Extremely Lightweight Neural Network for Large-Scale Image Compressive Sensing

Gang-Xuan Lin
Institute of Information Science
Academia Sinica, Taiwan, ROC
spybeiman@iis.sinica.edu.tw

Shih-Wei Hu
Institute of Information Science
Academia Sinica, Taiwan, ROC
vacuityhu@iis.sinica.edu.tw

Chun-Shien Lu
Institute of Information Science
Academia Sinica, Taiwan, ROC
lcs@iis.sinica.edu.tw

***Abstract*—In restoring a large-scale image from incomplete samples, one always introduces a blocking strategy to avoid a considerable memory burden. Nevertheless, certain applications, like medical imaging, reject to divide an image into patches. In this paper, we derive a variant of ADMM algorithm, which induces a non-convex optimization-based neural network architecture, for large-scale image compressive sensing. In the literature, the largest scale of natural images that can be reconstructed without blocking strategy is around 256×256, whereas our method is feasible for an image size as high as 1024×1024. In addition, our method can obtain better reconstruction performance and smaller model size than state-of-the-art methods using blocking strategy.**

I. Introduction

Compressive sensing (CS) [3], [9] has received considerable attention in diverse fields, including natural image reconstruction [28], [29] and medical image reconstruction [2], [10], [14], [16], [23], [26]. With the help of deep learning models, CS is able to achieve real-time recovery.

Formally, let $x_0 \in \mathbb{R}^n$ be the original signal and let $\Phi \in \mathbb{R}^{m \times n}$ be the sensing matrix, the CS encoding process can be formulated as:

$$y = \Phi x_0, \tag{1}$$

where $y \in \mathbb{R}^m$ is the measurement vector. The ratio $\frac{m}{n}$, ranging from 0 to 1, is defined as the measurement rate (MR).

A challenging problem is compressive sensing of large-scale images. Considering a large-scale image with the size of $w \times h$ and its vectorization $I_0 \in \mathbb{R}^{w \cdot h}$. We can see that the size of corresponding sensing matrix Φ will be $m \times wh$ and the memory storage cost of Φ is $m \cdot wh \cdot 4 \cdot 2^{-30}$ gigabytes, provided that "float" is used. The memory cost required for storing the sensing matrix Φ associated with different image sizes is shown in Table I under the MR $\frac{m}{n} = 10\%$.

TABLE I
Memory (in GB) required for storing sensing matrix Φ under different image sizes.

Image Size	Memory Size (GB)
256×256	1.60
512×512	25.6
1024×1024	409.6
2048×2048	6553.6

As indicated in Table I, the memory cost plus the derived computation cost are often not affordable. For example, if an ADMM-based algorithm [1] is adopted, its matrix-inverse operator requires huge memory storage for storing the inverse matrix. Besides, even the current learning-based methods heavily rely on multiple or powerful graphics processing units (GPUs) for training, it is still impracticable in some cases. Therefore, the objective of this paper is to study this large-scale compressive sensing problem.

Traditionally, we have two strategies for dealing with large-scale CS problems. The first and most common one is divide-and-conquer, also known as block-based CS (BCS) [12], [22]. BCS, however, does not meet the requirement of Magnetic Resonance Imaging (MRI) [10], [14], [16], [23], [26].

The second strategy relies on the use of structured sensing matrix (SRM) [7], which consists of three operators: a random sampling operator (D), an orthonormal linear operator (F), and a uniform random permutation operator (R). SRM allows the encoding process to operate without being explicitly represented in a matrix form.

In this work, we study a highly lightweight neural network for solving a large-scale CS problem. Main contributions of this paper are as follows:

- Different from traditional ADMM that leads to a calculation of matrix inverse operator, we derive ADMM in a different way to avoid matrix inverse operator.
- A non-convex ℓ_q-norm based convolutional neural network, dubbed QADM-Net, is proposed. To the best of our knowledge, QADM-Net is the first network architecture that can reconstruct a 2D image of the size larger than 1024×1024 from its incomplete samples with a single GPU without resorting to blocking strategy.

