Block-based Compressed Sensing of Images via Deep Learning

Amir Adler
MIT
McGovern Institute for Brain Research
Cambridge, MA 02319, USA
e-mail: adleram@mit.edu

David Boublil
Technion
Electrical Engineering Department
Haifa 32000, Israel
e-mail: davboub@tx.technion.ac.il

Michael Zibulevsky
Technion
Computer Science Department
Haifa 32000, Israel
e-mail: mzib@cs.technion.ac.il

***Abstract*—Compressed sensing (CS) is a signal processing framework for efficiently reconstructing a signal from a small number of measurements, obtained by linear projections of the signal. Block-based CS is a lightweight CS approach that is mostly suitable for processing very high-dimensional images and videos: it operates on local patches, employs a low-complexity reconstruction operator and requires significantly less memory to store the sensing matrix. In this paper we present a deep learning approach for block-based CS, in which a fully-connected network performs both the block-based linear sensing and non-linear reconstruction stages. During the training phase, the sensing matrix and the non-linear reconstruction operator are *jointly* optimized, and the proposed approach outperforms state-of-the-art both in terms of reconstruction quality and computation time. For example, at a 25% sensing rate the average PSNR advantage is 0.77dB and computation time is over 200-times faster.**

***Index Terms*—block-based compressed sensing, fully-connected neural network, non-linear reconstruction.**

I. Introduction

Compressed sensing [1], [2] is a mathematical framework that defines the conditions and tools for the recovery of a signal from a small number of its linear projections (i.e. measurements). In the CS framework, the measurement device acquires the signal in the linear projections domain, and the full signal is reconstructed by convex optimization techniques. CS has diverse applications including camera sensor design [3], radar imaging [4], Magnetic Resonance Imaging (MRI) [5], [6], spectrum sensing [7], indoor positioning [8], biosignals acquisition [9], and sensor networks [10]. In this paper we address the problem of block-based CS (BCS) [11], which employs CS on distinct low-dimensional segments of a high-dimensional signal. BCS is mostly suitable for processing very high-dimensional images and video, where it operates on distinct local patches. Our approach is based on a deep neural network [12], which jointly learns the linear sensing matrix and the non-linear reconstruction operator (a link to a software package for reproducing all presented results is provided in section IV).

The contributions of this paper are two-fold: (1) It presents for the first time, to the best knowledge of the authors, the utilization of a fully-connected deep neural network for the task of BCS; and (2) The proposed network performs both the linear sensing and non-linear reconstruction operators, and during training these operators are *jointly* optimized, leading to a significant advantage compared to state-of-the-art.

This paper is organized as follows: section II introduces CS concepts, and motivates the utilization of BCS for the acquisition of very high-dimensional images and video. Section III presents the deep neural network approach, and discusses structure and training aspects. Section IV evaluates the performance of the proposed approach for compressively sensing and reconstructing natural images, and compares it with state-of-the-art BCS methods and full-image Total Variation-based CS. Section V concludes the paper and discusses future research directions.

II. Compressed Sensing Overview

A. Full-Signal Compressed Sensing

Given a signal $\mathbf{x} \in \mathbf{R}^N$, an $M \times N$ sensing matrix Φ (such that $M \ll N$) and a measurements vector $\mathbf{y} = \Phi\mathbf{x}$, the goal of CS is to recover the signal from its measurements. The sensing rate is defined by $R = M/N$, and

The research leading to these results has received funding in part from the European Research Council under EU's 7th Framework Program, ERC under Grant 320649, and in part by Israel Science Foundation (ISF) grant no. 1770/14.

978-1-5090-3649-3/17/$31.00 ©2017 IEEE

since $R \ll 1$ the recovery of $\mathbf{x}$ is not possible in the general case. According to CS theory [1], [2], signals that have a sparse representation in the domain of some linear transform can be exactly recovered with high probability from their measurements: let $\mathbf{x} = \Psi\mathbf{c}$, where Ψ is the inverse transform, and $\mathbf{c}$ is a sparse coefficients vector with only $S \ll N$ non-zeros entries, then the recovered signal is synthesized by $\hat{\mathbf{x}} = \Psi\hat{\mathbf{c}}$, and $\hat{\mathbf{c}}$ is obtained by solving the following convex optimization program:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}'} \|\mathbf{c}'\|_1 \text{ subject to } \mathbf{y} = \Phi\Psi\mathbf{c}', \tag{1}$$

where $\|\alpha\|_1$ is the l_1-norm, which is a convex relaxation of the l_0 pseudo-norm that counts the number of non-zero entries of α. The exact recovery of $\mathbf{x}$ is guaranteed with high probability if $\mathbf{c}$ is sufficiently sparse and if certain conditions are met by the sensing matrix and the transform.

