

ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements

Kuldeep Kulkarni^{1,2}, Suhas Lohit¹, Pavan Turaga^{1,2}, Ronan Kerviche³, and Amit Ashok³

¹School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ

²School of Arts, Media and Engineering, Arizona State University, Tempe, AZ

³College of Optical Sciences, University of Arizona, Tucson, AZ

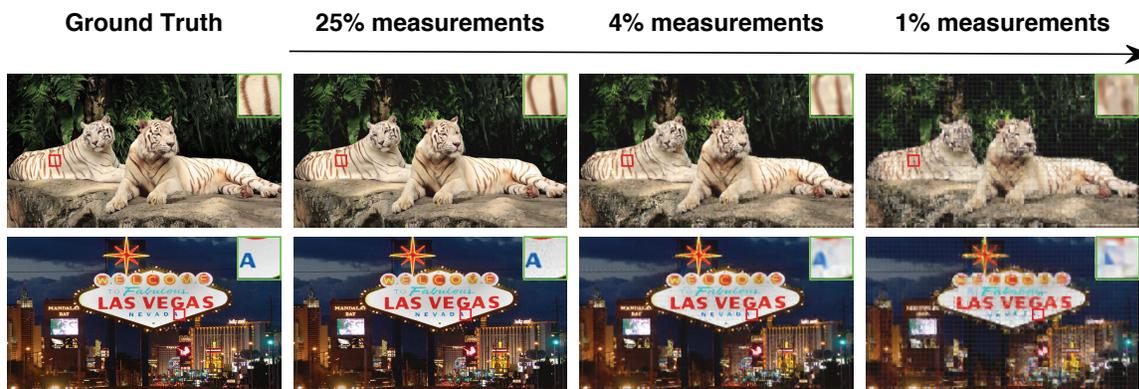


Figure 1: Given the block-wise compressively sensed (CS) measurements, our **non-iterative** algorithm is capable of high quality reconstructions. Notice how fine structures like tiger stripes or letter ‘A’ are recovered from only 4% measurements. Despite the expected degradation at measurement rate of 1%, the reconstructions retain rich semantic content in the image. For example, one can easily see that there are two tigers resting on rocks, although the stripes are blurry. This clearly points us to the possibility of CS based imaging becoming a resource-efficient solution in applications, where the final goal is high-level image understanding rather than exact reconstruction.

Abstract

The goal of this paper is to present a non-iterative and more importantly an extremely fast algorithm to reconstruct images from compressively sensed (CS) random measurements. To this end, we propose a novel convolutional neural network (CNN) architecture which takes in CS measurements of an image as input and outputs an intermediate reconstruction. We call this network, ReconNet. The intermediate reconstruction is fed into an off-the-shelf denoiser to obtain the final reconstructed image. On a standard dataset of images we show significant improvements in reconstruction results (both in terms of PSNR and time complexity) over state-of-the-art iterative CS reconstruction algorithms at various measurement rates. Further, through qualitative experiments on real data collected using our block single pixel camera (SPC), we show that our network is highly robust to sensor noise and can recover visually better quality images than competitive algorithms at extremely low sensing rates of 0.1 and 0.04. To demonstrate that our algorithm

can recover semantically informative images even at a low measurement rate of 0.01, we present a very robust proof of concept real-time visual tracking application.

1. Introduction

The easy availability of vast amounts of image data and the ever increasing computational power has triggered the resurgence of convolutional neural networks (CNNs) in the past three years and consolidated their position as one of the most powerful machineries in computer vision. Researchers have shown CNNs to break records in the two broad categories of long-standing vision tasks, namely: 1) high-level inference tasks such as image classification, object detection, scene recognition, fine-grained categorization and pose estimation [19, 13, 37, 35, 36] and 2) pixel-wise output tasks like semantic segmentation, depth mapping, surface normal estimation, image super resolution and dense optical flow estimation [21, 11, 32, 6, 31]. However, the benefits of CNNs have not been explored for one such important task belonging to the latter category, namely reconstruction of

images from compressively sensed measurements. In this paper we adapt CNNs to develop an algorithm to recover images from block CS measurements.

Motivation: The advances in compressive sensing theory [8, 3, 4] (for the benefit of the readers, a brief background on CS is provided later in the section) has led to the development of many novel imaging devices [23, 27]. The current CS imaging systems, such as the commercially available short-wave infrared single pixel camera, from Inview Technology Corporation, provide the luxury of reduced and fast acquisition of the image by taking only a small number random projections of the scene, thus enabling compression at the sensing level itself. Such characteristics of the acquisition system are highly sought-after in a) resource-constrained environments like UAVs where generally, computationally expensive methods are employed as a post-acquisition step to compress the fully acquired images, and b) applications such as Magnetic Resonance Imaging (MRI) [22] where traditional imaging methods are very slow. As an undesirable consequence, the computational load is now transferred to the decoding algorithm which reconstructs the image from the CS measurements or the random projections.