II. Related Works

Recent compressed sensing methods for natural image reconstruction are considered in deep learning-based paradigm with convolutional neural network, which achieve outstanding performance, compared to traditional iterative algorithms. The network architectures for natural image reconstruction can be classified into two categories: algorithm unrolling [20], [26]–[29] and heuristic design [15], [18], [21], [24], [25]. The main difference between them is that algorithm unrolling connects the network architecture with the traditional iterative

algorithm, which implies the trained network is interpretable [11], [19].

Nevertheless, all of the related studies commonly employ the blocking strategy to avoid the requirement of impracticable memory storage in a training process. For example, [15], [18], [27], [28] divided an image into sub-images of 33×33 pixels and merged the individual reconstructed results into an integrated one. [24], [25] first cut an image into multiple patches of size 96×96 and further divided each patch into 9 sub-images of size 32×32 for sensing. Under this setting, the GPU memory consumes around 8 GB during training (including the back-propagation process). It will become impracticable if an image of size larger than 128×128 is directly sensing without relying on the blocking strategy.

In general, an MR image has the size at least 128×128 (always 256×256 or larger) according to the scanned body parts. Moreover, since data acquisition of an MR image is carried out in k-space, which is an array of numbers representing spatial frequencies in the MR image or the Fourier transform of the MR image, we suffer from the unavailability of the blocking strategy. This explains the need of a compressive sensing method that can deal with large-scale images without resorting to the blocking strategy.

III. Proposed Method

A. Problem Formulation

Let $X_0 \in \mathbb{R}^{n_1 \times n_2}$ be an image of size $n_1 \times n_2$ and let $x_0 \in \mathbb{R}^n$ be its vector representation, where $n = n_1 \cdot n_2$. In order to reconstruct the original signal x_0 from the available measurement vector y, we consider the ℓ_q-norm minimization problem (referred as (ℓ_q)-problem) in the form

$$\min_x F(x) = \frac{1}{2} \|y - \Phi x\|_2^2 + \lambda \|\Psi x\|_q^q, \tag{2}$$

where λ is a positive constant, $0 < q < 1$, $\Psi \in \mathbb{R}^{n \times n}$ is a dictionary that allows x_0 to be sparsely represented, and $\|x\|_q = \sum_{i=1}^n (|x_i|^q)^{1/q}$ denotes the ℓ_q-norm. We adopt the non-convex ℓ_q-norm instead of convex ℓ_1-norm minimization problem (refer as LASSO) because the (ℓ_q)-problem has better reconstruction results under low measurement rates [4], [5].

It is noted that the discussions regarding an (ℓ_q)-problem or effective algorithms to solve it are largely ignored in the literature. In III-B, we first recall the traditional ADMM algorithm, which leads to a matrix-inverse operator. We then propose a modified version of ADMM algorithm in III-C to prevent from suffering the high-computation cost and high capacity of matrix-inverse operator. After modifying the ADMM algorithm, we finally propose an iterative algorithm to solve problem (2), which is called QADM algorithm.

B. ADMM algorithm

In the traditional ADMM algorithm [1], Problem (2) can be rewritten as a constrained minimization problem

$$\begin{cases} \min\limits_{x,z} & \frac{1}{2} \|y - \Phi x\|_2^2 + \lambda \|z\|_q^q \\ \text{s.t.} & z = \Psi x. \end{cases} \tag{3}$$

After some derivations, the x-subproblem of ADMM algorithm has the form:

$$x^{t+1} = \left(\Phi^T \Phi + \beta \Psi^T \Psi\right)^{-1} \left(\Phi^T y + \Psi^T \xi^t + \beta \Psi^T z^t\right).$$

