Block Compressed Sensing Image Reconstruction via Untrained Network Priors

Lan Li and Xiao-Long Qiu and Ming-Li Jing and Sha-Sha Pu

Abstract—**In this paper, we consider image reconstruction methods for block compressed sensing (BCS). We model images to lie in the range of an untrained network for every subblock and combine the projected gradient descent algorithm to improve the performance. This proposed method is named BCS_PGDNET. Meanwhile, sufficient condition is provided for algorithmic convergence. We compare with the following algorithm: OMP, COSAMP, IRLS, BCS_SPL_DDWT, BCS_FOCUSS, BCS_SPL_DCT. Experimental simulations show that BCS_PGDNET method have better recovery performance than main-stream CS methods and BCS methods with relative higher PSNR and SSIM, and tradeoffs between block size and the improved reconstruction performance are discussed. This is generation of compressed sensing (CS) with untrained network priors.**

***Index Terms*—block compressed sensing, untrained network, projection gradient descent, priors**

I. Introduction

With the development of the Internet and the advent of the era with big data, signal acquisition and transmission have received wide attention from scholars [1-4]. The traditional sampling method needs to satisfy the Shannon's sampling theory, and the original signal can be reconstructed when the signal sampling frequency is greater than two times of the highest frequency. However, due to the limitation of hardware facilities, it is difficult to sample and transmit for high-frequency signals. At the same time, for low-frequency signals, there is more redundant information with the large amount of sampled data. Reconstruction process leads to an increase in computational difficulty. To deal with the drawback, Donoho [1] proposes the CS theory. That is, when the signal is sparse, signal could be reconstructed from a few measurement values by a suitable measurement matrix in CS. In the subsequent research, CS achievements about sparse signal processing, reconstruction algorithms and measurement matrix design. CS is widely used in semantic segmentation [2], computational ghost imaging [3], depth determination of monocular camera [4], pattern recognition [5] and other fields. However, the classical CS reconstruction methods and the signal sampling with huge measurement matrix still impose a large computational consumption on the hardware devices. In order to reduce the computational burden. Lu [6] adopts block compressed sensing (BCS), which divides the image into several blocks with the same size. This method solves the problem of real-time transmission of large scale images and greatly reduces the computational complexity. The measurement matrix of every block image is much smaller than the whole image and it is easy to be stored. Mun et al. [7] implements the BCS_SPL algorithm, and the original image is reconstructed effectively. But lots of information is lost in the recovery process and this algorithm leads to the degradation of the recovery quality. BCS_FOCUSS algorithm is reported by Deepthi et al. [8] and this algorithm overcomes the drawback of some BCS algorithm in terms of lost information.

In recent years, driven by deep neural networks (DNNs) framework, a series of signal processing methods based on deep learning have been proposed. DNNs can provide a technical way to reconstruct signal with high quality. Adler et al. [9] combines simple structural DNNs with block measurement matrix to reconstruct the original signal faster. However, the artifacts exist in the reconstructed images. In this paper, a projection gradient descent (PGD) reconstruction algorithm based on untrained neural network is proposed. Firstly, the original signal/image is blocked. For every measurement signal, it is sampled by a Gaussian matrix. Then for the measured signal, we adopt an untrained network as priors and apply PGD to recover the original signal. The value of PSNR and SSIM for the reconstructed signal are compared with classical algorithms at different sampling rates and block size. Lastly, we discuss the effect of sampling rate and block size, respectively. Our contributions are as follows:

We draw a connection between PGD with the deep untrained network prior and BCS. Furthermore, we obtain the sufficient condition about the range of images S_B spanned by the deep untrained network. Meanwhile, we give the proof of convergence of the proposed algorithm.

In this paper, upper letters T indicates transpose, vectors are denoted by lower case bold letters such as $\boldsymbol{x}$ and matrices are expressed by upper case bold letters such as $\mathbf{A}$, respectively. We denote the inner product between $\boldsymbol{x}$, $\boldsymbol{y}$ as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

Manuscript received October 27, 2022; revised March 11, 2023.

This work is supported by Natural Science Basic Program of Shannxi (Program No.2021JM-399), the Graduate Innovation and Practical Ability Training Project of Xi'an Shiyou University (YCS21212149).

Lan Li is a professor of School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: lanli@xsyu.edu.cn).

Xiao-Long Qiu is a postgraduate student of School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: Pick_up_qiu@163.com).

Ming-Li Jing is an associate professor of School of Electronic Engineering. Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: mljingsy@xsyu.edu.cn).

Sha-Sha Pu is a postgraduate student of School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: pushashay@163.com).

II. The overview

CS is a new data acquisition method. That is, a sparse signal $\boldsymbol{x}$ can be acquired by incomplete information in CS.

A. Compressed Sensing

In CS model, the sampling process can be expressed as:

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

where $\boldsymbol{x} \in R^N$ is the original signal, and $\boldsymbol{y} \in R^m$ is the measurement values obtained by the measurement matrix $\boldsymbol{\Phi} \in R^{m \times N} (m \ll N)$, the sampling rate is $R = m/N$. This problem is ill-posed. It is well-known that if $\boldsymbol{\Phi}$ satisfies the restricted isometry property (RIP) condition, it is possible to reconstruct a sparse signal $\boldsymbol{x}$ by the following l_0 - minimization problem:

$$\min \|\boldsymbol{x}\|_0 \qquad \text{s.t. } \boldsymbol{y} = \boldsymbol{\Phi x}, \tag{2}$$

where $\|\bullet\|_0$ indicates the number of non-zero elements of the signal $\boldsymbol{x}$. The RIP concept is (see Definition 1) first introduced by Candès et al. [10].