Over the past decade, a plethora of reconstruction algorithms [2, 10, 26, 1, 20, 18, 34, 28, 24, 7] have been proposed. However, almost all of them are plagued by a number of similar drawbacks. Firstly, current approaches solve an optimization problem to reconstruct the images from the CS measurements. Very often, the iterative nature of the solutions to the optimization problems renders the algorithms computationally expensive with some of them even taking as many as 20 minutes to recover just one image, thus making real-time reconstruction impossible. Secondly, in many resource-constrained applications, one may be interested only in some property of the scene like ‘Where is a particular object in the image?’ or ‘What is the person in the image doing?’, rather than the exact values of all pixels in the image. In such scenarios, there is a great urge to acquire as few measurements as possible, and still be able to recover an image which retains enough information regarding the property of the scene that one is interested in. The current approaches, although slow, are capable of delivering high quality reconstructions at high measurement rates. However, their performance degrades appreciably as measurement rate decreases, yielding reconstructions which are not useful for any image understanding task. Motivated by these, in this paper we present a CS image recovery algorithm which has the desired features of being computationally light as well as being capable of delivering reasonable quality reconstructions useful for image understanding tasks, even at extremely low measurement rates of 0.01. The contributions of our paper are the following:

Contributions: a) We propose a **non-iterative** and ex-

tremely fast reconstruction algorithm for block CS imaging [12]. To the best of our knowledge, there exists no published work which achieves these desirable features. b) We introduce a novel class of CNN architectures called **ReconNet** which takes in CS measurements of an image block as input and outputs the reconstructed image block. Further, the reconstructed image blocks are arranged appropriately and fed into an off-the-shelf denoiser to recover the full image. c) Through experiments on a standard dataset of images, we show that, in terms of mean PSNR of reconstructed images, our algorithm beats the nearest competitor by considerable margins at measurement rates of 0.1 and below. Further, we validate the robustness of ReconNet to arbitrary sensor noise by conducting qualitative experiments on real-data collected using our block SPC. We achieve visually superior quality reconstructions than the traditional CS algorithms. d) We demonstrate that the reconstructions retain rich semantic content even at a low measurement rate of 0.01. To this end, we present a proof of concept real-time application, wherein object tracking is performed on-the-fly as the frames are recovered from the CS measurements.

Background: Compressive Sensing (CS) is a signal acquisition paradigm which provides the ability to sample a signal at sub-Nyquist rates. Unlike traditional sensing methods, in CS, one acquires a small number of random linear measurements, instead of sensing the entire signal, and a reconstruction algorithm is used to recover the original signal from the measurements. Mathematically, the measurements are given by $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathbb{R}^n$ is the signal, $\mathbf{y} \in \mathbb{R}^m$, known as the measurement vector, denotes the set of sensed projections, $\Phi \in \mathbb{R}^{m \times n}$ is called the measurement matrix defined by a set of random patterns, and $\mathbf{e} \in \mathbb{R}^m$ is the measurement noise. Reconstructing \mathbf{x} from \mathbf{y} when $m < n$ is an ill-posed problem. However, CS theory [8, 3] states that the signal \mathbf{x} can be recovered perfectly from a small number of $m = \mathcal{O}(s \log(\frac{n}{s}))$ random linear measurements by solving the optimization problem in Eq. 1, provided the signal is s -sparse in some sparsifying domain, Ψ .

$$\min_{\mathbf{x}} \quad \|\Psi \mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \Phi \mathbf{x}\|_2 \leq \epsilon. \quad (1)$$

Variants of the optimization problem with relaxed sparsity assumption in Eq. 1 have been proposed for the compressible signals as well. However, all such algorithms suffer from drawbacks as already discussed.

2. Related Work

The previous works can be divided into two broad categories, namely CS image reconstruction algorithms and CNNs for per-pixel output tasks.

CS image reconstruction: Several algorithms have been proposed to reconstruct images from CS measurements.

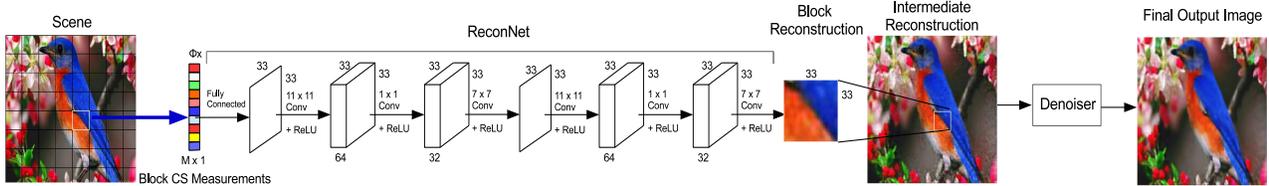


Figure 2: Overview of our non-iterative block CS image recovery algorithm.

The earliest algorithms leveraged the traditional CS theory described above [8, 3, 2] and solved the l_1 -minimization in Eq. 1 with the assumption that the image is sparse in some transform-domain like wavelet, DCT, or gradient. However, such sparsity-based algorithms did not work well, since images, though compressible, are not exactly sparse in the transform domain. This heralded an era of model-based CS recovery methods, wherein more complex image models that go beyond simple sparsity were proposed. Model-based CS recovery methods come in two flavors. In the first, the image model is enforced explicitly [10, 1, 18, 28], wherein in each iteration the image estimate is projected onto the solution set defined by the model. These models, often considered under the class of ‘structured-sparsity’ models, are capable of capturing the higher order dependencies between the wavelet coefficients. However, generally a computationally expensive optimization is solved to obtain the projection. In the second, the algorithms enforce the image model implicitly through a non-local regularization term in the objective function [26, 34, 7]. Recently, a new class of recovery methods called approximate message passing (AMP) algorithms [9, 30, 24] have been proposed, wherein the image estimate is refined in each iteration using an off-the-shelf denoiser. To the best of our knowledge there exists no published work which proposes a non-iterative solution to the CS image recovery problem. However, there has been one concurrent and independent investigation (paper on arXiv.org, but not yet peer-reviewed or published [25]) that presents stacked denoising auto-encoders (SDAs) based non-iterative approach for this problem. Different from this, in this paper we present a convolutional architecture, which has fewer parameters, and is easily scalable to larger block-size at the sensing stage, and also results in better performance than SDAs.

