.25	27.92	31.37	**33.57**	28.46	**33.98**
0.3	29.23	32.43	**34.76**	30.39	**35.26**
0.4	31.46	33.89	**36.89**	33.56	**37.37**
0.5	33.55	35.20	**38.76**	35.92	**39.45**
Avg.	25.76	28.45	**30.05**	24.94	**31.48**

Five deep-learning-based algorithms, namely, ReconNet [24], ISTA-Net$^+$ [4], CSNet$^+$ [17], SCSNet [40], and MR-CSGAN [25], are also compared at their default parameters. For fair comparison, these algorithms are tested on three datasets: Set5 (5 images), Set14 (14 images), and BSD100 (100 images). Both objective and subjective evaluations are performed.

The sampling rates of the image CS measurements are set as 0.01, 0.04, 0.1, 0.25, 0.3, 0.4 and 0.5 for ISTA-Net$^+$ [4], CSNet$^+$ [17], SCSNet [40], the sizes of the corresponding convolution kernels are set as 10, 41, 102, 256, 307, 410 and 512. And the sampling rates of the image CS measurements are set as 0.01, 0.04, 0.1 and 0.25 for ReconNet [24] and MR-CSGAN [25] due to studies in article [24] and article [25] only provided test results at these sampling rate.

TABLE 2. Average PSNR (dB) and SSIM comparisons of different image CS algorithms on Set5, Set14 and BSD100.

Algorithm		ReconNet [24]		ISTA-Net$^+$ [4]		CSNet$^+$ [17]		SCSNet [40]		MR-CSGAN [25]		TDCN*		TDCN_DCBI$^+$		TDCN	
Data	SR	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	0.01	18.07	0.41	19.59	0.51	24.18	0.65	24.21	0.65	**24.51**	0.67	**24.51**	0.67	24.27	0.65	**24.61**	0.68
	0.04	21.61	0.55	23.45	0.66	28.70	0.82	28.66	0.82	29.28	0.85	**29.38**	0.86	27.59	0.79	**29.45**	0.86
	0.1	24.58	0.68	29.97	0.85	32.59	0.91	32.77	0.91	33.38	0.93	**33.55**	0.93	32.61	0.91	**33.54**	0.93
	0.25	27.22	0.77	34.17	0.93	36.81	0.96	36.71	0.96	37.74	0.97	**37.83**	0.97	35.29	0.95	**37.85**	0.97
	0.3	-	-	36.03	0.94	38.25	0.96	38.45	0.97	-	-	**38.88**	0.97	38.17	0.97	**38.95**	0.97
	0.4	-	-	38.84	0.96	40.11	0.97	40.44	0,98	-	-	**40.90**	0.98	40.22	0.98	**40.91**	0.98
	0.5	-	-	40.77	0.97	41.79	0.98	42.22	0.98	-	-	**42.79**	0.98	42.14	0.98	**42.84**	0.98
Set14	0.01	18.09	0.39	19.29	0.46	22.83	0.56	22.87	0.56	**23.14**	0.58	23.09	0.58	22.85	0.56	**23.20**	0.58
	0.04	20.62	0.49	22.08	0.57	26.38	0.71	26.29	0.72	26.73	0.74	**26.83**	0.74	25.32	0.69	**26.82**	0.74
	0.1	22.91	0.60	27.28	0.75	29.13	0.82	29.22	0.82	29.68	0.84	**29.77**	0.85	29.15	0.83	**29.75**	0.85
	0.25	25.30	0.71	30.28	0.84	32.26	0.91	32.25	0.91	33.79	0.93	**33.80**	0.93	31.57	0.91	**33.81**	0.93
	0.3	-	-	32.68	0.89	34.34	0.93	34.51	0.93	-	-	**34.83**	0.94	34.15	0.94	**34.93**	0.94
	0.4	-	-	35.31	0.93	36.16	0.95	36.54	0.95	-	-	**36.88**	0.96	36.10	0.95	**36.93**	0.96
	0.5	-	-	37.33	0.95	37.89	0.96	**38.41**	0.97	-	-	**38.83**	0.97	38.00	0.97	**38.83**	0.97
BSD100	0.01	19.08	0.40	20.36	0.46	23.76	0.55	23.78	0.55	**23.89**	0.55	23.87	0.55	23.74	0.54	**23.95**	0.56
	0.04	21.26	0.49	22.26	0.55	26.25	0.67	26.36	0.69	26.49	0.70	**26.51**	0.70	25.64	0.66	**26.53**	0.70
	0.1	23.09	0.59	26.41	0.70	28.53	0.78	28.57	0.78	28.77	0.81	**28.86**	0.81	28.44	0.80	**28.82**	0.81
	0.25	25.20	0.70	28.99	0.80	31.91	0.89	31.93	0.90	**32.39**	0.91	**32.36**	0.91	30.77	0.89	32.35	0.91
	0.3	-	-	31.07	0.86	33.08	0.92	33.24	0.92	-	-	**33.35**	0.93	32.84	0.92	**33.41**	0.93
	0.4	-	-	33.26	0.91	34.91	0.94	35.21	0.95	-	-	**35.35**	0.95	34.76	0.95	**35.36**	0.95
	0.5	-	-	35.08	0.94	36.68	0.96	37.14	0.96	-	-	**37.27**	0.97	36.67	0.97	**37.29**	0.97