Definition 1. Assume a k -sparse vectors $\boldsymbol{x} \in R^N$. If there exists a constant $0 < \delta_k < 1$ and δ_k satisfies the following formula:

$$(1-\delta_k)\|\boldsymbol{x}\|_2^2 \le \|\boldsymbol{\Phi x}\|_2^2 \le (1+\delta_k)\|\boldsymbol{x}\|_2^2, \tag{3}$$

then $\boldsymbol{\Phi}_{m \times N}$ is said to satisfy RIP condition of order k. The smallest constant δ_k is called the restricted isometry constant (RIC) of order k for $\boldsymbol{\Phi}$.

However, for the large scale images in realistic scenes, the reconstruction process needs to cost more computing resources. In the next part, we consider BCS to overcome this drawback.

B. Block Compressed Sensing

Let the gray value matrix of a grayscale image be:

$$I = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{N1} \\ x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1N} & x_{2N} & \cdots & x_{NN} \end{bmatrix}^{\mathrm{T}}. \tag{4}$$

This matrix is vectorized as:

$$\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]^{\mathrm{T}}, \tag{5}$$

where $\boldsymbol{x}_j = [\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}, \cdots, \boldsymbol{x}_{j_N}]^{\mathrm{T}}$.

Let $\boldsymbol{x} = \left[\overbrace{x_1, \cdots, x_{l_1}}^{x_{[1]}}, \cdots, \overbrace{x_{(j-1)\times l_j+1}, \cdots, x_{j \times l_j}}^{x_{[j]}}, \cdots, \overbrace{x_{N-l_n+1}, \cdots, x_N}^{x_{[n]}} \right]^{\mathrm{T}}$, where the j -th subblock is written as $\boldsymbol{x}_{[j]}$ and the length of index set is $\Gamma = \{l_1, l_2, \cdots, l_n\}$. The size of original image is N dimension and the same block size is $l_j = b \times b$, $1 \le j \le n$, where $N = \sum_{j=1}^{n} l_j$. Especially, as n=1, the block compressed sensing is compressed sensing. The model of BCS can be sensing is compressed sensing. The model of BCS can be described:

$$\boldsymbol{y} = \left[\left\langle \boldsymbol{\Phi}_B, \boldsymbol{x}_{[1]} \right\rangle, \left\langle \boldsymbol{\Phi}_B, \boldsymbol{x}_{[2]} \right\rangle, \cdots, \left\langle \boldsymbol{\Phi}_B, \boldsymbol{x}_{[n]} \right\rangle \right]^{\mathrm{T}}, \tag{6}$$

where $\boldsymbol{\Phi}_B$ is the measurement matrix, and $\boldsymbol{y}_{[j]} = \boldsymbol{\Phi}_B \boldsymbol{x}_{[j]}$, $1 \le j \le n.$.

Definition 2. Let the original signal $\boldsymbol{x} \in R^N$ be block k -sparse. Assume the following formula satisfies a constant $0 < \delta_k < 1$:

$$(1-\delta_k)\|\boldsymbol{x}_{[j]}\|_2^2 \le \|\boldsymbol{\Phi x}_{[j]}\|_2^2 \le (1+\delta_k)\|\boldsymbol{x}_{[j]}\|_2^2, \tag{7}$$

then measurement matrix $\boldsymbol{\Phi}_B$ is said to satisfy the block restricted isometry property (Block_RIP) of order k. The Block_RIP concept is introduced by Junhong et al. [11].

Then, we give the flow chart of BCS, the details are seen in Figure.1:

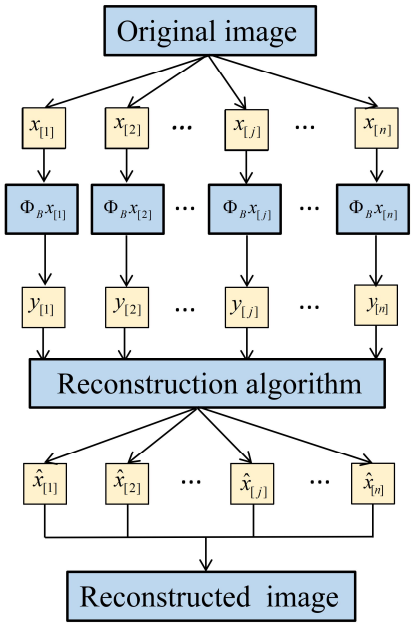


Fig. 1. The framework of block compressed sensing

II. Proposed algorithm and convergence

In this section, firstly, we describe sketch of the untrained neural network. Secondly, the BCS_PGDNET algorithm is presented. Finally, we give the convergence proof of the proposed algorithm.

