A Deep Learning Framework for Compressed Learning and Signal Reconstruction

Vinh Nguyen Xuan
Center for Sensorsystems
University of Siegen
Siegen, Germany
Email:vinh.nguyen@zess.uni-siegen.de

Otmar Loffeld
Center for Sensorsystems
University of Siegen
Siegen, Germany
Email: loffeld@zess.uni-siegen.de

Abstract—**Compressed Learning (CL) is a machine learning framework for a direct inference from a small number of compressive measurements without a perfect time-consuming signal reconstruction. However, in many computer vision and signal processing applications, e.g. image compression, besides the inference task, the signal reconstruction plays an important role. This paper aims to propose a new deep neural network (DNN) framework for performing simultaneously both compressed learning and signal reconstruction in two separate phases. This approach brings more advantages upon training and computation time with some benefits of compressed learning point of view. Through experiments on MNIST and CIFAR-10 datasets, our proposed model is proven to bring higher classification and image recovery performance.**

Index Terms—**compressed sensing, deep learning, classification, compressed learning, mutual coherence.**

I. INTRODUCTION

Recent compressed sensing (CS) studies have brought some advances in many applications of computational imaging and signal processing, e.g. image compression [1], lensless imaging [2], where the data $x \in \mathbb{R}^N$ can be represented as a sparse vector $c \in \mathbb{R}^L$ in a high dimensional space $x = \Psi c$ where Ψ is spasifying dictionary matrix. CS technique offers a framework for reconstructing this sparse vector that relies on linear dimensionality reduction [3], [4]. Instead of acquiring the data x directly as the traditional measuring method, it carries out $M < N$ linear compressive measurements

$$y = \Phi x = \Phi \Psi c \tag{1}$$

using a sensing matrix $\Phi \in \mathbb{R}^{M \times N}$ where $y \in \mathbb{R}^M$ is referred as the measurement vector. The ratio $R = M/N$ is referred as sensing rate of a CS framework. Finding the sparsest solution c of Eq 1 leads to the recovery of the required data x. Currently, compressive imagers, e.g. single-pixel cameras (SPC) [5], are practical systems producing compressive measurements for extracting visual information about a scene. Lensless imaging [2] is a typical application using such coded measurements for imaging at different resolutions or 3D imaging. Apparently, these compressive measurements become more popular in many practical computer vision or signal processing applications.

Compressed learning (CL) [6] aims to construct a machine learning model for inference tasks from compressive measurements without a perfect signal reconstruction. In principle, it is beneficial in both points of view of compressed sensing (CS) with reduced computation cost and machine learning (ML) with reduced training and inference time. Many publication have successfully used Support Vector Machine (SVM) [6] or further DNN [7], [8] for the realization of CL approach. Similarly, this paper proposes a new DNN model for a compressed classification learning. However, it comprises a less amount of hidden layers and hence potentially acquires shorter training or classification time. Furthermore, besides inference tasks, our proposed DNN is aimed at higher image recovery performance. Our idea approaches to many practical applications, e.g. image compression [1], which require both inference and image reconstruction tasks simultaneously from compressive measurements. A simple solution for this approach is to reconstruct the images before using them into the learning systems. However, this process consumes much abundant computation time for image recovery if we are only interested in inference results at one time point. In such scenarios, our model aims to perform these two tasks separately. The contributions of this paper is designing a DNN architecture for higher performance of both classification and image reconstruction based on compressive measurements. To the best of our knowledge, there has been no such a framework reported yet.

The outline of this paper is as follows: Section II mentions some state-of-art studies regarding CS and CL frameworks. Section III-A proposes a DNN with smaller computational cost. Besides, Section III-B modifies the loss function of the proposed DNN for optimizing the sensing matrix. Subsequently, Section IV provides experimental results to show the advantages of two proposed approaches for higher classification accuracy and image reconstruction performance. Section V concludes the paper.