B. Block-based Compressed Sensing

Consider applying CS to an image of $L \times L$ pixels: the techniques described above can be employed by column-stacking (or row-stacking) the image to a vector $\mathbf{x} \in \mathbb{R}^{L^2}$, and the dimensions of the measurement matrix Φ and the inverse transform Ψ are $M \times L^2$ and $L^2 \times L^2$, respectively. For modern high-resolution cameras, a typical value of L is in the range of 2000 to 4000, leading to overwhelming memory requirements for storing Φ and Ψ: for example, with $L = 2000$ and a sensing rate $R = 0.1$ the dimensions of Φ are $400,000 \times 4,000,000$ and of Ψ are $4,000,000 \times 4,000,000$. In addition, the computational load required to solve the CS reconstruction problem becomes prohibitively high. Following this line of arguments, a BCS framework was proposed in [16], in which the image is decomposed into non-overlapping blocks (i.e. patches) of $B \times B$ pixels, and each block is compressively sensed independently. The full-size image is obtained by placing each reconstructed block in its location within the reconstructed image canvas, followed by full-image smoothing. The dimensions of the block sensing matrix Φ_B are $B^2R \times B^2$, and the measurement vector of the i-th block is given by:

$$\mathbf{y}_i = \Phi_B\mathbf{x}_i, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^{B^2}$ is the column-stacked block, and Φ_B was chosen in [16] as an orthonormalized i.i.d Gaussian matrix. Following a per-block minimum mean squared error reconstruction stage, a full-image iterative hard-thresholding algorithm is employed for improving full-image quality. An improvement to the performance of this approach was proposed by [13], which employed the same BCS approach as [16] and evaluated the incorporation of directional transforms such as the Contourlet Transform (CT) and the Dual-tree Discrete Wavelet Transform (DDWT) in conjunction with a Smooth Projected Landweber (SPL) [17] reconstruction of the full image. The conclusion of the experiments conducted in [13] was that in most cases the DDWT transform offered the best performance, and we term this method as BCS-SPL-DDWT. A multi-scale approach was proposed by [15], termed MS-BCS-SPL, which improved the performance of BCS-SPL-DDWT by applying the block-based sensing and reconstruction stages in multiple scales and sub-bands of a discrete wavelet transform. A different block dimension was employed for each scale and with a 3-level transform, dimensions of $B = 64, 32, 16$ were set for the high, medium and low resolution scales, respectively. A multi-hypothesis approach was proposed in [14] for images and videos, which is suitable for either spatial domain BCS (termed MH-BCS-SPL) or multi-scale BCS (termed MH-MS-BCS-SPL). In this approach, multiple predictions of a block are computed from neighboring blocks in an initial reconstruction of the full image, and the final prediction of the block is obtained by an optimal linear combination of the multiple predictions. For video frames, previously reconstructed adjacent frames provide the sources for multiple predictions of a block. The multi-scale version of this approach provides the best performance among all above mentioned BCS methods. A survey of BCS theory and performance is provided in [11], which also describes applications such as BCS of multi-view images and video, and motion-compensated BCS of video.

III. The Proposed Approach

In this paper we propose an end-to-end deep learning solution for BCS, which processes each image block independently[1] as described in section II-B. The proposed approach *jointly* optimizes the sensing matrix Φ_B and the non-linear reconstruction operator:

$$\hat{\mathbf{x}} = \mathcal{R}_W(\Phi_B\mathbf{x}), \tag{3}$$

which is parameterized by a coefficients matrix W. The proposed approach provides a solution to the following joint optimization problem:

$$\{\Phi_B, W\} = \arg\min_{\Phi_B, W} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\mathcal{R}_W(\Phi_B\mathbf{x_i}), \mathbf{x_i}), \tag{4}$$

[1]In this paper we treat only block-based processing, and a full-image post-processing stage is not performed.

