# DR<sup>2</sup>-Net: **D**eep **R**esidual **R**econstruction **N**etwork for Image Compressive Sensing

Hantao Yao, Feng Dai, Dongming Zhang, Yike Ma, Shiliang Zhang, Member, IEEE, Yongdong Zhang, Member, IEEE, Qi Tian, Fellow, IEEE

Abstract—Most traditional algorithms for compressive sensing image reconstruction suffer from the intensive computation. Recently, deep learning-based reconstruction algorithms have been reported, which dramatically reduce the time complexity than iterative reconstruction algorithms. In this paper, we propose a novel Deep Residual Reconstruction Network (DR<sup>2</sup>-Net) to reconstruct the image from its Compressively Sensed (CS) measurement. The DR<sup>2</sup>-Net is proposed based on two observations: 1) linear mapping could reconstruct a high-quality preliminary image, and 2) residual learning could further improve the reconstruction quality. Accordingly, DR<sup>2</sup>-Net consists of two components, i.e., linear mapping network and residual network, respectively. Specifically, the fully-connected layer in neural network implements the linear mapping network. We then expand the linear mapping network to DR<sup>2</sup>-Net by adding several residual learning blocks to enhance the preliminary image. Extensive experiments demonstrate that the DR<sup>2</sup>-Net outperforms traditional iterative methods and recent deep learning-based methods by large margins at measurement rates 0.01, 0.04, 0.1, and 0.25, respectively. The code of DR<sup>2</sup>-Net has been released on: https://github.com/coldrainyht/caffe\_dr2

Index Terms—Image Compressive Sensing, DR<sup>2</sup>-Net, Convolutional Neural Networks

#### I. Introduction

Compressive sensing is an emerging technique to acquire and reconstruce digital data, *e.g.*, images and videos. It is to capture the data in the form of Compressively Sensed (CS) measurements, then reconstruct the original data from these CS measurements. Because the required number of measurements is far fewer than limited by the Nyquist theory, compressive sensing is very desirable in many applications such as cameras, medical scanners and so on.

By achieving perfect image reconstruction from CS measurements, some algorithms have been proposed in recently

Hantao Yao is with Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China, and also with the University of the Chinese Academy of Sciences, Beijing 100049, China, Email: yaohantao@ict.ac.cn

Feng Dai and Yike Ma is with Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China,

Dongming Zhang National Computer Network Emergency Response Technical Team/Coordination Center of China

Shiliang Zhang is with School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China, Email: slzhang.jdl@pku.edu.cn

Yongdong Zhang, is with Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China, and also with the Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing 100048, China, Email: zhyd@ict.ac.cn

Qi Tian is with Department of Computer Science University of Texas at San Antonio, San Antonio, USA, Email: qitian@cs.utsa.edu

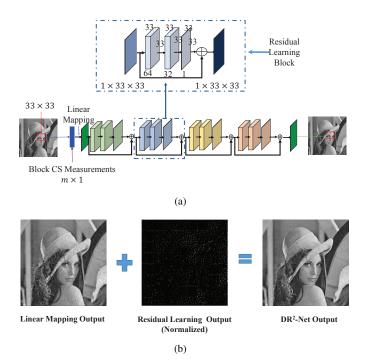


Fig. 1. (a) The framework of Deep Residual Reconstruction Network (DR<sup>2</sup>-Net). (b) Illustration of the two components for image reconstruction in DR<sup>2</sup>-Net. The right most image is the output of DR<sup>2</sup>-Net (Best viewed in color pdf).

years [1], [2], [3], [4], [5]. Most of these approaches commonly model the image signal with structured sparsity assumption and solve an optimization problem with iterative optimization strategies. The intensive computation of this iterative optimization has become the bottleneck for the application of image compressive sensing.

Deep neural networks [6] have exhibited a series of breakthroughs in computer vision tasks, *e.g.*, image classification [6], super-resolution [7], and restoration [8], *etc.* Recently, several deep neural networks have been proposed for compressive sensing image reconstruction. Thanks to the powerful learning ability, current deep learning-based methods effectively avoid the expensive computation in traditional approaches and achieve a promising reconstruction performance [9], [10].

This paper proposes a Deep Residual Reconstruction Network (DR<sup>2</sup>-Net), which further boosts the reconstruction quality from Compressively Sensed (CS) measurements with fast speed. The DR<sup>2</sup>-Net takes the CS measurement of an image

patch as input and consists of two components: linear mapping network and residual network. The linear mapping network is first used to obtain a preliminary reconstructed image, the residual network is then applied to infer the residual between the ground truth image and the preliminary reconstructed image. The residual finally updates the preliminary result as the output of DR<sup>2</sup>-Net. An intuitive description of DR<sup>2</sup>-Net is illustrated in Fig. 1.

Extensive experiments show our DR<sup>2</sup>-Net significantly outperforms existing iterative methods and deep learning-based methods. Our contributions could be summarized as follows:

- We propose a novel Deep Residual Reconstruction Network (DR<sup>2</sup>-Net) for image compressive sensing, which outperforms the existing works. With DR<sup>2</sup>, the deeplearning CS methods outperform traditional methods at all four measurement rates for the first time.
- The linear mapping generates a reasonably good preliminary reconstruction image with faster speed and lower computational cost. The linear mapping network is well-suited to image reconstruction task in computational resource and network bandwidth limited scenarios.
- The residual learning further improves the reconstruction quality. The residual learning is independent with the linear mapping stage. Therefore, the existing deep learningbased methods could flexibly fuse with residual learning to chase higher image reconstruction quality.

#### II. RELATED WORK

This work is related with image compressive sensing image reconstruction, deep learning for compressive sensing, and the deep residual network. In the following, we will review these three categories of related works.

