Image Compressed Sensing using Convolutional Neural Network

Wuzhen Shi, *Student Member, IEEE,* Feng Jiang, *Member, IEEE,* Shaohui Liu, *Member, IEEE,* and Debin Zhao, *Member, IEEE*

***Abstract*—In the study of compressed sensing (CS), the two main challenges are the design of sampling matrix and the development of reconstruction method. On the one hand, the usually used random sampling matrices (e.g. GRM) are signal independent, which ignore the characteristics of the signal. On the other hand, the state-of-the-art image CS methods (e.g. GSR and MH) achieve quite good performance, however with much higher computational complexity. To deal with the two challenges, we propose an image CS framework using convolutional neural network (dubbed CSNet) that includes a sampling network and a reconstruction network, which are optimized jointly. The sampling network adaptively learns the sampling matrix from the training images, which makes the CS measurements retain more image structural information for better reconstruction. Specifically, three types of sampling matrices are learned, i.e. floating-point matrix, {0,1}-binary matrix, and {-1,+1}-bipolar matrix. The last two matrices are specially designed for easy storage and hardware implementation. The reconstruction network, which contains a linear initial reconstruction network and a non-linear deep reconstruction network, learns an end-to-end mapping between the CS measurements and the reconstructed images. Experimental results demonstrate that CSNet offers state-of-the-art reconstruction quality, while achieving fast running speed. In addition, CSNet with {0,1}-binary matrix, and {-1,+1}-bipolar matrix gets comparable performance with the existing deep learning based CS methods, and outperforms the traditional CS methods. What's more, the experimental results further suggest that the learned sampling matrices can improve the traditional image CS reconstruction methods significantly.**

***Index Terms*—Compressed sensing, deep learning, convolutional neural network, sampling matrix, image reconstruction.**

I. Introduction

THE traditional image acquisition system usually first acquires a dense set of samples based on the Nyquist-Shannon sampling theorem [2], of which the sampling ratio is no less than twice the bandwidth of the signal, then compresses the signal to remove redundancy by a computationally complex compression method for storage or transmission. However, this kind of image acquisition system may not be favored in some image processing applications when the data acquisition devices must be simple (e.g. inexpensive resource-deprived sensors), or when oversampling can harm the object being captured (e.g. medical imaging). The emerging technology of compressed sensing (CS) depicts a new paradigm for image acquisition and reconstruction that implements the sampling and compression processes jointly. More specifically, the CS theory [3] shows that a signal can be recovered from many fewer measurements than suggested by the Nyquist-Shannon sampling theorem when the signal is sparse in some domain. It is well known that image has a large amount of redundant information and can be well sparsely represented. Therefore, image can be compressed and reconstructed efficiently according to CS theory.

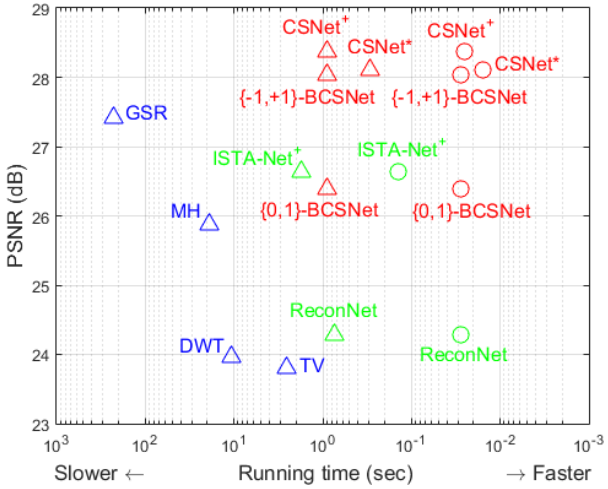

Fig. 1. The reconstruction quality and running speed comparison. △ and ○ represent the CPU implementation time and the GPU implementation time, respectively. The compared traditional methods are marked with blue font, and the compared deep learning methods are marked with green font. The chart is based on Set11 [1] results of sampling ratio 0.1 summarized in Table IV and VI.

Since the CS theory guarantees that a signal can be reconstructed with high quality at low sampling ratio when the signal is sparse in some domain, there has been significant interest in CS specifically tailored to image acquisition applicaitons. Some CS-based image acquisition devices [4], [5] have been developed. Among them, the most well-known CS device is the so-called single-pixel camera [4] developed at Rice University. Other efforts have been made in improving the imaging flexibility of CS-based camera [5] and exploring its possible use in mobile phones and other handheld devices [6]. Besides applying to image acquisition, some works apply the CS to image/video source coding [7], [8] and wireless broadcast [9], [10].

In the study of CS, the two main challenges are the design of sampling matrix and the development of reconstruction

W. Shi, F. Jiang, S. Liu and D. Zhao are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang, 150001 China, and Pengcheng Laboratory, Shenzhen, China, E-mail: {wzhshi, fjiang, shliu and dbzhao}@hit.edu.cn.

Manuscript received xx xx, xxxx; revised xx xx, xxxx.

method [11]. A lot of methods have been proposed to deal with these two challenges in the literature. To the first challenge, a variety of sampling matrices, such as the random matrix, the binary matrix [12], [13], and the structural matrix [8], [14], have been developed. However, these sampling matrices are all signal independent, which ignore the characteristics of the signal. To the second challenge, there have been a lot of sparsity-regularized-based methods proposed, including the convex-optimization algorithms (e.g. [15]–[17]), the greedy algorithms (e.g. [18]–[20]), and the iterative thresholding algorithms (e.g. [21]). For image CS, some works (e.g. TV [22], CoS [23] and GSR [24]) established more sophisticated models by exploring image priors, while others (e.g. DWT [25] and MH [26]) incorporated additional optimization criteria into the iterative thresholding algorithm. As shown in Fig. 1, some existing state-of-the-art CS reconstruction methods (e.g. GSR) cost several minutes to reconstruct one high quality image due to the need for repeated iterative operations, while other reconstruction methods (e.g. DWT and TV) have relatively lower reconstruction quality.

Recently, deep learning shows superior performance in computer vision problems. As far as we know, these also exist a few deep learning based image CS reconstruction methods, in which the random floating-point sampling matrix (e.g. Gaussian random matrix, GRM) is usually applied. These reconstruction methods include block-by-block methods (stacked denoising autoencoder (SDA) [27], non-iterative reconstruction using CNN (ReconNet) [1], and iterative shrinkage-thresholding algorithm based network (ISTA-Net) [28]) and postprocessing method (FompNet [29]). However, the block-by-block reconstruction methods [1], [27], [28] only use intra-block information to reconstruct a block resulting in blocking artifacts, therefore a postprocessing is usually needed. The postprocessing method [29] has relatively high computational complexity due to the use of orthogonal matching pursuit algorithm.

In this paper, a novel image CS framework using convolutional neural network (dubbed CSNet) is proposed to deal with the two main challenges, i.e. designing an efficient sampling matrix and developing a fast and effective reconstruction algorithm. CSNet contains a sampling network and a reconstruction network, which are optimized jointly. The sampling network adaptively learns the sampling matrix from the training images. This makes the CS measurements retain more image structural information for better reconstruction. The reconstruction network contains a linear initial reconstruction network and a non-linear deep reconstruction network to learn an end-to-end mapping between the CS measurements and the reconstructed images. The initial reconstruction network uses a convolution layer and a self-established combination layer to generate the initial reconstructed image. The non-linear deep reconstruction network further refines the initial reconstructed image to get better reconstruction quality by using deep residual network. In this work, three types of sampling matrices are learned, i.e. floating-point matrix, $\{0,1\}$-binary matrix, and $\{-1,+1\}$-bipolar matrix, especially the last two sampling matrices are designed for easy storage and hardware implementation. Experimental results demonstrate that CSNet offers state-of-the-art reconstruction quality, while achieving fast running speed. In addition, CSNet with $\{0,1\}$-binary matrix, and $\{-1,+1\}$-bipolar matrix gets comparable performance with the existing deep learning based CS methods, and outperforms the traditional CS methods. What's more, the experimental results further suggest that the learned sampling matrices can improve the traditional image CS reconstruction methods significantly.

The contributions of this work are mainly in four aspects:

- A novel image CS framework using CNN is proposed to deal with the two main challenges in CS: the design of sampling matrix and the development of reconstruction method.
- The sampling network is designed to learn the sampling matrix adaptively. This makes the CS measurements retain more image structural information for better reconstruction. Three types of sampling matrices have been learned, i.e. floating-point matrix, $\{0,1\}$-binary matrix, and $\{-1,+1\}$-bipolar matrix, especially the last two sampling matrices are designed for easy storage and hardware implementation.
- The end-to-end deep reconstruction network using residual learning is designed for recovering the image from the CS measurements. Compared with existing block-by-block reconstruction networks [1], [27], [28], the proposed reconstruction network can effectively utilize inter-block information and avoid blocking artifacts.
- The learned sampling matrices are further applied to the traditional image CS methods and can improve their performance significantly.