We observe that the solution to the x-subproblem involves a matrix-inverse operator and at least one of the measurement matrix Φ or the dictionary Ψ is a learning parameter. Under the circumstance, we have to calculate a matrix-inverse operation in each batch and in each iteration process. If the matrix size is large enough, the computation cost is not affordable. For example, if an image size is 64×64, the matrix size of $\left(\Phi^T \Phi + \beta \Psi^T \Psi\right)$ becomes $64^2 \times 64^2$ and the matrix-inverse operation calculated by MATLAB platform with Intel Core i7-7700K CPU needs 1.18 seconds. If the image size is 128×128, it costs 36.29 seconds. Moreover, if an image size is 256×256, the storage capacity of the matrix $\left(\Phi^T \Phi + \beta \Psi^T \Psi\right)$ costs 32 GB in total (including the inverse matrix operator). It can be seen that the ADMM algorithm is almost impracticable for general devices such as mobile phone or laptop provided the blocking strategy is not permitted for large images like MRI.

C. QADM algorithm

To reconstruct a large-scale image, we derive the ADMM algorithm in the different way to avoid the matrix-inverse operator. The problem (2) can be reformulated into the constrained minimization problem

$$\begin{cases} \min\limits_{x,z} & \frac{1}{2} \|z\|_2^2 + \lambda \|\Psi x\|_q^q \\ \text{s.t.} & z = y - \Phi x \end{cases} \tag{4}$$

in that z replaces the measurement noise term $y - \Phi x$. This indicates z is to control the error (or the measurement noise), and Ψx is to control the sparsity in a specified domain. Then, after some derivations, the ADMM procedure becomes

$$\begin{cases} z^{t+1} = \arg\min\limits_z \frac{1}{2} \|z\|_2^2 + \xi^{tT} z \\ \qquad\qquad + \frac{1}{2} \beta \|z - (y - \Phi x^t)\|_2^2 \\ x^{t+1} = \arg\min\limits_x \lambda \|\Psi x\|_q^q + \xi^{tT} \Phi x \\ \qquad\qquad + \frac{1}{2} \beta \left\|z^{t+1} - (y - \Phi x)\right\|_2^2 \\ \xi^{t+1} = \xi^t + \beta \left[z^{t+1} - (y - \Phi x^{t+1})\right]. \end{cases} \tag{5}$$

The iteration process in (5) consists of three parts: the z-subproblem, the x-subproblem, and the dual variable updates.

To solve z-subproblem, because it is a convex minimization problem, by Fermat's theorem, we have $z^{t+1} + \xi^t + \beta \left[z^{t+1} - (y - \Phi x^t)\right] = 0$, which implies

$$z^{t+1} = \frac{\beta}{1+\beta} \left((y - \Phi x^t) - \frac{1}{\beta} \xi^t\right). \tag{6}$$

To solve x-subproblem, the objective function can be rewritten as

$$\lambda \|\Psi x\|_q^q + \tfrac{1}{2} \beta \left\|\Phi x - \left(y - z^{t+1} - \tfrac{1}{\beta} \xi^t\right)\right\|_2^2 + C_1, \tag{7}$$

where C_1 is a constant.

Because the minimum solution to Eq. (7) cannot be obtained directly, we approximate (7) with a polynomial via the Taylor series expansion. Let

$$f(x) = \|\Phi x - \sigma\|_2^2,$$

where $\sigma = y - z^{t+1} - \frac{1}{\beta}\xi^t$. The Taylor series expansion of $f(x)$ at the point $\tilde{x}$ is

$$\begin{aligned} f(x) = &\|\Phi\tilde{x} - \sigma\|_2^2 + 2\left[\Phi^T(\Phi\tilde{x} - \sigma)\right]^T(x - \tilde{x}) \\ &+2(x - \tilde{x})^T\Phi^T\Phi(x - \tilde{x}). \end{aligned} \tag{8}$$