Compressive sensing image reconstruction: Compressive sensing is an emerging technique to acquire and process digital data like images and videos. It aims to recover the source signal  $\mathbf{x} \in \mathbf{R}^{n \times 1}$  from the randomized CS measurements, i.e.,  $\mathbf{y} = \mathbf{\Phi} \mathbf{x} (m \ll n)$ , where  $\Phi \in \mathbf{R}^{m \times n}$ , and  $\mathbf{y} \in \mathbf{R}^{m \times 1}$ . Since  $m \ll n$ , the equation has multiple solutions. Thus, there exists more than one  $\mathbf{x} \in \mathbf{R}^{n \times 1}$  that can yield the same CS measurements y. To reconstruct the original signals from CS measurements, the early recovery algorithms assume the original image signal has  $l_p$ -norm  $(0 \le p \le 1)$  sparsity. Based on this assumption, researchers propose several iterative algorithms, e.g., matching pursuit [4], orthogonal matching pursuit [5], iterative hard-thresholding [11], compressive sampling matching search [12], approximate message passing [13], and iterative soft-thresholding [1], etc. Furthermore, many elaborate structures, such as total variation sparsity [14], [15], non-local sparsity [3], [16], [3], block sparsity [17], wavelet tree sparsity [18], [19] are used as prior knowledge of the original image signals, which are beneficial to compressive sensing image reconstruction. All these recovery algorithms solve an optimization problem. The iterative solvers are often used and lead to expensive computation, which has become the bottleneck for the application of image compressive sensing. Recently, some deep learning-based approaches have been proposed for this problem. Stacked Denoising Auto-encoders

(SDAs) [10] and Convolutional Neural Network (CNN) [9] are applied to compressive sensing image reconstruction. Existing works [10], [9] have shown that the deep learning-based approaches are about 3 orders of magnitude faster than traditional reconstruction algorithms.

Deep learning for compressive sensing: Deep learning has exhibited promising performance in computer vision and image processing tasks, e.g., semantic segmentation [20], depth estimation [21], image super-resolution [7] and image denoising [22]. Recently, some works introduce the deep learning into image/video reconstruction. For compressive image recovery, Kulkarni etal. [9] propose a CNN-based ReconNet to achieve non-iterative compressive sensing image reconstruction. In [10], Stacked Denoising Auto-encoders (SDAs) are employed to learn a mapping between CS measurements and image blocks. For compressive video recovery, Iliadis etal. [23], [24] propose Deep Fully-Connected Network, where the encoder learns binary sensing mask and the decoder determines the reconstruction of the video. These approaches effectively avoid the expensive computation in traditional approaches and have achieved promising image/video reconstruction performance.

Deep Residual Network: Recently, the deep Residual Network (ResNets) [25] has achieved promising performance on several computer vision tasks. Compared with the traditional Convolutional Neural Network, e.g., AlexNet [6], VGG [26], and GoogleNet [27], the ResNets introduces identity shortcut connections that directly pass the data flow to later layers, thus effectively avoids signal attenuation caused by multiple stacked non-linear transformations. As a consequence, deeper network can be constructed with ResNets and faster training speed can be achieved. By going deeper in the network, ResNet generally gets better performance in comparison with other deep learning models.

Among the related works, the most similar work to ours is the one by Kulkarni etal. [9], who propose a ReconNet for compressive sensing image reconstruction. However, the proposed Deep Residual Reconstruction Network (DR<sup>2</sup>-Net) is different from the one in [9] in the aspects of both motivation and network structure. The method in [9] is the first work that employ CNN for compress sensing. The design of ReconNet [9] is inspired by the SRCNN [], and does not take the essential of CS into consideration. Therefore, it is not a carefully-designed network for CS image reconstruction. By give more analysis for compressive sensing problem, we found that simply using the linear mapping (fully-connected operation) could perfectly suitable to solve the CS, and it is not a optimal choice to train ReconNet in an end-to-end way. The DR<sup>2</sup>-Net firstly trains a linear mapping network to obtain a reasonably good preliminary reconstruction image, then trains the residual network to further improve the reconstruction quality. We find that this strategy is more suitable for applying CNN on image compressive sensing.

#### III. DEEP RESIDUAL RECONSTRUCTION NETWORK

As shown in Fig. 1(a), DR<sup>2</sup>-Net takes the Compressively Sensed (CS) measurements of the  $33 \times 33$  sized image patch

as input, and outputs a  $33 \times 33$  reconstructed image. The DR<sup>2</sup>-Net contains a linear mapping sub-network  $\mathcal{F}^f(\cdot)$  and a residual sub-network  $\mathcal{F}^r(\cdot)$ .  $\mathcal{F}^f(\cdot)$  generates a preliminary reconstructed image.  $\mathcal{F}^r(\cdot)$  infers the residual between the preliminary image and ground truth image. In the following, we firstly give details of linear mapping and residual learning. Then, we introduce the architecture of DR<sup>2</sup>-Net. Finally, we summarize the procedure that DR<sup>2</sup>-Net reconstructs an image.

#### A. Linear Mapping

Compressive sensing signal reconstruction recovers the signal  $\mathbf{x} \in \mathbf{R}^{n \times 1}$  from its CS measurement  $\mathbf{y} \in \mathbf{R}^{m \times 1}$ . The mapping from  $\mathbf{y}$  to  $\mathbf{x}$  can be regarded as an approximate linear mapping, *i.e.*,

$$\mathbf{x} = \mathbf{W}\mathbf{y},\tag{1}$$

where  $\mathbf{W} \in \mathbf{R}^{n \times m}$  is the mapping matrix. Eq. (1) is an overdetermined equation, therefore it has no accurate solution. However, we could estimate a mapping matrix  $\mathbf{W}^f$ , which makes  $||\mathbf{x} - \mathbf{W}^f \mathbf{y}||_2^2$  have the minimum error.

We denote the CS measurement and its corresponding source signal as  $\mathbf{y}_i \in \mathbf{R}^{m \times 1}$  and  $\mathbf{x}_i \in \mathbf{R}^{n \times 1}$ , respectively. The training data containing N training samples can be denoted as  $\{(\mathbf{y}_1, \mathbf{x}_1), (\mathbf{y}_2, \mathbf{x}_2), ....., (\mathbf{y}_N, \mathbf{x}_N)\}$ . Based on this training data, we could get  $\mathbf{W}^f$  by solving the following optimization, *i.e.*,

$$\mathbf{W}^f = arg \min_{\mathbf{W}} ||\mathbf{X} - \mathbf{W}\mathbf{Y}||_2^2, \tag{2}$$

where  $X = [x_1, x_2, ....., x_N]$ , and  $Y = [y_1, y_2, ....., y_N]$ .

