Application of deep learning and compressed sensing for reconstruction of images

Hanumanth P1,4, Bhavana P2,5 and Shreyanka Subbarayappa2,3,6

1Department of Computer Science and Engineering, M S Ramaiah University of Applied Sciences, #470-P, 4th Phase, Peenya Industrial Area, Bengaluru, KA 560058, India.
2Department of Electronic and Communication Engineering, M S Ramaiah University of Applied Sciences, #470-P, 4th Phase, Peenya Industrial Area, Bengaluru, KA 560058 India.
3Electrical Engineering Department, University of Texas at Arlington, Arlington, TX 76019, U.S.A

E-mail: 4hanumanth2609@gmail.com, 5pbhavana399@gmail.com, 6shreyanka.ec.et@msruas.ac.in

Abstract. Compressed sensing (CS) is a technique in signal processing which reconstructs any given signal at a rate less than that of Nyquist’s rate given that the signal is sparse and incoherent in nature. The main focus of CS is to find a random matrix which reconstructs the original signal using as few samples as possible. In recent years there has been a lot of interest in using CS to reconstruct 2D images. Several machine learning and deep learning algorithms have been proposed for finding the random matrix. However, using deep learning techniques to generate the random matrix is still an emerging concept. There are several papers exploring the concepts of deep learning with compressed sensing for images and have produced promising results. The results obtained in these papers warrant a comparison to analyze their performances against more traditional methods to determine the best approach to reconstruct images. These results are also compared with the popular JPEG and JPEG2000 codecs. This paper focuses on surveying deep learning algorithms for reconstruction of images using CS. The comparison reveals that deep learning algorithms perform significantly better than traditional methods and perform well when compared to any image compression codec like JPEG. Possible methods to improve the existing algorithms have been suggested.

1. Introduction and Background

Image compression has been carried out using traditional lossy and lossless compression algorithms for many years. Popular lossy compression methods like the discrete cosine transform [1] have been used in JPEG to achieve a 10:1 compression ratio. However, with the emergence of compressed sensing [2][3], for applications satisfying the sparse sensing conditions, it has proven to produce results with a much better compression ratio while sampling at a rate lower than Nyquist’s rate. In order to use compressed sensing to reconstruct images, two conditions need to be met. The signal must be sparse in any domain and the signal must be incoherent. For a signal to be sparse, the majority of the samples must be zero, with only a few non-zero samples.

For a given $x$, a real valued signal such that $x \in \mathbb{R}^N$ where $N$ is the number of samples, the aim is to find a random sensing matrix $\varphi \in \mathbb{R}^{M \times N}$ where $M \ll N$, then the sparse samples would be obtained using.
Where, $y \in \mathbb{R}^M$ is a vector with CS measurements.

The sensing matrix must be chosen carefully based on the application. Choosing a random matrix suffers unpredictable outcomes. It also has been proven to have high complexity along with huge memory requirements. Alternatively, matrices can be chosen specifically for an application so that the aforementioned limitations are reduced [4][5], however, generating the sensing matrix even with these approaches is difficult and somewhat tedious. Using learning algorithms to generate this matrix is advantageous as the matrix is generated specific to the application.

The next important step in CS is to successfully reconstruct the original image using the sensing matrix. There have been many compressed sensing algorithms based on reconstruction such as minimization of total variation [6][7][8], basis pursuit [9][46], group-based sparse representation [10], discrete wavelet transform [11], multihypothesis predictions [12], collaborative sparsity [13] and [57].

From [14] using DWT algorithm, it can be observed that the best reconstruction algorithm in terms of execution time was around 10 seconds. Almost all of these require an optimization problem to be solved in order to perform the reconstruction. This in turn means that the optimization problem would require hundreds of iterations in order to arrive at a suitable solution [47]. Leading to higher computation cost and poor execution time when it comes to deploying it in an application. Discrete wavelet transform (DWT) [56] and total variation (TV) algorithms have lower execution times at the cost of a lower quality reconstruction.

These algorithms are not suitable for real-world applications if each of their execution times are in magnitudes of 10. It simply wouldn’t be feasible. Further research in this field is being done to better the performance of the algorithm [53], but reducing the complexities of the algorithms [50] does not seem to be a priority. Deep learning has been very useful in extracting complicated information from images as previous researchers have suggested [15][16][52][54][60][62]. Neural network-based algorithms directly learn the inverse mapping [55] to the original image domain from the CS measurement domain. Thus, the computation cost decreases and the performance of the algorithms are relatively better.