A. Untrained Neural Network Model

Deep neural networks (DNNs) technique is widely applied in images reconstruction, such as denoising [12], inpainting [13], super-resolution [14], et al. However, traditional DNNs require a large datasets to train the model. Therefore, we explore untrained neural network as image priors to reconstruct image signal from the few measurement values. The j -th subblock $\boldsymbol{x}_{[j]}$ can be represented by the block untrained neural network $G_B(\mathbf{W}_j; \boldsymbol{z}_j)$, where $\mathbf{W}_j$ is the set of the weights of deep network, $\boldsymbol{z}_j$ is

the latent code and the set S_B captures the characteristics of $G_B(\mathbf{W}_j;\boldsymbol{z}_j)$, $1\le j\le n$ [15]. In other words, subblock $\boldsymbol{x}_{[j]}$ is said to obey an untrained neural network prior if it belongs to the set $S_B \coloneqq \{\boldsymbol{x}_{[j]} \mid \boldsymbol{x}_{[j]} = G_B(\mathbf{W}_j;\boldsymbol{z}_j)\}$. The j-th subblock $\boldsymbol{x}_{[j]}$ has a decoder prior and it is denoted by $G_B(\mathbf{W}_j;\boldsymbol{z}_j)$. The network has a piece-wise linear activation function of $\mathrm{Re\,Lu} = \max\{0,x\}$. If the true weights $\mathbf{W}_j$ is obtained for block signal $\boldsymbol{x}_{[j]}$ from $\mathbf{W}_j \leftarrow \arg\min_{\mathbf{W}_j}\left\|\boldsymbol{x}_{[j]} - G_B(\mathbf{W}_j;\boldsymbol{z}_j)\right\|$, then we need large datasets to train. However, for an untrained neural network, $\mathbf{W}_j$ is obtained without by training in advance and learning for any large datasets. Then we consider the neural network model:

$$\boldsymbol{x}_{[j]} = \mathbf{W}_j^L U^{L-1}\sigma(\mathbf{W}_j^{L-1}\boldsymbol{z}_j^{L-1}) = \mathbf{W}_j^L \boldsymbol{z}_j^L, \tag{8}$$

where $\sigma(\cdot)$ is the activate function operator $\mathrm{Re\,Lu}$, L is the number of layers for the neural network, and U represents the upsampling operator.

Definition 3. Suppose that the original signal $\boldsymbol{x}\in R^N$ and $\boldsymbol{x} = \left[\overbrace{x_1,\cdots,x_{l_1}}^{x_{[1]}},\cdots,\overbrace{x_{(j-1)\times l_j+1},\cdots,x_{j\times l_j}}^{x_{[j]}},\cdots,\overbrace{x_{N-l_n+1},\cdots,x_N}^{x_{[n]}}\right]^{\mathrm{T}}$ is block k-sparse. Let $\boldsymbol{\Phi}_B$ be Gaussian matrix with elements $N(0,1/n)$. For every $\boldsymbol{x}_{[j]}\in S_B, 1\le j\le n$, if there exists a constant $0<\delta_k<1$, the following formula satisfies:

$$(1-\delta_k)\left\|\boldsymbol{x}_{[j]}\right\|_2^2 \le \left\|\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}\right\|_2^2 \le (1+\delta_k)\left\|\boldsymbol{x}_{[j]}\right\|_2^2, \tag{9}$$

then the matrix $\boldsymbol{\Phi}_B$ satisfies block set-restricted eigenvalue condition with parameter δ_k (Block_SB_REC).

B. Reconstruction algorithm based on an untrained neural network

From formula (6), we obtain the measurement signal $\boldsymbol{y}_{[j]}$, $1\le j\le n$, from Gaussian matrix $\boldsymbol{\Phi}_B$. Next, we set an initial $\boldsymbol{x}_{[j]}^0 = G_B(\mathbf{W}_j^0;\boldsymbol{z}_j)$, then set learning rate η and the number of iterations T, respectively. Every $\boldsymbol{x}_{[j]}$, $1\le j\le n$ is reconstructed separately by BCS_PGDNET algorithm.

For fixed $\boldsymbol{x}_{[j]}, 1\le j\le n$, the algorithm is shown below:

Algorithm: BCS_PGDNET

Step 1: Input : $\boldsymbol{y}_{[j]}, \boldsymbol{\Phi}_B, \boldsymbol{x}_{[j]}^0 = G_B(\mathbf{W}_j^0;\boldsymbol{z}_j), \eta, T$

Step 2: for $t = 0,1,\cdots,T$ do

Step 3: $\boldsymbol{x}_{[j]}^t \leftarrow winner(\boldsymbol{x}_{[j]}^t)$

Step 4: $\boldsymbol{v}_j^t \leftarrow \boldsymbol{x}_{[j]}^t - \eta\boldsymbol{\Phi}_B^{\mathrm{T}}(\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t - \boldsymbol{y}_{[j]})$

Step 5: $\mathbf{W}_j^t \leftarrow \arg\min_{\mathbf{W}_j}\left\|\boldsymbol{v}_j^t - G_B(\mathbf{W}_j^t;\boldsymbol{z}_j)\right\|$

Step 6: $\boldsymbol{x}_{[j]}^{t+1} \leftarrow G_B(\mathbf{W}_j^t;\boldsymbol{z}_j)$

Step 7: Output : $\hat{\boldsymbol{x}}_{[j]} \leftarrow \boldsymbol{x}_{[j]}^T$

Finally, we obtain the output $\hat{\boldsymbol{x}} \leftarrow [\hat{\boldsymbol{x}}_{[1]},\hat{\boldsymbol{x}}_{[2]},\cdots,\hat{\boldsymbol{x}}_{[j]},\cdots,\hat{\boldsymbol{x}}_{[n]}]^{\mathrm{T}}$.