CNNs for per-pixel prediction tasks: Computer vision researchers have applied CNNs to per-pixel output tasks like semantic segmentation [21], depth estimation [11], surface normal estimation [32], image super-resolution [6] and dense optical flow estimation from a single image [31]. However, these tasks differ fundamentally from the one tackled in this paper in that they map a full-blown image to a similar-sized feature output, while in the CS reconstruction problem, one is required to map a small number of random linear measurements of an image to its estimate. Hence, we cannot use any of the standard CNN architectures that have

been proposed so far. Motivated by this, we introduce a novel class of CNN architectures for the CS recovery problem at any arbitrary measurement rate.

3. Overview of Our Algorithm

Unlike most computer vision tasks like recognition or segmentation to which CNNs have been successfully applied, in the CS recovery problem, the images are not inputs but rather outputs or labels which we seek to obtain from the networks. Hence, the typical CNN architectures which can map images to rich hierarchical visual features are not applicable to our problem of interest. How does one design a network architecture for the CS recovery problem? To answer this question, one can seek inspiration from the CNN-based approach for image super-resolution proposed in [6]. Similar to the character of our problem, the outputs in image super-resolution are images, and the inputs – lower-resolution images – are of lower dimension. In [6], initial estimates of the high-resolution images are first obtained from low-resolution input images using bicubic interpolation, and then a 3-layered CNN is trained with the initial estimates as inputs and the ground-truth of the desired outputs as labels. If we were to adapt the same architecture for the CS recovery problem, we will have to first generate the initial estimates of the reconstructions from CS measurements. A straightforward option would be to run one of the several existing CS recovery algorithms and obtain initial estimates. But how many iterations do we need to run to ensure a good initial estimate? Running for too many increases computational load, defeating the very goal of this paper of developing a fast algorithm, but running for too few could lead to extremely poor estimates.

Due to the aforementioned reasons, we relinquish the idea of obtaining initial estimates of the reconstructions, and instead propose a novel class of CNN architectures called ReconNet which can directly map CS measurements to image blocks. The overview of our ReconNet driven algorithm is given in Figure 2. The scene is divided into **non-overlapping** blocks. Each block is reconstructed by feeding in the corresponding CS measurements to ‘ReconNet’. The reconstructed blocks are arranged appropriately to form an intermediate reconstruction of the image, which is input to an off-the-shelf denoiser to remove blocky artifacts and obtain the final output image.

Network architecture: Here, we describe the proposed CNN architecture, ‘ReconNet’ shown as part of Figure 2.

The input to the network is an m -dimensional vector of compressive measurements, denoted by $\Phi\mathbf{x}$, where Φ is the measurement operator of size $m \times n$, m is the number of measurements and \mathbf{x} is the vectorized input image block. In our case, we train networks capable of reconstructing blocks of size 33×33 , hence $n = 1089$. This block size is chosen so as to reduce the network complexity and hence, the training time, while ensuring a good reconstruction quality.

The first layer is a fully connected layer that takes compressive measurements as input and outputs a feature map of size 33×33 . The subsequent layers are all convolutional layers inspired by [6]. Except the final convolutional layers, all the other layers use ReLU following convolution. All feature maps produced by all convolutional layers are of size 33×33 , which is equal to the block size. The first and the fourth convolutional layers use kernels of size 11×11 and generate 64 feature maps each. The second and the fifth convolutional layers use kernels of size 1×1 and generate 32 feature maps each. The third and the last convolutional layer use a 7×7 and generate a single feature map, which, in the case of the last layer, is also the output of the network. We use appropriate zero padding to keep the feature map size constant in all layers.

Denoising the intermediate reconstruction: The intermediate reconstruction (Figure 2) is denoised to remove the artifacts resulting due to block-wise processing. We choose BM3D [5] as the denoiser since it gives a good trade-off between computational complexity and reconstruction quality.

4. Learning the ReconNet

In this section, we discuss in detail training of deep networks for reconstruction of CS measurements. We use the network architecture shown in Figure 2 for all the cases.

Ground truth for training: We use the same set of 91 images as in [6]. We uniformly extract patches of size 33×33 from these images with a stride equal to 14 to form a set of 21760 patches. We retain only the luminance component of the extracted patches (For RGB images, during test time we use the same network to recover the individual channels). These form the labels of our training set. We obtain the corresponding CS measurements of the patches. These form the inputs of our training set. Experiments indicate that this training set is sufficient to obtain very competitive results compared to existing CS reconstruction algorithms, especially at low measurement rates.

Input data for training: To train our networks, we need CS measurements corresponding to each of the extracted patches. To this end, we simulate noiseless CS as follows. For a given measurement rate, we construct a measurement matrix, Φ by first generating a random Gaussian matrix of appropriate size, followed by orthonormalizing its

rows. Then, we apply $\mathbf{y} = \Phi\mathbf{x}$ to obtain the set of CS measurements, where \mathbf{x} is the vectorized version of the luminance component of an image patch. Thus, an input-label pair in the training set can be represented as $(\Phi\mathbf{x}, \mathbf{x})$. We train networks for four different measurement rates (MR) = 0.25, 0.10, 0.04 and 0.01. Since, the total number of pixels per block is $n = 1089$, the number of measurements $n = 272, 109, 43$ and 10 respectively.