Hence, deep learning can be used to reconstruct 2D images while producing better execution times than the existing non-learning algorithms.

The paper focuses on how deep learning solves two major problems in CS; generating the sensing matrix and reconstructing the image, while maintaining a better performance than the non-learning solutions and producing results that can be used in real-life applications. The performance will also be measured against standard image codecs like JPEG and JPEG2000 to quantify where it stands when compared with popular image codecs.

2. Our focus
We are focusing on three things:(1) to survey the latest research in CS using deep learning; (2) to compare the latest research in CS using deep learning with traditional methods evaluated based on PSNR metric, SSIM metric and run-time cycle; (3) to compare the results of the latest algorithms in CS using deep learning with image codecs like JPEG and JPEG2000.

3. Analysis of algorithms
We analyze deep learning algorithms for CS and their performances with special reference to quality metrics like PSNR, SSIM and their run time.

3.1. Deep learning sparse ternary projections [17]
This approach talks about hardware implementation of CS using DL algorithms [48] with the main focus being storage efficiency and fast computations. It simplifies the deep neural networks by using binary weights {-1, +1}. The algorithm uses mini-batch gradient descent method for training. The basic idea lies in obtaining the matrix with ternary values i.e., {-1, +1, 0} instead of real valued dense matrix.

The model learns the projection matrix and performs non-linear reconstruction. The algorithm imposes sparsity and binary constraint by taking sparse matrices composed of {0, -1, +1}. This makes it efficient since all the real valued entries will not be used. This is combined with binarization techniques [49] to produce highly sparse ternary projections. The network trains the system by using image patches and producing output images in a block-based manner.

The network consists of a sensing module and reconstruction module as shown in figure 1. The sensing module takes in vectorized patches of image of size \( N = S^2 \), where \( S \) is the sparse signal, and projects the input \( x \) of dimension \( N \) into a domain of dimension \( M \). This implies the number of units in this layer will be \( M = S^2 R \) and \( R \) is the sensing rate with values between 0 to 1 and in order to keep the projection matrix simple, they do not apply bias or non-linearity activation filter. This layer uses \{0, -1, +1\} as weights. The reconstruction layer consists of scaling layer, \( L \) hidden and an output layer. The layers are followed by a batch normalization layer at the end which helps at reducing covariance shift.

The scaling layer scales the output of the sensing module by the learnt factors \( \alpha \). All layers except the scaling layer are fully connected. After the scaling layer, there are \( L \) hidden layers which use ReLU activation function [18]. Using ReLU activation function has its own obvious reasons such as ease of computation, fast convergence, does not have vanishing gradient problem and is a sparsely activated network which implies that this layer is activated only when needed. The output layer is linear and has dimensions equal to input dimensions. The outputs are normalized using batch normalization layer [19].

![Figure 1: Deep learning framework to obtain sparse ternary projections. Here, \( \theta_{sb} \) represents sparse binary weights][17]

The training algorithm uses a non-conventional mini batch gradient descent method with sparsifying and binarization steps for training the sensing layer. The continuous weights in the sensing layer get converted to sparse weights of dimensions \( N \times M \), this procedure is referred to as \( top_k \_select \) function in a column wise manner where top weights are selected from the layer. The continuous sparse weights are mapped to sparse binary weights containing values \{-1, 0, +1\} in a step called binarization. The loss from binarizing the weights is compensated using hidden units in scaling layer connected one to one to hidden units in sensing layer [20]. This results in a sparse matrix with \( K \) non-zero entries of \{-1, +1\} and existing training algorithms are used for further training.

Overall, this algorithm simplifies the neural networks by binarization of the weights giving a highly sparse ternary projection matrix. This simplification allows the reconstruction module to be non-linear which helps in improving the performance. The results obtained indicate that the algorithm outperforms algorithms like O-NL-SDA [21] and BP [22] while using only 5% of non-zero binary entries.

3.2. CS framework using Convolutional Neural Networks (CSNet) [14]
This framework comprises a sampling network and a reconstruction network. Both these networks are jointly optimized to better their performance with respect to reconstruction quality and run time. CSNet uses CNN for block based compressed sensing. The proposed network as shown in figure 2, uses CNN for mainly three functions: Block based compressed sampling, initial reconstruction, and non-linear signal reconstruction.