TABLE I: Average reconstruction PSNR [dB] and SSIM vs. sensing rate (R=M/N): for each method and sensing rate, the result is displayed as PSNR | SSIM (each result is the average over the 10 test images).

Method	R = 0.1	R = 0.15	R = 0.2	R = 0.25	R = 0.3
Proposed (block-size = 16×16)	**28.21 | 0.916**	**29.73 | 0.948**	**31.03 | 0.965**	**32.15 | 0.976**	**33.11 | 0.983**
BCS-SPL-DDWT (16×16)[13]	24.92 | 0.789	26.12 | 0.834	27.17 | 0.873	28.16 | 0.898	29.02 | 0.917
BCS-SPL-DDWT (32×32)[13]	24.99 | 0.781	26.40 | 0.833	27.46 | 0.868	28.43 | 0.894	29.29 | 0.914
MH-BCS-SPL (16×16) [14]	26.01 | 0.827	27.92 | 0.888	29.46 | 0.919	30.69 | 0.939	31.69 | 0.952
MH-BCS-SPL (32×32) [14]	26.79 | 0.845	28.51 | 0.895	29.81 | 0.923	30.77 | 0.938	31.73 | 0.950
MS-BCS-SPL [15]	27.32 | 0.883	28.77 | 0.909	30.04 | 0.934	31.15 | 0.956	32.05 | 0.974
MH-MS-BCS-SPL [14]	27.74 | 0.889	29.10 | 0.919	30.78 | 0.947	31.38 | 0.960	32.82 | 0.979
TV (Full Image) [3]	27.41 | 0.867	28.57 | 0.890	29.62 | 0.909	30.63 | 0.926	31.59 | 0.939

where $\{\mathbf{x_i}\}_{i=1}^{N}$ is a training set of N image blocks $\mathbf{x_i}$. The loss function $\mathcal{L}(\bullet,\bullet)$ measures the distance between the ground-truth image block and the estimated one, provided by the reconstruction operator $\mathcal{R}_W(\bullet)$, whose input is the compressed measurements, denoted by $\Phi_B\mathbf{x_i}$. In this paper we employed the Mean Squared Error (MSE) loss function, which is commonly used for regression networks. Note that during training the sensing layer (matrix) and the subsequent layers, represented by $\mathcal{R}_W(\bullet)$, are treated as a single deep network. However, once training is complete, the sensing matrix is detached from the subsequent inference layers, and used for image block sensing. The input of the reconstruction operator, is therefore the second layer of the end-to-end learned network.

Our choice is motivated by the success of deep neural networks for the task of full-image denoising [18] in which a deep neural network achieved state-of-the-art performance by block-based processing. In our approach, the first hidden layer performs the linear block-based sensing stage (2) and the following hidden layers perform the non-linear reconstruction stage. The advantage and novelty of this approach is that during training, the sensing matrix and the non-linear reconstruction operator are *jointly* optimized, leading to better performance than state-of-the-art at a fraction of the computation time.

The proposed fully-connected network includes the following layers: (1) an input layer with B^2 nodes; (2) a compressed sensing layer with B^2R nodes, $R \ll 1$ (its weights form the sensing matrix); (3) $K \geq 1$ reconstruction layers with B^2T nodes, each followed by a ReLU [19] activation unit, where $T > 1$ is the redundancy factor; and (4) an output layer with B^2 nodes. Note that the performance of the network depends on the block-size B, the number of reconstruction layers K, and their redundancy T. We have evaluated[2] these parameters by a set of experiments that compared the average reconstruction PSNR of 10 test images, depicted in Figure 1, and by training the network with $N = 5,000,000$ distinct image blocks (patches), randomly extracted from 50,000 images in the LabelMe dataset [21]. The chosen optimization algorithm was AdaGrad [22] with a MSE criterion, learning rate of 0.005, batch size of 16, 100 epochs, and without sparsity constraints. Our study revealed that best[3] performance were achieved with a block size $B \times B = 16 \times 16$, $K = 2$ reconstruction layers and redundancy $T = 8$, leading to a total of 4,780,569 parameters (for $R = 0.1$). Table II compares reconstruction quality versus block size, between 8×8 to 20×20 (with 2 reconstruction layers and redundancy of 8), and indicates that block size of 16×16 provides the best results. Table III provides a comparison for varying the redundancy between 2 to 12 (with 2 reconstruction layers and block size of 16×16), and indicates that a redundancy of 8 provides the best results. Table IV provides a comparison for varying the number of hidden reconstruction layers between 1 to 8 (with redundancy of 8 and block size of 16×16), and indicates that two reconstruction layers provided the best performance. Further analysis of the network depth and redundancy is an important topic, which is left for future research.

