ConvCSNet: A Convolutional Compressive Sensing Framework Based on Deep Learning

Xiaotong Lu, Weisheng Dong, Member, IEEE, Peiyao Wang, Guangming Shi, Senior member, IEEE, and Xuemei Xie

Abstract—Compressive sensing (CS), aiming to reconstruct an image/signal from a small set of random measurements has attracted considerable attentions in recent years. Due to the high dimensionality of images, previous CS methods mainly work on image blocks to avoid the huge requirements of memory and computation, i.e., image blocks are measured with Gaussian random matrices, and the whole images are recovered from the reconstructed image blocks. Though efficient, such methods suffer from serious blocking artifacts. In this paper, we propose a convolutional CS framework that senses the whole image using a set of convolutional filters. Instead of reconstructing individual blocks, the whole image is reconstructed from the linear convolutional measurements. Specifically, the convolutional CS is implemented based on a convolutional neural network (CNN), which performs both the convolutional CS and nonlinear reconstruction. Through end-to-end training, the sensing filters and the reconstruction network can be jointly optimized. To facilitate the design of the CS reconstruction network, a novel two-branch CNN inspired from a sparsity-based CS reconstruction model is developed. Experimental results show that the proposed method substantially outperforms previous state-of-the-art CS methods in term of both PSNR and visual quality.

Index Terms—denoising-based image restoration, deep neural network, denoising prior, image restoration.

I. INTRODUCTION

Compressive sensing (CS) is a new image/signal acquisition framework acquiring only a few linear measurements [1], [2] with random measurement matrices. Such framework has the potentials of significantly improving the imaging speed and sensor energy efficiency in real applications. Based on CS, several new imaging systems, including single-pixel camera [3], compressive spectral imaging system [4], high-speed video camera [5], and fast Magnetic Resonance Imaging (MRI) system [6] have been developed. Perfect reconstruction from the linear measurements is guaranteed if the following two conditions are met, i.e., the sensing matrices have the restricted isometry property (RIP) and the images have sparse representation with respect to a dictionary. These two conditions can be easily satisfied as random Gaussian matrices have the RIP with high probabilities [1], [2] and natural images have sparse representation under many off-the-shelf or learned dictionaries (e.g., wavelets, learned dictionaries [7]). However, in practice the promise of CS is often offset by challenges relating to two conditions. First, it is difficult or even impossible to implement the random sensing matrix for a large image. Second, the sparsity-based CS reconstruction algorithms are very slow to converge for obtaining good estimates of the original images.

For a given image \(x \in \mathbb{R}^N\), the CS measurement of \(x\) can be expressed as \(y = \Phi x\), where \(\Phi \in \mathbb{R}^{M \times N}\) is the sensing matrix and \(M = rN\) \(0 < r \ll 1\). In practice, as the dimension of natural images are often very high, the memory requirement for storing the matrix \(\Phi\) is huge. For example, for a moderate image of size \(1,000 \times 1,000\) and \(r = 0.1\), the dimension of \(\Phi\) is \(100,000 \times 1,000,000\).

The computational complexity of the CS reconstruction with such large sensing matrix is also prohibitively high. To avoid these difficulties, block-based CS (BCS) methods have been proposed [8], [9], where an image is divided into many non-overlapping blocks and each block is sensed individually. As such, the sensing matrices \(\Phi\) become much smaller and images can be efficiently measured. However, as the blocks are sensed and reconstructed individually, the BCS methods lead to serious blocking artifacts. To reduce the blocking artifacts, post-processing is often required to improve the visual quality.

According to the CS theory [1], [2], the original images can be well-reconstructed by exploiting the sparsity prior of natural images. However, the sparsity-based CS methods recover the original images by solving an optimization algorithm, which is very slow to converge. Thus, the optimization-based CS methods cannot be used for real-time applications. Recently, inspired by the great successes of deep neural network (DNN) for computer vision tasks [10]–[12], DNN-based CS reconstruction methods have also been proposed [13]–[15]. With DNN, both the linear sensing and nonlinear reconstruction can be performed by a single neural network. Through end-to-end training, both the sensing matrix and the reconstruction method can be jointly optimized, and the speed is usually hundreds-times faster than optimization-based methods. However, to the best of our knowledge, current DNN based methods are all block-based, i.e., image blocks are sensed and reconstructed individually. Blocking artifacts can also be observed in the reconstructed images by the DNN-based methods [13], [14].