Eq. (2) presents a linear mapping function, which could be effectively simulated by fully-connected layer in deep learning. Consequently, we employ the linear mapping network  $\mathcal{F}^f(\cdot)$  composed of fully connected layer to infer the optimal mapping matrix  $\mathbf{W}^f$  with the loss function in Eq. (3), *i.e.*,

$$L(\{\mathbf{W}^f\}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathcal{F}^f(\mathbf{y}_i, \mathbf{W}^f)||_2^2,$$
(3)

where  $\mathcal{F}^f(\mathbf{y}_i, \mathbf{W}^f)$  is the output of linear mapping network for input  $\mathbf{y}_i$ .

The network  $\mathcal{F}^f(\cdot)$  contains one fully-connected layer with 1,089 neurons. By using Stochastic Gradient Descent (SGD) training algorithm, we could obtain an optimal mapping matrix  $\mathbf{W}^f$  corresponding to the minimum Eq. (2) on all training samples.

With the trained linear mapping network  $\mathcal{F}^f(\cdot)$ , a preliminary reconstructed image  $\widehat{\mathbf{x}}_i$  could be obtained for any given CS measurement  $\mathbf{y}_i$ . We denote the procedure of generating the preliminary reconstructed image as

$$\widehat{\mathbf{x}}_i = \mathcal{F}^f(\mathbf{y}_i, \mathbf{W}^f). \tag{4}$$

### B. Residual Learning

The linear mapping network is trained to chase an approximate solution to the source signal  $\mathbf{x}_i$ , because it is hard to get an accurate solution for Eq. (1). To further narrow down the gap between  $\hat{\mathbf{x}}_i$  and  $\mathbf{x}_i$ , the residual learning network is introduced to estimate the gap between the two signals. In

other words, the next step is to infer the residual  $d_i$  based on  $\widehat{\mathbf{x}}_i$ , *i.e.*,

$$\mathbf{d}_i = \mathbf{x}_i - \widehat{\mathbf{x}}_i. \tag{5}$$

Residual learning block is introduced in this part to estimate the residual  $d_i$  from  $\hat{x_i}$ . As illustrated in Fig. 1(b), the output of residual learning is fused with the output of linear mapping as the final image construction result.

We firstly give a brief introduction to residual learning. Given an input  $\mathcal{X}$ , we denote its desired underlying mapping performed by a few stacked layers in neural network as  $\mathcal{H}(\mathcal{X})$ . Most traditional methods expect to directly learn the mapping  $\mathcal{H}(\cdot)$ , e.g., using several convolutional operations to fit  $\mathcal{H}(\cdot)$  [26]. If the output  $\mathcal{H}(\mathcal{X})$  is similar to  $\mathcal{X}$ , we expect the  $\mathcal{H}(\cdot)$  is an identity mapping. However, it is difficult to optimize  $\mathcal{H}(\cdot)$  as the identity mapping in practice. On the other hand, for the case that the  $\mathcal{H}(\mathcal{X})$  is similar to  $\mathcal{X}$ , the residuals between  $\mathcal{H}(\mathcal{X})$  and  $\mathcal{X}$  would be closed to zero. As the weights of convolutional layers are always initialized to have zeromeans, the convolutional layers could be easily trained to approximate the residual  $\mathcal{F}(\mathcal{X})$  in Eq. (6). Therefore, rather than expect stacked layers to approximate  $\mathcal{H}(\cdot)$ , we explicitly let these layers approximate the residual function in Eq. (6). More details of residual learning can be found in [25].

$$\mathcal{F}(\mathcal{X}) := \mathcal{H}(\mathcal{X}) - \mathcal{X}. \tag{6}$$

As a consequence, the convolutional layers initialised with zero-means could be trained to estimate the residual  $\mathbf{d}_i$  in Eq. (5). We implement the residual learning with residual learning blocks shown in Fig. 1(a). Each block contains three convolution layers to simulate the complicated non-linear mappings. The residual network  $\mathcal{F}^r(\cdot)$  consists of several residual learning blocks.

Specifically, with  $\widehat{\mathbf{x}}_i$  as input, residual network  $\mathcal{F}^r(\widehat{\mathbf{x}}_i, \mathbf{W}^r)$  generates an estimated residual  $\widehat{\mathbf{d}}_i$ , where  $\mathbf{W}^r$  is the parameters for residual network. We denote this procedure as

$$\widehat{\mathbf{d}}_i = \mathcal{F}^r(\widehat{\mathbf{x}}_i, \mathbf{W}^r). \tag{7}$$

The DR<sup>2</sup>-Net takes the CS measurement  $\mathbf{y}_i$  as input, first obtains the preliminary reconstructed image  $\hat{\mathbf{x}}_i$ , then fuses it with the estimated residual  $\hat{\mathbf{d}}_i$  from residual network as the final result, *i.e.*,

$$\mathbf{x}_i^* = \widehat{\mathbf{x}}_i + \widehat{\mathbf{d}}_i. \tag{8}$$

By replacing the  $\hat{\mathbf{x}}_i$  and  $\hat{\mathbf{d}}_i$  with Eq. (4) and Eq. (7), the final  $\mathbf{x}_i^*$  is obtained by two networks, *i.e.*,

$$\mathbf{x}_{i}^{*} = \mathcal{F}^{f}(\mathbf{y}_{i}, \mathbf{W}^{f}) + \mathcal{F}^{r}(\mathcal{F}^{f}(\mathbf{y}_{i}, \mathbf{W}^{f}), \mathbf{W}^{r}). \tag{9}$$

The DR<sup>2</sup>-Net is trained with SGD according to the loss function in Eq. (10). The training procedure obtains the optimal parameters  $\mathbf{W}^f$  and  $\mathbf{W}^r$ , respectively.

$$L(\{\mathbf{W}^f, \mathbf{W}^r\}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{x}_i^*||_2^2.$$
 (10)

# C. DR<sup>2</sup>-Net Architecture

DR<sup>2</sup>-Net processes the CS measurement computed from a  $33 \times 33$  sized image block. The linear mapping network  $\mathcal{F}^f(\cdot)$  takes the CS measurement  $\mathbf{y} \in \mathbf{R}^{m \times 1}$  as input, where m is various with Measurement Rates (MRs), e.g., m =272, 109, 43, and 10 corresponding to MR = 0.25, 0.10, 0.04, and 0.01, respectively. The linear mapping network consists of one fully-connected layer with 1089 neurons. We reshape its 1089-dim output to the size of  $33 \times 33$  as the preliminary reconstructed image.