Learning algorithm details: All the networks are trained using Caffe [15]. The loss function is the average reconstruction error over all the training image blocks, given by $L(\{W\}) = \frac{1}{T} \sum_i ||f(\mathbf{y}_i, \{W\}) - x_i||^2$, and is minimized by adjusting the weights and biases in the network, $\{W\}$ using backpropagation. T is the total number of image blocks in the training set, x_i is the i^{th} patch and $f(\mathbf{y}_i, \{W\})$ is the network output for i^{th} patch. For gradient descent, we set the batch size to 128 for all the networks. For each measurement rate, we train two networks, one with random Gaussian initialization for the fully connected layer, and one with a deterministic initialization, and choose the network which provides the lower loss on a validation test. For the latter network, the j^{th} weight connecting the i^{th} neuron of the fully connected layer is initialized to be equal to $\Phi_{i,j}^T$. In each case, weights of all convolutional layers are initialized using a random Gaussian with a fixed standard deviation. The learning rate is determined separately for each network using a linear search. All networks are trained on a Nvidia Tesla K40 GPU for about a day each.

5. Experimental Results

In this section, we conduct extensive experiments on both simulated data and real data, and compare the performance of our CS recovery algorithm with state-of-the-art CS image recovery algorithms, both in terms of reconstruction quality and time complexity.

Baselines: We compare our algorithm with three iterative CS image reconstruction algorithms, TVAL3 [20], NLR-CS [7] and D-AMP [24]. We use the code made available by the respective authors on their websites. Parameters for these algorithms, including the number of iterations, are set to the default values. We use BM3D [5] denoiser since it gives a good trade-off between time complexity and reconstruction quality. The code for NLR-CS provided on author’s website is implemented only for random Fourier sampling. The algorithm first computes an initial estimate using a DCT or wavelet based CS recovery algorithm, and then solves an optimization problem to get the final estimate. Hence, obtaining a good estimate is critical to the success of the algorithm. However, using the code provided on the author’s website, we failed to initialize the reconstruction for random Gaussian measurement matrix. Similar observation was reported by [24]. Following the procedure outlined in [24], the initial image estimate for NLR-CS is obtained

Image Name	Algorithm	MR = 0.25		MR = 0.10		MR = 0.04		MR = 0.01	
		w/o BM3D	w/ BM3D						
Barbara	TVAL3 [20]	24.19	24.20	21.88	22.21	18.98	18.98	11.94	11.96
	NLR-CS [7]	28.01	28.00	14.80	14.84	11.08	11.56	5.50	5.86
	D-AMP [24]	25.89	25.96	21.23	21.23	16.37	16.37	5.48	5.48
	SDA [25]	23.19	23.20	22.07	22.39	20.49	20.86	18.59	18.76
	Ours	23.25	23.52	21.89	22.50	20.38	21.02	18.61	19.08
Fingerprint	TVAL3	22.70	22.71	18.69	18.70	16.04	16.05	10.35	10.37
	NLR-CS	23.52	23.52	12.81	12.83	9.66	10.10	4.85	5.18
	D-AMP	25.17	23.87	17.15	16.88	13.82	14.00	4.66	4.73
	SDA	24.28	23.45	20.29	20.31	16.87	16.83	14.83	14.82
	Ours	25.57	25.13	20.75	20.97	16.91	16.96	14.82	14.88
Flintstones	TVAL3	24.05	24.07	18.88	18.92	14.88	14.91	9.75	9.77
	NLR-CS	22.43	22.56	12.18	12.21	8.96	9.29	4.45	4.77
	D-AMP	25.02	24.45	16.94	16.82	12.93	13.09	4.33	4.34
	SDA	20.88	20.21	18.40	18.21	16.19	16.18	13.90	13.95
	Ours	22.45	22.59	18.92	19.18	16.30	16.56	13.96	14.08
Lena	TVAL3	28.67	28.71	24.16	24.18	19.46	19.47	11.87	11.89
	NLR-CS	29.39	29.67	15.30	15.33	11.61	11.99	5.95	6.27
	D-AMP	28.00	27.41	22.51	22.47	16.52	16.86	5.73	5.96
	SDA	25.89	25.70	23.81	24.15	21.18	21.55	17.84	17.95
	Ours	26.54	26.53	23.83	24.47	21.28	21.82	17.87	18.05
Mean PSNR	TVAL3	27.84	27.87	22.84	22.86	18.39	18.40	11.31	11.34
	NLR-CS	28.05	28.19	14.19	14.22	10.58	10.98	5.30	5.62
	D-AMP	28.17	27.67	21.14	21.09	15.49	15.67	5.19	5.23
	SDA	24.72	24.55	22.43	22.68	19.96	20.21	17.29	17.40
	Ours	25.54	25.92	22.68	23.23	19.99	20.44	17.27	17.55

Table 1: PSNR values in dB for 4 of the test images (see supplementary for the remaining) using different algorithms at different measurement rates. At low measurement rates of 0.1, 0.04 and 0.01, our algorithm yields superior quality reconstructions than the traditional iterative CS reconstruction algorithms, TVAL3, NLR-CS, and D-AMP. It is evident that the reconstructions are very stable for our algorithm with a decrease in mean PSNR of only 8.37 dB as the measurement rate decreases from 0.25 to 0.01, while the smallest corresponding dip in mean PSNR for classical reconstruction algorithms is in the case of TVAL3, which is equal to 16.53 dB.