All the subjective evaluation results in terms of average PSNR and SSIM are shown in TABLE 2, where the best result is marked in red and the runners-up is marked in blue.

The experimental results in TABLE 2 show that our proposed TDCN model has good performance at different sampling rates and improves differently from other algorithms in terms of PSNR. TDCN achieves the highest PSNR value and SSIM value than other methods at all sampling rate except one result for sampling rate of 0.25 on dataset BSD100.

In the subjective evaluation, we choose three standard images as the test images on Set11 and Set14 to demonstrate that the TDCN improves the visual performance of the reconstructed images. FIGURE 5, 6, and 7 show three visualization examples of images reconstructed using different methods at sampling rates of 0.01, 0.04, and 0.1, respectively. We can see that the TDCN presented here achieves the best visual effect at different sampling rates.

The comparison results of running speed are shown in TABLE 3, where the average running time (in seconds) and running conditions of the algorithms for reconstructing a 256×256 image are given in detail. The results of ReconNet are from their original paper [24] while the remaining methods are tested using our platform.

It can be seen from TABLE 3, traditional image CS algorithms take several seconds to several minutes to reconstruct a 256×256 image. In contrast, deep-learning-based methods, which take around one second on CPU or less than 0.08 second on GPU to reconstruct a 256×256 image, run faster than traditional algorithms.

Specifically, TDCN runs at a similar speed to CSNet$^+$ and SCSNet on a GPU and is much faster than other deep learning-based methods. And TDCN is about four times faster than MR-CSGAN because it is a smaller network than MR-CSGAN.

In summary, TDCN runs much faster than traditional CS algorithms and is comparable to existing deep learning-based CS algorithms.

D. ABLATION STUDY

In this section, ablation experiments are given to verify that our improvements are effective further.

1) We compared the performance differences between TDCN*, TDCN_DCBI$^+$ and TDCN.

TDCN* is a variant of our TDCN, where the convolution, reshaping and concatenating layers are used at preliminary reconstruction process instead of the deconvolution layer used in TDCN. In TDCN_DCBI$^+$ network, we replaced TSRBs in TDCN by residual blocks built from DCBI module [34] shown in FIGURE 1. The experimental results are also shown in TABLE 2 with the best result in red and the runners-up in blue.