As shown in Fig. 1(a), the residual network  $\mathcal{F}^r(\cdot)$  takes preliminary result as input, and outputs one image with size  $33\times33$ . The residual network contains four residual learning blocks and each block contains three convolutional layers. In each block, the first convolution layer uses kernel of size  $11\times1$  and generates 64 feature maps. The second layer produces 32 features maps with  $1\times1$  kernel. The third layer generates 1 feature map with  $7\times7$  kernel. To generate the feature map with size  $33\times33$ , we add corresponding padding on each layer. Inspired by previous work [25], we further add batch normalization [28] to the first two convolutional layers of each block. Each convolutional layer is followed by a ReLU layer except the last convolutional layer.

#### D. Reconstruction Procedure

Given an image, we firstly divide it into  $33 \times 33$  image patches with no overlap, and extract the CS measurement on each patch. Next, DR²-Net takes the CS measurements as input and outputs the reconstructed patches. The reconstructed patches compose an intermediate reconstructed image, which is finally processed with BM3D [29] to remove the artifacts caused by block-wise processing.

# IV. DR<sup>2</sup>-NET TRAINING

# A. Training Data Generation

We use the same set of 91 images used in [9] to generate the training data for DR<sup>2</sup>-Net. We first resize the original images to three scales *i.e.*, 0.75, 1, and 1.5, to construct a scale space. Then, on each scale, we extract  $33 \times 33$  sized image patches with stride 14. This process finally samples 86,656 image patches from 273 images. For each image patch, we firstly extract its luminance component, denoting the luminance component as  $\mathbf{x}_i$  and then compute its CS measurement  $\mathbf{y}_i = \mathbf{\Phi} \mathbf{x}_i$ , where  $\mathbf{\Phi}$  is a random Gaussian matrix.  $(\mathbf{y}_i, \mathbf{x}_i)$  is thus an input-output pair for DR<sup>2</sup>-Net training, where  $\mathbf{y}_i$  is the network input and  $\mathbf{x}_i$  is its ground truth label.

# B. Training Strategy

The training procedure of DR<sup>2</sup>-Net consists of two steps for two sub-networks  $\mathcal{F}^f(\cdot)$  and  $\mathcal{F}^r(\cdot)$ , respectively. The first step uses a relatively large learning rate with step strategy to train the network  $\mathcal{F}^f(\cdot)$ . Parameters like maximum number of iterations, learning rate, stepsize, and gamma are set as 1,000,000, 0.001, 200,000, and 0.5, respectively. With the

trained  $\mathbf{W}^f$ , the second step trains DR<sup>2</sup>-Net in an end-toend manner in up to 100,000 iterations with a smaller fixed learning rate set as 0.00001. Note that, the second step updates both the  $\mathbf{W}^f$  and  $\mathbf{W}^r$ .

#### V. EXPERIMENTS

In this section, we conduct a series of experiments to test the reconstruction performance of Deep Residual Reconstruction Network (DR<sup>2</sup>-Net).

# A. Implementation Details

We use Caffe [32] to implement and train the DR<sup>2</sup>-Net. The weights for  $\mathcal{F}^f(\cdot)$  are initialised using gaussian distribution with standard variance 0.01. The weights of convolutional layers in DR<sup>2</sup>-Net are initialised using gaussian distribution with standard variance 0.001. The batch-size is set as 128.

The experiments in Sect V-G are conducted on the ImageNet Val dataset [33]. The other experiments are conducted on the standard training and testing images described in [9].

## B. Comparison with Existing Methods

Firstly, we compare our DR<sup>2</sup>-Net with five existing methods, *i.e.*, TVAL3 [2], NLR-CS [30], D-AMP [31], SDA [10], and ReconNet [9]. The first three are iterative-based methods, and the last two are deep learning-based methods. The results are summarized in Table I, where the best results are highlighted in bold.

As shown in Table I, the proposed DR<sup>2</sup>-Net obtains the highest mean PSNR values at four Measurement Rates (MRs). Compared with the existing methods, DR<sup>2</sup>-Net has following advantages:

- 1) Different from the existing deep learning-based methods, *i.e.*, SDA [10] and ReconNet [9], DR<sup>2</sup>-Net is more robust for the CS measurement at higher MRs. From Table I, we could observe that the existing deep learning-based methods do not perform as good as DR<sup>2</sup>-Net at higher MRs. For example at MR 0.25, the highest PSNR for deep learning-based methods is 25.92dB [9], which is obviously lower than 28.19dB [30] of iterative-based method. Our DR<sup>2</sup>-Net outperforms both the other deep learning-based methods and iterative-based methods. For instance, at MR 0.25, the DR<sup>2</sup>-Net outperforms NLP-CS and ReconNet by 0.87dB and 3.14dB, respectively on PSNR.
- 2) Similar to the existing deep learning-based methods, DR<sup>2</sup>-Net is robust for the CS measurement at smaller MRs. As shown in Table I, the iterative-based methods do not work well at smaller MRs, *e.g.*, at MR 0.01 and 0.04. While the deep learning-based methods, *e.g.*, SDA and ReconNet, both achieve higher mean PSNR values. However, DR<sup>2</sup>-Net also constantly outperforms the other deep learning-based methods at lower MRs.

# C. Evaluation on Linear Mapping

To verify that the linear mapping is able to reconstruct a high-quality preliminary image, we compare linear mapping with ReconNet [9] from two aspects: PSNR value and testing

TABLE I

PSNR values in dB for testing images by different algorithms at different Measurement Rates. For TVAL3, NLR-CS, D-AMP, and SDA, we use the results reported in [9]. The reconstruction results for those images are shown in Fig. 8. "w/o BM3D" means without applying BM3D, and "w/ BM3D" denotes using BM3D. "Mean PSNR" is the mean PSNR value among all 11 testing images.