CSNet has other two advantages. Firstly, the learned floating-point sampling matrix has similar complexity of hardware implementation as the traditional random sampling matrix in the image acquisition system [4], and the learned $\{0,1\}$-binary and $\{-1,+1\}$-bipolar matrices are more easy for storage and hardware implementation. Based on [30], [31], using the $\{0,1\}$-binary matrix or $\{-1,+1\}$-bipolar matrix needs fewer memory size and fewer energy consumption of memory accesses, compared to the floating-point matrices. Secondly, the learned sampling matrix does not need to be transmitted from the encoder to the decoder in comparison with the traditional image CS methods.

A preliminary version of this work was presented earlier in [32]. This work improves the preliminary work in the following aspects. First, the residual learning based deep reconstruction network is designed for better reconstruction. Second, the binary and bipolar sampling matrices are designed, which are beneficial for easy storage and hardware implementation. Third, the learned sampling matrices are further applied to improve the performance of traditional image CS methods. Fourth, more comprehensive analysis and experiments are provided.

The remainder of this paper is organized as follows. The background on image CS and the related works are introduced in Section II. In Section III, the details of the image CS framework are presented. Section IV provides the experimental results. In Section V, we conclude the paper.

II. Background and Related works

The CS theory [3] permits linear projection of a signal into a dimension much lower than that of the original signal while allowing exact recovery of the signal from the projections, when the signal is sparse in some domain. Concretely, suppose that $x \in R^{N\times 1}$ is a real-valued signal and $\Phi \in R^{M\times N}$ is a sampling matrix, $M << N$, the CS measurements acquisition process is expressed as

$$y = \Phi x \tag{1}$$

where $y \in R^{M\times 1}$ is the CS measurement. Because the number of unknowns is much larger than the number of observations, recovering the original signal x from its corresponding measurements y is impossible in general due to the underdetermined property. However, if the signal x is sparse in some domain Ψ, the CS theory shows that correctly recovering x is possible. The most straightforward formulation of CS reconstruction can be expressed as

$$\min_x \|\Psi x\|_p, \quad s.t.\ y = \Phi x \tag{2}$$

where Ψx are the spare coefficients with respect to domain Ψ, and the subscript p is usually set to 1 or 0, characterizing the sparsity of the vector Ψx. There have been a large number of strategies proposed for solving this optimization problem in the literature. One kind of them is the convex optimization method, which translates the nonconvex problem into a convex one to get the approximate solution. Basis pursuit [33] is the most commonly used convex optimization method for CS reconstruction. It replaces the L0 norm constraint with the L1 norm one to get the solution by solving a linear programming problem. However, such convex-programming methods have very high computational complexity. As an alternative, the gradient-descent methods, such as iterative splitting and thresholding [15], sparse reconstruction via separable approximation [16], and gradient projection for sparse reconstruction [17], have been proposed to speed up the reconstruction process.

To reduce the computational complexity, some greedy algorithms have also been proposed for CS reconstruction. These algorithms include matching pursuit [18], orthogonal matching pursuit [19], and stage-wise orthogonal matching pursuit method [34]. Compared to convex-programming approaches, this kind of methods have relatively low computational complexity at the cost of lower reconstruction quality.

As an alternative to the matching pursuit class of CS reconstruction, techniques based on projected Landweber (PL) algorithm [35] are also proposed [11], [21], [26]. This kind of algorithms obtain the reconstructed image by successively projecting and thresholding. They not only reduce the computational complexity but also offer the possibility of easily incorporating additional optimization criteria.

For image CS, some existing works established more sophisticated models by exploring image priors. In [22], the total variation (TV) regularized constraint is used to replace the sparsity-based one for enhancing the local smoothness. In [23], Zhang et al. proposed collaborative sparsity (CoS) in an adaptive hybrid space-transform domain. In [24], Zhang et al. further proposed group sparse representation (GSR) for image compressed sensing recovery by enforcing image sparsity and non-local self-similarity simultaneously. Besides, some other image CS methods incorporated additional optimization criteria into the PL algorithm. In [11], Gan proposed block-based CS for natural images by incorporating Wiener filtering into the PL iteration, where image compressed sampling is conducted in a block-by-block manner by the same matrix. In recent years, some other improved PL-based image CS reconstruction methods [26], [36], [37] have also been proposed.

Although so many efforts have been made, the requirement of iterative computation makes these traditional methods have high computational complexity. As show in Fig. 1, these traditional image CS methods take several seconds to several minutes to reconstruct a high quality image. Besides, these sparse representation based image CS reconstruction methods need to know the sampling matrix first. That means the sampling matrix should be transmitted from the encoder to the decoder.

Recently, a few deep learning based methods have also been developed for image CS reconstruction [1], [27]–[29]. In [27], Mousavi et al. proposed a stacked denoising autoencoder (SDA) to capture statistical dependencies between the different elements of certain signals and improve signal revovery performance. However, SDA will have high computational complexity when the dimension of the signal increases because it is full connection between any two successive layers. In [1], Kulkarni et al. proposed a CNN-based reconstruction method (ReconNet), which can reduce computational complexity by weight sharing. In [28], Zhang et al. cast the iterative shrinkage-thresholding algorithm as CNN (ISTA-Net) for CS reconstruction. In [29], Bo et al. used a CNN network (FompNet) as the postprocessing of a fast orthogonal matching pursuit algorithm. FompNet has relatively higher computational complexity than the other deep learning based methods due to the use of orthogonal matching pursuit. Both SDA, ReconNet and ISTA-Net are block-by-block reconstruction methods, which may cause blocking artifact. As a result, they usually need additional deblocking. As show in Fig. 1, the deep learning based methods run faster than the traditional image CS methods.

In addition to the image reconstruction methods, another concern in the study of CS is the design of sampling matrix. In most works, the sampling matrix is a random matrix, such as a Gaussian or Bernoulli matrix, which satisfies the restricted isometry property with a large probability. Some other works design sampling matrices by taking some specific image properties into account [8], [14]. However, they always suffer high computation cost and vast storage. Recently, some researchers are interested in binary and bipolar sampling matrices that are easy to store and implement with hardware. Amini et al. [12] established the connection between the orthogonal optical codes and binary sampling matrix and introduced the bipolar sampling matrix. Lu et al. [13] proposed to determine the distribution of nonzero elements of binary sampling matrix by minimizing the coherence parameter [38] with a greedy method. However, these matrices are still signal independent, which ignores the characteristics of the signal.

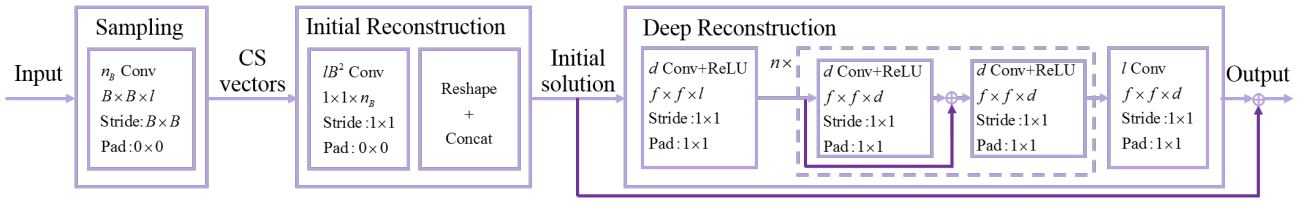

Fig. 2. The proposed framework of CSNet.

III. Convolutional Neural Network for Image Compressed Sensing

Fig. 2 shows the framework of CSNet. CSNet uses CNN to implement three functions, i.e. block-based compressed sampling, initial reconstruction and non-linear signal reconstruction, which correspond to the three operations of the block-based compressed sensing (BCS), respectively. CSNet has a sampling network and a reconstruction network. The sampling network is used to learn the sampling matrix and acquire CS measurements. The reconstruction network, which contains a linear initial reconstruction network and a non-linear deep reconstruction network, learns an end-to-end mapping from the CS measurements to the reconstructed images. The initial reconstruction network is a linear operation to obtain the initial reconstruction from the CS measurements. The deep reconstruction network is a non-linear operation to further refine the reconstruction quality.

In the training phase, the sampling network and the reconstruction network form an end-to-end network for joint optimization. Even though CSNet performs block-by-block sampling, it can effectively use both the intrablock information and the interblock correlation to optimize the reconstruceted image instead of an image block with the aid of the self-established combination layer in the initial reconstruction network. In the application phase, the sampling network is used as an encoder to generate CS measurements, and the reconstruction network is used as a decoder to reconstruct images.

A. Sampling Network

In BCS, the image is first divided into non-overlapping blocks of size $B \times B \times l$, where l represents the number of channels. Then, CS measurements are acquired using a sampling matrix Φ_B of size $n_B \times lB^2$. This process can be expressed as $y_j = \Phi_B x_j$. Intuitively, if each row of the sampling matrix Φ_B is considered as a filter, we can use a convolution layer to imitate this compressed sampling process. Since the size of each image block is $B \times B \times l$, the size of each filter in the sampling network is also $B \times B \times l$, so that each filter outputs one measurement. For the sampling ratio $\frac{M}{N}$, there are $n_B = \left\lfloor \frac{M}{N} lB^2 \right\rfloor$ rows in the sampling matrix Φ_B to obtain n_B CS measurements. Therefore, there are n_B filters of size $B \times B \times l$ in this layer. It should be noted that the stride of this convolution layer is $B \times B$ for non-overlapping sampling. Furthermore, there is no bias in each filter. We set $B = 32$ and $l = 1$ in our experiments as done by most BCS methods [11], [21], [36], [37]. Therefore, there are $n_B = \left\lfloor \frac{M}{N} lB^2 \right\rfloor = 102$ filters in this layer when the sampling ratio $\frac{M}{N} = 0.1$.

