RECONSTRUCTION-FREE DEEP CONVOLUTIONAL NEURAL NETWORKS FOR PARTIALLY OBSERVED IMAGES

Arun Nair^{1*}, *Luoluo Liu*^{1*}, *Akshay Rangamani*^{1}, *Peter Chin*^{2,1}, *Muyinatu A. Lediju Bell*^{1}, *Trac D. Tran*^{1}

^{1}Dept. of ECE, Johns Hopkins University, Baltimore, MD,21218
^{2} Dept. of CS, Boston University, Cambridge, MA, 02215

ABSTRACT

Conventional image discrimination tasks are performed on fully observed images. In challenging real imaging scenarios, where sensing systems are energy demanding or need to operate with limited bandwidth and exposure-time budgets, or defective pixels, where the data collected often suffers from missing information, and this makes the task extremely hard. In this paper, we leverage Convolutional Neural Networks (CNNs) to extract information from partially observed images. While pre-trained CNNs fail significantly even with such a small percentage of the input missing, our proposed framework demonstrates the ability to overcome it after training on fully-observed and partially-observed images at a few observation ratios. We demonstrate that our method is indeed *reconstruction-free*, *retraining-free* and *generalizable* to previously untrained-on observation ratios and it remains effective in two different visual tasks – image classification and object detection. Our framework performs well even for test images with only 10% of pixels available and outperforms the reconstruct-then-classify pipeline in these challenging scenarios for small observation fractions.

Index Terms**—** Deep Learning, Convolutional Neural Networks, Compressed Measurements, Image Classification, Object Detection

1. INTRODUCTION

In the era of big data, Convolutional Neural Networks (CNNs) today have become one of the most powerful methods for visual tasks. They have achieved success in problems such as image classification [1, 2, 3] and object detection [4]. Key to the success is the ability to learn rich feature hierarchies [5] , with low-level features like edges and colors learned at lower layers, which are combined together in the higher layers to detect complex shapes and patterns in a fully-differentiable end-to-end framework.

Traditionally, CNNs are trained on fully observed images. However, in challenging real imaging scenarios sensing systems are often energy demanding or need to operate with limited bandwidth and exposure-time budgets like in [6]. Or exposed to high level noise in communication channels, the collected data suffers from severe missing information. A decision framework based on inference from partially observed data is needed for more energy-efficient hardware system design and robust performance for noisy environment.

Compressed sensing (CS) theory [7, 8] guarantees the exact recovery of signals at sub-Nyquist sampling rates with sparsity assumptions. It provides theoretical foundations for designing CS hardware systems and reconstructing signals from compressed measurements [9]. Consequently, efficient systems have been developed for generating compressed measurements for demanding applications include underwater sensing [10], drone-based imaging [11, 6], satellite imaging [12], high-speed imaging [13, 14] and magnetic resonance imaging [15].

However, CS entails the need for slow iterative algorithms to perform recovery and/or inference on the sampled data. In addition, CS methods do not scale to the sizes of training data sets that the modern data deluge affords. To compensate for these disadvantages, CNN-based approaches have been considered to deal with compressed measurements.

Reconstruction algorithms such as ReconNet [16], DeepInverse [17] and classification algorithm [18] have shown promising performances. However, in those cases the sampling/sensing operators are assumed to be known, fixed a-priori and tied to the particular neural network being trained. In addition, they hinge on the availability of very specialized hardware like a Digital micro-mirror (DMD) array in order to allow efficient sensing implementations.

In this paper, we attempt to overcome these difficulties by directly performing classification on partially observed measurements. The test images here are various fraction of the image scene's pixels chosen randomly, which model measurements from CS hardware and partial observations due to noise. We demonstrate the sensitivity of pre-trained convolutional neural networks, which fail miserably with only a small portion of missing pixels and propose a framework to overcome it through making the network learn from fully-observed and compressed images in the training procedure. We also empirically verify that our approach generalizes to unseen observation ratios without retraining the network.

This work has been partially supported by NSF Grant CCF-1422995.

*These authors contributed equally to this work.