Because $f(x)$ is a second-degree polynomial, the reformulation in Eq. (8) is exact, that is, the remainder is zero. Further, the quadratic term in Eq. (8) has the upper bound

$$2\|\Phi(x - \tilde{x})\|_2^2 \le \tau\|x - \tilde{x}\|_2^2, \tag{9}$$

where $\tau = 2\|\Phi\|_{op}^2$, and $\|\cdot\|_{op}$ is an operator norm. Let

$$\begin{aligned} g(x) = &\|\Phi\tilde{x} - \sigma\|_2^2 + 2\left[\Phi^T(\Phi\tilde{x} - \sigma)\right]^T(x - \tilde{x}) \\ &+\tau\|x - \tilde{x}\|_2^2. \end{aligned} \tag{10}$$

Note that $g(x)$ is an upper bound of $f(x)$. Eq. (10) can be reformulated as

$$g(x) = \tau\|x - M\|_2^2 + C_2, \tag{11}$$

where both $M = \tilde{x} - \frac{1}{\tau}\Phi^T(\Phi\tilde{x} - \sigma)$ and C_2 are constants.

With the derivation of Eqs. (7), (8), and (11), the x-subproblem in (5) can be approximated as

$$x^{t+1} \approx \arg\min_x \lambda\|\Psi x\|_q^q + \tfrac{1}{2}\beta\tau\|x - M\|_2^2. \tag{12}$$

The optimal solution to problem (12) is approximated as

$$x^{t+1} = M + \frac{1}{\gamma}\Psi^T\left[\eta\left((\Psi M)_i; \frac{\gamma}{\beta\tau}w\right) - \Psi M\right], \tag{13}$$

where $w_i = \frac{\lambda}{(|\Psi M| + \varepsilon_i)^{1-q}} > 0$ and γ is a constant.

From Eqs. (6) and (13), the iteration process (5) can be written as

$$\begin{cases} z^{t+1} = \dfrac{\beta}{1+\beta}\left((y - \Phi x^t) - \frac{1}{\beta}\xi^t\right) \\ x^{t+1} = M + \frac{1}{\gamma}\Psi^T\left[\eta\left(\Psi M; \frac{\gamma}{\beta\tau}w\right) - \Psi M\right] \\ \xi^{t+1} = \xi^t + \beta\left[z^{t+1} - (y - \Phi x^{t+1})\right], \end{cases} \tag{14}$$

where $w_i = \frac{\lambda}{\left(|(\Psi M)_i| + \varepsilon_i\right)^{1-q}}$, $\forall i$, and $M = \tilde{x} - \frac{1}{\tau}A^T\left(A\tilde{x} - y + z^{t+1} + \frac{1}{\beta}\xi^t\right)$.

D. QADM-Net

To speedup reconstruction, we design a DNN architecture based on the QADM algorithm (14). First, we consider the dictionary Ψ as a convolutional neural network, as described in Sec. III-D1. Second, we unroll the specific parameters in (14), as described in Sec. III-D2. Moreover, to operate the network architecture in a large-scale setting, the structurally random matrix is adopted and discussed in Sec. III-D3.

1) Dictionary as Convolutional Neural Network: To recover the image X_0 via solving the optimization problem (2), Ψ plays the role of a dictionary in providing an image a sparse representation. Ψ is generally treated as an over-complete dictionary (*i.e.*, $\Psi \in \mathbb{R}^{N\times n}$ with $N > n$) to achieve better representation. However, as $N > n$, $\Psi \circ \Psi^T = \gamma_c I_N$ is not satisfied at all [19]. Thus, it is necessary to choose a $\Psi^\dagger$ satisfying $\Psi \circ \Psi^\dagger \approx \gamma I_N$ to replace Ψ^T. Following our previous study [19], the left inverse of Ψ exists, *i.e.*, $\tilde{\Psi} = \left(\Psi^T \circ \Psi\right)^{-1} \circ \Psi^T$ and $\tilde{\Psi} \circ \Psi = I_n$. Then, we have

$$\begin{aligned} \Psi^\dagger &= I_n \circ \Psi^\dagger = \left(\tilde{\Psi} \circ \Psi\right) \circ \Psi^\dagger \\ &= \tilde{\Psi} \circ \left(\Psi \circ \Psi^\dagger\right) \approx \tilde{\Psi} \circ \gamma I_N = \gamma\tilde{\Psi}, \end{aligned} \tag{15}$$

and further approximate the solution (13) as

$$x^{t+1} = M + \bar{\gamma}\tilde{\Psi}^t\left[\eta\left(\Psi^t M; \frac{\gamma}{\beta\tau}w\right) - \Psi^t M\right]. \tag{16}$$