by running D-AMP (with BM3D denoiser) for 8 iterations. Once the initial estimate is obtained, we use the default parameters and obtain the final NLR-CS reconstruction. We also compare with the unpublished concurrent work [25] which presents a SDA based non-iterative approach to recover from block-wise CS measurements. At the time of writing, the authors had not made either the training set or the pre-trained models publicly available. Here, we compare our algorithm with our own implementation of SDA, and show that our algorithm outperforms the SDA. For fair comparison, we denoise the image estimates recovered by baselines as well. The only parameter to be input to the BM3D algorithm is the estimate of the standard Gaussian noise, σ . To estimate σ , we first compute the estimates of the standard Gaussian noise for each block in the intermediate reconstruction given by $\sigma_i = \sqrt{\frac{\|y_i - \Phi x_i\|^2}{m}}$, and then take the median of these estimates.

5.1. Simulated data

For our simulated experiments, we use a standard set of 11 grayscale images, compiled from two sources ^{1,2}. We conduct both noiseless and noisy block-CS image reconstruction experiments at four different measurement rates

0.25, 0.1, 0.04 and 0.01.

Reconstruction from noiseless CS measurements: To simulate noiseless block-wise CS, we first divide the image of interest into non-overlapping blocks of size 33×33 , and then compute CS measurements for each block using the same random Gaussian measurement matrix as was used to generate the training data for the network corresponding to the measurement rate. The PSNR values in dB for both intermediate reconstruction (indicated by w/o BM3D) as well as final denoised versions (indicated by w/ BM3D) for the measurement rates are presented in Table 1. It is clear from the PSNR values that our algorithm outperforms traditional reconstruction algorithms at low measurement rates of 0.1, 0.04 and 0.01. Also, the degradation in performance with lower measurement rates is more graceful.

Further, in Figure 3, we show the final reconstructions of parrot and house images for various algorithms at measurement rate of 0.1. From the reconstructed images, one can notice that our algorithm, as well as SDA are able to retain the finer features of the images while other algorithms fail to do so. NLR-CS and DAMP provide poor quality reconstruction. Even though TVAL3 yields PSNR values comparable to our algorithm, it introduces undesirable artifacts in the reconstructions.

¹<http://dsp.rice.edu/software/DAMP-toolbox>

²http://see.xidian.edu.cn/faculty/wsdong/NLR_Exps.htm

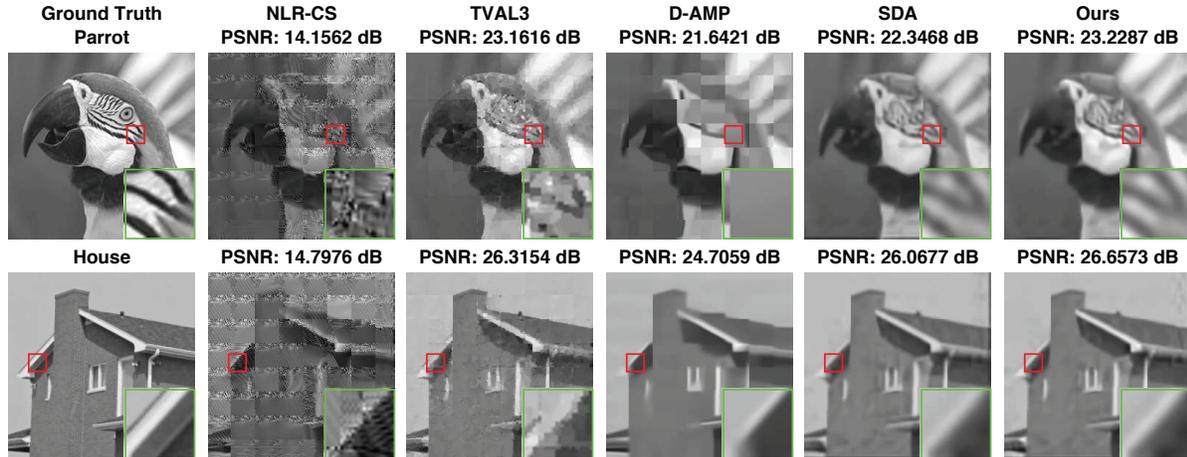


Figure 3: Reconstruction results for parrot and house images from noiseless CS measurements at measurement rate of 0.1. It is evident that our algorithm recovers more visually appealing images than other competitors. Notice how fine structures are recovered by our algorithm.

Algorithm	MR = 0.25	MR = 0.10	MR = 0.04	MR = 0.01
TVAL3	2.943	3.223	3.467	7.790
NLR-CS	314.852	305.703	300.666	314.176
D-AMP	27.764	31.849	34.207	54.643
ReconNet	0.0213	0.0195	0.0192	0.0244
SDA	0.0042	0.0029	0.0025	0.0045

Table 2: Time complexity (in seconds) of various algorithms (without BM3D) for reconstructing a single 256×256 image. By taking only about 0.02 seconds at any given measurement rate, ReconNet can recover images from CS measurements in real-time, and is 3 orders of magnitude faster than traditional reconstruction algorithms.