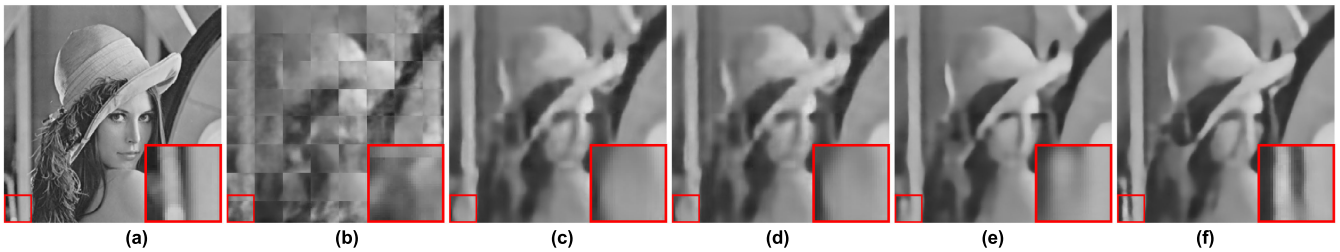

FIGURE 5. Comparison of reconstruction effect on Lena from Set11 with 0.01 sampling rate, (a) Original (PSNR / SSIM). (b) ISTA-Net$^+$ (18.54 dB / 0.2557). (c) CSNet$^+$ (22.43 dB / 0.6179). (d) SCSNet (22.41 dB / 0.6159). (e) MR-CSGAN (22.84 dB / 0.6446). (f) TDCN (23.33 dB / 0.6646).

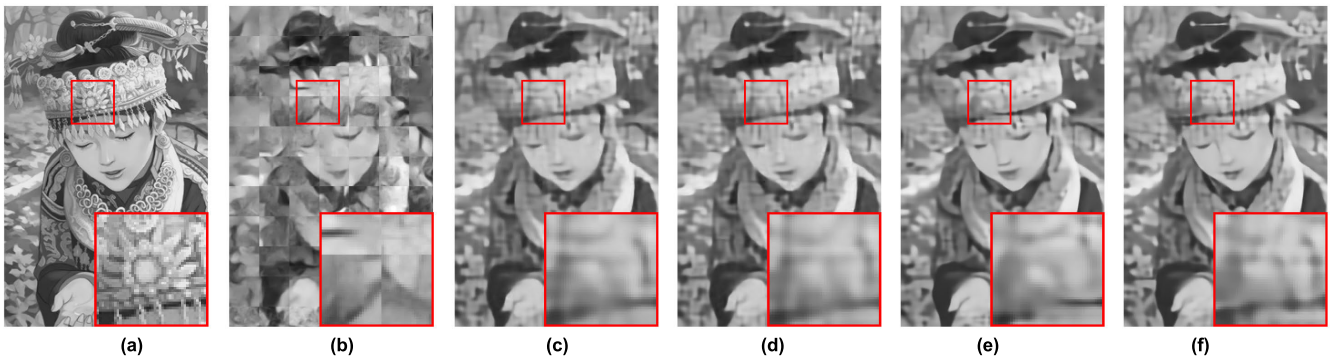

FIGURE 6. Comparison of reconstruction effect on Comic from Set14 with 0.04 sampling rate, (a) Original (PSNR / SSIM). (b) ISTA-Net$^+$ (17.60 dB / 0.4306). (c) CSNet$^+$ (21.81 dB / 0.5791). (d) SCSNet (21.79 dB / 0.5766). (e) MR-CSGAN (21.93 dB / 0.6301). (f) TDCN (22.03 dB / 0.6412).

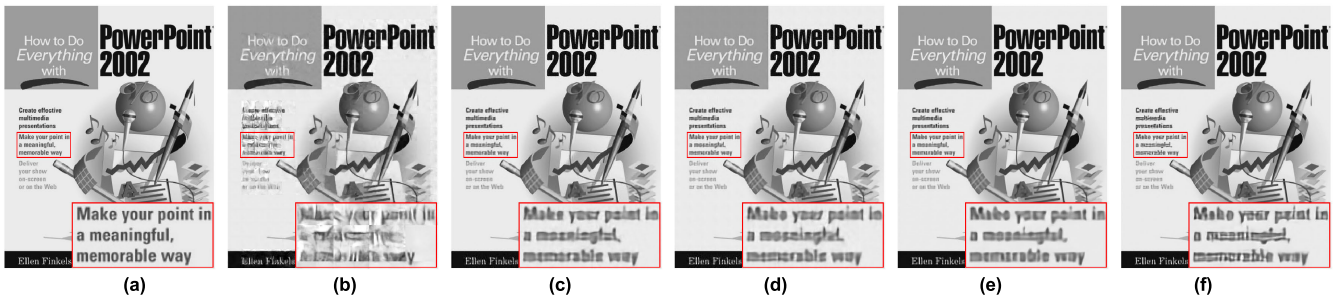

FIGURE 7. Comparison of reconstruction effect on PPT3 from Set14 with 0.1 sampling rate, (a) Original (PSNR / SSIM). (b) ISTA-Net$^+$ (24.92 dB / 0.8826). (c) CSNet$^+$ (27.94 dB / 0.9493). (d) SCSNet (28.01 dB / 0.9480). (e) MR-CSGAN (28.70 dB / 0.9650). (f) TDCN (28.83 dB / 0.9680).