Formally, the block-based compressed sampling network (referred as S) can be expressed as an operation $S(x)$:

$$y = S(x) = W_s * x \tag{3}$$

where $*$ represents convolution operation, x is the input image, y is the CS measurement, W_s corresponds to n_B filters of support $B \times B \times l$. It is worth noting that there is no bias in this layer, and no activation function after this layer. Intuitively, the output is composed of n_B feature maps, and each column of the output is the n_B measurements of an image block.

In the training phase, the sampling network learns the sampling matrix with training images adaptively. When the weights of W_s are constrained to floating-point value, we get a floating-point matrix. When the weights of W_s are constrained to $\{0,1\}$ or $\{-1,+1\}$, we get a $\{0,1\}$-binary matrix or a $\{-1,+1\}$-bipolar matrix. The learned sampling matrices can efficiently utilize the characteristic of images, and make the CS measurements retain more image structural information for better reconstruction. In the application phase, the learned sampling matrices are used as encoder to generate CS measurements.

B. Reconstruction Network

Since natural image can usually be well sparsely represented, the CS theory [3] guarantees the image can be correctly recovered from the CS measurements. We propose to recover the image by using a reconstruction network (referred as R) that contains an initial reconstruction network (referred as I) and a deep reconstruction network (referred as D). Given the CS measurements y, the reconstruction process is formulated as

$$R(y) = D(I(y)) \tag{4}$$

Initial Reconstruction Network. Given the CS measurements, BCS usually uses a pseudo-inverse matrix to obtain the initial reconstructed image [24], [25]. That is, given the CS measurements y_j of the j^{th} block, its initial reconstruction result is $\tilde{x}_j = \tilde{\Phi}_B y_j$. Obviously, $\tilde{\Phi}_B$ is a matrix of size $lB^2 \times n_B$. Similar to the compressed sampling process, we can also use a convolution layer with special kernel size and stride to implement the initial reconstruction process. Different with previous BCS [24], [25], the matrix $\tilde{\Phi}_B$ in CSNet is optimized adaptively in a network instead of the pseudo-inverse matrix of Φ_B.

The initial reconstruction of the image can be expressed as an operation $\tilde{I}(y)$:

$$\tilde{I}(y) = W_{int} * y \tag{5}$$

where y is the CS measurement, and W_{int} is the filters. To an image block, the output of the sampling network is a $1 \times 1 \times$

n_B vector, so the size of each convolution filter in the initial reconstruction layer is $1 \times 1 \times n_B$. Besides, each image block has lB^2 elements. Therefore, W_{int} corresponds to lB^2 filters of support $1 \times 1 \times n_B$. We set the stride of this convolution layer as 1×1 to reconstruct each block. The bias is also ignored. Intuitively, each column of $\tilde{I}(y)$ is a $1 \times 1 \times lB^2$ vector corresponding to an image block of size $B \times B \times l$.

In summary, we use lB^2 convolution filters of size $1 \times 1 \times n_B$ to obtain each initial reconstructed block. However, the reconstructed output of each block is still a vector. The traditional BCS methods will reshape and concatenate these reconstructed vectors to get the initial reconstructed image. We design a combination layer, which contains a reshape function and a concatenation function, to obtain the initial reconstructed image. This layer first reshapes each $1 \times 1 \times lB^2$ reconstructed vector to a $B \times B \times l$ block, then concatenates all blocks to get an initial reconstructed image. This process is expressed as an operation $I(y)$:

$$\tilde{x} = I(y) = \kappa \begin{pmatrix} \gamma\left(\tilde{I}_{11}(y)\right) & \cdots & \gamma\left(\tilde{I}_{1w}(y)\right) \\ \vdots & \ddots & \vdots \\ \gamma\left(\tilde{I}_{h1}(y)\right) & \cdots & \gamma\left(\tilde{I}_{hw}(y)\right) \end{pmatrix} \quad (6)$$

where $\tilde{I}_{ij}(y)$ is a $1 \times 1 \times lB^2$ vector, i and j are the space indices of $\tilde{I}(y)$, h and w represent the numbers of blocks in row and column respectively, $\gamma(\cdot)$ is the reshape function that converts the $1 \times 1 \times lB^2$ vector to a $B \times B \times l$ block, $\kappa(\cdot)$ is the concatenation function that concatenates all these blocks to obtain the initial reconstructed image. The initial reconstruction provides chance to optimize the entire image rather than an independent image block, which makes our method can make full use of both intra-block and inter-block information for better reconstruction. Since there is no activation layer in the initial reconstruction network, it is a linear signal reconstruction network.

Deep Reconstruction Network. In this work, we use a residual learning based deep reconstruction network to implement the non-linear signal reconstruction process for better reconstruction. This network includes three operations: feature extraction, non-linear mapping, and feature aggregation.

The feature extraction operation is used to produce the high dimensional feature from the local receptive field. It is a convolution layer followed with an activation layer. Since the convolution layer operates on the initial reconstruction output, it has d filters of size $f \times f \times l$. This operation is expressed as an operation $D_e(\tilde{x})$:

$$D_e(\tilde{x}) = Act(W_e * \tilde{x} + B_e) \quad (7)$$

where $\tilde{x}$ is the initial reconstructed result of Eq. (6), W_e corresponds to d filters of size $f \times f \times l$, B_e is the biases of size $d \times 1$, and $Act(\cdot)$ is a specific activation function. In our experiments, we apply the most common Rectified Linear Unit (Relu, max(0,x)) [39] as the activation function.

After getting the high dimensional image feature, the deep reconstruction network alternatively cascades residual block, convolution layer and activation layer, which increases the network non-linear and its receptive field.This non-linear mapping operation is expressed as

$$D^i_{m1}(\tilde{x}) = Act\left(D^{i-1}_{m2}(\tilde{x}) + W^i_{m1} * D^{i-1}_{m2}(\tilde{x}) + B^i_{m1}\right) \quad (8)$$

$$D^i_{m2}(\tilde{x}) = Act\left(W^i_{m2} * D^i_{m1}(\tilde{x}) + B^i_{m2}\right) \quad (9)$$

where $i \in \{1, 2, \cdots, n\}$, Eq. (8) is the residual block, in which there is a short skip connection between the input and the output of a convolution layer. W^i_{m1} and W^i_{m2} contain d filters of size $f \times f \times d$, B^i_{m1} and B^i_{m2} are the biases of size $d \times 1$, $Act(\cdot)$ is also a Relu activation function. $D^0_{m2}(\tilde{x}) = D_e(\tilde{x})$.

To generate the final output, a feature aggregation operation is used to reconstruct the image from the high dimensional features. This process is expressed as an operation $D_a(\tilde{x})$:

$$D_a(\tilde{x}) = W_a * D^n_{m2}(\tilde{x}) + B_a \quad (10)$$

where W_a corresponds to l filters of size $f \times f \times d$, and B_a is the bias of size $l \times 1$. To further accelerate the network convergence, a long ship connection [40] between the initial reconstructed image $\tilde{x}$ and the output $D_a(\tilde{x})$ of the deep reconstruction network is added. As a result, the final reconstructed image is

$$D(\tilde{x}) = \tilde{x} + D_a(\tilde{x}) \quad (11)$$

C. Training Binary and Bipolar Sampling Matrices

To train the floating-point sampling matrix, we just need to calculate the gradient of the parameters and then update each parameter normally. In this subsection, we introduce how to train the binary and bipolar sampling matrices. The deterministic and stochastic binarization functions have been proposed in [31] that transforms the floating-point variables into either +1 or -1. In our work, we follow [12], [13] call {0,1}-matrix as binary matrix, and {-1,+1}-matrix as bipolar matrix.

The stochastic binarization is more appealing than the deterministic binarization, but harder to implement as it requires the hardware to generate random bits when quantizing [31]. Therefore, we apply the deterministic method to quantize the elements of the sampling matrices. The binarization function is formulated as:

$$w^b = \text{Binary}(w) = \begin{cases} 1 & \text{if } w > 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The bipolarization function is formulation as:

$$w^b = \text{Bipolar}(w) = \begin{cases} +1 & \text{if } w \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (13)$$

where w^b is the binarized or bipolarized variable.

The training process is illustrated in Algorithm 1, where λ is the learning rate, L is the number of layers, * represents convolution, and $\circ$ represents element-wise multiplication. For convenience, we ignore the combination layer in the initial reconstruction network and the ReLu layer in the deep reconstruction network because they have no parameters.

In the forward propagation, the filter w_1 of the first convolution layer (sampling layer) are first quantized using Eq.12 or Eq.13, and then the binary or bipolar filters w^b_s are used

to convolve the images as shown in Step 1 and 2. The regular forward propagation is shown in Step 3 to 5. In the backward propagation, we compute the gradients with the network prediction x_L and the target x^* in Step 6 before the regular backward propagation shown in Step 7 to 10. Then, we compute the gradients for the quantized filter w_s^b rather than the floating-point filter w_1.