Image Name	Methods	MR=0.25		MR=0.10		MR=		MR=0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
	TVAL3 [2]	24.19	24.20	21.88	22.21	18.98	18.98	11.94	11.96
Barbara	NLR-CS [30]	28.01	28.00	14.80	14.84	11.08	11.56	5.50	5.86
	D-AMP [31]	25.08	25.96	21.23	21.23	16.37	16.37	5.48	5.48
	SDA [10]	23.19 23.25	23.20 23.52	22.07 21.89	22.39 22.50	20.49 20.38	20.86 21.02	18.59 18.61	18.76 19.08
	ReconNet [9] DR <sup>2</sup> -Net	25.23	25.32 25.99	21.89 22.69	22.30 22.82	20.38	21.02	18.65	19.08 <b>19.10</b>
	TVAL3	22.70	22.71	18.69	18.70	16.04	16.05	10.35	10.37
Fingerprint	NLR-CS	23.52	23.52	12.81	12.83	9.66	10.03	4.85	5.18
	D-AMP	25.17	23.87	17.15	16.88	13.82	14.00	4.66	4.73
	SDA	24.28	23.45	20.29	20.31	16.87	16.83	14.83	14.82
	ReconNet	25.57	25.13	20.75	20.97	16.91	16.96	14.82	14.88
	DR <sup>2</sup> -Net	27.65	27.75	22.03	22.45	17.40	17.47	14.73	14.95
Flinstones	TVAL3	24.05	24.07	18.88	18.92	14.88	14.91	9.75	9.77
	NLR-CS	22.43	22.56	12.18	12.21	8.96	9.29	4.45	4.77
	D-AMP	25.02	24.45	16.94	16.82	12.93	13.09	4.33	4.34
	SDA	20.88	20.21	18.40	18.21	16.19	16.18	13.90	13.95
	ReconNet	22.45	22.59	18.92	19.18	16.30	16.56	13.96	14.08
	DR <sup>2</sup> -Net	26.19	26.77	21.09	21.46	16.93	17.05	14.01	14.18
	TVAL3	28.67	28.71	24.16	24.18	19.46	19.47	11.87	11.89
	NLR-CS	29.39	29.67	15.30	15.33	11.61	11.99	5.95	6.27
Lena	D-AMP	28.00	27.41	22.51	22.47	16.52	16.86	5.73	5.96
	SDA	25.89	25.70	23.81	24.15	21.18	21.55	17.84	17.95
	ReconNet	26.54	26.53	23.83	24.47	21.28	21.82	17.87	18.05
	DR <sup>2</sup> -Net	29.42	29.63	25.39	25.77	22.13	22.73	17.97	18.40
	TVAL3 [2]	27.77	27.77	21.16	21.16	16.73	16.73	11.09	11.11
	NLR-CS [30]	25.91	26.06	14.59	14.67	11.62	11.97	6.38	6.71
Monarch	D-AMP [31]	26.39	26.55	19.00	19.00	14.57	14.57	6.20	6.20
	SDA [10]	23.54	23.32	20.95	21.04	18.09	18.19	15.31	15.38
	ReconNet [9]	24.31	25.06	21.10	21.51	18.19	18.32	15.39	15.49
	DR <sup>2</sup> -Net	27.95	28.31	23.10	23.56	18.93	19.23	15.33	15.50
Parrot	TVAL3	27.17	27.24	23.13	23.16	18.88	18.90	11.44	11.46
	NLR-CS	26.53	26.72	14.14	14.16	10.59	10.92	5.11	5.44
	D-AMP	26.86	26.99	21.64	21.64	15.78	15.78	5.09	5.09
	SDA ReconNet	24.48 25.59	24.36 26.22	22.13 22.63	22.35 23.23	20.37 20.27	20.67 21.06	17.70 17.63	17.88 18.30
	DR <sup>2</sup> -Net	28.73	20.22 29.10	23.94	24.30	21.16	21.85	18.01	18.30 18.41
	TVAL3	28.81	28.81	23.86	23.86	19.20	19.20	11.86	11.88
	NLR-CS	29.11	29.27	14.82	14.86	10.76	11.21	5.38	5.72
Boats	D-AMP	29.26	29.26	21.95	21.95	16.01	16.01	5.34	5.34
Douis	SDA	26.56	26.25	24.03	24.18	21.29	21.54	18.54	18.68
	ReconNet	27.30	27.35	24.15	24.10	21.36	21.62	18.49	18.83
	DR <sup>2</sup> -Net	30.09	30.30	25.58	25.90	22.11	22.50	18.67	18.95
	TVAL3	25.69	25.70	21.91	21.92	18.30	18.33	11.97	12.00
	NLR-CS	24.88	24.96	14.18	14.22	11.04	11.43	5.98	6.31
Cameraman	D-AMP	24.41	24.54	20.35	20.35	15.11	15.11	5.64	5.64
	SDA	22.77	22.64	21.15	21.30	19.32	19.55	17.06	17.19
	ReconNet	23.15	23.59	21.28	21.66	19.26	19.72	17.11	17.49
	DR <sup>2</sup> -Net	25.62	25.90	22.46	22.74	19.84	20.30	17.08	17.34
	TVAL3	35.42	35.54	28.69	28.74	20.63	20.65	10.97	11.01
Foreman	NLR-CS	35.73	35.90	13.54	13.56	9.06	9.44	3.91	4.25
	D-AMP	35.45	34.04	25.51	25.58	16.27	16.78	3.84	3.83
	SDA	28.39	28.89	26.43	27.16	23.62	24.09	20.07	20.23
	ReconNet	29.47	30.78	27.09	28.59	23.72	24.60	20.04	20.33
	DR <sup>2</sup> -Net	33.53	34.28	29.20	30.18	25.34	26.33	20.59	21.08
House	TVAL3	32.08	32.13	26.29	26.32	20.94	20.96	11.86	11.90
	NLR-CS	34.19	34.19	14.77	14.80	10.66	11.09	4.96	5.29
	D-AMP	33.64	32.68	24.84	24.71	16.91	17.37	5.00	5.02
	SDA	27.65	27.86	25.40	26.07	22.51	22.94	19.45	19.59
	ReconNet	28.46	29.19	26.69	26.66	22.58	23.18	19.31	19.52
	DR <sup>2</sup> -Net	31.83	32.52	27.53	28.40	23.92	24.70	19.61	19.99
Peppers	TVAL3 NLR-CS	<b>29.62</b> 28.89	<b>29.65</b> 29.25	22.64 14.93	22.65 14.99	18.21 11.39	18.22 11.80	11.35	11.36 6.10
	D-AMP	28.89	29.25 28.58	21.39	21.37	16.13	16.46	5.77 5.79	5.85
	SDA	29.84 24.30	28.58 24.22	21.39 22.09	22.34	19.63	19.89	16.93	17.02
	ReconNet	24.30	24.22 25.16	22.09	22.34 22.67	19.63	20.00	16.93	17.02
	DR <sup>2</sup> -Net	28.49	25.16	22.15 23.73	24.28	20.32	20.00 20.78	16.82	16.96 17.11
	TVAL3	28.49	27.87	23.73	22.86	18.39	18.40	11.31	11.34
	NLR-CS	28.05	28.19	14.19	14.22	10.58	10.98	5.30	5.62
Mean PSNR	D-AMP	28.03	28.19	21.14	21.09	15.49	15.67	5.19	5.23
IVICALI I SIVIN	SDA	24.72	24.55	22.43	22.68	19.96	20.21	17.29	17.40
	ReconNet	25.54	25.92	22.68	23.23	19.99	20.21	17.27	17.55
	DR <sup>2</sup> -Net	28.66	29.06	24.32	24.71	20.80	21.29	17.44	17.80