Time complexity: In addition to competitive reconstruction quality, for our algorithm without the BM3D denoiser, the computation is real-time and is about 3 orders of magnitude faster than traditional reconstruction algorithms. To this end, we compare various algorithms in terms of the time taken to produce the intermediate reconstruction of a 256×256 image from noiseless CS measurements at various measurement rates. For traditional CS algorithms, we use an Intel Xeon E5-1650 CPU to run the implementations provided by the respective authors. For ReconNet and SDA, we use a Nvidia GTX 980 GPU to compute the reconstructions. The average time taken for the all algorithms of interest are given in table 2. Depending on the measurement rate, the time taken for block-wise reconstruction of a 256×256 for our algorithm is about 145 to 390 times faster than TVAL3, 1400 to 2700 times faster than D-AMP, and 15000 times faster than NLR-CS. It is important to note that the speedup achieved by our algorithm is not solely because of the utilization of the GPU. It is mainly because unlike traditional CS algorithms, our algorithm being CNN based relies on much simpler convolution operations, for which very fast implementations exist. More importantly, the non-iterative nature of our algorithm makes it amenable to parallelization. SDA, also a deep-learning based non-iterative algorithm shows significant speedups over traditional algorithms at all measurement rates.

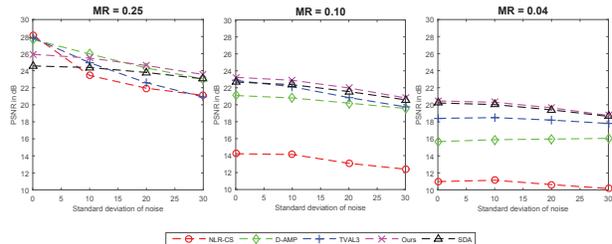


Figure 4: Comparison of different algorithms in terms of mean PSNR (in dB) for the test set in presence of Gaussian noise of different standard deviations at MR = 0.25, 0.10 and 0.04.

Performance in the presence of noise: To demonstrate the robustness of our algorithm to noise, we conduct reconstruction experiments from noisy CS measurements. We perform this experiment at three measurement rates - 0.25, 0.10 and 0.04. We emphasize that for ReconNet and SDA, we **do not** train separate networks for different noise levels but use the same networks as used in the noiseless case. To first obtain the noisy CS measurements, we add standard random Gaussian noise of increasing standard deviation to the noiseless CS measurements of each block. In each case, we test the algorithms at three levels of noise corresponding to $\sigma = 10, 20, 30$, where σ is the standard deviation of the Gaussian noise distribution. The intermediate reconstructions are denoised using BM3D. The mean PSNR for various noise levels for different algorithms at different measurement rates are shown in Figure 4. It can be observed that our algorithm beats all other algorithms at high noise levels. This shows that the method proposed in this paper is extremely robust to all levels of noise.

5.2. Experiments with real data

The previous section demonstrated the superiority of our algorithm over traditional algorithms for simulated CS mea-

measurements. Here, we show that our networks trained on simulated data can be readily applied for real world scenario by reconstructing images from CS measurements obtained from our block SPC. We compare our reconstruction results with other algorithms.

Scalable Optical Compressive Imager Testbed: We implement a scalable optical compressive imager testbed similar to the one described in [17, 16]. It consists of two optical arms and a discrete micro-mirror device (DMD) acting as a spatial light modulator as shown in Figure 5. The first arm, akin to an imaging lens in a traditional system, forms an optical image of the scene in the DMD plane. It has a 40° field of view and operates at F/8. The DMD has a resolution of 1920 × 1080 micro-mirror elements, each of size 10.8μm. However, in our system the field of view (FoV) is limited to an image circle of 7.5mm, which is approximately 700 DMD pixels. The DMD micro-mirrors are bi-stable and each is either oriented half-way toward the second arm or in the opposite direction (when the flux is discarded). The micro-mirrors can be switched in either direction at a very high rate to effectively achieve 8 bits gray-scale modulation via pulse width modulation. The optically modulated scene on the DMD plane is then imaged (by the second arm) and spatially integrated by a 1/3", 640 × 480 CCD focal plane array with a measurement depth of 12 bits. In the CCD plane, the field of view is 3mm in diameter (≈ 400 CCD pixels). Thus, in effect, this testbed implements several single pixel cameras [29] in parallel. Each block on the DMD effectively maps to a super pixel (e.g. 2 × 2 binned pixels) on the CCD. The DMD sequences (in time) through m projections, implementing the m rows of the $m \times n$ projection matrix Φ , where each projection vector appears as a $\sqrt{n} \times \sqrt{n}$ block pattern, replicated across the scene FoV. Before data acquisition, a calibration step is performed to map the DMD blocks to CCD detector pixels to characterize any deviation from the idealized system model.

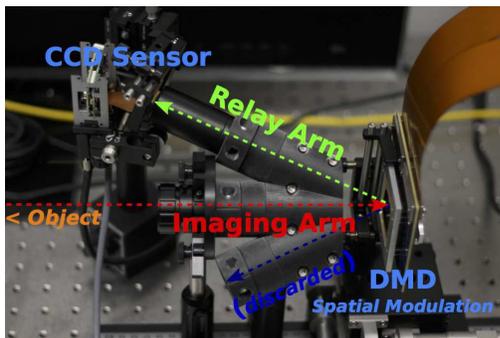


Figure 5: Compressive imager testbed layout with the object imaging arm in the center, the two DMD imaging arms are on the sides.