After getting the gradients of all variables, a parameter updating method [41] is used to accumulate the parameter gradients. Note that we do not update the quantized filter w_s^b but the floating-point filter w_1 using the gradients $g_{w_s^b}$ of w_s^b as shown in Step 13. The other parameters are updated regularly as shown in Step 14 to 16.

When the network is well trained, we reshape each filter of size $B \times B \times l$ of the sampling layer into a $1 \times lB^2$ vector, and all these vectors form a binary/bipolar sampling matrix Φ_B of size $n_B \times lB^2$.

Algorithm 1 Training binary or bipolar matrices.

Input: a minibatch of inputs and targets (x_0, x^*), previous weights w^t, learning rate λ;
Output: updated weights w^{t+1};

A. Computing the parameters gradients:
A1. Forward propagation:
1: $w_s^b \leftarrow \mathrm{B}(w_1)$;
2: $x_1 \leftarrow w_s^b * x_0$;
3: **for** k = 2 to L **do**
4: $x_k \leftarrow w_k * x_{k-1}$;
5: **end for**

A2. Backward propagation:
6: Compute $g_{x_L} = \frac{\partial C}{\partial x_L}$ knowing x_L and x^*;
7: **for** k = L to 2 **do**
8: $g_{w_k} \leftarrow g_{x_k} \circ \frac{\partial x_k}{\partial w_k}$;
9: $g_{x_{k-1}} \leftarrow g_{x_k} \circ \frac{\partial x_k}{\partial x_{k-1}}$;
10: **end for**
11: $g_{w_s^b} \leftarrow g_{x_1} \circ \frac{\partial x_1}{\partial w_s^b}$;
12: $g_{x_0} \leftarrow g_{x_1} \circ \frac{\partial x_1}{\partial x_0}$;

B. Accumulating the parameters gradients:
13: $w_1^{t+1} \leftarrow \mathrm{Update}\left(w_1^t, g_{w_s^b}, \lambda\right)$;
14: **for** k = 2 to L **do**
15: $w_k^{t+1} \leftarrow \mathrm{Update}\left(w_k^t, g_{w_k}, \lambda\right)$;
16: **end for**

D. Training

The sampling network and the reconstruction network introduced above form a CNN-based framework for image CS. Given the input image *x*, our goal is to obtain the CS measurements *y* by using the sampling network *S*, and then recover the original input image *x* accurately from *y* by using the reconstruction network *R*. Since the output of *S* is the input of *R*, they can be merged into an end-to-end network for joint optimization without the need to consider what *y* is. Therefore, the input and the label are all image *x* itself for training CSNet. That is, the training dataset can be represented as $\{x_i, x_i\}_{i=1}^K$.

The mean square error is adopted as the cost function of CSNet. We have two objectives to minimize: the initial reconstructed image and the final reconstructed image. For the initial reconstructed image, we have the loss function

$$l_{int}(\theta, \phi) = \frac{1}{2K} \sum_{i=1}^{K} \|I(S(x_i; \theta); \phi) - x_i\|_F^2 \qquad (14)$$

where θ and ϕ are the parameters of the sampling network and the reconstruction network needed to be trained, respectively, $S(x_i; \theta)$ are CS measurements and $I(S(x_i; \theta); \phi)$ is the initial reconstructed output with respect to image x_i.

For the final output $R(S(x_i; \theta); \phi)$, we have

$$l_{deep}(\theta, \phi) = \frac{1}{2K} \sum_{i=1}^{K} \|R(S(x_i; \theta); \phi) - x_i\|_F^2 \qquad (15)$$

Training is carried out by optimized Eq. (14) and Eq. (15) simultaneously using adaptive moment estimation (Adam) [41]. It should be noted that we train the sampling network and the reconstruction network jointly, but they can be used independently.

IV. Experimental Results

In the experiments, CSNet with the floating-point matrix is named as CSNet$^+$, CSNet with {0,1}-binary sampling matrix is named as {0,1}-BCSNet, and CSNet with {-1,+1}-bipolar sampling matrix is names as {-1,+1}-BCSNet, respectively. The network in our preliminary version [32] is named as CSNet*. The test codes can be downloaded at the website: https://github.com/wzhshi/TIP-CSNet.

A. Training Details

Our training dataset has 400 images that are the 200 training images and 200 test images from the BSDS500 database [44]. The same with other work [40], the data augmentation technology is used to increase the training dataset. Specifically, we use eight data augmentation methods, i.e. the original image, flipped, rotation 90, rotation 90 plus flipped, rotation 180, rotation 180 plus flipped, rotation 270, and rotation 270 plus flipped. The training images are prepared as 96×96 pixel sub-images by cropping images with stride of 57. Finally, we randomly select 89600 sub-images for network training.

For CSNet$^+$, {0,1}-BCSNet, and {-1,+1}-BCSNet, the network parameters are set as follows: the block size in the sampling process is the same with the traditional BCS methods, i.e. $B = 32$ and $l = 1$, the spacial size of a kernel is $f = 3$, the number of feature maps in the deep reconstruction network is $d = 64$, and the amount of non-linear mapping layer in the deep reconstruction network is $n = 5$. CSNet$^+$ uses the method described in [45] to initialize weights, which is a theoretically sound procedure for networks utilizing rectified linear units. {0,1}-BCSNet, and {-1,+1}-BCSNet initialize weights by the learned weights of CSNet$^+$. For other hyper-parameters of Adam, we use the default setting. We train our model for 100 epoches and each epoch iterates 1400 times with batch size of 64. The learning rates of the first 50 epoches, the 51 to 80 epoches, and the other 20 epoches are 10^{-3}, 10^{-4} and 10^{-5}, respectively. CSNet* is implemented using the SimpleNN wrapper of MatConvNet package [46], while CSNet$^+$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2928136, IEEE Transactions on Image Processing

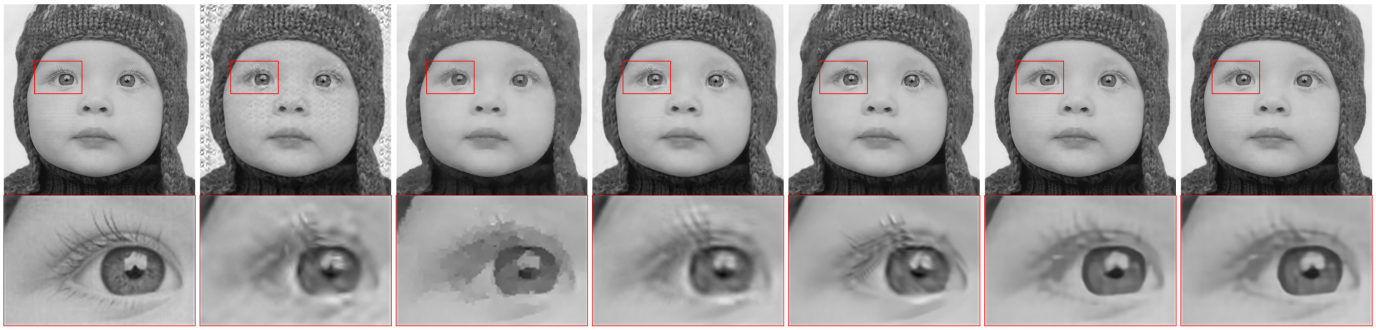

Fig. 3. Visual quality comparisons of CS recovery on *Baby* from Set5 [42] in the case of sampling ratio = 0.1.

Fig. 4. Visual quality comparisons of CS recovery on *PPT3* from Set14 [43] in the case of sampling ratio = 0.2.

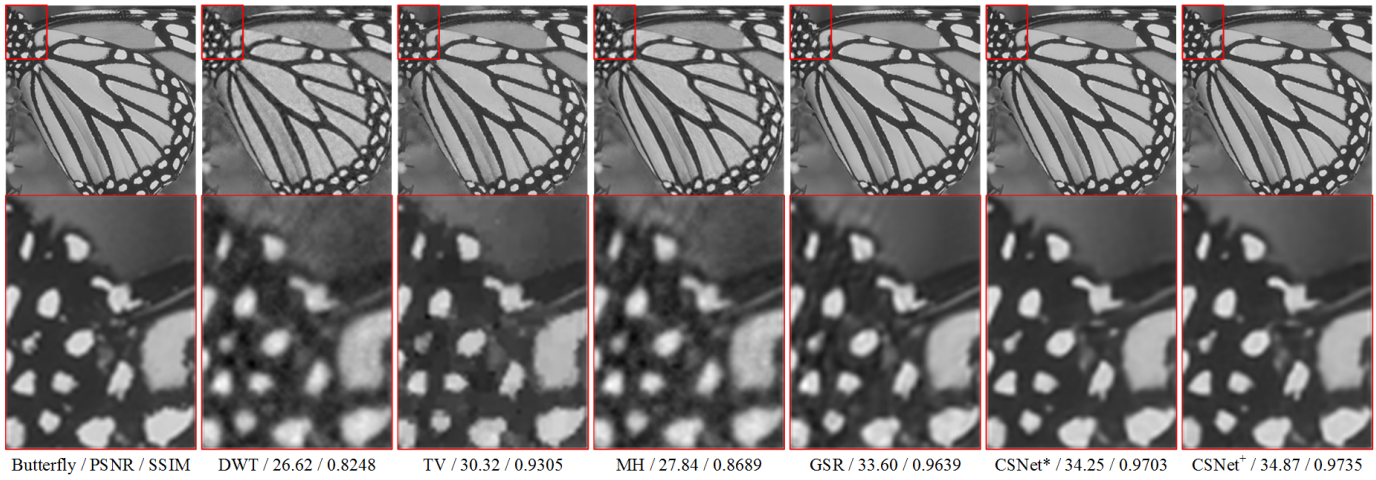

Fig. 5. Visual quality comparisons of CS recovery on *Butterfly* from Set5 [42] in the case of sampling ratio = 0.3.