II. RELATED WORK

A. Compressed sensing

For a fast processing, solving the sparse problem respect to Eq 1 is a complex task and hence relaxed to the following

l_1-minimization program:

$$\hat{\boldsymbol{c}} = arg \min_{\boldsymbol{c}} ||\boldsymbol{c}||_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi\Psi c} \qquad (2)$$

There are some restricting conditions, e.g. null space property (NSP), restricted isometry property (RIP), coherence property[3], [4], on the sensing and dictionary matrices $\boldsymbol{A} = \boldsymbol{\Phi\Psi}$ to guarantee for the coincidence between l_0-norm and l_1-norm solutions. In this paper, we use mutual coherence

$$\mu(\boldsymbol{A}) = \max_{1 \le i \ne j \le N} \frac{\boldsymbol{a}_i \boldsymbol{a}_j}{||\boldsymbol{a}_i||_2 ||\boldsymbol{a}_j||_2} \qquad (3)$$

as the maximum value of inner products between two different columns of matrix $\boldsymbol{A}$, for designing sensing matrix $\boldsymbol{\Phi}$. The goal of this design work is to minimize this mutual coherence for a higher signal reconstruction performance of CS framework [9]. Notably, the set of all above inner products forms Gram matrix of matrix $\boldsymbol{A}$. Additionally, there is a lower bound on the number of compressive measurements [10]:

$$M \ge CKL\mu^2(\boldsymbol{A}) \log(L/\gamma) \qquad (4)$$

for fixed values of $\gamma < 1$, C, C', to guarantee for an exact K-sparse signal recovery using l_1-minimization algorithm with a probability at least $1 - \gamma$.

B. Compressed learning

Calderbank et al. [6] provided the theoretical evidence of CL that direct inference from compressive measurements is able to acquire a high classification accuracy under certain conditions. In particular, their studies proceeded an analysis of a linear Support Vector Machine (SVM) classifier operating in the CS domain. For a more complicated classification model, a deep learning (DL) approach to CL was introduced in [7] for a higher classification accuracy. Through a linear transformation using the transpose version of Gaussian random sensing matrix, compressive measurements are inversely converted to one vector $\boldsymbol{z} = \boldsymbol{\Phi}^T \boldsymbol{y}$ with the same dimensionality as the original signal. Afterwards, the authors used a Convolutional Neural Network (CNN) architecture training on this converted vector for a classification task. Their experimental results on MNIST and ImageNet datasets indicated that the performance of this CNN in the CS domain with a high sensing ratio approximately exceeds the performance of a CNN operating in the image domain. However, if this ratio becomes extremely small, then their learning model acquires a high classification error level. Recently, Adler et al. [8] has proved that these unexpected results come from the usage of Gaussian random matrix and its transpose version. They proposed an end-to-end deep neural network (DNN) as described in Fig 1 with two additional fully-connected (FC) layers

$$\boldsymbol{v} = \max(0, \tilde{\boldsymbol{\Phi}}\boldsymbol{x}) \quad \text{and} \quad \boldsymbol{v} = \max(0, \tilde{\boldsymbol{\Xi}}\boldsymbol{x}) \qquad (5)$$

where $\tilde{\boldsymbol{\Phi}} \in \mathbb{R}^{M \times N}$ is the optimized sensing matrix and $\tilde{\boldsymbol{\Xi}} \in \mathbb{R}^{N \times M}$ is the optimized inverse transformation matrix, for impressively higher classification accuracy on the MNIST dataset of handwritten digits. Typically, as sensing ratio is as
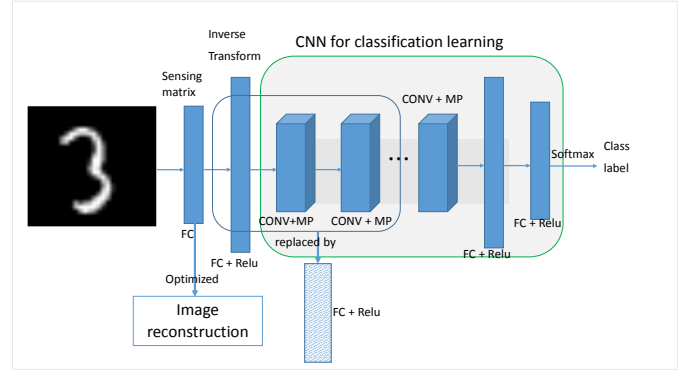

Fig. 1. Our modified CL framework for classification task.

small as $R = 0.01$, the classification error of their model is only 6.46%. Nevertheless, there is currently no advance in the ML point of view since the input to these CNN models has the same dimensionality as the original data. Furthermore, these models can cause obstacles in ML threatening the computation time of the classification task with the curse of data dimensionality.