TABLE II: Reconstruction PSNR [dB] vs. block size ($B \times B$)

Training Examples	$B = 8$	$B = 12$	$B = 16$	$B = 20$
5×10^6	27.21	27.66	**28.21**	27.73

[2]The network was implemented using Torch7 [20] scripting language, and trained on NVIDIA Titan X GPU card.

[3]Note that by increasing significantly the training set, slightly different values of B, K, and T may provide better results, as discussed in [18].

TABLE III: Reconstruction PSNR [dB] vs. network redundancy

Training Examples	$T=2$	$T=4$	$T=8$	$T=12$
5×10^6	27.99	28.11	**28.21**	28.15

TABLE IV: Reconstruction PSNR [dB] vs. no. of reconstruction layers

Training Examples	$K=1$	$K=2$	$K=4$	$K=8$
5×10^6	27.98	**28.21**	28.18	27.55

IV. Performance Evaluation

This section provides performance results of the proposed approach[4] vs. the leading BCS approaches: BCS-SPL-DDWT [13], MS-BCS-SPL [15], MH-BCS-SPL [14] and MH-MS-BCS-SPL [14], using the original code provided by their authors. The proposed approach was employed with block size 16×16, BCS-SPL-DDWT with block sizes of 16×16 and 32×32 (the optimal size for this method), MH-BCS-SPL with block sizes of 16×16 and 32×32 (the optimal size for this method). MS-BCS-SPL and MH-MS-BCS-SPL utilized a 3-level discrete wavelet transform with block sizes as indicated in section II-B (their optimal settings). In addition, we also compared to the classical full-image Total Variation (TV) CS approach of [3] that utilizes a sensing matrix with elements from a discrete cosine transform and Noiselet vectors. Reconstruction performance was evaluated for sensing rates in the range of $R=0.1$ to $R=0.3$, using the average PSNR and SSIM [23] over the collection of 10 test images (512×512 pixels), depicted in Figure 1. Reconstruction results are summarized in Table I, and reveal a consistent advantage of the proposed approach vs. all BCS methods as well as the full-image TV approach. Visual quality comparisons (best viewed in the electronic version of this paper) are provided in Figures 2-5, and demonstrate the high visual quality of the proposed approach. Computation time comparison at a sensing rate $R=0.25$, with a MATLAB implementation of all tested methods (without GPU), is provided in Table V and demonstrates that the proposed approach is over $\times200$ faster than state-of-the-art (MH-MS-BCS-SPL), and over $\times1600$ faster than full-image TV CS.

[4]A MATLAB package for reproducing all the results is available at: http://www.cs.technion.ac.il/~adleram/BCS_DNN_2016.zip

TABLE V: Computation time at R=0.25 (512×512 images):

Method	Time [seconds]
Proposed	0.80
BCS-SPL-DDWT (16×16) [13]	13.57
BCS-SPL-DDWT (32×32) [13]	13.10
MH-BCS-SPL (16×16) [14]	144.61
MH-BCS-SPL (32×32) [14]	69.73
MS-BCS-SPL [15]	6.39
MH-MS-BCS-SPL [14]	207.32
TV (Full Image) [3]	1675.09

V. Conclusions

This paper presents a deep neural network approach to BCS, in which the sensing matrix and the non-linear reconstruction operator are jointly optimized during the training phase. The proposed approach outperforms state-of-the-art both in terms of reconstruction quality and computation time, which is two orders of magnitude faster than the best available BCS method. Our approach can be further improved by using the SSIM metric as a loss function for the deep network training procedure, thus explicitly maximizing visual quality of the reconstructed image blocks.