{0,1}-BCSNet, and {-1,+1}-BCSNet are implemented using the DagNN wrapper of MatConvNet package.

B. Comparisons with State-of-the-Art Image CS Methods

In this section, we investigate the performance of CSNet in terms of both image reconstruction quality and running speed. First, we compare CSNet$^+$, {0,1}-BCSNet, {-1,+1}-BCSNet with four state-of-the-art traditional methods and three deep learning based methods, respectively. CSNet* is also listed for comparison. Then, we compare the running speeds of different methods. All the experiments are implemented in Matlab 2015a on Windows 7 system, and runs on desktop computer with 4 cores CPU at 3.4 GHz and 12 GB RAM, except some results are provided by the authors of the compared methods.

1) Comparisons with Traditional Methods: The four state-of-the-art traditional image CS methods to be compared are wavelet method (DWT) [25], total variation (TV) method [22], multi-hypothesis (MH) method [26] and group sparse representation (GSR) method [24]. The comparison with CoS [23] is not provided, as CoS is much time-consuming and its performance is better than MH and worse than GSR. The implementation codes of the compared methods are down-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2928136, IEEE Transactions on Image Processing

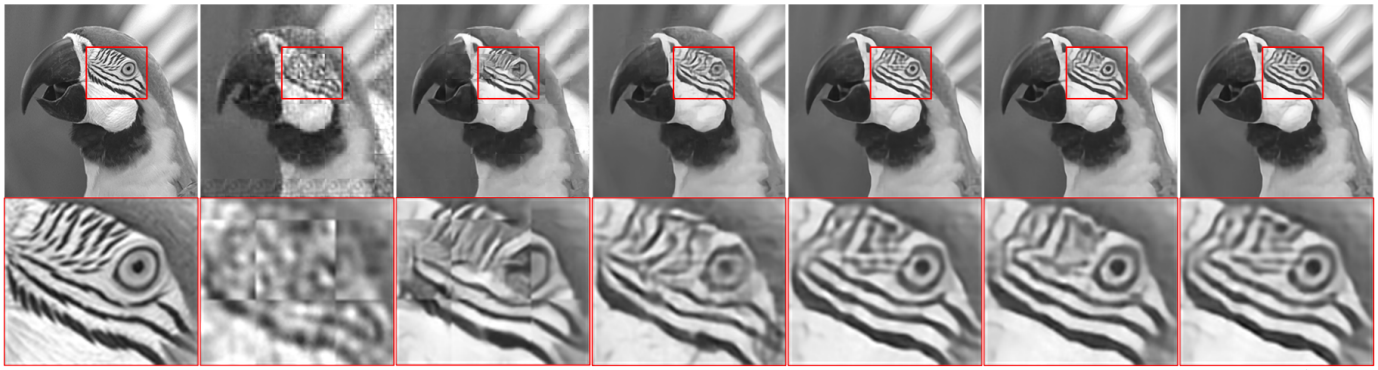

Fig. 6. Visual quality comparisons of CS recovery on *Parrots* from Set11 [1] in the case of sampling ratio = 0.1.

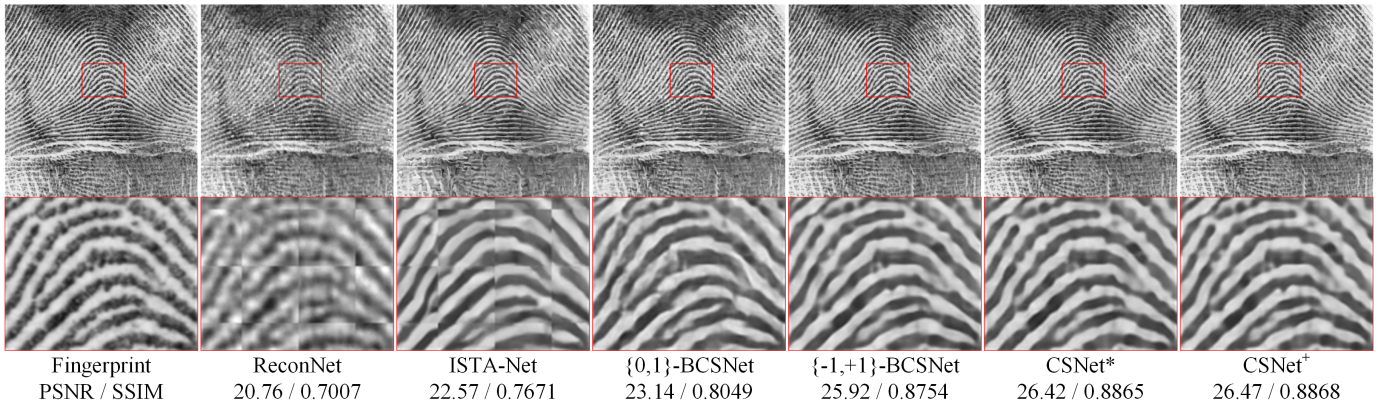

Fig. 7. Visual quality comparisons of CS recovery on *Fingerprint* from Set11 [1] in the case of sampling ratio = 0.1.

loaded from the authors' websites and the default parameters are used in the experiments. We test these methods on three datasets: Set5 [42] (5 images), Set14 [43] (14 images), and BSD100 [47] (100 images) that are widely used in the other image restoration works. Both objective evaluation and subjective evaluation are given.

Objective evaluation. The PSNR and SSIM comparisons on Set5, Set14 and BSD100 are shown in Table IV-B1, Table II and Table III in case of sampling ratio from 0.01 to 0.5, respectively. We mark the best results in bold font. As shown in Table IV-B1, CSNet$^+$ achieves the best PSNR and SSIM in comparison with CSNet*, {0,1}-BCSNet, {-1,+1}-BCSNet, and the other state-of-the-art traditional image CS methods. Compared to the best traditional method, i.e. GSR, CSNet$^+$ can improve roughly 5.31 dB, 4.79 dB, 2.60 dB, 1.88 dB, 1.42 dB, 1.30 dB and 1.14 dB on average PSNR with respect to sampling ratios of 0.01, 0.05, 0.1,0.2.0.3,0.4,0.5, respectively, and the average SSIM gains are 0.1561, 0.1215, 0.0408, 0.0224, 0.0152, 0.0114 and 0.0079, respectively. From Table II, CSNet$^+$ can improve roughly 6.82 dB, 4.66 dB, 3.69 dB and 1.88 dB on average PSNR over seven sampling ratios, in comparison with DWT, TV, MH and GSR, respectively. From Table III, CSNet$^+$ achieves the highest average PSNR and SSIM on this 100 test image dataset. All PSNR and SSIM values also show CSNet$^+$ have a better performance than CSNet*. Furthermore, the experimental results demonstrate that {0,1}-BCSNet and {-1,+1}-BCSNet still outperform the other four traditional methods.

Subjective evaluation. Three visual examples of the reconstructed images of various methods are shown in Fig. 3 to Fig. 5 with respect to sampling ratios 0.1, 0.2 and 0.3, respectively. In Fig. 3, we can see that both CSNet$^+$ and CSNet* can recover finer details than the other methods. In Fig. 4, the reconstructed results of CSNet$^+$ and CSNet* show that the text is easy to identify as shown in the enlarged part, while the results of the other methods are blurry. In Fig. 5, it is obvious that the reconstructed results of CSNet$^+$ and CSNet* are smoother, clearer and sharper than the other methods.