III. THE PROPOSED APPROACH

A. The proposed DNN architecture

Fig 1 describes the structure of a CNN for image classification comprising many convolutional layers (CONV), max-pooling layers (MP) followed by FC layers [11]. In this paper, we use a Relu activation function for CONV and FC layers, a linear activation function for MP layers and a softmax activation function for the output layer. The output of the softmax function represents for the probabilities with that an image belongs to classes.

Adler et al. [8] added two FC layers at the beginning of a CNN as the compressive sensing stage $\tilde{\boldsymbol{\Phi}}$ and the inverse transformation stage $\tilde{\boldsymbol{\Xi}}$ (see Eq 5). Firstly, the non-linear operator in the first FC layer or sensing matrix stage can lose some information of compressive measurements. The number of compressive measurements is limited but still reduced through this layer and hence this non-linear operator degrades the performance of classification. Our CNN model eliminates this non-linear operator from the first FC layer.

Secondly, the inverse transformation stage in the model of Adler et al. plays a role as image recovery. Nevertheless, in mathematical principle, this operator only performs a linear combination of linear measurements followed by a Relu operator. The following CONV layers also performs a partly linear combination of the output of the inverse transformation stage. From these reasons, we propose to simplify this model by combining the second FC layer and the first following (possibly one or more) CONV layers followed by MP operators into one multi-dimensional FC layer, as described in Fig 1, for extracting directly the features from compressive measurements. For two-dimensional (2D) compressive measurements, e.g.

gray-scale images, a multi-dimensional FC layer is equivalent to a conventional FC layer performing a matrix multiplication operation. Whereas, for multi-dimensional inputs, e.g. RGB color images, it performs a tensordot (TD) operation of Tensorflow framework [12] which sums the products of elements in all channels. Additionally, we add a dropout (DP) layer to the output of this multi-dimensional FC layer to reduce over-fitting problem. In general, our proposed model acquires the reduced training and classification time, compared to the state-of-art Adler's model.

B. The proposed loss function

Given z_{out} be the output vector of the proposed CNN. For conventional multi-class image classification, the softmax function on z_{out}

$$h_\theta(z_{out}) = \frac{1}{1 + e^{-\theta^T z_{out}}} \tag{6}$$

is used to define the loss function

$$L(\theta) = -\Big[\sum_{i=1}^m l^{(i)} \log(h_\theta(z_{out}^{(i)})) + (1-l^{(i)}) \log(1 - h_\theta(z_{out}^{(i)})) \Big] \tag{7}$$

where m is the number of labeled samples, $l^{(i)}$ is label of training samples. The aim of this CNN framework is to find the parameters θ of all hidden layers for minimizing the above loss function.

$$\theta_{opt} = \arg\min_\theta L(\theta) \tag{8}$$

As mentioned before, the mutual coherence property $\mu(\mathbf{\Phi\Psi})$ plays an important role for improving the sparse signal reconstruction performance in CS frameworks. One wishes to design sensing matrix with a smallest mutual coherence value. As a result, we add a regularization term to the above loss function as follows:

$$\theta_{opt} = \arg\min_\theta L(\theta) + \beta\mu(\tilde{\mathbf{\Phi}}\mathbf{\Psi}) \tag{9}$$

where β is a regularization factor, to improve signal recovery quality. A large value of β represents for a big concern about image reconstruction performance.

IV. EXPERIMENTAL RESULTS

In this section, we carry out experiments on two common datasets, i.e. MNIST and CIFAR-10, to show the effectiveness of our proposed model with higher classification and image reconstruction performance. All networks are trained using Tensorflow [1] [12] on GPU NVIDIA Tesla P100 [2].

A. Image classification performance

We compare our proposed framework with the one of Adler et al. which is to our best knowledge regarded as the current state-of-art CNN model for CL classification. For each dataset, we employ the training phase on the training set using Adam algorithm with an annealing learning rate and the loss function in Eq 8, then evaluate its performance on the testing set. Test, training accuracies and mutual coherence of sensing matrix are obtained after 2000 epochs. For statistical results, we train each CNN five times for each dataset. All achieved results listed in Table I and Table II, are averages of five training models.