In this paper, we propose a novel convolutional compressive sensing (ConvCS) framework based on deep convolutional neural network (DCNN). In the proposed ConvCS network, the first layer senses an input image by convolving the whole image with a set of random filters, followed by subsampling. The advantage of the proposed ConvCS is that the whole image can be efficiently sensed with a set of small filters that are easy to store, and effectively reconstructed without introducing blocking artifacts. The remaining layers of the proposed ConvCS network perform the nonlinear reconstruction of the whole image from the measurements. To design the reconstruction network, the domain knowledge of the sparsity-based CS reconstruction is incorporated, leading to a
novel CNN for image reconstruction. By end-to-end training, both the convolutional sensing filters and the reconstruction CNN are jointly optimized. Experimental results show that the proposed method substantially outperforms current state-of-the-art CS methods in terms of PSNR and visual quality.

II. RELATED WORKS

A. Background of CS

Instead of sampling the entire signal, the CS theory samples a signal $x \in \mathbb{R}^N$ by taking only $M$ linear measurements, i.e., $y = \Phi x + e$, $y \in \mathbb{R}^M$, where $\Phi \in \mathbb{R}^{M \times N}$, $M < N$ is the sensing matrix and $e \in \mathbb{R}^M$ is the measurement noise. Since $M < N$, recovering $x$ from $y$ is generally an ill-posed inverse problem. However, the CS theory guarantees perfect reconstruction of $x$ if $x$ is sparse in some sparsifying spaces and $\Phi$ holds the RIP. It has been proven that the Gaussian random matrices have the RIP with very high probabilities. Standard CS methods recover $x$ by solving a minimization problem,

$$\alpha = \arg \min_{\alpha} \| y - \Phi \alpha \|_2^2 + \lambda \| \alpha \|_1, \quad (1)$$

where $D$ is the dictionary and $\alpha$ are the sparse codes of $x$. After estimating the sparse codes $\alpha$, $x$ can be reconstructed as $x = D \alpha$. It has been proven in [1], [2] that $x$ can be faithfully recovered from $M = O(K \log(N/K))$ measurements, where $K$ denotes the number of nonzero coefficients of $\alpha$. The $\ell_1$-minimization problem can be solved by many optimization algorithms [16], [17]. However, as mentioned above these methods are very slow to converge.

B. Block-based image CS

When applying CS to images, the image can be sensed by representing it into a vector and measured as $y = \Phi x$. However, for images of high dimensionality, the sensing matrix $\Phi$ becomes very large and it is impossible to store it and compute with such large matrix $\Phi$. To avoid such difficulties, block-based CS (BCS) methods have been proposed [8], [9]. In these methods, the input image is divided into many non-overlapped blocks and each block is sensed independently using a much smaller matrix $\Phi$. The full image is recovered by placing back the reconstructed blocks, followed by full-image smoothing. To reduce the blocking artifacts, full-image iterative shrinkage algorithm was proposed [9], and improvements can be achieved by using more advanced transforms, such as contourlets and dual-tree discrete wavelet transforms [8]. In addition to the sparsity prior, model-based CS recovery algorithms have also been developed to exploit the high-order dependencies between the wavelet coefficients [18], leading to better performance. The effective nonlocal self-similarity prior has also been integrated into the objective function through nonlocal low-rank regularization [19].

C. Structured CS for images

To overcome the drawback of the BCS, structured CS operators [20]–[22] have been proposed. The convolutional CS methods, which performs the sensing by first convoluting a signal with a random filter and then subsampling, have been proposed in [20], [22]. These CS methods are easy to implement and have many potential applications, such as Fourier optics [20], Radar imaging [21] and coded aperture imaging [23]. In addition to the random filter, deterministic filter that is more convenient to implement has also been proposed in [22]. Though the convolution-based CS are more easy to implement, the use of only one random filter makes the reconstruction problem more difficult [20]. Also, all these convolution-based CS use iterative algorithms to reconstruct the original images, which are very slow to converge.