2) Comparisons with Other Deep Learning based Methods: The three deep learning based methods to be compared are SDA [27], ReconNet [1] and ISTA-Net [28]. For a fair comparison, we follow [1], [28] to use Set11 [1] as the test images. Table IV shows the average PSNR of different deep learning based methods. As shown, CSNet$^+$ obtains the best performance at all sampling ratios. Furthermore, the experimental results show that {0,1}-BCSNet and {-1,+1}-BCSNet still obtain comparable performance with these existing deep learning based methods. Fig. 6 and Fig. 7 are two visual examples of various deep learning based methods. There are significant blocking artifacts in the reconstruction results of ReconNet

TABLE I
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT IMAGE CS ALGORITHMS ON SET5 [42]

	DWT		TV		MH		GSR		{0,1}-BCSNet		{-1,+1}-BCSNet		CSNet*		CSNet$^+$	
Ratio	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.01	9.27	0.1402	15.53	0.4554	18.08	0.4472	18.87	0.4909	23.79	0.6358	24.07	0.6449	24.02	0.6378	**24.18**	**0.6478**
0.05	14.27	0.3559	23.16	0.6678	23.67	0.6566	24.95	0.7270	28.57	0.8164	29.39	0.8398	29.32	0.8354	**29.74**	**0.8485**
0.1	24.74	0.7680	27.07	0.7865	28.57	0.8211	29.99	0.8654	29.99	0.8512	32.20	0.8981	32.30	0.9015	**32.59**	**0.9062**
0.2	30.83	0.8749	30.45	0.8709	32.08	0.8881	34.17	0.9257	32.31	0.8977	35.24	0.9390	35.63	0.9451	**36.05**	**0.9481**
0.3	33.61	0.9050	32.75	0.9107	34.06	0.9158	36.83	0.9492	36.44	0.9545	37.22	0.9559	37.90	0.9630	**38.25**	**0.9644**
0.4	35.32	0.9249	34.89	0.9363	35.65	0.9337	38.81	0.9626	38.24	0.9672	38.62	0.9640	39.89	0.9736	**40.11**	**0.9740**
0.5	36.87	0.9409	36.75	0.9540	37.21	0.9482	40.65	0.9724	38.69	0.9701	39.23	0.9671	40.96	0.9784	**41.79**	**0.9803**
Avg.	26.42	0.7014	28.66	0.7974	29.90	0.8015	32.04	0.8419	32.57	0.8704	33.71	0.8870	34.29	0.8907	**34.67**	**0.8956**

TABLE II
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT IMAGE CS ALGORITHMS ON SET14 [43]

	DWT		TV		MH		GSR		{0,1}-BCSNet		{-1,+1}-BCSNet		CSNet*		CSNet$^+$	
Ratio	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.01	8.97	0.0989	15.26	0.3890	17.23	0.4218	17.87	0.4337	22.48	0.5529	22.74	0.5617	22.73	0.5556	**22.83**	**0.5630**
0.05	14.52	0.2933	22.24	0.5815	21.64	0.6528	22.54	0.6140	26.09	0.6941	26.67	0.7242	26.65	0.7238	**26.93**	**0.7331**
0.1	24.16	0.6798	25.24	0.6887	26.38	0.7433	27.50	0.7705	27.36	0.7564	28.78	0.8050	28.91	0.8119	**29.13**	**0.8169**
0.2	28.13	0.7882	28.07	0.7844	29.47	0.8278	31.22	0.8642	29.25	0.8158	31.55	0.8800	31.86	0.8908	**32.15**	**0.8941**
0.3	30.38	0.8389	30.12	0.8424	31.37	0.8732	33.74	0.9071	32.68	0.9119	33.47	0.9156	34.00	0.9276	**34.34**	**0.9297**
0.4	31.99	0.8753	32.03	0.8837	33.03	0.9084	35.78	0.9336	34.52	0.9383	34.81	0.9342	35.95	0.9495	**36.16**	**0.9502**
0.5	33.54	0.9044	33.84	0.9148	34.52	0.9314	37.66	0.9522	35.01	0.9450	35.56	0.9440	37.05	0.9607	**37.89**	**0.9631**
Avg.	24.53	0.6398	26.69	0.7264	27.66	0.7655	29.47	0.7822	29.63	0.8021	30.51	0.8235	31.02	0.8314	**31.35**	**0.8357**

TABLE III
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT IMAGE CS ALGORITHMS ON BSD100 [47]

	DWT		TV		MH		GSR		{0,1}-BCSNet		{-1,+1}-BCSNet		CSNet*		CSNet$^+$	
Ratio	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.01	9.63	0.1067	15.98	0.3995	18.21	0.4076	18.90	0.4431	23.49	0.5411	23.70	0.5473	23.69	0.5441	**23.76**	**0.5484**
0.05	14.81	0.2935	23.05	0.5690	21.36	0.5169	22.16	0.5682	26.04	0.6581	26.55	0.6885	26.61	0.6908	**26.78**	**0.6976**
0.1	23.46	0.6343	25.46	0.6612	25.16	0.6673	25.91	0.7071	27.05	0.7219	28.21	0.7698	28.40	0.7787	**28.53**	**0.7834**
0.2	27.26	0.7516	27.58	0.7557	28.09	0.7746	29.18	0.8156	28.65	0.7854	30.50	0.8546	30.88	0.8681	**31.05**	**0.8721**
0.3	29.23	0.8108	29.27	0.8191	29.85	0.8307	31.33	0.8723	31.67	0.8943	32.28	0.8993	32.89	0.9146	**33.08**	**0.9171**
0.4	30.72	0.8524	30.86	0.8660	31.35	0.8695	33.20	0.9096	33.41	0.9282	33.67	0.9245	34.13	0.9250	**34.91**	**0.9443**
0.5	32.17	0.8862	32.46	0.9019	32.86	0.9012	34.94	0.9359	33.95	0.9371	34.57	0.9396	36.09	0.9587	**36.68**	**0.9618**
Avg.	23.90	0.6194	26.38	0.7103	26.70	0.7097	27.95	0.7503	29.18	0.7809	29.93	0.8034	30.38	0.8114	**30.68**	**0.8178**

TABLE IV
PSNR COMPARISONS OF DIFFERENT DEEP LEARNING BASED IMAGE CS ALGORITHMS ON SET11 [1] TEST IMAGES.

Algorithm	Sampling Ratio					
	0.5	0.4	0.3	0.1	0.01	Avg.
SDA	28.95	27.79	26.63	22.65	17.29	24.66
ReconNet	31.50	30.58	28.74	24.28	17.27	26.47
ISTA-Net	37.43	35.36	32.91	25.80	17.30	29.76
ISTA-Net$^+$	38.07	36.06	33.82	26.64	17.34	30.39
{0,1}-BCSNet	35.05	34.61	32.57	26.39	20.62	29.85
{-1,+1}-BCSNet	35.57	34.94	33.42	28.03	20.93	30.58
CSNet*	37.51	36.10	33.86	28.10	20.94	31.30
CSNet$^+$	**38.52**	**36.48**	**34.30**	**28.37**	**21.03**	**31.74**

and ISTA-Net because they are block-by-block reconstruction methods. In contrast, CSNet$^+$ gets better visual effect without blocking artifacts. In [1], the authors use BM3D [48] denoiser to further deal with the blocking artifacts. Table V compares CSNet$^+$ with SDA and ReconNet that with or without BM3D [48] denoiser, in which SDA and ReconNet results are taken from [1]. As shown, CSNet$^+$ outperforms SDA and ReconNet with or without BM3D denoiser significantly.

3) Running Speed Comparisons: Table VI shows the running time comparisons. For a fair comparison, we report the running times that the authors reported in their papers for SDA, ReconNet and ISTA-Net, as well as our running results by using the authors' released codes. The running times for the traditional methods are obtained by the codes download from the authors' websites. The average running times on CPU/GPU and the their implementation platform for reconstructing a 256×256 image are listed in Table VI. As shown, traditional image CS methods take roughly several seconds to several minutes to reconstruct a 256×256 image. In contrast, deep learning based methods run faster than traditional methods, which take less than one second on CPU or 0.05 second on GPU to reconstruct a 256×256 image. Specifically, CSNet$^+$, {0,1}-BCSNet and {-1,+1}-BCSNet run faster than ISTA-Net and ISTA-Net$^+$. CSNet* runs faster than ReconNet, ISTA-Net and ISTA-Net$^+$. CSNet* also runs faster than CSNet$^+$, {0,1}-BCSNet and {-1,+1}-BCSNet. This is because CSNet* is a smaller network than CSNet$^+$, {0,1}-BCSNet and {-1,+1}-BCSNet, and CSNet* is implemented using the SimpleNN wrapper of MatConvNet package [46], while CSNet$^+$, {0,1}-BCSNet, and {-1,+1}-BCSNet are implemented using

TABLE V
PSNR COMPARISONS OF DIFFERENT DEEP LEARNING BASED IMAGE CS ALGORITHMS ON THE 4 TEST IMAGES OF SET11 [1].

Ratio	Algorithm	Barbara	Fingerprint	Flintstones	Lena	Average
0.01	SDA	18.59	14.83	13.90	17.84	17.29
	SDA + BM3D	18.76	14.82	13.95	17.95	17.40
	ReconNet	18.61	14.82	13.96	17.87	17.27
	ReconNet + BM3D	19.08	14.88	14.08	18.05	17.55
	CSNet$^+$	**21.79**	**16.29**	**16.65**	**22.45**	**19.30**
0.1	SDA	22.07	20.29	18.40	23.81	22.43
	SDA + BM3D	22.39	20.31	18.21	24.15	22.68
	ReconNet	21.89	20.75	18.92	23.83	22.68
	ReconNet + BM3D	22.50	20.97	19.18	24.47	23.23
	CSNet$^+$	**24.41**	**26.47**	**24.31**	**29.15**	**26.08**

the DagNN wrapper of MatConvNet package. The DageNN wrapper is slightly slower than the SimpleNN wrapper [46]. In conclude, CSNet runs much faster than the traditional CS methods, and have comparable running speed with the existing deep learning based CS methods.