1) MNIST: is a dataset containing handwritten digits of size $28 \times 28 = 784$ pixels in grayscale. It is divided into two parts - training and testing sets of 50000 and 10000 images respectively. Each image is labeled with one of 10 classes - digits 0 through 9. The batch size is 50. The learning rate is initialized with 0.0005 and then annealed by a factor of 0.996/epoch. The input to CNNs is a 2D array of size 28×28 without data augmentation, e.g. cropping, rotating, scaling.

For Adler's method, besides two FC layers for sensing matrix and inverse transformation matrix stages, we use the state-of-art CNN architecture for MNIST dataset classification [13], [11] described as follows: 1x784-MFC-784FC-1x28x28-32C3-MP2-64C5-MP3-150FC-DP0.5-10FC. This description represents a net with an input image vector of size 1x784, a FC layer with M hidden units (sensing matrix), a FC layer with L hidden units (inverse transformation matrix), a reshape operator to an image 28x28, a CONV layer with 32 features and 3x3 filters, a MP layer over non overlapping regions of size 2x2, a CONV layer with 64 features and 5x5 filters, a MP layer over non overlapping regions of size 3x3, a FC layer with 150 hidden units, a DP layer with a dropout rate of 50% and a FC output layer with 10 units (one per class). This representation will be used for the short description of next CNNs. For our proposed method, we modified the above model as follows: 1. the first FC layer without Relu operator, 2. the above CNN is simplified to 1x784-MFC-(32x9x9)FC-DP0.2-64C4-MP2-150FC-DP0.5-10FC. Notably, another DP layer with a dropout rate of 20% is added after multi-dimensional (32x9x9)FC layer for reducing overfitting.

We trained two above networks at five different sensing rates of R = 0.25, 0.1, 0.05 and 0.01 or equivalently the measurement number of M= 196, 78, 39 and 8. The test and training accuracies for all scenarios are summarized in Table I. These achieved results apparently indicate that our proposed model brings higher test accuracies for all cases of different sensing rates. As the sensing rate much decreases from 0.25 to 0.05, the classification performance of our proposed CNN architecture becomes slightly lower. Especially, the digit classification with approximately 4% test error can be obtained with a small sensing rate of only 0.01 or only 8 compressive measurements.

2) CIFAR-10: is a dataset consisting of 60000 32x32 color images (RGB channels) or 3x1024 input tensors in 10 classes, with 6000 images per class. There are 50000 training

Table I. Classification accuracy (%) and mutual coherence for the MNIST dataset vs. sensing rate R = M/N. For CNN on the original dataset, the state-of-art performance is 99.52% without data augmentation

Sensing rate	M	CNN model	Test Accuracy	Train Accuracy	Mutual Coherence	Training time/epoch
0.25	196	Adler et al.	$98.48 \pm 0.04\%$	100%	0.79	$\approx$4.67s/epoch
		Our proposed CNN with Eq 8	**98.68 ± 0.06%**	99.99%	0.863	$\approx$4.6s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$98.57 \pm 0.02\%$	99.98%	**0.087**	$\approx$4.8s/epoch
0.1	78	Adler et al.	$98.29 \pm 0.02\%$	99.99%	0.868	$\approx$4.67s/epoch
		Our proposed CNN with Eq 8	**98.60 ± 0.04%**	99.95%	0.866	$\approx$4.6s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$98.06 \pm 0.07\%$	99.87%	**0.18**	$\approx$4.8s/epoch
0.05	39	Adler et al.	$98.09 \pm 0.09\%$	99.99%	0.907	$\approx$4.67s/epoch
		Our proposed CNN with Eq 8	**98.57 ± 0.03%**	99.98%	0.922	$\approx$4.6s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$97.33 \pm 0.11\%$	99.79%	**0.283**	$\approx$4.8s/epoch
0.01	8	Adler et al.	$95.18 \pm 0.04\%$	97.19%	0.9719	$\approx$4.5s/epoch
		Our proposed CNN with Eq 8	**95.87 ± 0.1%**	97.28%	0.9728	$\approx$4.5s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$91.32 \pm 0.3\%$	93.34%	**0.78**	$\approx$4.8s/epoch