Reconstruction experiments: We use the set up described above to obtain the CS measurements for 383 blocks (size of 33 × 33) of the scene. Operating at MR's of 0.1 and

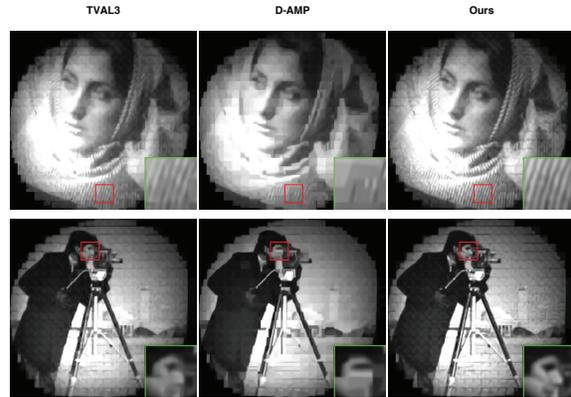


Figure 6: The figure shows reconstruction results on 3 images collected using our block SPC operating at measurement rate of 0.1. The reconstructions of our algorithm are qualitatively better than those of TVAL3 and D-AMP.



Figure 7: The figure shows reconstruction results on 3 images collected using our block SPC operating at measurement rate of 0.04. The reconstructions of our algorithm are qualitatively better than those of TVAL3 and D-AMP.

0.04, we implement the 8-bit quantized versions of measurement matrices (orthogonalized random Gaussian matrices). The measurement vectors are input to the corresponding networks trained on the simulated CS measurements to obtain the block-wise reconstructions as before and the intermediate reconstruction is denoised using BM3D. Figures 6 and 7 show the reconstruction results using TVAL3, D-AMP and our algorithm for three test images at MR = 0.10 and 0.04 respectively. It can be observed that our algorithm yields visually good quality reconstruction and preserves more detail compared to others, thus demonstrating the robustness of our algorithm.

5.3. Training strategy for a different Φ

In the experimental results presented earlier in this section, we assumed that the measurement matrix used to obtain the measurements of a test example is the same as the measurement matrix used to obtain the measurements of the training examples. However, in a practical scenario, this may not always be true, wherein one may wish to recon-

struct the images from CS measurements obtained using an arbitrarily different random Φ . Training a new network for the new Φ of a desired MR, as noted above, generally takes about 1 day, and hence may not be a feasible solution. To circumvent this problem, we propose a suboptimal, yet effective and computationally light training strategy outlined below, ideally suited to scenarios such as above, which will eliminate the need to train the network from scratch. Specifically, we adapt the convolutional layers (C1-C6) of a pre-trained network for the same or slightly higher MR, henceforth referred to as the *base network*, and train **only** the fully connected (FC) layer with random initialization for 1000 iterations (or equivalent time of around 2 seconds on a Titan X GPU), while keeping C1-C6 **fixed**. The mean PSNR (without BM3D) for the test-set at various MRs, the time taken to train models and the MR of the base network are given in table 3. From the table, it is clear that the overhead

New Φ MR	0.1	0.08	0.04	0.01
Base network MR	0.25	0.1	0.1	0.25
Mean PSNR (dB)	21.73	20.99	19.66	16.60
Training Time (seconds)	2	2	2	2

Table 3: Networks for a new Φ can be obtained by training only the FC layer of the base network at minimal computational overhead, while maintaining comparable PSNRs.

in computation for new Φ is trivial, while the mean PSNR values are comparable to the ones presented in table 1. We note that it may be possible to obtain better quality reconstructions at the cost of more training time if C1-C6 layers are also fine-tuned along with FC layer.

6. Real-time high level vision from CS imagers

In the previous section, we have shown how our approach yields good quality reconstruction results in terms of PSNR over a broad range of measurement rates. Despite the expected degradation in PSNR as the measurement rate plummets to 0.01, our algorithm still yields reconstructions of 15-20 dB PSNR and rich semantic content is still retained. As stated earlier, in many resource-constrained inference applications the goal is to acquire the least amount of data required to perform high-level image understanding. To demonstrate how CS imaging can be applied in such scenarios, we present an example proof of concept real-time high level vision application - tracking. To this end we simulate video CS at a measurement rate of 0.01 by obtaining frame-wise block CS measurements on 15 publicly available videos [33] (see supplementary for the list of videos) used to benchmark tracking algorithms. Further, we perform object tracking on-the-fly as we recover the frames of the video using our algorithm without the denoiser. For object tracking we use a state-of-the-art algorithm based on kernelized correlation filters [14]. We call the aforementioned pipeline, ReconNet+KCF. For comparison, we conduct tracking on original videos as well. Figure 8 shows the

average precision curve over the 15 videos, in which each datapoint is the mean percentage of frames that are tracked correctly for a given location error threshold. Using a location error threshold of 20 pixels, the average precision over 15 videos for ReconNet+KCF at 1% MR is 65.02%, whereas tracking on the original videos yields an average precision value of 83.01%. ReconNet+KCF operates at around 10 Frames per Second (FPS) for a video with frame size of 480×720 to as high as 56 FPS for a frame size of 240×320 . This shows that even at an extremely low MR of 1%, using our algorithm, effective and real-time tracking is possible by using CS measurements. More results can be found in the supplementary material.

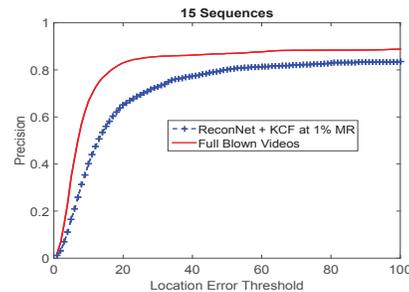


Figure 8: The figure shows the variation of average precision with location error threshold for ReconNet+KCF and original videos. For a location error threshold of 20 pixels, ReconNet+KCF achieves an impressive average precision of 65.02%.