In the sampling network, three sampling matrices are learned. They are floating-point matrix, binary \{0, 1\}, and bipolar \{+1, -1\}. Binary and bipolar matrices are learned to allow hardware implementation and easy storage. Conventionally, block based compressed sensing uses a pseudo inverse matrix [10][23] to acquire an initial reconstructed matrix, this paper proposes a network which uses adaptive optimization in the network for reconstruction. The algorithm learns the sampling matrix from the training set to retain more structural information. Deterministic binarization is used instead of stochastic binarization to quantize the elements of the sampling matrix. Reconstruction network comprises of linear initial reconstruction network that generates an initial reconstructed image using a convolution layer and a combination layer [51]. A non-linear deep reconstruction network learns end-to-end mapping using a deep residual network to improve reconstruction quality of the images from CS measurements. Multiple convolutional filters are used to obtain initial reconstructed blocks, these blocks are reshaped and concatenated using a combination layer which contains functions to do the same. The initial reconstruction optimizes the entire image which enables utilization of both intra block and inter block information. The deep reconstruction network uses residual learning for non-linear signal reconstruction. Reconstruction is done using three operations: feature extraction, non-linear mapping and feature aggregation. Feature extraction is carried out using a convolution layer followed by an activation layer (ReLU), this produces high dimensional feature from the local receptive field. After the extraction of the feature, the deep reconstruction network cascades residual block, convolution layer and activation layer for non-linear mapping and increase receptive fields. Finally, the feature aggregation produces the reconstructed image using high dimensional features which were priorly extracted.

![Figure 2: Framework for CSNet. [14]](image)

For training the floating-point matrix, gradient of parameters is calculated and updated. For training the bipolar and binary matrices, deterministic binarization is used as suggested in [24][25] and elements of sampling matrix are quantized to binary and bipolar forms. During training, the sampling network and reconstruction network form an end-to-end network. During the application phase, the sampling network behaves as an encoder and the reconstruction network as a decoder.

3.3. Cascaded reconstruction networks (CSRNet and ASRNet) [26]
Compatibly sampling reconstruction network reconstructs high-quality images using random samples (CS measurements) from a random sampling matrix. The output from the reconstruction network of CSRNet is passed through CNN based residual network module to obtain higher quality images. CSRNet is not a complete CS image reconstruction network as it only reconstructs the image, it does not contain a sampling network.

CSRNet as shown in figure 3, contains three modules and they are Initial reconstruction module, deep reconstruction module and residual reconstruction module. Initial reconstruction module reconstructs
initial image directly from CS measurements. Deep reconstruction module reconstructs images using three convolution layers, the first two convolution layers are followed by ReLU layer and produce an image of good quality which is used as an input for the third layer. The first layer uses kernel of size 11x11 and generates 64 feature maps. The second layer generates 32 features maps and uses kernel of size 1x1. Finally, the third layer uses kernel of size 7x7 and generates 1 feature map which is the output of the second module. Residual reconstruction module reconstructs a higher quality image by learning the residual information between input data and original image.

![Figure 3: Framework for CSRNet. [26]](image)

Adaptively sampling reconstruction network is a complete compressed sensing network containing a sampling module and a reconstruction module unlike CSRNet. ASRNet learns the sampling matrix using fully connected layers in the automatically sampling module and matches it with the residual reconstruction module.

ASRNet, as shown in figure 4, has three modules; Sampling module which uses a fully connected layer to achieve traditional compressive sampling. Initial reconstruction module uses a fully connected layer to generate an initial reconstructed image where the sensing matrix is automatically learned instead of being computed. The residual reconstruction module is same as that of CSRNet. ASRNet is also useful because the entire network is end to end and they can be trained together.

![Figure 4: Framework for ASRNet. [26]](image)

BM3D denoiser is used in the final stage to remove the artifacts from patch processing. Both ASRNet and CSRNet achieve better results than state of the art algorithms while the former achieves more than 1dB gain than CSRNet.