D. Deep learning based CS for images

Recently, inspired by the successes of the deep neural networks, non-iterative CS reconstruction methods have been proposed [13]–[15]. In [13], a stacked auto-encoder denoising network was developed for image CS. Similarly, a full-connected neural network has been proposed in [15]. Convolutional neural network has also been proposed for this task, where a BM3D denoising [24] stage is adopted to further improve the reconstruction performance [14]. Through end-to-end training, both the sensing matrices and the reconstruction network can be jointly optimized for better performance. However, all those methods perform CS and reconstruction on image blocks, leading to limited reconstruction performance.

In this paper, we propose a new convolutional compressive sensing (denoted as ConvCS) framework using deep convolutional neural network (DCNN), where the first layer implements the sensing by convolving the input image with a set of random filters followed by subsampling. The remaining layers reconstruct the input image from the linear measurements. Furthermore, inspired from the sparsity-based reconstruction model, a novel CNN containing two branches is proposed for CS reconstruction. By performing sensing and reconstruction on the whole image, the proposed method significantly outperforms the previous CS methods. Different from previous convolution-based CS methods [20], [22], both the sensing filters and the reconstruction algorithm can be jointly optimized. Experimental results show that the proposed method outperforms existing state-of-the-art CS methods by a large margin.

III. PROPOSED CONVOLUTIONAL COMPRESSIVE SENSING USING DEEP LEARNING

A. Proposed convolutional compressive sensing

Unlike existing BCS and convolutional CS, we propose to sense an image by convolving it with a set of random filters, followed by spatial subsampling of the convolved images. Specifically, for a given image $x \in \mathbb{R}^N$, we convolve it with a set of random filters $\Phi_i$, of size $L \times L$, $i = 1, 2, \cdots, m$, to generate the CS measurements. Mathematically, the sensing matrix can be expressed as

$$\Phi = [(S \Phi_1)^T, (S \Phi_2)^T, \cdots, (S \Phi_m)^T]^T, \quad (2)$$

where $\Phi_i \in \mathbb{R}^{N \times N}$ is set to be a sparse matrix such that $\Phi_i x$ is equivalent to convolving $x$ with filter $\phi_i$, and $S \in \mathbb{R}^{m_0 \times N}$ is a subsampling matrix. Then, the proposed convolutional CS
As convolution can also be implemented with matrix-vector multiplication, the proposed ConvCS is equivalent to the CS process that first extracts blocks of size $L \times L$ with sliding step $s$ and then measures the blocks with the sensing matrix, whose $i$-th rows are composed with the vectorized filter coefficients of $\phi_i$. Thus, the proposed ConvCS matrix $\Phi$ of Eq. (2) still holds the advantages of the random Gaussian matrix for CS (i.e., the RIP). However, the proposed ConvCS is clearly distinct from the previous BCS methods in two aspects. First, the proposed ConvCS for images of large dimensions is much easier to implement. Second, the convolutional nature of the proposed ConvCS makes the joint optimization of the sensing filters $\phi_i$ and reconstruction of the whole image much more effective, without introducing any blocking artifacts.

The proposed ConvCS can be easily implemented using a convolutional neural network, as shown in Fig. 1 (a). The first layer convolves the input image $x$ with a set of $m$ random filters of size $L \times L$ with stride $s$. The measurements can be obtained by representing the obtained feature maps into a $M$-dimensional vector $y$. Note that nonlinearity is not involved in the CS process. More convolutional layers with nonlinear activity function could be added into the ConvCS process, which may lead to better performance. However, for simplicity, here we only use one linear layer to obtain the measurements. As shown in Sec. IV, the simple ConvCS can already lead to excellent CS recovery performances.

Essentially, the ConvCS encodes the visual information of images and plays a similar role as autoencoders. In the context of CS, the stacked denoising autoencoders (SDA) [13] have been proposed to encode the images. However, the SDA method performs CS encoding at image block levels, resulting in serious blocking artifacts.