C. Convergence of the proposed algorithm

In this part, to validate the convergence of the algorithm, the following proof is given. Firstly, we give the error function at t-time iteration for the j-th subblock $\boldsymbol{x}_{[j]}$:

$$E_B(\boldsymbol{x}_{[j]}^t) = \left\|\boldsymbol{y}_{[j]} - \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\|_2^2. \tag{10}$$

For $\boldsymbol{x}^t$ at the t-time iteration, then we have:

$$E(\boldsymbol{x}^t) = \sum\nolimits_{j=1}^n E_B(\boldsymbol{x}_{[j]}^t). \tag{11}$$

Combining equation (10), we get:

$$\begin{aligned}
&E_B(\boldsymbol{x}_{[j]}^{t+1}) - E_B(\boldsymbol{x}_{[j]}^t)\\
&= \left(\left\|\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^{t+1}\right\|_2^2 - \left\|\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\|_2^2\right) - 2\left\langle\boldsymbol{y}_{[j]}, \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^{t+1}\right\rangle + 2\left\langle\boldsymbol{y}_{[j]}, \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\rangle\\
&= \left(\left\|\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^{t+1}\right\|_2 - \left\|\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\|_2\right)^2 - 2\left\langle\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t, \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\rangle + 2\left\langle\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^{t+1}, \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\rangle\\
&\qquad - 2\left\langle\boldsymbol{y}_{[j]}, \boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right)\right\rangle\\
&= \left\|\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right\|_2^2 - 2\left\langle\boldsymbol{y}_{[j]}, \boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right)\right\rangle\\
&\qquad + 2\left\langle\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t, \boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right)\right\rangle.
\end{aligned} \tag{12}$$

Next, according to the BCS_PGDNET algorithm, we have:

$$\boldsymbol{v}_j^t \leftarrow \boldsymbol{x}_{[j]}^t - \eta\boldsymbol{\Phi}_B^{\mathrm{T}}(\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t - \boldsymbol{y}_{[j]}). \tag{13}$$

Subsequently, $\mathbf{W}_j^t$ of the proposed algorithm lies in the range of the decoder $G_B(\mathbf{W}_j^t;\boldsymbol{z}_j)$, $G_B(\mathbf{W}_j^t;\boldsymbol{z}_j)$ is the closest to $\boldsymbol{v}_j^t$, and $\mathbf{W}_j^t$ satisfies:

$$\left\|G_B(\mathbf{W}_j^t;\boldsymbol{z}_j) - \boldsymbol{v}_j^t\right\| \le \left\|G_B(\mathbf{W}_j;\boldsymbol{z}_j) - \boldsymbol{v}_j^t\right\|. \tag{14}$$

Denote $G_B(\mathbf{W}_j^t;\boldsymbol{z}_j) = \boldsymbol{x}_{[j]}^{t+1}$ and $G_B(\mathbf{W}_j;\boldsymbol{z}_j) = \boldsymbol{x}_{[j]}$. Therefore, we obtain the following formula:

$$\left\|\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{v}_j^t\right\|_2^2 \le \left\|\boldsymbol{x}_{[j]} - \boldsymbol{v}_j^t\right\|_2^2, \tag{15}$$

Substitution of formula (13) into the formula (15) gives:

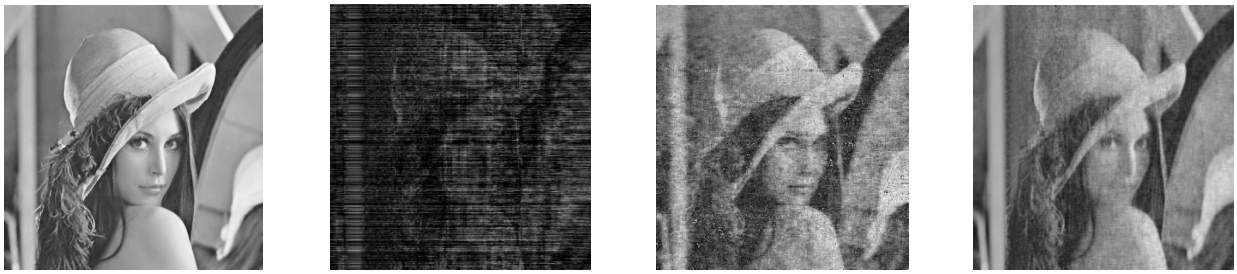
$$\begin{aligned}
&\left\|\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t + \eta\boldsymbol{\Phi}_B^{\mathrm{T}}\left(\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t - \boldsymbol{y}_{[j]}\right)\right\|_2^2\\
&\qquad \le \left\|\boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^t + \eta\boldsymbol{\Phi}_B^{\mathrm{T}}\left(\boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t - \boldsymbol{y}_{[j]}\right)\right\|_2^2.
\end{aligned} \tag{16}$$

Combining the same internal terms of formula (16), we obtain:

$$\begin{aligned}
&\left\|\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right\|_2^2 - 2\eta\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right)\right)^{\mathrm{T}}\left(\boldsymbol{y}_{[j]} - \boldsymbol{\Phi}_B\boldsymbol{x}_{[j]}^t\right) \le\\
&\qquad \left\|\boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^t\right\|_2^2 - 2\eta\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^t - \boldsymbol{x}_{[j]}\right)\right)^{\mathrm{T}}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^t - \boldsymbol{x}_{[j]}\right)\right).
\end{aligned} \tag{17}$$