References

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

[2] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[3] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14–20, March 2008.

[4] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, "Sparsity and compressed sensing in radar imaging," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1006–1020, June 2010.

[5] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing mri," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, March 2008.

[6] M. Murphy, M. Alley, J. Demmel, K. Keutzer, S. Vasanawala, and M. Lustig, "Fast l_1 -spirit compressed sensing parallel imaging mri: Scalable parallel implementation and clinically feasible runtime," *IEEE Transactions on Medical Imaging*, vol. 31, no. 6, pp. 1250–1262, June 2012.

[7] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio : State-of-the-art and recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101–116, May 2012.

[8] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, pp. 1983–1993, Dec 2012.

[9] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ecg and emg wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, April 2012.

Fig. 1: Test images (512×512): 'lena', 'bridge', 'barbara', 'peppers', 'mandril', 'houses', 'woman', 'boats', 'cameraman' and 'couple'.

Fig. 2: Reconstruction of 'couple' at R = 0.1 (PSNR [dB] |SSIM): (a) Original, (b) Full image TV (27.1691 | 0.8812), (c) MS-BCS-SPL, (d) MH-MS-BCS-SPL (27.1804 | 0.8877); and (e) Proposed (28.5902 | 0.9414).

Fig. 3: Reconstruction of 'houses' at R = 0.2 (PSNR [dB] | SSIM): (a) Original, (b) Full image TV (31.0490 | 0.9304), (c) MS-BCS-SPL, (d) MH-MS-BCS-SPL (31.7030 | 0.9544); and (e) Proposed (32.9328 | 0.9766).

Fig. 4: Reconstruction of 'lena' at R = 0.25 (PSNR [dB] | SSIM): (a) Original, (b) Full image TV (35.4202 | 0.9718), (c) MS-BCS-SPL, (d) MH-MS-BCS-SPL (35.7346 | 0.9825; and (e) Proposed (36.3734 | 0.9910).

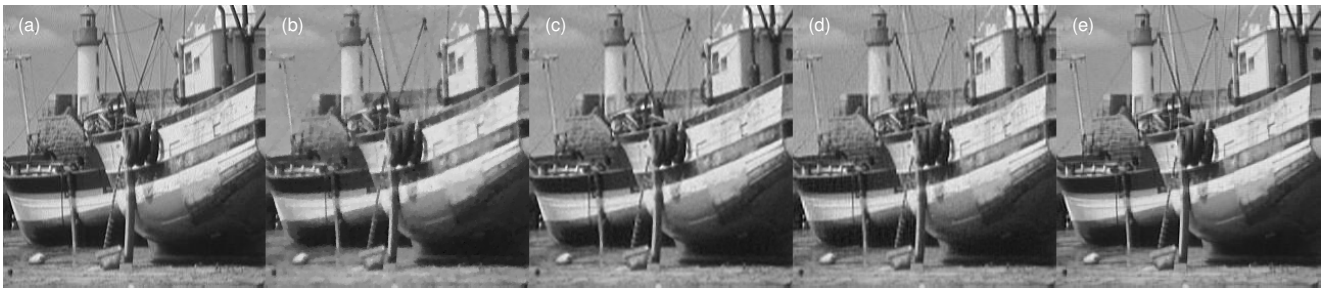

Fig. 5: Reconstruction of 'boats' at R = 0.25 (PSNR[dB] | SSIM): (a) Original, (b) Full image TV (31.5676 | 0.9493), (c) MS-BCS-SPL, (d) MH-MS-BCS-SPL (31.1675 | 0.9641); and (e) Proposed (33.0422 | 0.9873).

[10] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, Nov 2013.

[11] J. E. Fowler, S. Mun, and E. W. Tramel, "Block-based compressed sensing of images and video," *Foundations and Trends in Signal Processing*, vol. 4, no. 4, pp. 297–416, 2012.

[12] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

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

[14] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov 2011, pp. 1193–1198.

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

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

[17] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, CRC Press, 1998.

[18] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2392–2399.

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

[20] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[21] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 157–173, May 2008.

[22] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, July 2011.

[23] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.