B. Sparse model inspired DCNN for CS reconstruction

After obtaining the CS measurements, we aim to recover the original image from the measurements. In recent years, DCNN has shown very promising performances for many low-level image processing tasks, e.g., image super-resolution [25], [26]. However, the problem of reconstructing images from the ConvCS measurements is different from previous image SR problem, and existing reconstruction network can not be applied directly for this task. To facilitate the design of the CS reconstruction network, we propose to incorporate the domain knowledge of the sparsity-based CS reconstruction. Specifically, we first propose the following analysis sparse representation model for CS reconstruction,

$$\min_{x, \alpha_k} ||y - \Phi x||^2_2 + \eta \sum K_k \{||w_k \ast x - \alpha_k||^2_F + J(\alpha_k)\},$$  (3)

where $w_k \in \mathbb{R}^{n \times n}$ is the analysis filter, $\ast$ denotes the 2D convolution, and $J(\cdot)$ denotes a regularization term imposed on sparse codes $\alpha_k$. Classic sparsity enforcing regularizers, e.g., $\| \cdot \|_p (0 \leq p \leq 1)$, as well as the non-negative indicator function $I(\cdot)$ (as inspired by the ReLU function) can be adopted. Solving Eq. (3) amounts to alternatively solving two subproblems, i.e.,

$$x = \arg\min_x ||y - \Phi x||^2_2 + \eta \sum K_k \{||w_k \ast x - \alpha_k||^2_F\},$$

$$\alpha_k = \arg\min_{\alpha_k} ||w_k \ast x - \alpha_k||^2_F + J(\alpha_k),$$  (4)

where $k = 1, 2, \cdots, K$. Both sub-problems can be easily optimized. With a fixed estimate $x(t)$ obtained at $t$-th iteration, $\alpha_k$ can be solved in closed-form for several commonly used sparsity regularizer, as

$$\alpha_k^{(t+1)} = S_\tau(w_k \ast x(t)), \text{ or } [w_k \ast x(t)]_+$$  (5)

for $\ell_1$-norm sparse regularizer and non-negative regularizer, respectively, where $S_\tau(\cdot)$ denotes the soft-thresholding with threshold $\tau$. The $x$-subproblem is a quadratic optimization problem and can be solved in closed-form. However, to avoid large matrix inversion, we prefer to solve it via a gradient descent method. With a fixed estimate of $\alpha_k$, we iteratively update $x$ as

$$x^{(t+1)} = x(t) - \delta(\Phi^T (\Phi x(t) - y) + \eta \sum K_k (W_k^T (W_k x(t) - \alpha_k^{(t+1)}))),$$  (6)

where $W_k$ is the sparse matrix such that $W_k x$ is equivalent to convolving $x$ with $w_k$, and $\delta$ is the predefined constant. By alternatively updating $x$ and $\alpha$, the iterative process will converge. However, the convergence speed is very slow.

In this paper, we propose to convert the alternative update of $x$ and $\alpha$ into a deep network. For simplicity, we rewrite Eq. (6) as

$$x^{(t+1)} = x(t) - \delta(\Lambda x(t) - y + \eta \sum K_k (W_k x(t) - \alpha_k^{(t+1)})), $$  (7)

where $\Delta x = 0 \Phi^T \Phi x$ denotes the process that first senses $x$ with $\Phi$ and then reconstructs $x$ from the CS measurements by back-projection. For simplicity, we let $\Lambda = I$ and $W_k W_k = I$. Then, Eq. (7) can be approximated as

$$x^{(t+1)} \approx \rho x^{(t)} + \delta x^{(0)} + \gamma x^{(t+1/2)},$$  (8)

where $\rho = (1 - \delta (1 + \eta)), \gamma = \delta \eta, x^{(0)}$ denotes the initial reconstruction of $x$ from $y$, and $x^{(t+1/2)} = \sum K_k W_k \alpha_k^{(t+1)}$ denotes the reconstructed $x$ from $\alpha_k^{(t+1)}$.