Substituting formula (12) and (17), and formula (17) is divided by η of both sides, we have:

$$\begin{aligned}
&\left(\frac{1}{\eta}\left\|\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right\|_2^2 - \left\|\boldsymbol{\Phi}_B\left(\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^t\right)\right\|_2^2\right) + E_B(\boldsymbol{x}_{[j]}^{t+1}) - E_B(\boldsymbol{x}_{[j]}^t)\\
&\qquad \le \left(\frac{1}{\eta}\left\|\boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^t\right\|_2^2\right) - 2E_B(\boldsymbol{x}_{[j]}^t).
\end{aligned} \tag{18}$$



(A) ORIGINAL (B) COSAMP (C) OMP (D) IRLS

(E) BCS_SPL_DDWT (F) BCS_SPL_DCT (G) BCS_FOCUSS (H) BCS_PGDNET

Fig. 2. Reconstructed results from different algorithms at sampling rate 0.3 for Lena image

Therefore, we obtain:

$$E_B(\boldsymbol{x}_{[j]}^{t+1}) + E_B(\boldsymbol{x}_{[j]}^{t}) \leq \left\| \boldsymbol{\Phi}_B \left(\boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^{t} \right) \right\|_2^2 - \frac{1}{\eta} \left\| \boldsymbol{x}_{[j]}^{t+1} - \boldsymbol{x}_{[j]}^{t} \right\|_2^2 + \frac{1}{\eta} \left\| \boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^{t} \right\|_2^2. \quad (19)$$

According to Definition 3, since $\boldsymbol{x}_{[j_1]}, \boldsymbol{x}_{[j_2]} \in S_B$, we have:

$$\begin{aligned} (1-\delta_k) \left\| \boldsymbol{x}_{[j_1]} - \boldsymbol{x}_{[j_2]} \right\|_2^2 &\leq \left\| \boldsymbol{\Phi}_B \left(\boldsymbol{x}_{[j_1]} - \boldsymbol{x}_{[j_2]} \right) \right\|_2^2 \\ &\leq (1+\delta_k) \left\| \boldsymbol{x}_{[j_1]} - \boldsymbol{x}_{[j_2]} \right\|_2^2. \end{aligned} \quad (20)$$

Combining the left side of formula (20) and formula (19), we obtain:

$$(1-\delta_{k_1}) \left\| \boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^{t} \right\|_2^2 \leq \left\| \boldsymbol{\Phi}_B \left(\boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^{t} \right) \right\|_2^2. \quad (21)$$

Similarly, we obtain:

$$\left\| \boldsymbol{\Phi}_B \left(\boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^{t} \right) \right\|_2^2 \leq (1+\delta_{k_2}) \left\| \boldsymbol{x}_{[j]} - \boldsymbol{x}_{[j]}^{t} \right\|_2^2. \quad (22)$$

If $\left(1-\delta_{k_1}\right) < 1/\eta < 2\left(1-\delta_{k_1}\right)$ and $\left(1+\delta_{k_1}\right) < 1/\eta$, where δ_{k_1} and δ_{k_2} are constants.

Substitution of formula (21) and formula (22) into equation (19), we have the following formula:

$$E_B(\boldsymbol{x}_{[j]}^{t+1}) < \varepsilon E_B(\boldsymbol{x}_{[j]}^{t}). \quad (23)$$

For reconstructed image $\hat{\boldsymbol{x}}$, we obtain:

$$\sum_{j=1}^{n} E(\boldsymbol{x}^{t+1}) < \varepsilon E(\boldsymbol{x}^{t}), \quad 0 < \varepsilon < 1. \quad (24)$$

Therefore, the proposed algorithm is convergent.

III. Experiment and Analysis

This section will verify the effect of different sampling rates and block sizes on the reconstruction effect.

A. Evaluation Criteria and Experimental Environment

For BCS_PGDNET method, the experiment simulation is based on Intel-i5-10300H (CPU) processor and graphics card NIDIA GeForce GTX1650 in Windows 10 operating system. The code runs on the Pytorch platform. Among the classical CS methods and BCS algorithms runs on the Matlab 2015a environment, 64-bit operating system.

To verify the effectiveness of the proposed method, the sampling rate is set at 0.1, 0.15, 0.2, 0.25 and 0.3 to perform the experiments, respectively. Meanwhile, we compare conventional CS methods and classical BCS methods. Among the classical CS methods include OMP [16], COSAMP [17] and IRLS [18], the BCS algorithms include BCS_SPL_DDWT [7], BCS_SPL_DCT [7] and BCS_ FOCUSS [8].

We test the performance of BCS_PGDNET algorithm on different images. The performance is evaluated by peak signal to noise ratio (PSNR) and structural similarity (SSIM) value.

B. Simulation results and discussion

When the ground-truth $\boldsymbol{x}$ is reconstructed through the BCS algorithms, the block size has an impact on the experimental results. As the block size is larger, the running time is significantly longer. On the contrary, as the block size is smaller, the running time is shorter, but there exists obviously block effect. Therefore, we choose the block size of 16×16 by overall consideration to perform the experiments.

To explore the effectiveness of the BCS reconstruction algorithm, Lena image is selected for the experiments with sampling rate at 0.3. The results are shown in Figure 2. We find classical CS algorithms does not perform well with artifacts. To explore the effect of different sampling rates on the reconstruction results, we plot TABLE I. From TABLE I, the SSIM and PSNR value of BCS algorithms are generally better than CS algorithms, greater than approximately 0.3 and 10dB, respectively. Meanwhile, it can be seen from Figure 3 that the PSNR shows an increasing trend when the sampling rate increases. From Figure 4, the results of BCS_PGDNET has the relatively high SSIM value, and the proposed algorithms is competitive for other methods.