TABLE 3. Average running time (in seconds) of various algorithms for reconstructing a 256 × 256 image.

Algorithm	Ratio=0.01		Ratio=0.1		Programming Language	Platform
	CPU	GPU	CPU	GPU		
TV	0.8510	-	0.9240	-	Matlab	Intel Core i9-9900k
MH	15.060	-	11.9460	-		
GSR	586.9530	-	584.3840	-		
D-AMP	6.3180	-	8.3590	-		
ReconNet(Author)	0.5193	0.0244	0.5258	0.0195	Matlab+Caffe	Intel Xeon E5-1650 CPU+NVIDIA GTX980 GPU
ISTA-Net$^+$	1.3750	0.0470	1.3750	0.0470	Python+Pytorch	Intel Core i9-9900k CPU+ NVIDA RTX2080Ti GPU
CSNet$^+$	1.1691	0.0103	1.1808	0.0102	Matlab+Matconvnet	
SCSNet	0.7295	0.0101	0.7383	0.0201		
MR-CSGAN	1.9508	0.0708	2.0241	0.0722	Python+Pytorch	
TDCN(Ours)	0.7191	0.0160	0.7081	0.0170		

TABLE 2 shows that TDCN has very tiny advantages than TDCN* in terms of PSNR values. This result shows that adopting deconvolution layer at the preliminary reconstruction process instead of the linear convolution, reshaping

and concatenating layers can bring comparable, even slightly better performance for TDCN.

The fact that TDCN_DCBI$^+$ produced worst results than TDCN in TABLE 2 is an experimental proof that tree-structured dilated convolutions in TSRB perform better than DCBI [34] that adopts dilated convolutions with different dilation rates to extract image features simultaneously.

2) To investigate the effect of the loss function on network performance, a variant of TDCN, i.e., TDCN$^+$ is built, where L2 loss is used instead of L1 loss.

They are both trained at the sampling rate of 0.1. The training process for TDCN and TDCN$^+$ are shown in FIGURE 8, which demonstrates that training with L1 loss performs better than with L2 loss under the same condition of TDCN structure.

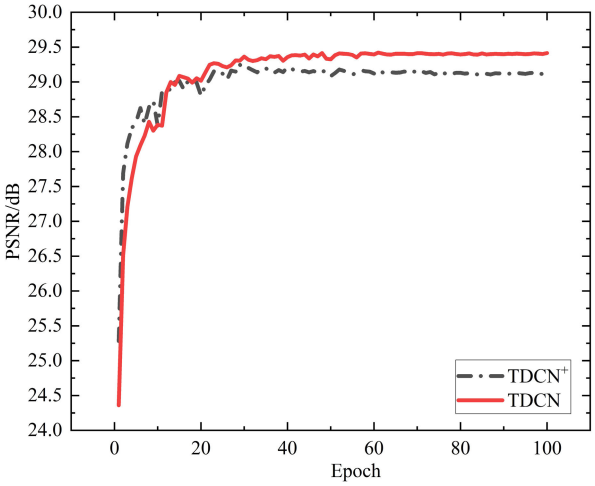

FIGURE 8. PSNR comparison among TDCN and TDCN$^+$ on Set11. The sampling rate is 0.1.

3) To investigate the effect of BN layer, KConv and DConv layer on network performance, three variants of TDCN are built as shown in TABLE 4, namely TDCN_TFA, TDCB_TFA$^+$ and TDCN_TFA^{++}.

TABLE 4. Different measures of improvement for TFA, TFA$^+$, TFA^{++}, and TDCN.