### 3.4. ReconNet [29]

Computer vision tasks like recognition where convolution neural networks have been applied, use images as input to the network. However, images need to be obtained as outputs from the CNN used for reconstruction purposes. So, a typical CNN architecture which maps images to features cannot be used. This paper uses [30] as inspiration. [30] uses bicubic interpolation to obtain initial estimates of high-resolution images from the low-resolution images. It further uses a 3-layer CNN which uses these initial estimates as input and uses the ground truth of the required output as labels. However, in this paper, they propose ReconNet which does not try to obtain initial estimates, instead, it proposes a CNN architecture which can directly map CS measurements to image blocks.
This method proposes a non-iterative algorithm using a CNN that takes CS entries as input and produces an intermediate reconstructed image. This intermediate image is fed to a denoiser to obtain the final reconstructed image. The algorithm is demonstrated to be very effective and perform better than its competitors even at low measurement rates of 0.1 or lower.

As shown in figure 8, the first layer of ReconNet is a fully connected layer that takes CS measurements as input and produces feature maps of size 33 x 33 as output. All subsequent layers are convolution layers that are followed by the ReLU activation function except the final layer. The final layer generates the intermediate reconstructed image. Finally, the noise in the intermediate image is removed using BM3D \cite{31} denoiser.

Patches of size 33x33 are extracted from the input to obtain a set of 21760 patches where the stride is equal to 14. Stride is essentially the number of bytes that needs to be added to the first pixel of a row to reach the address of the first pixel in the next row. Only the luminance component of each image is extracted which form the labels of the training set. The CS measurements of these patches form the inputs of the training set. To obtain the CS measurements, a random Gaussian matrix is generated and a measurement matrix $\Phi$ is constructed. Then each of its rows are orthonormalized, followed by applying $y = \Phi x$ to obtain the measurements. Here, $x$ is the luminance component of a patch after it is vectorized. Finally, the networks are trained using Caffe \cite{32,63}.

3.5. Multi-channel deep networks for block-based CS (BCSNet) \cite{33}

In order to efficiently deal with high dimensional natural images, block-based compressed sensing was proposed. Here, an image is partitioned into blocks and each block is sampled and reconstructed independently. The problem that block-based CS brings is that image quality is reduced because of the blocking artifacts which are discussed in \cite{34} and \cite{35}. A few methods were proposed in \cite{34,35} and \cite{36} which uses an iterative algorithm where at each iteration, an approximate of each block is obtained by using the projection operation and the denoising operation \cite{59} is used on the image that is reassembled using all the approximate blocks. Results proved that although the results obtained were satisfactory, the reconstruction time was too long. Many deep neural networks-based algorithms were also proposed \cite{27}, \cite{29}, \cite{37}, \cite{38} but it turned out that the blocking artifacts still existed especially when the sampling rates were low. Also, reconstruction accuracy was affected because the neural network-based methods ignore the structural insight of CS reconstruction algorithms and are trained as a black box. Image CS using deep learning has its performance reduced by using different sampling rates for each block in an image. This paper proposes multi-channel deep networks for block-based CS where sampling rates are assigned to each block and blocking artifacts are removed at the model-level. It is called BCSNet as it uses the block-based CS algorithm in the learning network.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{Framework for BCSNet. \cite{33}}
\end{figure}

Each of the blocks with the multiple sampling rates are given to the deep reconstruction module as input. The blocking artifacts are removed by taking advantage of the inter-block correlation. The reconstruction network is divided into residual layers. Each of those layers correspond to an iteration of the BCS algorithm. This approach is very interesting because it makes use of both types of algorithms to achieve its goal of high-quality reconstruction. It makes use of the deep learning networks to learn the structure of BCS to produce good quality outputs.
3.6. Deep Residual Reconstruction Network for Image CS [39]

DR2Net comprises of two modules; the linear mapping network and the residual network. The linear mapping network is implemented using a fully connected layer and generates a preliminary image. The preliminary image needs to be enhanced to achieve better image quality. This is done in the residual network using residual learning blocks [28][61]. The algorithm divides the initial image into 33 x 33 patches. CS entries are recorded for each of these patches and DR2Net uses these entries to reconstruct the image as shown in figure 6.

Once the linear mapping network generates the preliminary image, the residual network works on finding the residual. It does so by using four residual learning blocks. Each of the four blocks contains three convolution layers and the first two convolution layers are followed by the ReLU activation function. The first convolution layer in each of the blocks generates 64 feature maps using kernel of size 11x11. The second layer generates 32 feature maps with kernel size 1x1. Finally, the third layer uses kernel of size 7x7 to generate 1 feature map which is the output.
The reconstruction of the image is performed by first extracting CS measurement from the 33x33 image patches that are obtained. The CS measurements are taken as input to DR2Net and it outputs the reconstructed patches. The reconstructed patches form an intermediate reconstructed image which is fed to BM3D denoiser to remove any artifacts.