TABLE I COMPARISON IN PSNR AND SSIM BASED ON DIFFERENT ALGORITHMS FOR LENA IMAGE

Methods	0.1	0.15	0.2	0.25	0.3
COSAMP	10.85 | 0.1871	11.75 | 0.2082	12.54 | 0.2196	15.08 | 0.2297	17.32 | 0.2301
OMP	13.93 | 0.2042	14.70 | 0.2139	16.67 | 0.2387	18.51 | 0.2601	20.17 | 0.3551
IRLS	17.01 | 0.2495	19.36 | 0.3292	21.54 | 0.4074	22.71 | 0.4624	24.04 | 0.5156
BCS_SPL_DCT	27.23 | 0.4468	29.01 | 0.5206	30.15 | 0.5856	31.05 | 0.6323	32.18 | 0.6710
BCS_SPL_DDWT	27.72 | 0.4873	29.68 | 0.5654	30.82 | 0.6236	31.96 | 0.6715	32.80 | 0.7078
BCS_FOCUSS	**28.07** | 0.5236	29.54 | **0.6836**	31.26 | 0.6958	**32.55** | 0.7135	33.07 | 0.7587
BCS_PGDNET	27.35 | **0.6535**	**29.92** | 0.6743	**31.81** | **0.7989**	32.41 | **0.8423**	**33.71** | **0.8654**

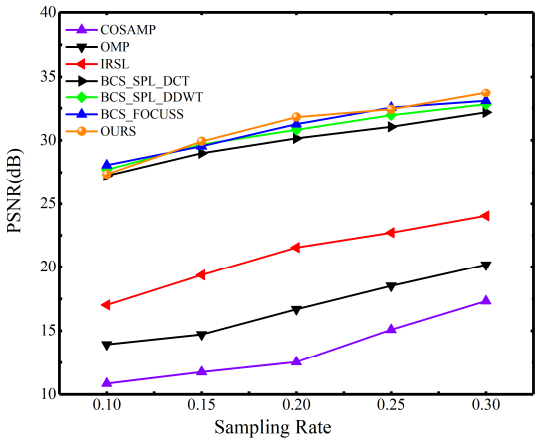


Fig. 3. Comparison in PSNR between seven algorithms

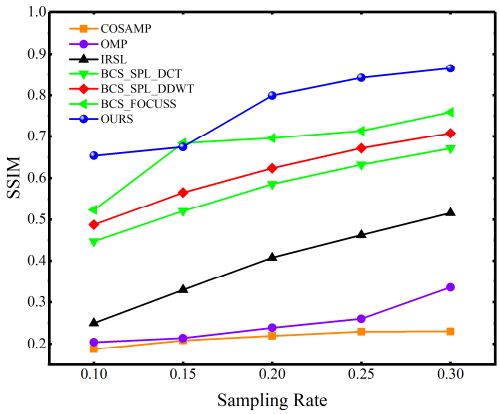
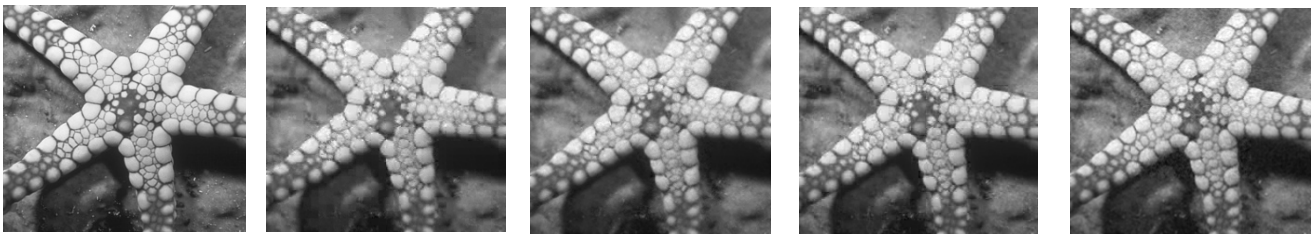


Fig. 4. Comparison in SSIM between seven algorithms

To further verify the effectiveness of BCS_PGDNET algorithm, we set the sampling rate 0.1, 0.15, 0.2, 0.25 and 0.3, respectively. Then, we choose different images, such as Cameraman (512×512), Barbara(512×512), Fishstar(512×512) and Ship(512×512) to progress experiments.

(A) ORIGINAL (B) BCS_SPL_DCT (C) BCS_SPL_DDWT (D) BCS_FOCUSS (E) BCS_PGDNET

Fig. 5. Reconstructed results from different algorithms at sampling rate 0.3 for Barbara image



(A) ORIGINAL (B) BCS_SPL_DCT (C) BCS_SPL_DDWT (D) BCS_FOCUSS (E) BCS_PGDNET

Fig. 6. Reconstructed results from different algorithms at sampling rate 0.3 for Fishstar image

(A) ORIGINAL (B) BCS_SPL_DCT (C) BCS_SPL_DDWT (D) BCS_FOCUSS (E)BCS_PGDNET

Fig. 7. Reconstructed results from different algorithms at sampling rate 0.3 for Ship image

(A) ORIGINAL (B) BCS_SPL_DCT (C) BCS_SPL_DDWT (D) BCS_FOCUSS (E)BCS_PGDNET

Fig. 8. Reconstructed results from different algorithms at sampling rate 0.3 for Cameraman image