	TDCN_TFA	TDCN_TFA$^+$	TDCN_TFA^{++}	TDCN
BN	√		√	
Avgpool	√	√		
DConv	√	√	√	√

TDCN_TFA adopts both BN layer and KConv that includes Avgpool layer and DConv layer; TDCN_TFA$^+$ adopts both Avgpool layer and DConv layer; TDCN_TFA^{++} adopts both BN layer and DConv layer. We must also note that TDCN uses only DConv layer compared with its variants. These three variants of TDCN are trained separately at the sampling rate of 0.1. The test results of PSNR vs. Epoch for

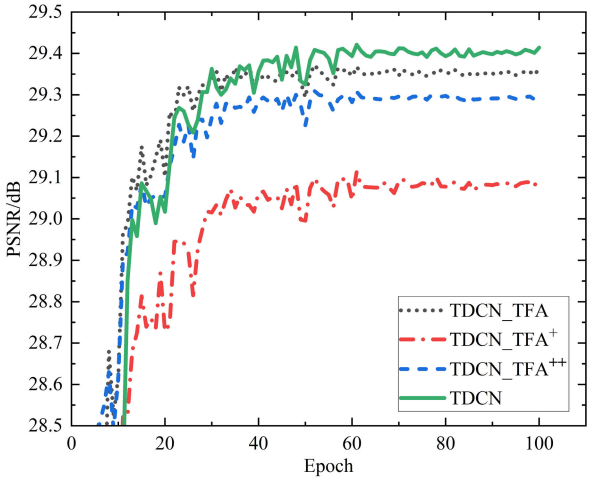

FIGURE 9. PSNR comparison among TDCN_TFA, TDCN_TFA$^+$, TDCN_TFA^{++} and TDCN on Set11. The sampling rate is 0.1.

TDCN_TFA, TDCN_TFA$^+$, TDCN_TFA^{++} and TDCN on Set11 are shown in FIGURE 9.

Experimental results show that TDCN_TFA performs better than TDCN_TFA$^+$ and TDCN_TFA^{++}. However, our proposed TDCN outperforms other variants in PSNR value. It proved that using DConv only is the best option compared with other assumed situations in Table 4.

V. CONCLUSION

In this study, we propose a tree-structured dilated convolutional network for image compressed sensing. The algorithm uses a tree-structured residual block to recover the detailed image features in deep reconstructed networks fully. Meanwhile, we use L1 loss rather than L2 loss to train the network. All experimental results demonstrate that the reconstructed images of TDCN have more detailed structural information and a sharper appearance. The proposed TDCN outperforms the current algorithms in both the PSNR and SSIM metrics, and the running speed of the algorithm is comparable to that of the current algorithms. In the future, we will consider applying TDCN to CS in hyperspectral remote sensing images and study an algorithm that utilizes interspectral correlation to obtain higher reconstruction quality.

REFERENCES

[1] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
[2] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal Process.*, vol. 86, no. 3, pp. 549–571, Mar. 2006.
[3] C. E. Shannon, "Communication in the presence of noise," *Proc. Inst. Radio Eng.*, vol. 37, no. 1, pp. 10–21, Jan. 1949.
[4] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1828–1837.
[5] R. Baraniuk and P. Steeghs, "Compressive radar imaging," in *Proc. IEEE Radar Conf.*, Apr. 2007, pp. 128–133.
[6] L. Zhang, M. Xing, C. Qiu, J. Sheng, Y. Li, and Z. Bao, "Resolution enhancement for inversed synthetic aperture radar imaging under low SNR via improved compressive sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3824–3838, Oct. 2010.

[7] C. Zheng, K. Liao, S. Ouyan, and C. Li, "Distributed computing method for synthetic aperture radar compressed sensing imaging based on MapReduce," in *Proc. IEEE 3rd Int. Conf. Electron. Inf. Commun. Technol. (ICEICT)*, Nov. 2020, pp. 541–544.

[8] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Proc. Int. Conf. Image Process. (ICIP)*, Nov. 2006, pp. 1273–1276.

[9] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "CS-MUVI: Video compressive sensing for spatial-multiplexing cameras," in *Proc. IEEE Int. Conf. Comput. Photogr. (ICCP)*, Seattle, WA, USA, Apr. 2012, pp. 1–10.