Table II. Classification accuracy (%) and mutual coherence for the CIFAR-10 dataset vs. sensing rate R = M/N. For CNN on the original dataset, the performance of the chosen network is approximately 78% without data augmentation

Sensing rate	M	CNN model	Test Accuracy	Train Accuracy	Mutual Coherence	Training time
0.25	256	Adler et al.	$57.12 \pm 0.40\%$	100%	0.496	$\approx$10.7s/epoch
		Our proposed CNN with Eq 8	**60.026 ± 0.4%**	99.97%	0.568	$\approx$3.5s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$58.468 \pm 0.23\%$	99.97%	**0.082**	$\approx$3.7s/epoch
0.125	128	Adler et al.	$56.04 \pm 0.52\%$	100%	0.551	$\approx$10.7s/epoch
		Our proposed CNN with Eq 8	**59.684 ± 0.357%**	99.96%	0.611	$\approx$3.5s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$57.31 \pm 0.35\%$	99.94%	**0.132**	$\approx$3.7s/epoch
0.0625	64	Adler et al.	$55.34 \pm 0.73\%$	93.75%	0.75	$\approx$10.7s/epoch
		Our proposed CNN with Eq 8	**59.568 ± 0.3%**	99.97%	0.699	$\approx$3.5s/epoch
		Our proposed CNN with Eq 9 - $\beta = 100$	$55.96 \pm 0.29\%$	99.88%	**0.21**	$\approx$3.7s/epoch

images and 10000 test images. For this dataset, we construct a variant of AlexNet [14] with 5 CONV layers followed by 3 FC layers. In particular, a CNN comprising 1x3x1024-MFC-1024FC-1x3x32x32-64C5-BN-MP2-64C5-BN-MP2-128C3-128C3-128C3-BN-MP2-384FC-192FC-10FC is used for Adler's method. Notably, BN represents for local response normalization operator in Tensorflow. The batch size is 128, the learning rate is initialized with 0.0002 and then annealed by a factor of 0.996/epoch. The input to the network is a 3D array of RGB image without data augmentation.

For our proposed method, we use a CNN model comprising 1x3x1024-MFC-(32x8x8)TD-DP0.2-128C3-128C3-128C3-BN-MP2-384FC-192FC-10FC. In other words, two CONV layers followed by MP layers for extracting 64 features of 8x8 pixels in the above Adler's model are replaced by a multi-dimensional FC layer with TD operator and (32x8x8) hidden units. Similarly to MNIST model, the first FC layer has no Relu operator and a DP layer with a dropout rate of 20% is added after the muti-dimensional (32x8x8)TD layer for reducing overfitting as well as training time. Through many experiments, we verified that replacing more than two

CONV layers gave the approximate performance, but the above model is the best one.

Similarly, we trained each network five times on CIFAR-10 at three different sensing rates of $R = 0.25$, 0.125 and 0.0625, as described in Table II. The reconstruction of natural images from compressive measurements relying on only one FC layer is apparently inaccurate. Whereas, our proposed modifications not only simplify CNN but also effectively generates main features with a higher accuracy. From this reason, the outperformance of our network on CIFAR-10 in Table II is sufficiently high in comparison to the learning process on MNIST.

B. Image reconstruction performance

We retrained our proposed models for MNIST and CIFAR-10 datasets with different loss function in Eq 9 with a regularization factor of $\beta = 100$. For more simplification, we assume that the verified datasets are sparse in the image domain and hence there exists no sparsifying dictionary matrix Ψ in this paper. According to the comparison between results without and with sensing matrix optimization in Table I and Table II, the models using the proposed loss function in Eq 9
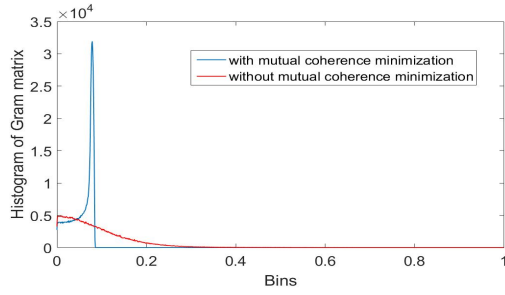
Fig. 2. Histogram of all absolute elements in Gram matrices of two sensing matrices from our proposed CL model without (red line) and with mutual coherence minimization ($\beta = 100$, blue line) for MNIST dataset with $N = 784$, $M = 196$.
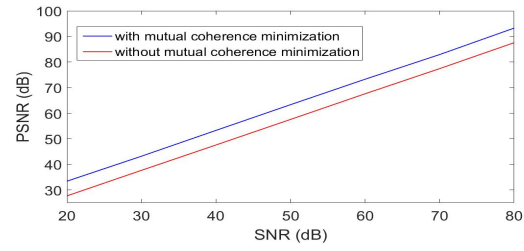