4) Color Image CS: Our method can be applied directly to color image by setting $l = 3$. We have trained four models on the RGB color space for color image CS. For color image, the average PSNR values on Set5 are 24.35 dB, 29.23 dB, 32.08 dB, 35.19 dB at the sampling ratio of 0.01, 0.05, 0.1, 0.2, respectively. Fig.8 shows three visual examples of the reconstruction results of CSNet$^+$ at sampling ratio of 0.1. As shown, CSNet$^+$ obtains good color image CS reconstruction.

C. Traditional Methods using the Learned Sampling Matrix

As discussed before, the learned sampling matrix is signal dependent and takes the advantages of the characteristics of the given signal, which makes the CS measurements retains more image structural information for better reconstruction. In this subsection, we apply the learned sampling matrix to replace the commonly used random one to improve the traditional image CS methods.

1) Traditional Methods using the Learned Floating-Point Sampling Matrix: The learned floating-point sampling matrix is applied to four state-of-the-art image CS methods, i.e. DWT, TV, MH and GSR. Fig. 9 shows the average PSNR and SSIM comparisons between various CS methods with GRM and the learned floating-point sampling matrix on Set5. As shown, these four methods with the learned floating-point sampling matrix have significant gain on both PSNR and SSIM. Fig. 10 shows the visual quality comparison with GRM and the learned floating-point sampling matrix on *Women* from Set5 at sampling ratios of 0.1 and 0.2, respectively. As shown in Fig. 10, these four traditional image CS methods with the learned floating-point sampling matrix has significant visual quality improvement in comparison with that with GRM.

2) Traditional Methods using the Learned binary and bipolar Matrices: In this experiments, we compare these four traditional methods with {0,1}-random, {0,1}-learned, {-1,+1}-random, and {-1,+1}-learned sampling matrices, respectively. Fig. 11 shows the quantitative comparisons between different methods with different sampling matrices on Set5 in the case of the sampling ratio of 0.1 and 0.2, respectively. As shown, these four traditional methods with {0,1}-learned sampling matrix get higher PSNR than that with {0,1}-random sampling matrix. As we expect that these methods with {-1,+1}-learned sampling matrix also outperform that with {-1,+1}-random sampling matrix.

Fig. 12 provides visual quality comparisons between different methods with the four kinds of sampling matrices on *Baby* from Set5 in the case of sampling ratio of 0.1. The first row is the result that with {0,1}-random and {-1,+1}-random sampling matrices, while the second low is the result that with {0,1}-learned and {-1,+1}-learned sampling matrices. We zoom in on the part of the eye to highlight their visual differences. Obviously, each compared method using the learned sampling matrices has better visual quality than that using the corresponding random sampling matrices.

V. CONCLUSION

In this paper, a novel image CS framework (dubbed CSNet) using CNN is proposed to deal with the two challenges, i.e. the design of sampling matrix and the development of reconstruction method. CSNet includes a sampling network and a reconstruction network. The sampling network adaptively learns the sampling matrix from the training images. This results in that the learned sampling matrix takes the advantages of the characteristics of the training images to make the CS measurements retain more image structural information for better reconstruction. Three types of sampling matrices have been learned, i.e. floating-point matrix, {0,1}-binary matrix, and {-1,+1}-bipolar matrix. The last two matrices are specially designed for easy storage and hardware implementation. The reconstruction network learns an end-to-end mapping between CS measurements and the reconstructed images. By optimizing the sampling network and the reconstruction network jointly, CSNet offers state-of-the-art reconstruction quality, while achieving fast running speed. What's more, the learned sampling matrices have been successfully applied to improve the traditional image CS methods.

ACKNOWLEDGMENT

This work has been supported in part by the Major State Basic Research Development Program of China (973 Program 2015CB351804), the National Science Foundation of China under Grants No. 61872116, 61572155.

TABLE VI
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		
DWT	10.3176	-	10.5539	-	Matlab	Intel Core i7-3770 CPU
TV	2.3349	-	2.5871	-		
MH	23.1006	-	19.0405	-		
GSR	235.6297	-	230.4755	-		
SDA	-	0.0045	-	0.0029	Matlab + caffe	Intel Xeon E5-1650 CPU + NVIDIA GTX980 GPU
ReconNet(author)	0.5193	0.0244	0.5258	0.0195		
ISTA-Net(author)	0.9230	0.0390	0.9230	0.0390	Python + TensorFlow	Intel Core i7-6820 CPU + NVIDIA GTX1060 GPU
ISTA-Net$^+$(author)	1.3750	0.0470	1.3750	0.0470		
ReconNet	0.7320	0.0145	0.7348	0.0278		Intel Core i7-3770 CPU + NVIDIA GTX960 GPU
ISTA-Net$^+$	1.6894	0.1403	1.7647	0.1409		
{0,1}-BCSNet	0.9066	0.0275	0.9081	0.0281	Matlab + Matconvenet	
{-1,+1}-BCSNet	0.9115	0.0274	0.9065	0.0276		
CSNet	0.2950	0.0157	0.2941	0.0155		
CSNet+	0.8960	0.0262	0.9024	0.0257		

[a] The results of SDA and ReconNet (author) are taken from [1]. The results do not contains the running time of BM3D denoiser.
[b] The results of ISTA-Net (author) and ISTA-Net$^+$ (author) are the average runing time of seven sampling ratios taken from [28].

Fig. 8. Color visual results reconstructed by CSNet$^+$ at sampling ratio of 0.1 on Set5.

REFERENCES

[1] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 449–458.

[2] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] M. F. Duarte, M. A. Davenport, D. Takbar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[5] R. Kerviche, N. Zhu, and A. Ashok, "Information-optimal scalable compressive imaging system," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2014, pp. CM2D–2.

[6] Y. Oike and A. El Gamal, "Cmos image sensor with per-column $\sigma\delta$ adc and programmable compressed sensing," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 318–328, 2013.

[7] S. Mun and J. E. Fowler, "DPCM for quantized block-based compressed sensing of images," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1424–1428.

[8] 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 *2015 Data Compression Conference*. IEEE, 2015, pp. 133–142.

[9] W. Yin, X. Fan, Y. Shi, R. Xiong, and D. Zhao, "Compressive sensing based soft video broadcast using spatial and temporal sparsity," *Mobile Networks and Applications*, vol. 21, no. 6, pp. 1002–1012, 2016.

[10] C. Li, H. Jiang, P. Wilford, Y. Zhang, and M. Scheutzow, "A new compressive video sensing framework for mobile broadcast," *IEEE Transactions on Broadcasting*, vol. 59, no. 1, pp. 197–205, 2013.

[11] L. Gan, "Block compressed sensing of natural images," in *2007 15th International Conference on Digital Signal Processing*. IEEE, 2007, pp. 403–406.

[12] A. Amini and F. Marvasti, "Deterministic construction of binary, bipolar, and ternary compressed sensing matrices," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2360–2370, 2011.

[13] W. Lu, T. Dai, and S.-T. Xia, "Binary matrices for compressed sensing," *IEEE transations on signal processing*, vol. 66, no. 1, p. 77, 2018.

[14] K. Q. Dinh, H. J. Shim, and B. Jeon, "Measurement coding for compressive imaging using a structural measuremnet matrix," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 10–13.

[15] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[16] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[17] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[18] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[19] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[20] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[21] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[22] C. Li, W. Yin, and Y. Zhang, "TVAL3: Tv minimization by augmented lagrangian and alternating direction agorithm 2009," Available: http://www.caam.rice.edu/$\sim$optimization/L1/TVAL3/.

[23] J. Zhang, D. Zhao, C. Zhao, R. Xiong, S. Ma, and W. Gao, "Image compressive sensing recovery via collaborative sparsity," *IEEE Journal*

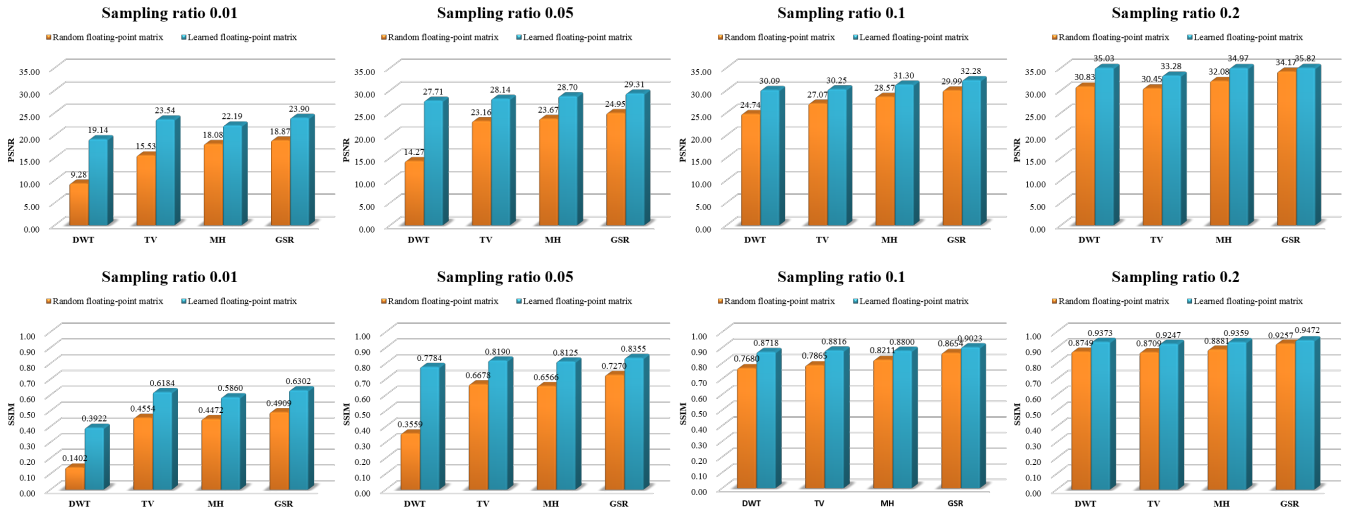

Fig. 9. Average PSNR and SSIM comparisons between various CS methods using GRM and the learned floating-point sampling matrix on Set5 [42].