[10] M. Murad, M. Bilal, A. Jalil, A. Ali, K. Mehmood, and B. Khan, "Efficient reconstruction technique for multi-slice CS-MRI using novel interpolation and 2D sampling scheme," *IEEE Access*, vol. 8, pp. 117452–117466, 2020.

[11] U. Molnar, J. Nikolov, O. Nikolić, N. Boban, V. Subašić, and V. Till, "Diagnostic quality assessment of compressed SENSE accelerated magnetic resonance images in standard neuroimaging protocol: Choosing the right acceleration," *Phys. Medica*, vol. 88, pp. 158–166, Aug. 2021.

[12] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ECG via block sparse Bayesian learning," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 2, pp. 300–309, Feb. 2013.

[13] X. Gao, J. Zhang, W. Che, X. Fan, and D. Zhao, "Block-based compressive sensing coding of natural images by local structural measurement matrix," in *Proc. Data Compress. Conf. (DCC)*, Apr. 2015, pp. 133–142.

[14] A. Amini and F. Marvasti, "Deterministic construction of binary, bipolar, and ternary compressed sensing matrices," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2360–2370, Apr. 2011.

[15] W. Lu, T. Dai, and S.-T. Xia, "Binary matrices for compressed sensing," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 77–85, Jan. 2018.

[16] K. Q. Dinh, H. J. Shim, and B. Jeon, "Measurement coding for compressive imaging using a structural measurement matrix," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 10–13.

[17] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," *IEEE Trans. Image Process.*, vol. 29, pp. 375–388, 2020.

[18] Y. Kim, M. S. Nadar, and A. Bilgin, "Compressed sensing using a Gaussian scale mixtures model in wavelet domain," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 3365–3368.

[19] 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, Jul. 2013.

[20] J. Zhang, C. Zhao, D. Zhao, and W. Gao, "Image compressive sensing recovery using adaptively learned sparsifying basis via L_0 minimization," *Signal Process.*, vol. 103, pp. 114–126, Oct. 2014.

[21] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[22] 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.

[23] A. Adler, D. Boublil, and M. Zibulevsky, "Block-based compressed sensing of images via deep learning," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

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

[25] J. Tian, W. Yuan, and Y. Tu, "Image compressed sensing using multi-scale residual generative adversarial network," *Vis. Comput.*, pp. 1–10, Sep. 2021, doi: 10.1007/s00371-021-02288-y.

[26] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[27] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[28] T. Wu, S. Tang, R. Zhang, J. Cao, and J. Li, "Tree-structured Kronecker convolutional network for semantic segmentation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 940–945.

[29] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.

[30] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L.-A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 135-1–135-10.

[31] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.* Berlin, Germany: Springer, 2010, pp. 711–730.

[32] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[34] W. Shi, F. Jiang, and D. Zhao, "Single image super-resolution with dilated convolution based multi-scale information learning inception module," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 977–981.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[37] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.

[38] C. Li, W. Yin, and Y. Zhang. (2013). *TVAL3: TV Minimization by Augmented Lagrangian and Alternating Direction Algorithm 2009*. [Online]. Available: https://www.caam.rice.edu/~optimization/L1/TVAL3/

[39] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *Proc. Conf. Rec. 45th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 1193–1198.

[40] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable convolutional neural network for image compressed sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12282–12291.

RUI LU was born in Jiangxi, China, in 1996. He is currently pursuing the M.Sc. degree with the College of Science, Jiangxi University of Science and Technology. His current research interests include deep learning image processing and compressed sensing.

KUNTAO YE received the B.S. degree in physics from Nankai University, China, in 1994, and the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 2006.

He conducted his postdoctoral research in the MOEMS field for a short period at McMaster University, Hamilton, ON, Canada, in 2007. From 2007 to 2009, he was a Product Engineer and the Chief Technology Officer with IntelliSense Corporation, USA. Since 2009, he has been an Associate Professor at the Jiangxi University of Science and Technology, Jiangxi, China. He was also a Visiting Scholar at the University of Nevada Las Vegas, in 2017 and McMaster University, in 2019. His research interests include MEMS, image processing, and optoelectronic instrumentation.