7. Conclusion

We have presented a CNN-based non-iterative solution to the problem of CS image reconstruction. We showed that our algorithm provides high quality reconstructions on both simulated and real data for a wide range of measurement rates in real time. We note that the non-iterative and parallelizable nature of our algorithm lends itself to further reduction in its computational complexity as more powerful GPUs emerge. Through a proof of concept real-time tracking application at the very low measurement rate of 0.01, we demonstrated the possibility of CS imaging becoming a resource-efficient solution in applications where the final goal is high-level image understanding rather than exact reconstruction. However, the existing CS imagers are not capable of delivering real-time video. We hope that this work will give the much needed impetus to building of more practical and faster video CS imagers.

8. Acknowledgements

The work of KK, SL, and PT was supported by ONR Grant N00014-12-1-0124 sub-award Z868302. We thank Charles Collins for installing Caffe, the anonymous reviewers, Rushil Anirudh, Suren Jayasuriya and Arjun Jauhari for their valuable suggestions.

References

- [1] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. Inf. Theory*, 56(4):1982–2001, 2010. **2, 3**
- [2] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006. **2, 3**
- [3] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, 52(12):5406–5425, 2006. **2, 3**
- [4] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, pages 21 – 30, 2008. **2**
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007. **4**
- [6] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Euro. Conf. Comp. Vision*, pages 184–199. Springer, 2014. **1, 3, 4**
- [7] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang. Compressive sensing via nonlocal low-rank regularization. *Image Processing, IEEE Transactions on*, 23(8):3618–3632, 2014. **2, 3, 4, 5**
- [8] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006. **2, 3**
- [9] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009. **3**
- [10] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk. Wavelet-domain compressive signal reconstruction using a hidden markov tree model. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5137–5140. IEEE, 2008. **2, 3**
- [11] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Adv. Neural Inf. Proc. Sys.*, pages 2366–2374, 2014. **1, 3**
- [12] L. Gan. Block compressed sensing of natural images. In *Digital Signal Processing, 2007 15th International Conference on*, pages 403–406. IEEE, 2007. **2**
- [13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 580–587. IEEE, 2014. **1**
- [14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):583–596, 2015. **8**
- [15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. **4**
- [16] R. Kerviche, N. Zhu, and A. Ashok. Information optimal scalable compressive imager demonstrator. In *IEEE Conf. Image Process.*, 2014. **7**
- [17] R. Kerviche, N. Zhu, and A. Ashok. Information-optimal scalable compressive imaging system. In *Classical Optics 2014*. Optical Society of America, 2014. **7**
- [18] Y. Kim, M. S. Nadar, and A. Bilgin. Compressed sensing using a gaussian scale mixtures model in wavelet domain. In *IEEE Conf. Image Process.*, pages 3365–3368. IEEE, 2010. **2, 3**
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inf. Proc. Sys.*, pages 1097–1105, 2012. **1**
- [20] C. Li, W. Yin, H. Jiang, and Y. Zhang. An efficient augmented lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3):507–530, 2013. **2, 4, 5**
- [21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comp. Vision and Pattern Recog*, June 2015. **1, 3**
- [22] M. Lustig, D. Donoho, and J. M. Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007. **2**
- [23] M.B. Wakin, J.N. Laska, M.F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K.F. Kelly and R.G. Baraniuk. An architecture for compressive imaging. In *IEEE Conf. Image Process.*, 2006. **2**
- [24] C. A. Metzler, A. Maleki, and R. G. Baraniuk. From denoising to compressed sensing. *arXiv preprint arXiv:1406.4175*, 2014. **2, 3, 4, 5**
- [25] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. *arXiv preprint arXiv:1508.04065*, 2015. **3, 5**
- [26] G. Peyré, S. Bougleux, and L. Cohen. Non-local regularization of inverse problems. In *Euro. Conf. Comp. Vision*, pages 57–68. Springer, 2008. **2, 3**
- [27] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk. Cs-muvi: Video compressive sensing for spatial-

- multiplexing cameras. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–10. IEEE, 2012. [2](#)
- [28] S. Som and P. Schniter. Compressive imaging using approximate message passing and a markov-tree prior. *Signal Processing, IEEE Transactions on*, 60(7):3439–3448, 2012. [2](#), [3](#)
- [29] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006. [7](#)
- [30] J. Tan, Y. Ma, and D. Baron. Compressive imaging via approximate message passing with image denoising. *Signal Processing, IEEE Transactions on*, 63(8):2085–2092, 2015. [3](#)
- [31] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. *arXiv preprint arXiv:1505.00295*, 2015. [1](#), [3](#)
- [32] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2015. [1](#), [3](#)
- [33] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015. [8](#)
- [34] J. Zhang, S. Liu, R. Xiong, S. Ma, and D. Zhao. Improved total variation based image compressive sensing recovery by nonlocal regularization. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2836–2839. IEEE, 2013. [2](#), [3](#)
- [35] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *Euro. Conf. Comp. Vision*, pages 834–849. Springer, 2014. [1](#)
- [36] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 1637–1644. IEEE, 2014. [1](#)
- [37] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Adv. Neural Inf. Proc. Sys.*, pages 487–495, 2014. [1](#)