Inspired from the alternative update of $\alpha_k$ and $x$, we propose a novel DCNN for CS reconstruction. As shown in Fig. 1 (b), the proposed reconstruction network contains two branches. The first branch implements a conventional CNN for generating the feature maps $\alpha_k$. The first branch taking the measurements vector $y$ as input back projects $y$ into the feature maps, where all entries are zero except the sampled set of pixels (as marked red shown in Fig. 1 (b)). The first layer uses kernels of size $3 \times 3$ and generates $m$ feature maps, while the remaining 14 layers use kernels of size $3 \times 3$ and generate 96 feature maps. The ReLU function is applied following convolution. Compared to the
160 collected natural images from the ImageNet dataset [27] and for CS reconstruction. To train the proposed networks, we branch of the reconstruction network shown in Fig. 2 (b) verification the effectiveness of the proposed two-branch reconstruction network, we also implement a variant of the ConvCSNet for an updated estimate \( x^{(t+1)} \), which mimics the computations of Eqs. (7) and (8). In each layer of the second branch, the feature map \( \alpha^{(t+1)} \) from the CNN branch is fed into a convolutional layer to produce an image, which is further added with previously reconstructed \( x^{(t)} \) and \( x^{(0)} \) for the network parameters (including the sensing filters) and \( T \) the total number of training patches. The proposed network is trained with ADAM optimizer [28] by setting \( \beta_1 = 0.9 \), \( \beta_2 = 0.999 \) and \( \epsilon = 10^{-8} \). The minibatch size is set to 64. We initialize the learning rate as \( 10^{-4} \) and halved at every 2 \( \times \) \( 10^5 \) minibatch updates. We implemented the proposed network with TensorFlow framework and training them using 4 NVIDIA 1080Ti GPUs. It takes one day to train ConvCSNet. Currently, we trained one model for each sensing rates. In future, a general reconstruction network may be trained for different measurement rates.

### IV. Experimental results

#### A. Training details

To achieve better performance, the sensing layers and the remaining reconstruction layers are jointly trained, and thus the sensing filters \( \phi_k \) and the reconstruction network can be jointly optimized. Let ConvCSNet denote the proposed network performing convolutional CS and reconstruction. To verify the effectiveness of the proposed two-branch reconstruction network, we also implement a variant of the ConvCSNet (denoted as ConvCSNet-baseline), which uses only the first branch of the reconstruction network shown in Fig. 2 (b) for CS reconstruction. To train the proposed networks, we collected natural images from the ImageNet dataset [27] and extracted the central 160 \( \times \) 160 part of each image. The extracted patches are converted into grayscale and augmented via horizontal and vertical flips and 90 rotations. Finally, we obtained a training set containing 4000,000 image patches. We empirically set the parameters of the convolutional sensing layer as: \( L = 17, m = 8, s = 8 \) for rate 0.05; \( L = 17, m = 6, s = 8 \) for rate 0.1, \( L = 11, m = 5, s = 5 \) for rate 0.2, and \( L = 11, m = 8, s = 5 \) for rate 0.3, where \( L, \ m, \) and \( s \) denotes the filter size, number of filters and the convolutional stride, respectively. The parameters of the reconstruction layers are the same for different measurement rates, except the first layer of the reconstruction part of the ConvCSNet. In that layer, we use corresponding filter size of \( L \times L \) used in the sensing layer.

The proposed network is trained using the \( \ell_2 \) loss function, as 

\[
L(\Theta) = \frac{1}{T} \sum_i \| f(x_i, \Theta) - x_i \|^2_2,
\]

where \( \Theta \) denotes all the network parameters (including the sensing filters) and \( T \) is the number of training patches. The proposed network is trained with ADAM optimizer [28] by setting \( \beta_1 = 0.9 \), \( \beta_2 = 0.999 \) and \( \epsilon = 10^{-8} \). The minibatch size is set to 64. We initialize the learning rate as \( 10^{-4} \) and halved at every 2 \( \times \) \( 10^5 \) minibatch updates. We implemented the proposed network with TensorFlow framework and training them using 4 NVIDIA 1080Ti GPUs. It takes one day to train ConvCSNet. Currently, we trained one model for each sensing rates. In future, a general reconstruction network may be trained for different measurement rates.

#### B. Comparison with state-of-the-art methods

We compare the proposed ConvCSNet method with two iterative CS methods, including the total variation method (denoted as TV [29]), the denoising-based approximate message passing (D-AMP) method [30], and one recently developed fully-connected network based CS method (denoted as FCN-CS) [15]. Note that both TV [29] and D-AMP methods [30] conduct compressive sensing on the whole image, using very large measurement matrices. For an input image of size 255 \( \times \) 255 and measurement rate 0.2, the measurement matrix is of size 13005 \( \times \) 65032, requiring about 7GB memory for storing it in Matlab platform. The computational complexity of the iterative CS reconstruction using such large sensing matrices also become very slow. However, the advantage of using such large measurement matrices is that the reconstruction quality can be much improved. Also note that the D-AMP method [30] uses the well-known BM3D denoising method [24] in its iterative reconstruction process. As the BM3D denoising method is very effective in suppressing noise and artifacts, the D-AMP method achieves the state-of-the-art CS performance. The deep learning based FCN-CS method [15] performing the sensing and reconstruction based on 16 \( \times \) 16 blocks. As the authors of FCN-CS method only provided the...
TABLE I: PSNR results on the set of images of Fig. 2 by the test methods at different measurement rates.