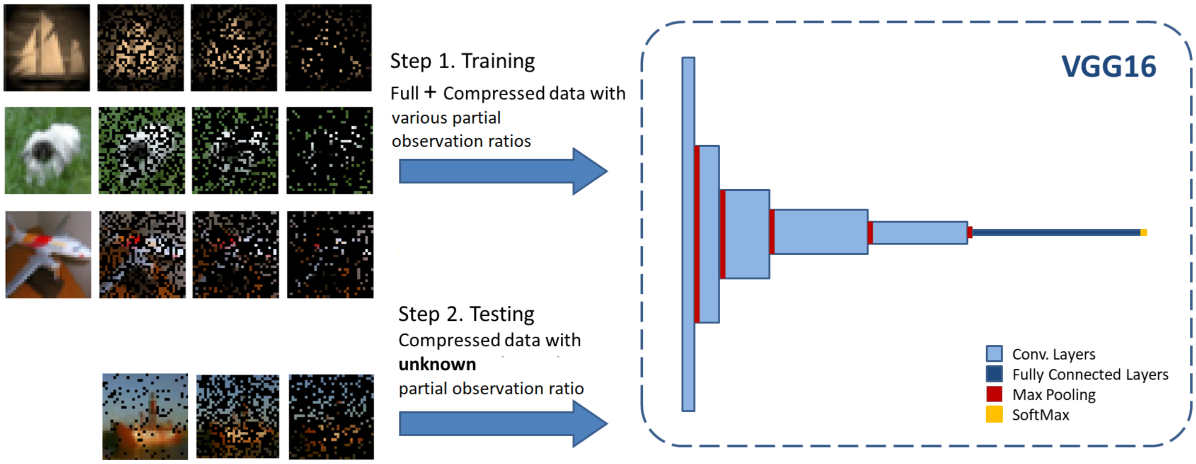

Step 1. Training
Full + Compressed data with various partial observation ratios
Step 2. Testing
Compressed data with **unknown** partial observation ratio
VGG16
Conv. Layers
Fully Connected Layers
Max Pooling
SoftMax

Fig. 1: Overview of our framework in image classification. During training, we input full images and images with missing pixel ratios of .5, .25, .125 to a VGG16 [2] network. The test data to the network with partial observation ratio randomly generated between $(0, 1]$

Our framework is low-cost, efficient, and hardware friendly. It has several advantages: *(i)* Reconstruction-free in discriminative applications; *(ii)* Robust to changes in the partial observation mask; *(iii)* Retraining-free and generalizable to test data with unseen partial observation ratios; *(iv)* Transfers across visual tasks. *(v)* Efficiently deals with missing and incomplete data as long as the label information is correct.

2. RELATED WORK

2.1. Reconstructing CS measurements via CNNs

The CS measurements $\boldsymbol{y} \in \mathbb{R}^m$ of a signal $\boldsymbol{x} \in \mathbb{R}^n$, are generated using $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$ with a smaller dimension than the signal dimension, where the sensing matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ is a random matrix [7]. Recent work such as ReconNet [16] and DeepInverse [17] propose using CNNs to perform reconstruction from CS measurements. In ReconNet [16], CNNs are then employed to reconstruct the CS measurements of each image block. All reconstructed blocks are then arranged and fed into a denoiser. DeepInverse [17] on the other hand proposes using CNNs to learn the inverse transformation of $\boldsymbol{\Phi}$ to invert CS measurements $\boldsymbol{y}$ to signals $\boldsymbol{x}$.

There are several drawbacks to either employing a reconstruction algorithm before CNNs for partially-observed data or incorporating the reconstruction network into the entire framework. First, reconstruction is power-consuming. Second, the reconstruction network does not generalize well for test images with unseen and various partial observation ratios.

2.2. Classification on CS measurements using CNNs

To the best of our knowledge, the only work that proposes a classification algorithm on CS measurements using CNNs is [18]. Instead of reconstructing images from compressive measurements $\boldsymbol{y}$ before feeding to CNNs, they perform a projection on the measurements $\boldsymbol{\Phi}^T\boldsymbol{y}$, and which is then resized into the original image size. Their framework performs well on MNIST and ImageNet with low measurement rates.