Inspired by the representation power of CNN [8] and the design of NN architecture [27], dictionary Ψ^t is adopted as

$$\Psi^t = \mathcal{C}_2^t \circ \mathrm{ReLU} \circ \mathcal{C}_1^t, \tag{17}$$

where all $\mathcal{C}_1^t$ and $\mathcal{C}_2^t$ are convolutional operators, and ReLU and is a rectified linear unit. To exhibit a "left-inverse" structure of Ψ^t, the $\tilde{\Psi}^t$ in Eq. (16) is adopted as

$$\tilde{\Psi}^t = \mathcal{C}_4^t \circ \mathrm{ReLU} \circ \mathcal{C}_3^t, \tag{18}$$

which has the same structure as that in Eq. (17). All $\mathcal{C}_3^t$ and $\mathcal{C}_4^t$ are convolutional operators. Based on the relaxation $\frac{1}{\gamma}\Psi = \bar{\gamma}\tilde{\Psi}$, we will present a loss function appropriately to ensure the left-inverse relation between Ψ^t and $\tilde{\Psi}^t$ in Sec. III-D4.

2) Parameter Unrolling: We describe how to unroll parameters in an NN model. After plugging Eq. (16) into (14), the NN architecture is constructed by unrolling the specific parameters at t^{th} $(0 \le t < T)$ layer as

$$\begin{cases} z^{t+1} &= \beta_{z_1}\left((y - \Phi x^t) - \beta_{z_2}\xi^t\right) \\ M^{t+1} &= x^t - \beta_{M_1}\Phi^T\left(\Phi x^t - y + z^{t+1} + \beta_{M_2}\xi^t\right) \\ w_i^t &= \frac{\lambda^t}{\left(\left|(\Psi^t M^{t+1})_i\right| + \varepsilon_i\right)^{1-q}},\ \forall i, \\ x^{t+1} &= M^{t+1} + \beta_x\tilde{\Psi}^t\left[\eta\left(\Psi^t M^{t+1}; w^t\right) - \Psi^t M^{t+1}\right] \\ \xi^{t+1} &= \xi^t + \beta_\xi\left[z^{t+1} - (y - \Phi x^{t+1})\right], \end{cases} \tag{19}$$

and the set of unrolling parameters is

$$\left\{\beta_{z_1}, \beta_{z_2}, \beta_{M_1}, \beta_{M_2}, \beta_x, \beta_\xi\right\}.$$

Finally, the learning parameters in QADM-Net is summarized as $\left\{\beta_{z_1}^t, \beta_{z_2}^t, \beta_{M_1}^t, \beta_{M_2}^t, \beta_x^t, \beta_\xi^t, \lambda^t, \Psi^t, \tilde{\Psi}^t\right\}_{t=1}^T$ and the flowchart is shown in Fig. 1.

3) Sensing Matrix: In order to recover a large-scale image without relying on the blocking strategy, we adopt the Structurally Random Matrix (SRM) [7] as the sensing matrix:

$$\Phi = DFR, \tag{20}$$

where $D \in \mathbb{R}^{m\times n}$ is a random sampling matrix, $F \in \mathbb{R}^{n\times n}$ is an orthonormal matrix, and $R \in \mathbb{R}^{n\times n}$ is a uniform random permutation matrix. This Φ leads to low computation cost