3.7. Interpretable Optimization-Inspired deep network for image CS [38]
This algorithm is dubbed ISTA-Net, shown in figure 7. It uses Iterative shrinkage-thresholding algorithm (ISTA) and maps it onto a deep network. Each of the phases in this algorithm are like iterations in ISTA. ISTA-Net uses back-propagation to learn parameters like shrinkage thresholds, nonlinear transforms, etc., and they are learned end to end. Essentially, all the parameters involved in ISTA are learned automatically by the deep network instead of being fixed manually. Nonlinear sparsifying transform has a proximal mapping problem that is solved efficiently by ISTA-Net that allow's for other optimization algorithms to be mapped to deep learning networks. Residuals of natural images and videos are more compressible and this fact is used to design a more enhanced ISTA-Net called ISTA-Net+. This enhanced algorithm strives to better CS performance and from the results obtained, it performs even better than ISTA-Net in terms of speed and quality.

4. Comparison
In this section, we have compiled PSNR and SSIM values for traditional algorithms like total variation minimization by augmented Lagrangian and alternating direction algorithms (TVAL3) [40], distributed approximate message passing (DAMP) [41][58], stacked denoising autoencoder (SDA) [42] along with algorithms surveyed in the previous section: ReconNet, DR2-Net, CSRNet, ASRNet, ISTA-Net, CSNet (binary, bipolar, floating point), and BCSNet. All images used for comparison are grayscale images with different measurement rates (MR) (compression ratios). The metric values are obtained without the BM3D denoiser.

In figure 9 and figure 10, PSNR and SSIM values have been compiled for the images (from the standard data set) Parrot of dimensions 512*512 with a MR 0.1 and a block size of 256*256. Monarch has been used with a dimension of 256*256. For MR 0.25. Fingerprint of size 256*256 with MR 0.1. For Parrot, BCSNet slightly outperforms CSNet (floating point) with both PSNR and SSIM and these algorithms significantly outperform all the other networks.

Figure 11 illustrates the output of these networks. For Monarch, ISTA-Net outperforms the other algorithms. However, for a MR of 0.3 CSNet (floating point) performs better. Figure 12 illustrates the output of these networks.

For Fingerprint, BCSNet and CSNet (floating point) have similar performance with respect to PSNR and SSIM. It is observed that they perform much better than other networks and the traditional methods.

In figure 13, the average runtimes have been computed for all the networks along with the traditional algorithms using a standard 256*256 image with MR 0.1. It is observed that SDA has the fastest execution time, although the results produced aren’t as good as the networks surveyed in previous sections. All the networks have executions times ranging from 0.01s to 1.37s, with ReconNet being the fastest algorithm. These execution times are quite fast with high reconstruction quality.

From comparisons with respect to PSNR and SSIM, it can be concluded that BCSNet and CSNet (floating point) outperform all the other networks. BCSNet does well because it uses a dynamic sampling rate depending on different blocks within an image. It also combines the traditional BCS with neural networks resulting in better PSNR and SSIM values. CSNet (floating point) does well because it uses neural networks to implement BCS [43][44].

Figure 14 and figure 15, compares JPEG and JPEG2000 with all the surveyed networks to compare these reconstruction algorithms with complete image codecs. Barbara (from the standard image data set) is
used for this comparison with different quality factors. It is quite clear that JPEG and JPEG2000 perform significantly better than the networks with respect to PSNR and SSIM. Figure 16 and figure 17 illustrates the output of these networks. However, it is not always the case that these metrics are perfect models to represent human perception. From the outputs in [45], the images are subjectively of higher quality. There are several reasons why JPEG quantifiably performs better than these networks.

JPEG and JPEG 2000 are complete codecs with filtering and post processing techniques which weigh in for higher PSNR and SSIM values obtained whereas the PSNR and SSIM compiled for the surveyed networks only reconstruct the images without any post processing and still produce acceptable results. It should also be considered that JPEG uses a sample rate (higher than Nyquist’s’ rate) much higher than that used by each of these networks. As discussed in Section IV, the networks use around 0.5% of the samples to reconstruct the image.