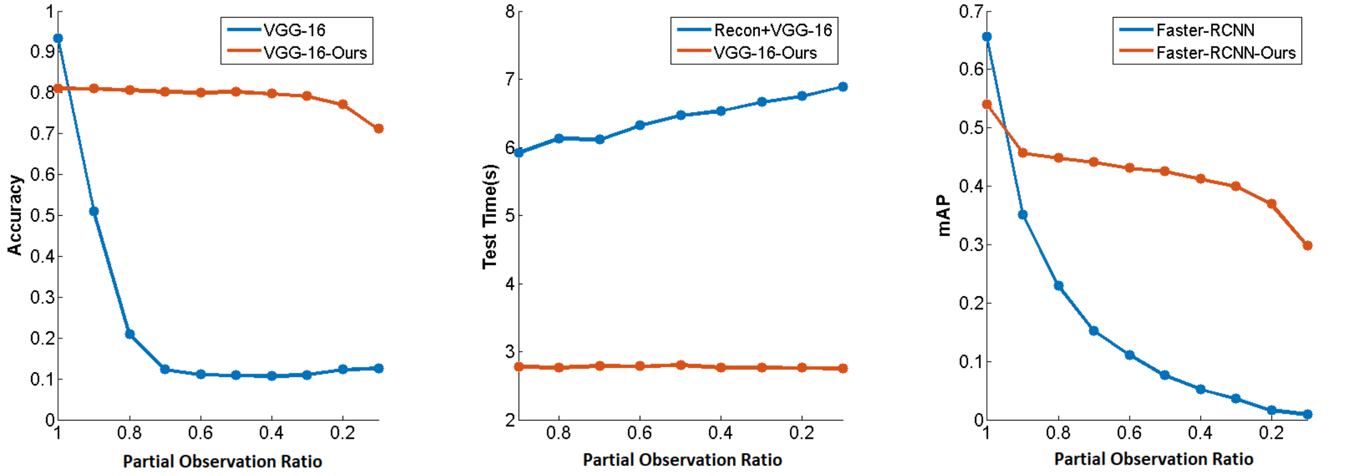
Their work demonstrates the promise of classification directly on CS Measurements using CNNs. However, the huge disadvantage of this framework is that the sensing matrix $\boldsymbol{\Phi}$ is fixed. In several image sensing models, the sensing operation of training data varies each time. Also, it does not generalize to new unseen sampled data without re-training the network.

3. METHOD: EXTRACTING INFORMATION FROM PARTIALLY OBSERVED IMAGES

We propose a framework to extract information from visual data with an unknown fraction of pixels missing using CNNs, without performing reconstruction or re-training the neural network for every possible partial observation ratio.

We first generate partially observed training data corresponding to k ratios between 0 and 1. In this paper, we use the original fully-observed data, along with data observed at three ratios of $0.5, 0.25$ and 0.125. We then train neural network with the enlarged training set. The ratio of the randomly observed pixels in the testing data need not match the ratios used during training, so as the random observation masks. An overview of our framework in image classification is shown in Figure 1.

In the image classification task, the neural network that we use is the VGG-16 [2] network. Our proposed framework is also tested on object detection, in which Faster-RCNN, based on VGG-16 features, is employed.



(a) Averaged classification accuracies for VGG-16 [2] network and our method denoted as VGG-16-Ours

(b) Testing times on data for reconstruction plus VGG-16 network (Recon+VGG-16) and our reconstruction-free method

(c) Mean Average Precision(mAP) for Faster-RCNN and our framework: Faster-RCNN-Ours

Fig. 2: (a) and (c) Classification accuracy and object detection performance for classical CNNs and ours with various partial observation ratios. (b) Testing times for doing reconstruction algorithm vs ours with various observation ratios

Partial Observation Ratios	**1.0**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**	**Random**
VGG-16	**0.93**	0.51	0.21	0.12	0.11	0.11	0.11	0.11	0.12	0.13	0.19
VGG-16-Ours	0.81	0.81	0.81	0.80	0.80	0.80	0.80	**0.79**	**0.77**	**0.71**	**0.76**
Recon+VGG-16	**0.93**	**0.93**	**0.93**	**0.92**	**0.91**	**0.89**	**0.86**	**0.79**	0.65	0.35	-
Recon+VGG-16-Ours	0.81	0.81	0.81	0.81	0.81	0.80	0.80	0.78	0.74	0.62	-

Table 1: Averaged classification accuracies with various partial observationratios for four different methods: *(i)* VGG-16; *(ii)* VGG-16-Ours; *(iii)* Recon+VGG-16: reconstruct first and then use VGG-16; and *(iv)* Recon+VGG-16-Ours: reconstruct first and then use the network trained by our method. "Random" denotes the case that each test datum is randomly partially-observed by a ratio generated from $(0, 1]$ uniformly at random. The two dashes $(-)$ in the last column denote that the experiments are not preformed since the reconstruction method is not robust to unknown partial observation ratios.