Fig. 10. Visual quality comparisons between various methods with GRM and the learned floating-point sampling matrix on *Women* from Set5 [42]. R-DWT, R-TV, R-MH and R-GSR are the results of DWT, TV, MH and GSR with GRM, respectively. L-DWT, L-TV, L-MH and L-GSR are the results of DWT, TV, MH and GSR with the learned floating-point sampling matrix, respectively.

on Emerging and Selected Topics in Circuits and Systems, vol. 2, no. 3, pp. 380–391, 2012.

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

[25] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 3021–3024.

[26] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 1193–1198.

[27] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Communication, Control, and Computing*, 2016, pp. 1336–1343.

[28] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1828–1837.

[29] L. Bo, H. Lu, Y. Lu, J. Meng, and W. Wang, "FompNet: Compressive sensing reconstruction with deep learning over wireless fading channels," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2017, pp. 1–6.

[30] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *IEEE International Solid-state Circuits Conference Digest of Technical Papers*, 2014.

[31] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training neural networks with weights and

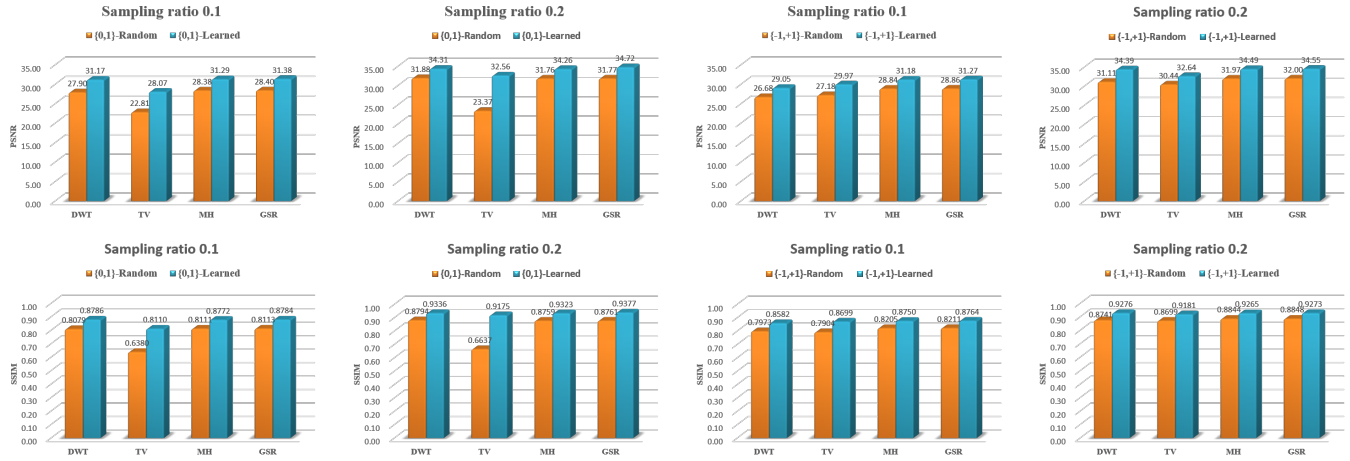

Fig. 11. Comparisons between the traditional CS methods using the two-values random sampling matrices and that using the learned ones on Set5 [42].

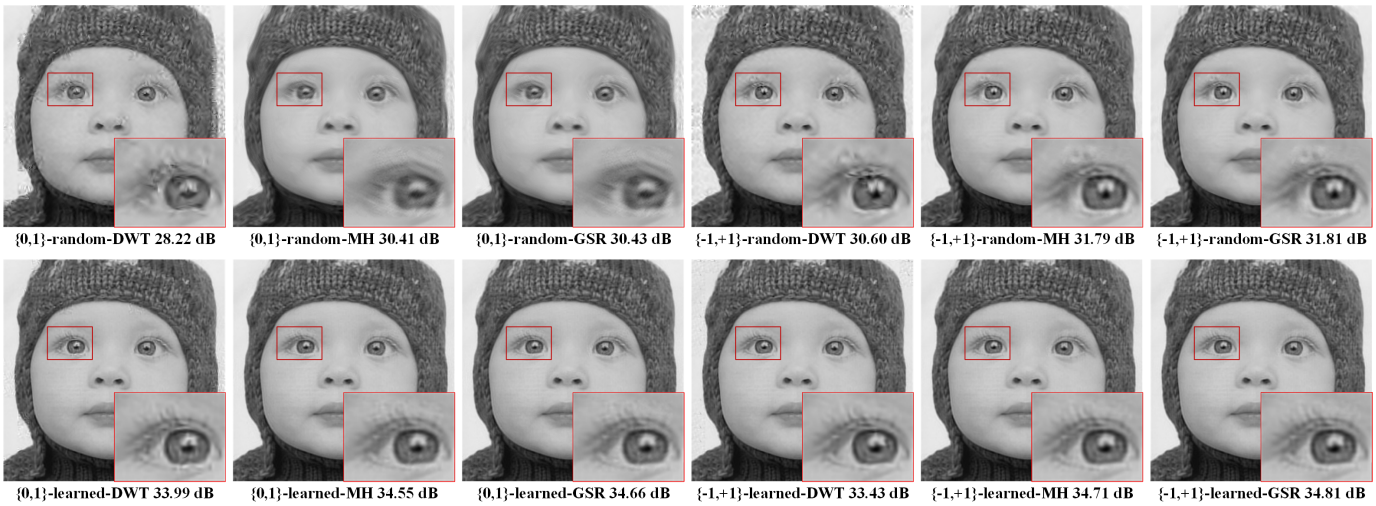

Fig. 12. Visual quality comparisons between various methods with the two-values random sampling matrices and that using the learned ones on *Baby* from Set5 [42] in the case of sampling ratio = 0.1.

activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[32] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 877–882.

[33] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[34] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[35] M. Bertero and P. Boccacci, *Introduction to inverse problems in imaging*. CRC press, 1998.

[36] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *2011 Data Compression Conference*. IEEE, 2011, pp. 183–192.

[37] J. E. Fowler, S. Mun, and E. W. Tramel, "Multiscale block compressed sensing with smoothed projected landweber reconstruction," in *Signal Processing Conference, 2011 19th European*. IEEE, 2011, pp. 564–568.

[38] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE transactions on Information theory*, vol. 49, no. 12, pp. 3320–3325, 2003.

[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[40] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," *arXiv preprint arXiv:1511.04587*, 2015.

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

[42] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *British Machine Vision Conference*, 2012.

[43] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*. Springer, 2010, pp. 711–730.

[44] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[46] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM, 2015, pp. 689–692.

[47] 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 *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.

[48] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions*

on Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.

Wuzhen Shi is now a Ph.D. candidate in the School of Computer Science and Technology, Harbin Institute of Technology (HIT), Harbin, China. He received Master degree from Northwest A & F University, Yangling, Shaanxi, China, in 2014, and received Bachelor degree from Shenyang Agricultural University, Shenyang, China, in 2012. His research interests include image processing, computer vision, and image/video coding and transmission.

Feng Jiang received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2001, 2003, and 2008, respectively. He is now an Associated Professor in the Department of Computer Science, HIT and a visiting scholar in the School of Electrical Engineering, Princeton University. His research interests include computer vision, pattern recognition and image and video processing.

Shaohui Liu received the B.S., M.S., and Ph.D. degrees in computation mathematics and its application software, computatioin mathematics, computer science from Harbin Institute of Technology(HIT), Harbin , China, in 1999, 2001, and 2007, respectively. He is now an Associated Professor in the School of Computer Science and Technology, HIT and his research interests include data compression, pattern recognition, image and video processing and multimedia security.

Debin Zhao received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 1985, 1988, and 1998, respectively.

He is now a Professor in the Department of Computer Science, Harbin Institute of Technology(HIT). He has published over 200 technical articles in refereed journals and conference proceedings in the areas of image and video coding, video processing, video streaming and transmission, and pattern recognition.