![Figure 9: Graphical representation of different traditional methods and surveyed networks with respect to PSNR for Parrot, Monarch and Fingerprint content.](image)

![Figure 10: Graphical representation of different traditional methods and surveyed networks with respect to SSIM for Parrot, Monarch and Fingerprint content.](image)
Figure 11: (a) Original image of Parrot, (b) TVAL3 Parrot [26], (c) DAMP Parrot [26], (d) ReconNet Parrot [26], (e) DR2-Net Parrot [26], (f) CSRNet Parrot [26], (g) ASRNet Parrot [26], (h) BCSNet Parrot [14].

Figure 12: (a) Original image of Monarch, (b) TVAL3 Monarch [26], (c) DAMP Monarch [26], (d) ReconNet Monarch [26], (e) DR2-Net Monarch [26], (f) CSRNet Monarch [26], (g) ASRNet Monarch [26].

Figure 13: Graphical representation of different traditional methods and surveyed networks with respect to time for Fingerprint content.

Figure 14: Graphical representation of different traditional methods and surveyed networks with respect to PSNR for Barbara content.
5. Conclusion

In this paper, we have surveyed all the algorithms for Deep Learning in CS for 2-D images. It is quite clear that Deep Learning coupled with CS is a potent combination achieving both fast execution times and high-quality reconstructions. Where traditional algorithms fail in terms of producing quality results within acceptable times, Deep Learning in CS succeeds by producing both acceptable run-times as well as subjectively good reconstructed images. On comparison, it was observed that BCSNet and CSNet (floating point) were the best performing networks. Additionally, all the surveyed networks outperformed traditional methods. It can be concluded that using Deep Learning in CS is the best way forward for
reconstructing 2-D images. These networks were outperformed by JPEG and JPEG2000 but it should be considered that these networks use only a fraction of the samples from the original image whereas JPEG uses a sample rate higher than Nyquist’s rate.

As part of the future work in this field, incorporating these networks in these codecs for reconstruction of images would increase the performance of the codecs while exploiting the advantages of CS; it collects only a fraction of data instead of discarding information after collecting the entire data leading to efficient data storage.