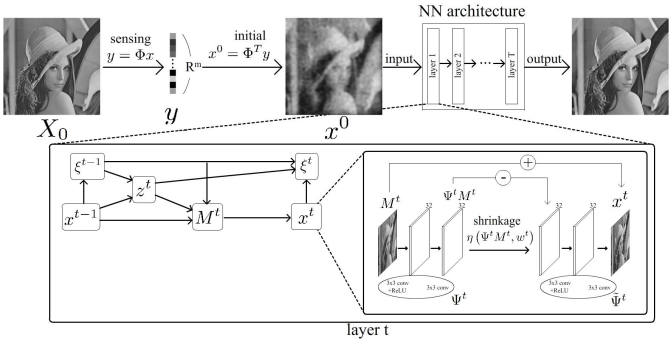

Fig. 1. Flowchart of QADM-Net.

and storage capacity by adopting the corresponding operators associated with D, F, and R, instead of using the original matrix representations.

We observe that if the signal size of x is n, the computation costs of $Rx = R(x)$, $FRx = F(Rx)$, and $D(FRx)$ are n, $n(4n-2)$, and m, respectively, provided the orthonormal matrix F adopts the DFT (discrete Fourier transformation) operator. Moreover, the computation cost of matrix-vector multiplication Φx is $m(2n-1)$. But the storage cost can be further saved as m, 0, and n for the matrices D, F, and R, respectively, provided the orthonormal matrix F adopts DFT (discrete Fourier transformation) operator. Based on the above considerations, we set F to the Discrete Cosine Transform (DCT) due to its fast computation and cost-effectiveness.

4) Loss Function: Since the QADM algorithm aims to find a feasible minimum solution to problem (4) and QADM-Net is the network architecture associated with QADM algorithm, it is desired to have the output x^T of QADM-Net close to the ground-truth x^0. Therefore, the loss function is designed as:

$$\mathcal{L}_{\text{tMSE}} = \sum_{t=1}^{T} \left\| x^t - x^0 \right\|_2^2,$$

where x^t is the output of t^{th} layer of the network. We call it total MSE (tMSE) loss function.

On the other hand, we further follow our prior study [19] to consider the tMSE-loss function in specifying the left-inverse relation between Ψ^t and $\tilde{\Psi}^t$, as mentioned in Sec. III-D1, as

$$\mathcal{L}_{\text{trans}} = \sum_{t=1}^{T} \left\| \tilde{\Psi}^t(\Psi^t(M^t)) - M^t \right\|_2^2.$$

Overall, the loss function for QADM-Net is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{tMSE}} + \delta \cdot \mathcal{L}_{\text{trans}}, \tag{21}$$

where $\delta > 0$ is a parameter balancing $\mathcal{L}_{\text{tMSE}}$ and $\mathcal{L}_{\text{trans}}$.

IV. Experiments

We demonstrate the performance of QADM-Net in reconstructing large-scale images.

A. Parameter Setting

The constant parameters in QADM-Net were $q = 0.3$ and $\varepsilon = 10^{-6} \cdot \mathbf{1}_n$. The training parameters of QADM-Net were initialized as $\beta_{z_1}^t = 0.1, \beta_{z_2}^t = 0.1, \beta_{M_1}^t = 0.1, \beta_{M_2}^t = 0.1, \beta_x^t = 0.1, \beta_\xi^t = 0.1$, and $\lambda^t = 0.1$, whereas $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, and $\mathcal{C}_4$ in Ψ and $\tilde{\Psi}$ were were initialized using Xavier initializer [13]. For $\mathcal{C}_i$, the number of filters was set to $n_f = 32$ with a kernel size 3×3.

The experiments were conducted on a PC with Intel Core i7-7700 CPU, a NVIDIA GeForce GTX 1080 Ti GPU, and Python with TensorFlow version 1.15.0.

B. Natural Image Reconstruction

1) Datasets for Training and Testing: We created the training dataset by collecting the images of large sizes from ImageNet by Step 1 below. The training dataset consisting of images of size 1024×1024 is generated by Step 3 below. The testing dataset was generated by Step 2 below.