TABLE II

Comparison of loss and PSNR between  $\mathrm{DR}^2$ -Net and ReconNet [9].  $fc_{1089}$  denotes the linear mapping network with one fully-connected layer containing 1089 neurons.  $Res_n$  denotes that further adding n residual learning blocks to the linear mapping network. "MR" denotes Measurement Rate.

Models	MR=0.01		MR=0.04		MR=0.10		MR=0.25	
	Loss	PSNR/BM3D	Loss	PSNR/BM3D	Loss	PSNR/BM3D	Loss	PSNR/BM3D
ReconNet	5.307	17.27/17.55	2.34	19.98/20.44	1.139	22.6793/23.23	0.517	25.53/25.92
$fc_{1089}$	5.42	17.26/17.60	2.25	20.05/20.46	1.01	23.10/23.39	0.398	26.81/27.33
$fc_{1089}$ - $Res_1$	5.2066	17.37/17.71	1.9946	20.59/21.03	0.7991	24.08/24.55	0.2880	28.41/28.81
$fc_{1089}$ - $Res_2$	5.1831	17.40/17.79	1.9586	20.68/21.05	0.7756	24.28/24.71	0.2814	28.62/29.01
$fc_{1089}$ - $Res_3$	5.1896	17.44/17.80	1,9638	20.75/20.93	0.7696	<b>24.32</b> /24.71	0.2801	28.64/ <b>29.09</b>
$fc_{1089}$ - $Res_4$	5.1983	17.41/17.75	1.9452	20.80/21.29	0.7650	24.25/ <b>24.71</b>	0.2788	<b>28.66</b> /29.06

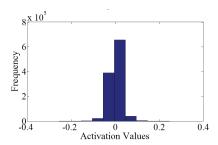


Fig. 2. Histogram of the activation values for residual learning.

loss. Note that, the ReconNet contains one fully-connected layer and six convolutional layers, and our linear mapping network  $\mathcal{F}^f(\cdot)$  only contains one fully-connected layer with 1089 neurons, denoted by  $fc_{1089}$ . The related results are summarized in Table II.

From Table II, we could see that the linear mapping network  $fc_{1089}$  outperforms ReconNet both for PSNR and testing loss at MRs 0.25, 0.10, and 0.04, respectively. At MR 0.01, the testing loss for network  $fc_{1089}$  is 5.42, which is larger than the 5.307 of ReconNet. However, the network  $fc_{1089}$  achieves a comparable PSNR value with ReconNet, *i.e.*, 17.26dB vs 17.27dB of ReconNet. We further compare the time complexity between  $fc_{1089}$  and ReconNet. As shown in Table III, the network  $fc_{1089}$  is ten times faster than ReconNet. Therefore, we could conclude that the linear mapping could generate a reasonably good preliminary reconstruction image with fast speed.

#### D. Evaluation on Residual Learning

As the linear mapping network has produced a preliminary reconstruction, the residual estimated by the residual learning should be small. As shown in Fig. 2, we could observe that residual learning network reasonably infers many small residual values, *e.g.*, more than 95% of the activation values are within the range of -0.05 to 0.05.

The linear mapping network is first trained. Then, DR<sup>2</sup>-Net is trained in an end-to-end manner. To show the validity of this training strategy, we analyze the network loss on the validation set during training, and summarize results in Fig. 3. As shown in Fig. 3, the loss for linear mapping network, *i.e.*, the blue solid lines, keeps stable during training. We also observe that DR<sup>2</sup>-Net loss, *i.e.*, the red lines, decreases as the iteration number increases. Note that, the DR<sup>2</sup>-Net loss is composed of

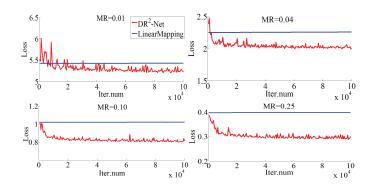


Fig. 3. The network losses on validation set during training procedure. The blue line and red line denote the loss of linear mapping network and DR<sup>2</sup>-Net, respectively.

the linear mapping network loss and residual learning network loss. As the DR<sup>2</sup>-Net loss keeps decreasing and linear mapping network loss is stable, it is easy to infer that the residual learning network is effectively optimized during the network training procedure.

Next, we show the effect of residual learning on reconstruction quality. As shown in Table II, adding residual block substantially reduces the testing loss and improve the reconstruction quality. For example, compared to linear mapping network  $fc_{1089}$ , the deep residual network  $fc_{1089}$ - $Res_1$  achieves 0.09 dB, 0.54dB, 0.92dB, 1.6dB improvement for MR 0.01, 0.04, 0.10, 0.25, respectively.

We finally evaluate the performance of DR<sup>2</sup>-Net with different number of residual learning blocks. As shown in Table II, we could observe that introducing more residual learning blocks increases the PSNR performance and reduces the loss for most cases, e.g., using four residual blocks achieves the highest performance at MR 0.04, 0.10, 0.25. It is also clear that the improvement becomes relatively smaller as more residual blocks are added to DR<sup>2</sup>-Net. For example, when MR = 0.25,  $fc_{1089}$ -Res<sub>4</sub> only shows 0.25dB improvement over  $fc_{1089}$ -Res<sub>1</sub>, which is smaller than 1.6dB improvement achieved by  $fc_{1089}$ -Res<sub>1</sub> over  $fc_{1089}$ . Therefore, residual learning is important for improving the reconstruction quality of DR<sup>2</sup>-Net. However, it is not necessary to add too many residual blocks into the DR<sup>2</sup>-Net. We add four residual blocks in DR<sup>2</sup>-Net and use the network  $fc_{1089}$ -Res<sub>4</sub> as DR<sup>2</sup>-Net.