TABLE II COMPARISON IN PSNR AND SSIM BASED ON DIFFERENT ALGORITHMS FOR BARBARA IMAGE

Methods	0.1	0.15	0.2	0.25	0.3
BCS_SPL_DCT	22.71 | 0.4082	23.37 | 0.4814	24.08 | 0.5364	24.75 | 0.5948	25.41 | 0.6423
BCS_SPL_DDWT	22.73 | 0.4404	23.22 | 0.5068	23.90 | 0.5598	24.74 | 0.6192	25.40 | 0.6636
BCS_FOCUSS	22.96 | 0.4620	23.79 | **0.5428**	24.94 | 0.6085	25.94 | 0.6693	26.79 | 0.7099
BCS_PGDNET	**23.35** | **0.5135**	**24.92** | 0.5403	**25.81** | **0.6119**	**26.41** | **0.6723**	**27.71** | **0.7154**

TABLE III COMPARISON IN PSNR AND SSIM BASED ON DIFFERENT ALGORITHMS FOR FISHSTAR IMAGE

Methods	0.1	0.15	0.2	0.25	0.3
BCS_SPL_DCT	21.75 | 0.4436	23.05 | 0.5286	24.20 | 0.5898	25.03 | 0.6357	26.11 | 0.6850
BCS_SPL_DDWT	22.72 | 0.5158	23.95 | 0.5962	25.15 | 0.6573	25.97 | 0.7014	27.17 | 0.7489
BCS_FOCUSS	**22.43** | 0.5236	23.85 | 0.6120	25.12 | 0.6742	**26.24** | 0.7198	27.27 | 0.7657
BCS_PGDNET	22.35 | **0.6535**	**23.92** | **0.6743**	**25.51** | **0.7829**	26.11 | **0.7523**	**28.25** | **0.7954**

TABLE IV COMPARISON IN PSNR AND SSIM BASED ON DIFFERENT ALGORITHMS FOR SHIP IMAGE

Methods	0.1	0.15	0.2	0.25	0.3
BCS_SPL_DCT	24.29 | 0.3650	25.57 | 0.4452	26.80 | 0.5226	27.76 | 0.5787	28.87 | 0.6299
BCS_SPL_DDWT	25.05 | 0.3975	**26.43** | 0.4750	27.50 | 0.5426	28.44 | 0.5992	29.43 | 0.6475
BCS_FOCUSS	24.67 | 0.4117	26.20 | **0.4920**	27.54 | 0.5628	28.30 | **0.6111**	29.67 | 0.6546
BCS_PGDNET	**25.35** | **0.4535**	26.32 | 0.4743	**28.81** | **0.5789**	**30.41** | 0.6023	**31.71** | **0.6654**

TABLE V COMPARISON IN PSNR AND SSIM BASED ON DIFFERENT ALGORITHMS FOR CAMERAMAN IMAGE

Methods	0.1	0.15	0.2	0.25	0.3
BCS_SPL_DCT	26.22 | 0.3624	27.43 | 0.4425	29.51 | 0.5225	30.96 | 0.5876	32.13 | 0.6383
BCS_SPL_DDWT	26.53 | 0.4266	27.80 | 0.5165	**29.76** | 0.6029	31.41 | 0.6732	33.17 | 0.7278
BCS_FOCUSS	26.10 | 0.4955	28.52 | **0.5861**	30.63 | 0.6658	**32.97** | 0.7329	34.23 | 0.7842
BCS_PGDNET	**27.27** | **0.5235**	**28.92** | 0.5743	**31.81** | **0.6989**	32.01 | **0.7403**	**34.71** | **0.7954**

We test different images for different texture structure on the simulations. From Figure. 5 to Figure. 8, the BCS algorithms are effective to reconstruct the original image, where the left is the original image and the right are the reconstructed images for different algorisms. Meanwhile, we plot the data tables for different images. From TABLE II to TABLE V, it can be seen that when the sampling rate increases, both the PSNR and SSIM of the reconstructed results both show an increasing trend. For the Barabara and Fishstar images with complex texture, we can clearly find that the proposed method has a relatively high SSIM value in TABLE II and TABLE III. But for the Ship and Cameraman images without complex texture structure, BCS_PGDNET method is generally better than other algorithms even though the sampling rate is 0.1 with lower measurements. In general, the proposed BCS_PGDNET method is largely superior to other BCS methods with the increase of sampling rate.

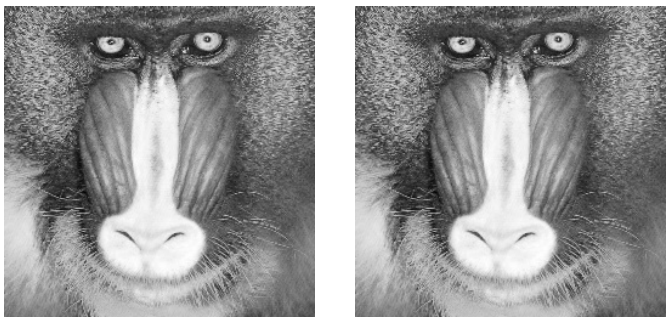
C. Tradeoffs between reconstruction quality and block size

In this part, we discuss the effect of different block size on the quality for reconstruction results. The sampling rate is set at 0.3. For the Peppers(512×512) and Mandrill(256×256) images are chosen to progress experiments at different block size, respectively. The reconstruction results are shown as follows.