Fig. 3. Average PSNR of Kalman filter with null space method using two sensing matrices from our proposed CL model without (red line) and with mutual coherence minimization ($\beta = 100$, blue line) for the sparsified 500 images ($K \in [15, 35]$) of EMNIST dataset with $N = 784$, $M = 196$.

can generate sensing matrices with lower mutual coherence values as well as preserve approximately high classification accuracies. Tensorflow framework [12] uses automatic differentiation for gradient computation of all functions including the max function in mutual coherence (in Eq 3).

In this part, we carry out a Monte Carlo simulation to compare the image reconstruction capabilities of two our proposed models using different loss functions in Eq 8 and Eq 9 respectively. In particular, two sensing matrices are picked from the final training models for MNIST dataset with M=196. Firstly, the histograms of their Gram matrices are demonstrated in Fig 2 with the red and blue lines respectively. The sensing matrix without optimization acquires a mutual coherence value of 0.857 and the one with optimization acquires the considerably lower value of 0.086.

For a diverse evaluation, a set of 500 images are drawn from the extended MNIST dataset, which has not been trained yet in our classification model. For each image, only K largest-magnitude pixels are kept for the generation of K-sparse signal. The value of K is randomly chosen from [15,35] so that the K-sparse signals can be reconstructed exactly, according to Eq 4, through 196 compressive measurements using our l_1-minimization method, i.e. Kalman filter with null space [15]. We generate noisy compressive measurements based on two sensing matrices and then add Gaussian noises to them with different SNR levels from 20 dB to 80 dB. The metric of PSNR values between the reconstructed and originally sparsified images are demonstrated in Fig 3. A comparison between the achieved results in Fig 3 indicates that the optimized sensing matrix from our proposed model using the loss function Eq 9 (blue line) with a lower mutual coherence brings higher PSNRs or higher image recovery quality in all cases of different noise variances.

V. CONCLUSION

Our paper proposed a new DNN framework for performing both classification and image reconstruction separately from compressive measurements. This combination between CL and CS is promising for many practical applications where the size of original data is extremely large. According to our research, the data can be stored under an appropriate

format with a high compression rate. Our future work aims to use this framework for natural images through exploring the appropriate sparsifying dictionary matrix.

REFERENCES

[1] G. Li, X. Li, S. Li, H. Bai, Q. Jiang, and X. He, "Designing robust sensing matrix for image compression," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5389–5400, Dec 2015.

[2] G. Satat, M. Tancik, and R. Raskar, "Lensless imaging with compressive ultrafast sensing," *CoRR*, vol. abs/1610.05834, 2016.

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

[4] H. Rauhut, "Compressive sensing and structured random matrices," *Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter, Radon Series Comp. Appl. Math.*, 2011.

[5] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, March 2008.

[6] R. Calderbank and S. Jafarpour, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," 2009.

[7] S. Lohit, K. Kulkarni, and P. Turaga, "Direct inference on compressive measurements using convolutional neural networks," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1913–1917.

[8] A. Adler, M. Elad, and M. Zibulevsky, "Compressed learning: A deep neural network approach," *CoRR*, vol. abs/1610.09615, 2016.

[9] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, 2011.

[10] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," 2006.

[11] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012.

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, ..., Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[13] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1058–1066.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[15] O. Loffeld, A. Seel, M. H. Conde, and L. Wang, "A nullspace based l1 minimizing kalman filter approach to sparse cs reconstruction," in *Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar*, June 2016, pp. 1–5.