TABLE III Time (in seconds) for reconstructing a single  $256\times256$  image.

Models	MR=0.01	MR=0.04	MR=0.10	MR=0.25
SDA	0.0045	0.0025	0.0029	0.0042
ReconNet	0.0326	0.0301	0.0309	0.0461
$fc_{1089}$	0.0028	0.0032	0.0037	0.0041
$fc_{1089}$ - $Res_1$	0.0190	0.0200	0.0198	0.0185
$fc_{1089}$ - $Res_2$	0.0344	0.0315	0.0349	0.0365
$fc_{1089}$ - $Res_3$	0.0508	0.0480	0.0433	0.0505
$fc_{1089}$ - $Res_4$	0.0600	0.0576	0.0565	0.0557

# E. Time Complexity

As the time complexity is a key factor for image compressive sensing, we analysis the time complexity of DR<sup>2</sup>-Net. The related results are summarized in Table III. As the deep learning-based methods have been sped up 100 times over traditional iterative algorithms [9], we only compare the time complexity between DR<sup>2</sup>-Net and other deep-learning based methods.

Firstly, we show the time complexity of DR<sup>2</sup>-Net with different structures. From Table III, we could observe that the time complexity is linearity correlated with the depth of network, *i.e.*, deeper DR<sup>2</sup>-Net requires more running time. Among the five networks, the network  $fc_{1089}$  contains only one fully-connected layer, thus shows the fastest speed, *e.g.*, only needs 3-4ms to reconstruct a  $256 \times 256$  image. The deepest network  $fc_{1089}$ -Res<sub>4</sub> contains one fully-connected layer and 12 convolutional layers, and is about 30 times slower than  $fc_{1089}$ .

Next, we make comparison with other deep learning-based methods: SDA[10] and ReconNet[9]. Compared with SDA and ReconNet, the linear mapping network  $fc_{1089}$  spends less time to achieve a comparable performance. Therefore, when the running time is an important concern, simply using network  $fc_{1089}$  could be a good choice. From Table III and Table II, we could observe that DR<sup>2</sup>-Net with structure  $fc_{1089}$ -Res<sub>1</sub> not only achieves a faster speed, but also obtains a higher performance than ReconNet. As the other three DR<sup>2</sup>-Nets contain more layers, they are slower than ReconNet.

#### F. Robustness to Noise

To show the robustness of DR<sup>2</sup>-Net to noise, we investigate the reconstruction performance under the presence of measurement noise. We firstly add the standard Gaussian noise to the CS measurements of testing set, *i.e.*, we add five levels of noise corresponding to  $\sigma = 0.01, 0.05, 0.1, 0.25$  and 0.5, where  $\sigma$  is the standard variance for the Gaussian noise. Then, the DR<sup>2</sup>-Net trained on the noiseless CS measurements takes the noisy CS measurements as input, and outputs the reconstruction images. Aiming to make a fair comparison, we do not use BM3D [29] to denoise the reconstruction images. The results are summarized in Fig. 4.

It could be observed that the DR<sup>2</sup>-Net outperforms the ReconNet for the  $\sigma$  = 0.01, 0.05, 0.1 at four MRs. For the cases that  $\sigma$  = 0.25 and 0.5, DR<sup>2</sup>-Net and ReconNet both do not work well because intense noises generated by large  $\sigma$  significantly distract the original CS measurements.

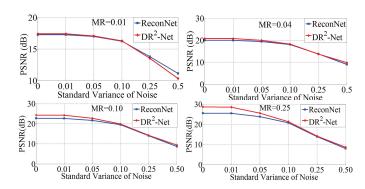


Fig. 4. Comparison of robustness to Gaussian noise between ReconNet and  $\ensuremath{\mathsf{DR}}^2\text{-Net}.$ 

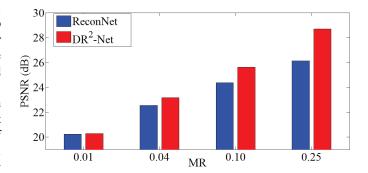


Fig. 5. Comparison between DR<sup>2</sup>-Net and ReconNet on ImageNet Val dataset at four MRs.

#### G. Scalability on Large-scale Dataset

To verify the scalability of DR<sup>2</sup>-Net, we conduct experiments to compare DR<sup>2</sup>-Net with ReconNet on the large-scale ImageNet Val dataset [33], which contains 50,000 images from 1,000 classes. Note that, DR<sup>2</sup>-Net and ReconNet are both trained based on the standard 91 images in [9].

As shown in Fig. 5, DR<sup>2</sup>-Net gets better performance than ReconNet at four MRs. Especially for cases at higher MRs, DR<sup>2</sup>-Net substantially outperforms ReconNet by large margins, *e.g.*, DR<sup>2</sup>-Net achieves nearly 3dB improvement over ReconNet at MR 0.25.

Fig. 6 shows several testing images and the quality of their reconstruction by DR<sup>2</sup>-Net and ReconNet, respectively at MR 0.10. It is clearly that DR<sup>2</sup>-Net constantly outperforms ReconNet. We notice that images with higher reconstruction quality are generally smooth while those with lower reconstruction quality contain richer textures. Therefore, the texture complexity affects the construction quality of both DR<sup>2</sup>-Net and ReconNet.

## VI. CONCLUSION

Inspired by the existing works that deep learning-based methods could dramatically reduce the time complexity than iterative reconstruction algorithms. We propose a Deep Residual Reconstruction Network to reconstruct the image from its Compressively Sensed (CS) measurements. The proposed network is composed of a linear mapping network and several



(a) Images with high quality reconstruction.