1. Collect all of the images with sizes exceeding 1024×1024 from the testing dataset of ImageNet [6]. In this step, we have 846 images.
2. Divide the images from Step 1 into two parts: the first 820 images was designated as the training dataset whereas the remaining 26 images was the testing dataset, which is called ImageNet26 here.
3. Crop each image in the training dataset into patches of size 1024×1024 with a stride of 256.

2) Training Details: During training, we adopted the Adam optimizer [17] with a learning rate of 0.0001. The network was trained for 100 epochs with a batch size of 1. The number of layers was set to 5.

3) Performance Comparison: We compared QADM-Net with state-of-the-art learning-based methods, including ISTA-Net$^+$ [27], CSNet$^+$ [24], SCSNet [25], OPINE-Net [28], and AMP-Net [29], that are block-based network models. Due to space limit, we only show the comparison results in Table II for dataset BSD68. We can see similar results for other datasets. On the other hand, the reconstruction result for ImageNet26, which is described in Sec. IV-B1, is shown in Table III, where the comparisons are absent because, for example, OPINE-Net [28] exhibits out-of-memory to test ImageNet26.

C. Medical Image Reconstruction

We adopted ADMM-Net [26] as the baseline and employed the dataset from open-source code of [26], which is available on GitHub, as training and test dataset.

1) Datasets for Training and Testing: We only compared with ADMM-Net [26] because either the test datasets or source codes of state-of-the-art methods were not released. The image size of in datasets, Brain1 and Brain2, is 256×256.

2) Training Details: During training, we adopted Adam optimizer [17] with a learning rate of 0.0001. The network was trained for 100 epochs with a batch size of 16. The number of layers were set to 5.

3) Performance Comparison: We compared QADM-Net with ADMM-Net [26], as shown in Table IV.

TABLE II
AVERAGE PSNR (DB) COMPARISONS UNDER DIFFERENT MEASUREMENT RATES ON BSD68. THE BEST RESULTS ARE MARKED IN BOLD RED AND THE SECOND ONES ARE MARKED IN BOLD BLUE.

Measurement Rate	50%	40%	30%	20%	10%
ISTA-Net$^+$ [27]	34.01	32.21	30.34	-	25.33
CSNet [24]	34.89	32.53	31.45	-	27.10
SCSNet [25]	35.77	33.86	31.87	-	27.28
AMP-Net-9-BM [29]	**36.82**	**34.86**	**32.84**	30.63	**27.86**
OPINE-Net [28]	36.32	34.33	32.46	**30.38**	27.81
QADM-Net	**41.05**	**38.58**	**35.44**	**30.66**	**28.75**

TABLE III
AVERAGE PSNR OF QADM-NET ON IMAGENET26.

ImageNet26	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Measurement Rate	50%		30%		10%	
QADM-Net	43.53	0.975	38.88	0.945	33.27	0.849

TABLE IV
AVERAGE PSNR (DB) COMPARISONS ON MRI IMAGES.

Images	Brain1/Brain2		
Measurement Rate	50%	30%	10%
ADMM-Net [26]	36.62/33.55	34.03/30.55	28.40/24.88
QADM-Net	41.96/37.06	38.22/33.15	30.30/25.24

D. Model Size Comparison

Table V shows a comparison of parameter number and model size of the network architecture for various methods under measurement rate 10%. Evidently, the memory consumption of training parameters in QADM-Net is at most 34% of other methods.

TABLE V
COMPARISON OF TRAINING PARAMETER NUMBER AND MODEL SIZE WITH MEASUREMENT RATE 10%.

Methods	Parameter #	Model Size
ISTA-Net$^+$ [27]	336978	1.29MB
CSNet$^+$ [24]	578688	2.21MB
SCSNet [25]	796416	3.04MB
AMP-Net-9-BM [29]	579555	2.21MB
OPINE-Net [28]	274845	1.05MB
QADM-Net	95075	0.36MB

V. CONCLUSIONS

In this paper, we presented a deep learning model for compressive sensing of large-scale images. The key is to introduce variable-splitting and propose an alternative ADMM algorithm to avoid matrix inverse operator. Experimental results validate the proposed method.