| Images | Ratio | Noiseless | Noisy |
|--------|-------|-----------|-------|
|        | 0.05  | 0.1       | 0.2   | 0.3   | 0.05  | 0.1       | 0.2   | 0.3   |
| butterfly | TV [29] | 19.01 | 22.73 | 28.03 | 31.99 | 18.56 | 21.88 | 25.11 | 25.28 |
|          | FCN-CS [15] | 22.93 | 25.78 | 29.07 | 30.87 | 22.82 | 24.83 | 27.01 | 29.40 |
|          | D-AMP [30] | 22.70 | 26.55 | 31.38 | 34.32 | 20.52 | 23.40 | 27.75 | 28.93 |
|          | ConvCSNet-baseline | 24.17 | 27.50 | 31.44 | 34.72 | 23.99 | 27.11 | 29.85 | 32.32 |
|          | ConvCSNet | 25.63 | 28.48 | 32.91 | 35.36 | 25.35 | 28.46 | 30.70 | 33.49 |
| parthenon | TV [29] | 22.75 | 24.90 | 27.28 | 29.18 | 22.16 | 23.76 | 25.21 | 25.95 |
|          | FCN-CS [15] | 25.28 | 27.18 | 29.93 | 31.96 | 25.21 | 26.59 | 28.58 | 30.49 |
|          | D-AMP [30] | 24.59 | 27.43 | 30.72 | 33.11 | 23.78 | 25.50 | 27.62 | 28.90 |
|          | ConvCSNet-baseline | 25.34 | 26.80 | 30.01 | 33.35 | 24.99 | 26.46 | 28.70 | 31.23 |
|          | ConvCSNet | 25.79 | 27.39 | 30.94 | 33.89 | 25.65 | 27.31 | 29.53 | 32.44 |
| starfish | TV [29] | 22.07 | 24.31 | 27.58 | 30.24 | 21.85 | 23.50 | 25.37 | 26.57 |
|          | FCN-CS [15] | 24.90 | 27.35 | 30.91 | 33.25 | 24.94 | 27.16 | 29.76 | 31.11 |
|          | D-AMP [30] | 23.87 | 27.44 | 31.73 | 34.46 | 22.71 | 25.20 | 27.82 | 29.25 |
|          | ConvCSNet-baseline | 25.04 | 27.29 | 32.61 | 36.17 | 24.62 | 26.94 | 30.27 | 33.19 |
|          | ConvCSNet | 25.79 | 28.79 | 34.10 | 37.51 | 25.67 | 28.72 | 31.77 | 35.00 |
| flower | TV [29] | 23.48 | 25.81 | 28.85 | 31.33 | 22.9 | 24.39 | 26.03 | 26.88 |
|          | FCN-CS [15] | 26.00 | 28.08 | 30.91 | 32.83 | 25.92 | 27.81 | 29.90 | 31.83 |
|          | D-AMP [30] | 24.96 | 28.34 | 32.49 | 36.18 | 24.07 | 26.13 | 28.20 | 29.64 |
|          | ConvCSNet-baseline | 26.10 | 27.88 | 32.14 | 35.67 | 25.72 | 27.45 | 29.22 | 30.82 |
|          | ConvCSNet | 26.80 | 29.14 | 33.38 | 36.11 | 26.60 | 29.07 | 31.45 | 34.49 |
| girl | TV [29] | 28.78 | 30.41 | 32.20 | 33.47 | 27.00 | 27.78 | 28.31 | 28.46 |
|          | FCN-CS [15] | 31.29 | 32.87 | 34.58 | 35.76 | 31.24 | 32.68 | 33.95 | 35.02 |
|          | D-AMP [30] | 30.35 | 31.87 | 33.43 | 34.42 | 28.18 | 29.69 | 30.87 | 31.59 |
|          | ConvCSNet-baseline | 31.55 | 33.10 | 34.69 | 34.5 | 31.05 | 32.22 | 32.72 | 33.64 |
|          | ConvCSNet | 31.97 | 33.73 | 35.22 | 36.55 | 31.57 | 33.16 | 34.26 | 35.11 |
| leaves | TV [29] | 17.64 | 20.10 | 24.50 | 25.08 | 17.27 | 19.81 | 23.02 | 24.94 |
|          | FCN-CS [15] | 22.13 | 24.67 | 28.82 | 31.3 | 22.06 | 24.31 | 27.27 | 30.12 |
|          | D-AMP [30] | 18.69 | 26.80 | 33.00 | 36.77 | 20.20 | 24.42 | 27.77 | 29.2 |
|          | ConvCSNet-baseline | 22.09 | 24.76 | 30.93 | 34.73 | 21.82 | 24.44 | 29.18 | 33.15 |
|          | ConvCSNet | 23.51 | 27.03 | 33.32 | 36.98 | 23.38 | 27.05 | 30.96 | 35.20 |
| raccoon | TV [29] | 24.08 | 25.66 | 27.33 | 29.12 | 23.60 | 24.60 | 25.83 | 26.47 |
|          | FCN-CS [15] | 26.17 | 27.53 | 29.82 | 31.66 | 26.16 | 27.53 | 29.43 | 31.25 |
|          | D-AMP [30] | 25.16 | 27.12 | 29.44 | 31.77 | 24.34 | 25.87 | 27.48 | 28.48 |
|          | ConvCSNet-baseline | 25.68 | 27.12 | 30.73 | 33.38 | 25.44 | 26.73 | 29.45 | 31.29 |
|          | ConvCSNet | 26.27 | 27.95 | 31.20 | 34.17 | 26.14 | 27.87 | 29.71 | 32.58 |
| Average | TV [29] | 22.54 | 24.85 | 28.00 | 30.49 | 21.88 | 23.68 | 25.55 | 26.51 |
|          | FCN-CS [15] | 25.54 | 27.65 | 30.58 | 32.52 | 25.48 | 27.27 | 29.41 | 31.46 |
|          | D-AMP [30] | 24.33 | 27.94 | 31.74 | 34.43 | 23.40 | 25.96 | 28.22 | 29.43 |
|          | ConvCSNet-baseline | 25.71 | 27.78 | 31.79 | 34.65 | 25.38 | 27.34 | 30.08 | 32.52 |
|          | ConvCSNet | 26.54 | 28.89 | 33.01 | 35.80 | 26.34 | 28.81 | 31.07 | 33.91 |
TABLE II: Average PSNR results of different methods on test set of BSD100.