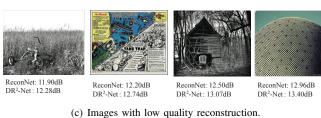
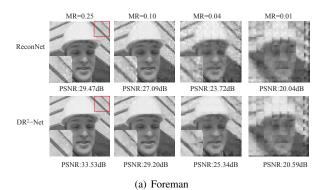


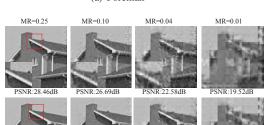
Fig. 6. Images with different reconstruction quality. The results show the texture complexity affects the reconstruction quality of both DR<sup>2</sup>-Net and ReconNet. The PSNR values are reported at MR 0.10.

residual learning blocks. The fully-connected layer in neural network is taken as the linear mapping to produce the preliminary reconstruction. The residual learning blocks are trained to boost the preliminary result by inferring the residual between preliminary reconstruction and the groudtruth. Extensive experiments show that DR<sup>2</sup>-Net outperforms the traditional iterative-based and deep learning-based methods by large margins in the aspects of both speed and quality.

#### REFERENCES

- [1] 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.
- [2] C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented lagrangian method with applications to total variation minimization," Computational Optimization and Applications, vol. 56, no. 3, pp. 507-530, 2013.
- [3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in ICCV. IEEE, 2009, pp. 2272-
- [4] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," TSP, vol. 41, no. 12, pp. 3397-3415, 1993.
- [5] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," TIT, vol. 53, no. 12, pp. 4655-4666, 2007.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS, 2012, pp. 1097-
- [7] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in ECCV. Springer, 2014, pp.
- [8] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," arXiv preprint arXiv:1605.00366, 2016.
- [9] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed random measurements," CVPR, 2016.





PSNR · 23 92dB

PSNR 19 99dB

PSNR:31 83dB PSNR:27 53dB

DR2-Net

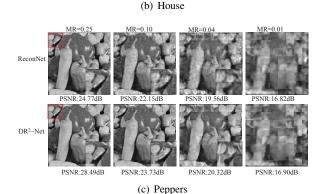


Fig. 7. Reconstruction results from noiseless CS measurements without BM3D. It is obvious that DR<sup>2</sup>-Net recovers more visually appealing images (Best viewed in color pdf).

- [10] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2015, pp. 1336-1343.
- [11] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," Applied and Computational Harmonic Analysis, vol. 27, no. 3, pp. 265-274, 2009.
- [12] D. Needell and J. A. Tropp, "Cosamp: iterative signal recovery from incomplete and inaccurate samples," Communications of the ACM, vol. 53, no. 12, pp. 93-100, 2010.
- [13] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," NAS, vol. 106, no. 45, pp. 18914-18 919, 2009.
- [14] Y. Xiao, J. Yang, and X. Yuan, "Alternating algorithms for total variation image reconstruction from random projections," Inverse Problems and Imaging, vol. 6, no. 3, pp. 547-563, 2012.
- C. Li, W. Yin, and Y. Zhang, "Tval3: Tv minimization by augmented lagrangian and alternating direction agorithm 2009."
- W. Dong, G. Shi, X. Li, L. Zhang, and X. Wu, "Image reconstruction with locally adaptive sparsity and nonlocal robust regularization," Signal Processing: Image Communication, vol. 27, no. 10, pp. 1109-1122, 2012
- [17] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," TSP, vol. 58, no. 6, pp.

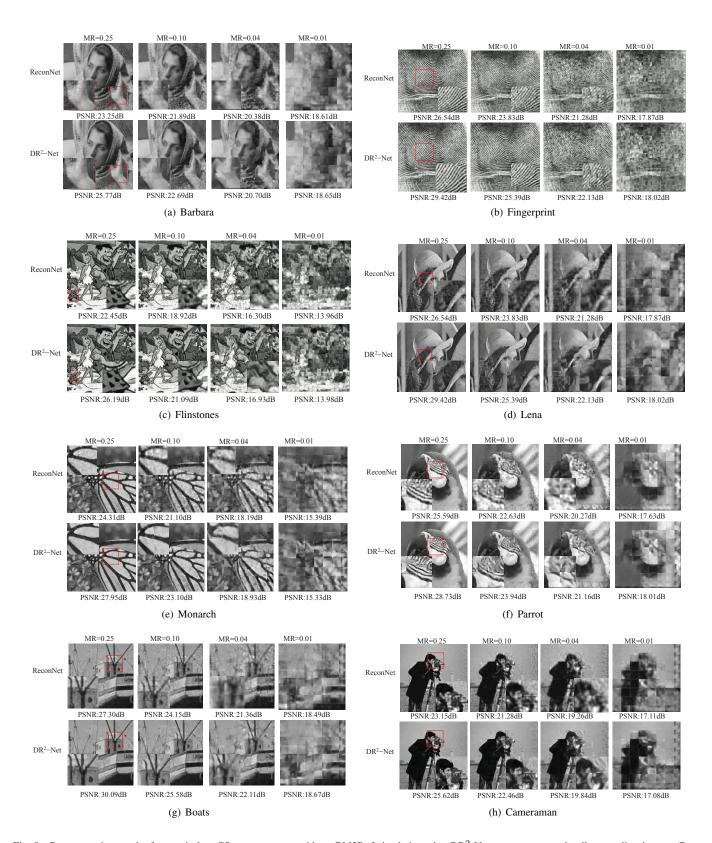


Fig. 8. Reconstruction results from noiseless CS measurements without BM3D. It is obvious that  $DR^2$ -Net recovers more visually appealing images (Best viewed in color pdf).

- 3042-3054, 2010.
- [18] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," TIT, vol. 56, no. 4, pp. 1982–2001, 2010.
- [19] C. Hegde, P. Indyk, and L. Schmidt, "A fast approximation algorithm for tree-sparse recovery," in ISIT. IEEE, 2014, pp. 1842–1846.
- [20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in CVPR, 2015, pp. 3431–3440.
- [21] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in NIPS, 2014, pp. 2366–2374.
- [22] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in NIPS, 2012, pp. 341–349.
- [23] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deepbinarymask: Learning a binary mask for video compressive sensing," arXiv preprint arXiv:1607.03343, 2016.
- [24] —, "Deep fully-connected networks for video compressive sensing," arXiv preprint arXiv:1603.04930, 2016.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, 2016.
- [26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in CVPR, 2015, pp. 1–9.
- [28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ICML*, 2015.
  [29] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising
- [29] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *TIP*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [30] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *TIP*, vol. 23, no. 8, pp. 3618–3632, 2014.
- [31] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," 2014.
- [32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.
- [33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.