VI. ACKNOWLEDGMENT

This work was supported by Ministry of Science and Technology, ROC, under Grants MOST 110-2221-E-001–020-MY2 and NSTC 111-2634-F-006-022.

REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.
[2] A. Bustin, N. Fuin, R. M. Botnar, and C. Prieto. From compressed-sensing to artificial intelligence-based cardiac MRI reconstruction. *Front. Cardiovasc. Med.*, 2020.
[3] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.
[4] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14(10):707–710, 2007.
[5] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. *In ICASSP*, 2008.
[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database problem. *CVPR*, 2009.
[7] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran. Fast and efficient compressive sensing using structurally random matrices. *IEEE Trans. Signal Process.*, 60(1), Jan. 2012.
[8] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. *In Proc. Eur. Conf. Comput. Vision (ECCV)*, pages 184–199, 2014.
[9] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.
[10] T. Eo, Y. Jun, T. Kim, J. Jang, H.-J. Lee, and D. Hwang. KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance images. *Magn. Reson. Medicine*, 80(5):2188–2201, Nov. 2018.
[11] F. L. Fan, J. Xiong, M. Li, and G. Wang. On interpretability of artificial neural networks: A survey. *IEEE Trans. Radiat. Plasma Med. Sci.*, 2021.
[12] L. Gan. Block compressed sensing of natural images. *In Proc. Int. Conf. Digit. Signal Process.*, Jul. 2007.
[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *In Proc. Int. Conf. on Artif. Intell. Statist. (AISTATS)*, 2010.
[14] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstructionof accelerated MRI data. *Magn. Reson. Medicine*, 79:3055–3071, 2018.
[15] H.Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu. DR^2-Net: Deep residual reconstruction network for image compressive sensing. *Neurocomputing*, 359:483–493, Sep. 2019.
[16] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo. Deep learning for undersampled MRI reconstruction. *Phys. Med. Biol.*, 63, Jun. 2018.
[17] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
[18] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. *In CVPR*, Jun. 2016.
[19] Gang-Xuan Lin, Shih-Wei Hu, and Chun-Shien Lu. Qista-imagenet: A deep compressive image sensing framework solving ℓ_q-norm optimization problem. *ECCV*, Oct. 2022.
[20] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.*, 38(2):18–44, Mar. 2021.
[21] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. *In Proc. Annu. Allerton Conf. Commun., Control, and Comput. (Allerton)*, Sep. 2015.
[22] S. Mun and J. E. Fowler. Block compressed sensing of images using directional transforms. *In ICIP*, Nov. 2009.
[23] S. J. Park, C. G. Lim, and C.-B. Ahn. Network slimming for compressed-sensing cardiac cine MRI. *Electron. Lett.*, 57(7):297–299, Jan. 2021.
[24] W. Shi, F. Jiang, S. Liu, and D. Zhao. Image compressed sensing using convolutional neural network. *IEEE Trans. Image Process.*, 29:375–388, Jul. 2019.
[25] W. Shi, F. Jiang, S. Liu, and D. Zhao. Scalable convolutional neural network for image compressed sensing. *In CVPR*, Jun. 2019.
[26] Y. Yang, J. Sun, H. Li, and Z. Xu. Deep admm-net for compressive sensing mri. *In Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2016.
[27] J. Zhang and B. Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. *In CVPR*, Jun. 2018.
[28] J. Zhang, C. Zhao, and W. Gao. Optimization-inspired compact deep compressive sensing. *IEEE J. Sel. Topics Signal Process.*, 14:765–774, May 2020.
[29] Z. Zhang, Y. Liu, J. Liu, F Wen, and C. Zhu. Amp-net: Denoising-based deep unfolding for compressive image sensing. *IEEE Trans. Image Process.*, 30:1487–1500, Dec. 2020.