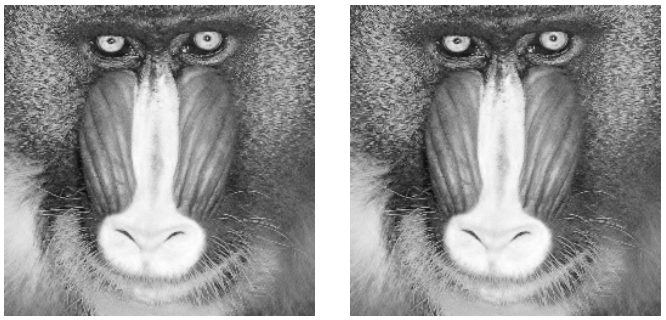
Size is 8×8 Size is 16×16

Size is 32×32 Size is 64×64

Fig. 9. Reconstructed results from different block size for Peppers image



Size is 8×8 Size is 16×16



Size is 32×32 Size is 64×64

Fig. 10. Reconstructed results from different block size for Mandrill image

TABLE VI PSNR OF DIFFERENT BLOCK SIZE FOR PEPPERS IMAGE

	8×8	16×16	32×32	64×64
BCS_PGDNET	28.25	29.45	30.16	31.28

TABLE VII PSNR OF DIFFERENT BLOCK SIZE FOR MANDRILL IMAGE

	8×8	16×16	32×32	64×64
BCS_PGDNET	21.84	22.79	22.06	22.38

From TABLE VI to TABLE VII, for the Peppers image, the reconstruction results has become worse as the block size decreases. The reason is that for the images with simple texture, the smaller the block size and the larger the number of blocks, the results with more block effect. For the Mandrill image, the PSNR value does not change significantly as the block size becomes smaller. Therefore, we select the appropriate block size and sampling rate to reconstruct different images by repeated experiments.

IV. CONCLUSIONS

In this paper, we combine an untrained neural network as prior information with projected gradient descent method for iterative reconstruction. By comparing different reconstruction algorithms with various images, we find that BCS_PGDNET algorithm has better reconstruction results. In the future work, we will investigate the effect of different block shapes on the reconstruction results.

REFERENCES

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

[2] A. G. Garcia, S. O. Escolano, S. Oprea, V. V. Martinez, P. G. Martinez, J. G. Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," Applied Soft Computing, vol. 70, pp. 41-65, 2018.

[3] T. Shimobaba, Y. Endo, T. Nishitsuji, T. Takahashi, Y. Nagahama, S. Hasegawa, M. Sano, R. Hirayama, T. Kakue, A. Shiraki, "Computational ghost imaging using deep learning," Optics Communications, vol. 413, pp. 147-151, 2018.

[4] S. Bazrafkan, H. Javidnia, J. Lemley, "Semiparallel deep neural network hybrid architecture: first application on depth from monocular camera," Journal of Electronic Imaging, vol. 27, no. 4, pp. 41-43, 2018.

[5] H, Bendjenna, A. Meraoumia, Chergui O, "Pattern recognition system: From classical methods to deep learning techniques," Journal of Electronic Imaging, vol. 27, no. 3, pp. 13-25, 2018.

[6] L. Gan, "Block Compressed sensing of natural images," International Conference on Digital Signal Processing, Cardiff, pp. 403-406, 2007.

[7] S. Mun, J. E. Fowler, "Block compressed sensing of images using directional transforms," International Conference on Image Processing, Cairo, pp. 3021-3024, 2009.

[8] A. S. Unde, P. P. Deepthi, "Fast BCS-FOCUSS and DBCS-FOCUSS with augmented Lagrangian and minimum residual methods," Journal of Visual Communication and Image Representation, vol. 52, pp. 99-100, 2018.

[9] A. Adler, D. Boublil, M. Zibulevsky, "Block-based compressed sensing of images via deep learning," Workshop Multimedia Signal Process, vol. 49, no. 51, pp. 753-766, 2017.

[10] E. J. Candès, T. Tao, "Decoding by linear programming," IEEE Transactions on Information Theory, vol. 51, no. 12, pp. 4023-4215, 2005.

[11] L. Junhong, L. Song, "Block Sparse Recovery via Mixed l2/l1 Minimization," Acta Mathematica Sinica, English Series, vol. 29, no. 7, pp. 1401-1412, 2013.

[12] K. Ko, Y. J. Koth, Kim, S. Chang, "Blind and Compact Denoising Network Based on Noise Order Learning," IEEE Transactions on Image Processing, vol. 126, pp. 1657-1670, 2022.

[13] J. Qin, H. Bai, Y. Zhao, "Multi-level augmented inpainting network using spatial similarity," Pattern Recognition: The Journal of the Pattern Recognition Society, vol. 126, no. 7, pp. 108547-1-108547-17, 2022.

[14] S. Wang, C. Qiao, A. Jiang, D. Li, "Instant multicolor super-resolution microscopy with deep convolutional neural network," Biophysics Reports, vol. 7, no. 4, pp. 304-312, 2021.

[15] G. Jagatap, H. Chin, "Algorithmic guarantees for inverse imaging with untrained network priors," International Conference on Neural Information Procesing Systems, Vancouver, pp. 14832-14842, 2019.

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

[17] D. Needell, J. Tropp, "CoSaMP: iterative signal recovery from incomeplete and inaccurate samples," Applied and Computation Harmonic Analysis, vol. 26, pp. 301-321, 2009.

[18] R. Chartrand, W. Yin, "Iteratively Reweighted Algorithms for Compressed Sensing," Computational Optimization and Applications, vol. 76, no. 3, pp. 57-69, 2008.