4. EXPERIMENTS

In this section, we test with our method on two common visual tasks - image classification and object detection. For image classification, we evaluate our network on the standard CIFAR-10 [19] dataset, while we use the Pascal-VOC 2007 [20] dataset for testing object detection.

4.1. Image Classification

The CIFAR-10 [19] dataset contains 60000 images equally split between 10 object categories, with 50000 images marked as training and 10000 as test. Each image has 32×32 pixels.

We first train the VGG-16 CNN with default parameters from [2] on the dataset with full images, and as we see from Fig. 2a and Table 1, the classification accuracy is 0.93. We call this model 'VGG-16' henceforth. However, testing it on partially observed data, the classification accuracy drops sharply. To remedy this, we retrain the VGG-16 CNN network on full images as well as partially observed images. We used SGD with momentum= 0.9, learning rate= 0.1, learning rate decay= 10^{-6}, batch size= 128 for 250 epochs with data augmentation through random translations, flips and rotations. We noted that only three partial observation ratios of $0.5, 0.25$ and 0.125 were sufficient to ensure robustness to such corruption as displayed in Fig. 2a. These three partial observation ratios were chosen empirically as a trade-off between magnitude of training data required and robustness to random missing. This is interesting, as it suggests the CNN has learned to generalize to randomly missing data. We term the network trained as such 'VGG-16-Ours' for purposes of discussion.

We see from Fig. 2a and Table 1 that as as we miss more data, performance degrades. However, even in the challenging scenario of having available a mere 10% of pixels, the network is achieves a classification accuracy of 0.71.

In order to compare our solution with the standard paradigm of reconstruct-then-classify, we train a set of 9 denoising convolutional autoencoders on the CIFAR-10 dataset, one for each partial observation ratio from $0.1, 0.2, ...0.9$. We then passed test images with each of those ratios being observed through the corresponding autoencoder to reconstruct it, and fed the output to a pre-trained VGG-16 network. We term this experimental pipeline 'Recon+VGG-16' and refer to it as the same. As seen from the results in Table 1, this pipeline outperforms VGG-16-Ours, trained to classify on the partially-observed data directly at high observation ratios. However, for the more challenging cases of $0.3.$, 0.2 and especially for the 0.1 case, VGG-16-Ours performs just as well if not substantially better than the standard reconstruction pipeline. Keeping our end-goal of sensing as little as we can get away with in mind, our results suggest that discrimination should be performed directly on the compressed data.

In addition, we also study the time required for the Recon+VGG-16 processing pipeline versus VGG-16-Ours in Fig. 2b. As both the networks are feed-forward neural networks, they each process the data by repeatedly applying basic arithmetic operations like multiplication, addition and a simple non-linearity on the data, once trained. This means testing performance for both pipelines is quite fast and the room of improvement is small. However, we note that direct classification on the compressed data is accomplished twice as fast (taking only 2.78 seconds averaging across all partial observation ratios) for classifying all 10000 test examples in CIFAR-10 as compared to Recon+VGG-16 (which takes 6.42 seconds averaging across all partial observation ratios). This obvious advantage stems from skipping the unnecessary step of reconstruction, speeding up the imaging as well as classification processes.

In order to better understand the behavior of VGG-16-Ours, we also used it to classify compressed data after reconstruction. Interestingly, as the results in Table 1 confirm, it would appear that the act of training VGG-16-Ours on partially observed data as well (which we term 'Recon+VGG-16-Ours' in the table) has made it more robust to perturbations in the input space, leading to a much higher classification performance in the challenging 0.1 observation ratio case (obtaining 0.62 classification accuracy) versus just passing the partially-observed data through the convolutional autoencoder and classifying it using a VGG-16 network trained only on fully-sampled data (yielding just 0.35 classification accuracy). The reconstructed data from the autoencoder loses a lot of high frequency information. This experiment suggests that including data with various observatio ratios has the added bonus of making a neural network robust to blur.