6. References
[1] Suhail M and Obaidat M 2003 Digital watermarking-based DCT and JPEG model IEEE Transactions on Instrumentation and Measurement 52 1640-47
[2] Donoho D 2006 Compressed sensing IEEE Transactions on Information Theory 52 1289-306
[3] Kutyniok G and Eldar Y 2012 Compressed sensing: Theory and applications Cambridge university press 36 79-101
[4] Rani M, Dhok S and Deshmukh R 2018 A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications IEEE Access 6 4875-94
[5] Gong P and Jieping Y 2015 A modified orthant-wise limited memory quasi-Newton method with convergence analysis International Conference on Machine Learning
[6] Han W, Yu H and Wang G 2009 A General Total Variation Minimization Theorem for Compressed Sensing Based Interior Tomography International Journal of Biomedical Imaging 2009 1-3
[7] Wang Y, Yang J, Yin W and Zhang Y 2008 A New Alternating Minimization Algorithm for Total Variation Image Reconstruction SIAM Journal on Imaging Sciences 1 248-72
[8] Chambolle A and Lions P 1997 Image recovery via total variation minimization and related problems Numerische Mathematik 76 167-88
[9] Yin W, Osher S, Goldfarb D and Darbon J 2008 Bregman Iterative Algorithms for $l_1$ Minimization with Applications to Compressed Sensing SIAM Journal on Imaging Sciences 1 143-68
[10] Zhang J, Zhao D and Gao W 2014 Group-Based Sparse Representation for Image Restoration IEEE Transactions on Image Processing 23 3336-51
[11] Shin D and Moon Y 2014 Super-Resolution Image Reconstruction Using Wavelet Based Patch and Discrete Wavelet Transform Journal of Signal Processing Systems 81 71-81
[12] Chen C, Tramel E W and Fowler J E 2011 Compressed-sensing recovery of images and video using multi hypothesis predictions conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR) IEEE
[13] Zhang J, Zhao D, Zhao C, Xiong R, Ma S and Gao W 2012 Image Compressive Sensing Recovery via Collaborative Sparsity IEEE Journal on Emerging and Selected Topics in Circuits and Systems 2 380-91
[14] Shi W, Jiang F, Liu S and Zhao D 2019 Image Compressed Sensing Using Convolutional Neural Network IEEE Transactions on Image Processing 29 375-88
[15] Chen Y, Jiang H, Li C, Jia X and Ghamisi P 2016 Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks IEEE Transactions on Geoscience and Remote Sensing 54 6232-51
[16] Liu W, Wang Z, Liu X, Zeng N, Liu Y and Alsaadi F 2017 A survey of deep neural network architectures and their applications Neurocomputing 234 11-26
[17] Nguyen DM, Tsiligianni E and Deligiannis N2017 Deep learning sparse ternary projections for
compressed sensing of images IEEE Global Conference on Signal and Information Processing
1125-29
[18] Glorot X, Bordes A and Bengio Y 2011 Deep sparse rectifier neural networks Proceedings of the
fourteenth international conference on artificial intelligence and statistics 315-23
[19] Ioffe S and Szegedy C 2015 Batch normalization: Accelerating deep network training by reducing
internal covariate shift arXiv preprint arXiv
[20] Rastegari M, Ordonez V, Redmon J and Farhadi A 2016 Xnor-net: ImageNet classification using
binary convolutional neural networks European conference on computer vision Springer Cham.,
525-42
[21] Mousavi A, Patel AB and Baraniuk RG 2015 A deep learning approach to structured signal
recovery 53rd annual allerton conference on communication, control, and computing
(Allerton) 1336-43
[22] Chen S, Donoho D and Saunders M 2001 Atomic Decomposition by Basis Pursuit SIAM Review 43
129-59
[23] Mun S and Fowler JE 2009 Block compressed sensing of images using directional transforms16th
IEEE international conference on image processing (ICIP) 3021-24
[24] Amini A and Marvasti F 2011 Deterministic Construction of Binary, Bipolar, and Ternary
Compressed Sensing Matrices IEEE Transactions on Information Theory 57 2360-70
[25] Lu W, Dai T and Xia ST 2017 Binary matrices for compressed sensing IEEE Transactions on
Signal Processing 66(1) 77-85
[26] Wang Y, Bai H, Zhao L and Zhao Y 2018 Cascade reconstruction network for compressive image
sensing EURASIP Journal on Image and Video Processing 2018
[27] Shi W, Jiang F, Liu Sand Zhao D 2018 Multi-scale deep networks for image compressed sensing
25th IEEE International Conference on Image Processing (ICIP) 46-50
[28] HeK, Zhang X, Ren Sand Sun J 2016 Deep residual learning for image recognition Proceedings of the
IEEE conference on computer vision and pattern recognition 770-8
[29] Kulkarni K, Lohit S, Turaga P, Kerviche R and Ashok A 2016 ReconNet: Non-iterative
reconstruction of images from compressively sensed measurements Proceedings of the IEEE
Conference on Computer Vision and Pattern Recognition 449-58
[30] Dong C, Loy CC, He K and Tang X 2014 Learning a deep convolutional network for image super-
resolution European conference on computer vision 184-99 Springer Cham
[31] Dabov K, Foi A, Katkovnik V and Egiazarian K 2007 Image Denoising by Sparse 3-D Transform-
Domain Collaborative Filtering IEEE Transactions on Image Processing 16 2080-95
[32] VedaldiA, Jia Y, Shlens J, Donahue J, Karayev S, Long J and Darrell T 2014 Convolutional
architecture for fast feature embedding Cornell University arXiv: 1408.5093 v12014
[33] Zhou S, He Y, Liu Yand Li C 2019 Multi-Channel Deep Networks for Block-Based Image
Compressive Sensing arXiv preprint arXiv:1908.11221
[34] Fowler JE, Mun S and Tramel EW 2012 Block-based compressed sensing of images and
video Foundations and Trends in Signal Processing 4(4) 297-416
[35] Dinh KQ, Shim H and Jeon B 2013 Measurement coding for compressive imaging using a
structural measurement matrix IEEE International Conference on Image Processing 10-13 IEEE.
[36] Gan L 2007 Block compressed sensing of natural images 15th Intl. conf. on digital signal
processing. IEEE 403-06
[37] Lohit S, Kulkarni K, Kerviche R, Turaga P and Ashok A 2018 Convolutional neural networks for
noniterative reconstruction of compressively sensed images IEEE Transactions on
Computational Imaging 4 326-40
[38] Zhang J and Bernard Ghanem 2018 ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing Proceedings of the IEEE conf. on computer vision and pattern recognition 1828-37