| Ratio  | Noiseless   | Noisy     |
|--------|-------------|-----------|
|        | 0.05 | 0.1 | 0.2 | 0.3 | 0.05 | 0.1 | 0.2 | 0.3 |
| TV [29] | 24.09 | 26.00 | 28.46 | 30.45 | 21.96 | 22.95 | 24.17 | 25.16 |
| FCN-CS [15] | 25.92 | 27.54 | 29.85 | 31.57 | 25.89 | 27.29 | 29.03 | 30.74 |
| D-AMP [30] | 25.48 | 27.66 | 30.65 | 33.12 | 22.63 | 23.72 | 24.95 | 25.81 |
| ConvCSNet-baseline | 26.25 | 28.12 | 30.36 | 33.08 | 25.80 | 27.66 | 28.96 | 31.15 |
| ConvCSNet | 26.47 | 28.19 | 31.03 | 33.49 | 26.22 | 28.09 | 29.63 | 32.25 |

Fig. 3: Reconstructed results for Leaves and Butterfly images from noiseless CS measurements at sensing rate 0.05. The PSNR results: (b) TV [29] (17.64 dB, 19.01 dB); (c) FCN-CS [15] (22.13 dB, 22.93 dB); (d) D-AMP [30] (18.69 dB, 22.70 dB); (e) Proposed ConvCSNet (23.51 dB, 25.63 dB)

Fig. 4: Reconstructed results for Starfish images from noisy CS measurements at sensing rate 0.1. The PSNR results: (b) TV [29] (23.50 dB); (c) FCN-CS [15] (27.16 dB); (d) D-AMP [30] (25.20 dB); (e) Proposed ConvCSNet (28.72 dB)

TABLE III: Running time (second) of different methods on a test image of size 255 × 255 with measurement rate 0.2.

| Methods   | TV [29] | FCN-CS [15] | D-AMP [30] | ConvCSNet |
|-----------|---------|-------------|------------|-----------|
| Time      | 82.47   | 0.41        | 60.79      | 0.08      |

reconstructed by the proposed ConvCSNet method is significantly better than other competing methods. The proposed method can generate visually pleasant images at measurement rate 0.05, while other methods produce images with severe visual artifacts. For more visual comparisons, please refer to the supplementary material.