For the last column in Table 1, we randomly take a fraction of pixels in each test image by an unknown fraction $s_i \in (0, 1]$, and then passed the test images through VGG-16 and VGG-16-Ours. As expected, VGG-16-Ours obtained a much higher classification accuracy (a respectable 0.76, close to the average of the accuracies on the different partial observation ratios previously tested) on the test data than VGG-16 (0.19). We did not run Recon+VGG-16 and Recon+VGG-16-Ours experiments on this test set as the partial observation ratio was not constrained to match with the 9 partial observation ratios for which we had trained the autoencoders.

4.2. Object Detection

The Pascal VOC 2007 detection dataset [20] contains images corresponding to 20 different object categories as part of various natural scenes, with close to 5000 images provided for training and cross-validation, with approximately another 5000 provided for testing.

Aiming to understand if the phenomenon of CNNs learning to handle arbitrary partial observation ratios extends to tasks besides object classification, we train a Faster RCNN network on the Pascal VOC 2007 dataset for object detection. We use the mean average precision (mAP) as our evalutaion metric. Details for the measure are contained in the original paper [4]. Similar to Section 4.1, we initially train the network purely on fully-observed images. We term this trained network 'Faster-RCNN'. Then, we train another model with the same Faster-RCNN architecture on the fully-observed images as well as partially-observed images at observation ratios of $0.5, 0.25$ and 0.125. We then tested both networks on object detection from partially-observed test images of various unseen partial observation ratios.

The results of the experiments are displayed in Fig. 2c. We note the trends here mirror those of the classification case, with Faster-RCNN's performance dropping quickly in the presence of random missing , while the performance of Faster-RCNN-Ours remains much more stable. In addition, we again observe that Faster-RCNN-Ours generalizes to unseen partial observation ratios here as well.

5. CONCLUSION

This paper presents an efficient, reconstruction-free training paradigm to extract information from sparsely-sensed images, overcoming the sensitivity that CNNs naturally have to such input mismatches. Moreover, the proposed method generalizes to different, unseen, arbitrary partial observation ratios without retraining. Our method outperforms the pre-trained CNNs and reconstruction-first-classify-later technique in challenging cases with small observation ratios.

Future work includes developing a neuron visualization tool to better understand the differences in neurons learned by such a network when compared to the ones in traditionally trained CNNs. Investigation of the role of missing information may play in making a network more robust to adversarial interference is another interesting open question. Finally, extending our framework to handle missing/incomplete or partially corrupted data and sensor failure is a possible future research direction.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[6] Jie Zhang, Tao Xiong, Trac Tran, Sang Chin, and Ralph Etienne-Cummings, "Compact all-cmos spatiotemporal compressive sensing video camera with pixel-wise coded exposure," *Optics Express*, vol. 24, no. 8, pp. 9013–9024, 2016.

[7] Emmanuel J Candes and Terence Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

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

[9] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[10] Fatemeh Fazel, Maryam Fazel, and Milica Stojanovic, "Random access compressed sensing for energy-efficient underwater sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1660–1670, 2011.

[11] Karan Shetti and Asha Vijayakumar, "Evaluation of compressive sensing encoding on ar drone," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 204–207.

[12] Julien Michel, Gwendoline Blanchet, Julien Malik, and Rosario Ruiloba, "Compressed sensing for earth observation with high resolution satellite imagery," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*. IEEE, 2012, pp. 412–415.

[13] Toshiki Sonoda, Hajime Nagahara, Kenta Endo, Yukinobu Sugiyama, and Rin-ichiro Taniguchi, "High-speed imaging using cmos image sensor with quasi pixel-wise exposure," in *Computational Photography (ICCP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–11.

[14] Dikpal Reddy, Ashok Veeraraghavan, and Rama Chellappa, "P2c2: Programmable pixel compressive camera for high speed imaging," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 329–336.

[15] Michael Lustig, David Donoho, and John M Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[16] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 449–458.

[17] Ali Mousavi and Richard G Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2272–2276.

[18] Suhas Lohit, Kuldeep Kulkarni, and Pavan Turaga, "Direct inference on compressive measurements using convolutional neural networks," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1913–1917.

[19] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge 2007 (VOC2007) Results," .