[39] Yao H, Dai F, Zhang S, Zhang Y, Tian Q and Xu C 2019 Dr2-net: Deep residual reconstruction network for image compressive sensing Neurocomputing 359 483-493

[40] Li C, Yin W and Zhang Y 2009 User’s guide for TVAL3: TV minimization by augmented lagrangian and alternating direction algorithms CAAM report 20 46-47

[41] Han P, Niu R and Ren M 2014 Distributed approximate message passing for compressed sensing arXiv preprint arXiv:1404.3766

[42] Gondara L 2016 Medical image denoising using convolutional denoising autoencoders 2016 IEEE 16th ICDMW 241-46

[43] Fowler J E, Mun S and Tramel E W 2012 Block-based compressed sensing of images and video Foundations and Trends® in Signal Processing 4.4 297-416

[44] Jha M, Yummam AS and Raju D 2014 Comparison between image codecs: JPEG and JPEG2000 2014 Intl. Conf. on Computing for Sustainable Global Development (INDIACom). IEEE

[45] Sidky E Y and Pan X 2008 Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization Physics in Medicine & Biology 53,174777

[46] Chen S and Donoho D 1994 Basis pursuit Proceedings of 1994 28th Asilomar Conf. on Signals, Systems and Computers, IEEE1 41-4

[47] Kingma D P and Ba J 2014 Adam: A method for stochastic optimization arXiv preprint arXiv:1412.6980

[48] Han S, Pool J, Tran J and Dally W 2015 Learning both weights and connections for efficient neural network Advances in neural info. proc. systems 1135-43

[49] Courbariaux M, Hubara I, Soudry D, El-Yaniv R and Bengio Y 2016 Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1 arXiv preprint arXiv:1602.02830

[50] Donoho DL, Tsaig Y, Drori I and Starck JL2012Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit IEEE transactions on Info. Theory 58.2 1094-121.

[51] Kim J, Lee J K and Mu Lee K 2016 Accurate image super-resolution using very deep convolutional networks Proceedings of the IEEE conf. on computer vision and pattern recognition 1646-54

[52] Wang J, Zhu S and Gong Y 2010 Resolution enhancement based on learning the sparse association of image patches Pattern Recognition Letters 31.1 1-10

[53] Li P, Page T, Luo G, Zhang W, Wang P, Zhang P, Maass P, Jiang M and Cong J 2014 Fpga acceleration for simultaneous medical image reconstruction and segmentation 2014 IEEE 22nd Annual Intl. Symposium on Field-Programmable Custom Computing Machines 172-172

[54] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks Advances in neural info. proc. systems 1097-105

[55] Mousavi A and Baraniuk R G 2017 Learning to invert: Signal recovery via deep convolutional networks 2017 IEEE ICASSP 2272-76

[56] Kim Y, Nadar M S and Bilgin A 2010 Compressed sensing using a Gaussian scale mixtures model in wavelet domain 2010 IEEE Intl. Conf. on Imag. Proc. 3365-68

[57] Wakin MB, Laska JN, Duarte MF, Baron D, Sarvotham S, Takhar D, Kelly KF and Baraniuk RG 2006 An architecture for compressive imaging 2006 IEEE Intl. Conf. on Imag. Proc. 1273-76
[58] Tan J, Ma Y and Baron D 2015 Compressive imaging via approximate message passing with image denoising IEEE Transactions on Signal Processing 63.8 2085-92

[59] Xie J, Xu L and Chen E 2012 Image denoising and inpainting with deep neural networks Advances in neural info. proc. Systems 341-49

[60] Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L 2009 Imagenet: A large-scale hierarchical image database 2009 IEEE conf. on computer vision and pattern recognition 248-55

[61] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv preprint arXiv:1409.1556

[62] Liu D, Wang Z, Wen B, Yang J, Han W and Huang TS 2016 Robust single image super-resolution via deep networks with sparse prior IEEE Transactions on Img. Proc. 25.7 3194-207

[63] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S and Darrell T 2014 Caffe: Convolutional architecture for fast feature embedding Proceedings of the 22nd ACM international conference on Multimedia 675-8