Regarding the computational complexity of the test methods, we report the running time of the test methods on a test image of size 255 × 255 for the noiseless case, as shown in Table III. For TV [29] and D-AMP [30] methods, the computer with Intel i7-6700 3.4GHz and 64G memory is used to run these algorithms provided by the authors. For the FCN-CS [15] and the proposed methods, a Nvidia GTX 1080Ti GPU is used to compute the CS reconstructions. From Table III, it can be seen that the time taken by the proposed ConvCSNet for reconstructing a 255 × 255 image is about 1030 times faster than the TV method [29] and 760 times faster than the D-AMP method [30]. Note that the high computational complexity of TV and D-AMP is not only due to slow convergence, but also the use of huge measurement matrices. When compared to the block-based FCN-CS method, our method is also about 5
times faster.

V. CONCLUSIONS

Compressive sensing is a new image/signal acquisition paradigm, which has potentials in high-speed and energy efficient imaging applications. However, a practical issue in the use of CS theory is the huge memory and computation requirements for sensing a whole image. While performing CS on image block level leads to efficient measurement, it will significantly decrease the reconstruction performance. In this paper, we propose a novel convolutional compressive sensing (ConvCS) method based on deep learning. In the proposed ConvCS network, the first layer conducts sensing of the whole image using a set of convolutional filters and the remaining layers performs the reconstruction of the whole image. For better CS reconstruction, a novel two-branch convolutional neural network is proposed. Through end-to-end training, both the sensing filters and the reconstruction network can be jointly optimized. Experimental results show that the proposed method significantly outperforms existing iterative and deep learning based CS methods.

REFERENCES

[1] E. J. Candes, J. K. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489–509, 2006.
[2] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.
[3] M. Duarte, M. Davenport, D. Takbar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, “Single-pixel imaging via compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, 2008.
[4] M. E. Gehm, R. John, D. Brady, R. Willett, and T. J. Schulz, “Single-shot compressive spectral imaging with a dual-disperser architecture,” Optics Express, vol. 15, no. 21, pp. 14 013–14 027, 2007.
[5] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, “Video from a single coded exposure photograph using a learned over-complete dictionary,” in Proc. of the IEEE ICCV, 2011, pp. 287–294.
[6] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, “Compressed sensing mri,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 72–82, 2008.
[7] M. Elad and M. Aharon, “Image denoising via sparse and redundant representation over learned dictionaries,” IEEE Transactions on Image Processing, vol. 15, no. 12, pp. 3736–3745, 2006.
[8] J. E. Fowler, S. Mun, and E. W. Tramel, “Block-based compressed sensing of images and video,” Foundations and Trends in Signal Processing, vol. 4, no. 4, pp. 297–416, 2012.
[9] S. Mun and J. E. Fowler, “Block compressed sensing of images using directional transforms,” in Proc. of the ICIP, 2009, pp. 3021–3024.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. of the NIPS, 2012, pp. 1097–1105.
[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. of the CVPR, 2014, pp. 580–587.
[12] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. of the CVPR, 2015, pp. 3431–3440.
[13] A. Mousavi, A. B. Patel, and R. G. Baraniuk, “A deep learning approach to structured signal recovery,” in Proc. of Annual Allerton Conf. on Comm., Control, and Computing, 2015, pp. 1336–1343.
[14] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, “Reconnet: non-iterative reconstruction of images from compressively sensed random measurements,” in Proc. of CVPR, 2016, pp. 449–458.
[15] A. Adler, D. Bourdel, M. Elad, and M. Zibulevsky, “A deep learning approach to block-based compressed sensing of images,” in Proc. of ICASSP, 2017.
[16] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” Commun. Pure Appl. Math., vol. 57, no. 11, pp. 1413–1457, 2004.