

## Deep Learning for Lensless Compressive Imaging

Xin Yuan<sup>1</sup> and Yunchen Pu<sup>2</sup>

<sup>1</sup> Nokia Bell Labs, Murray Hill, NJ.

<sup>2</sup> Department of ECE, Duke University, Durham, NC.

We investigate the deep learning techniques for lensless compressive imaging. A block-wise lensless compressive camera has been built, which captures the scene compressively and in parallel using multiple sensors. A deep learning based algorithm is developed to reconstruct the scene from compressed measurements in real time.

The lensless compressive camera [1], enjoys low-cost, low-power properties and has demonstrated excellent results using advanced reconstruction algorithms [2]. Furthermore, the same architecture can be used for imaging of visible spectrum, and other spectra such as infrared and millimeter waves. The architecture can also be used to capture hyperspectral images [3] and polarized images [4] by integrating related hardware. However, two disadvantages exist in the current lensless compressive camera, and both are related to speed. Firstly, the capture speed is slow, which limits the applications of the lensless compressive camera. The second one is that the reconstruction algorithm is slow such that user needs to wait, e.g. several minutes, to obtain the reconstructed image.

To mitigate the first slow capture issue, we build a new camera, called the block-wise lensless camera (Fig. 1) to capture the scene in parallel. Specifically, multiple sensors are employed in our camera and each sensor is capturing a fraction of the scene. These sensors are performed in parallel. The camera consists of three components: a) the sensor board which contains multiple sensors and each one corresponding to one block, b) the isolation chamber which prevents the light from other blocks, and c) the aperture assembly. Note that the block sizes can be different such that each block can reconstruct different resolution image fractions, thus leading to multi-scale compressive camera. The pattern used for each block can also be different and can be adapted to the content of the image part, thus leading to adaptive compressive sensing. We have built our prototype using a 3.5-inch programmable transmission LCD and a 4x4 array sensor board.

A generative model (Fig. 2) based on deep convolutional neural networks (CNN) is developed for real time reconstruction [5]. Specifically, we first use a fully-connected layer to convert the measurement to a 4096-dimension vector, and then reshape it to a 4x4x256 tensor. Following this, a series of deconvolutional (a.k.a transposed convolutional) layers along with the batch normalization (BN) and leaky rectified activation (Leaky ReLU) with different number of neurons are used to generate the desired image patch with size of 16x 16 pixels, which fits our hardware implementations. All the deconvolutional layers are associated with zero padding to make the dimension of input and output feature maps be the same. Results of digital data captured by our camera are shown in Fig.2(e) at various compressive sensing ratio (CSr), which is defined as measurement number/pixel number.

### References:

- [1] G Huang *et al*, IEEE International Conference on Image Processing (2013), p. 2101.
- [2] X Yuan *et al*, IEEE Sens. J. **16** (2016), p. 8091.
- [3] X Yuan *et al*, IEEE J. Sel. Top. Sig. **9** (2015), p. 964.
- [4] T-H Tsai, X Yuan and D J Brady, Opt. Express **23** (2015), p. 11912.

[5] X Yuan and Y Pu, Opt. Express 26 (2018), p. 1962.

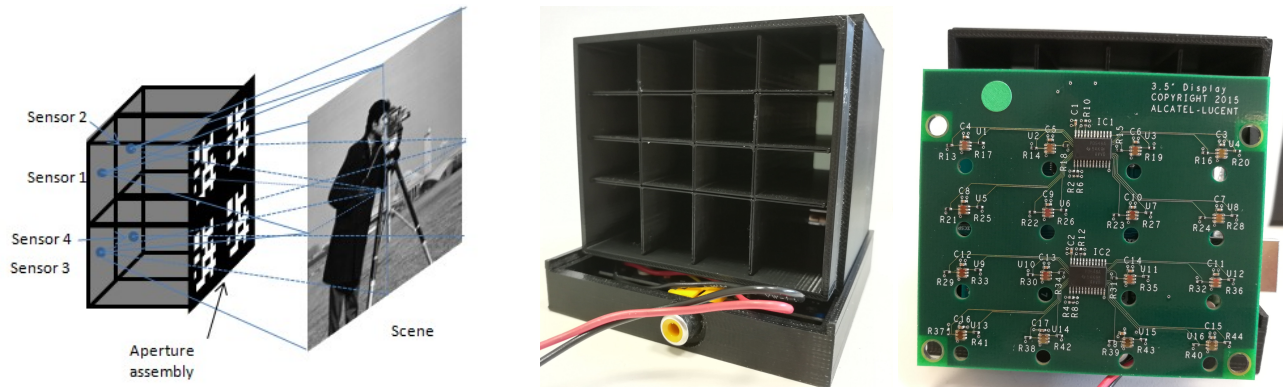


Figure 1. Left: principle of the block-wise lensless camera. Middle and Right: the built camera.

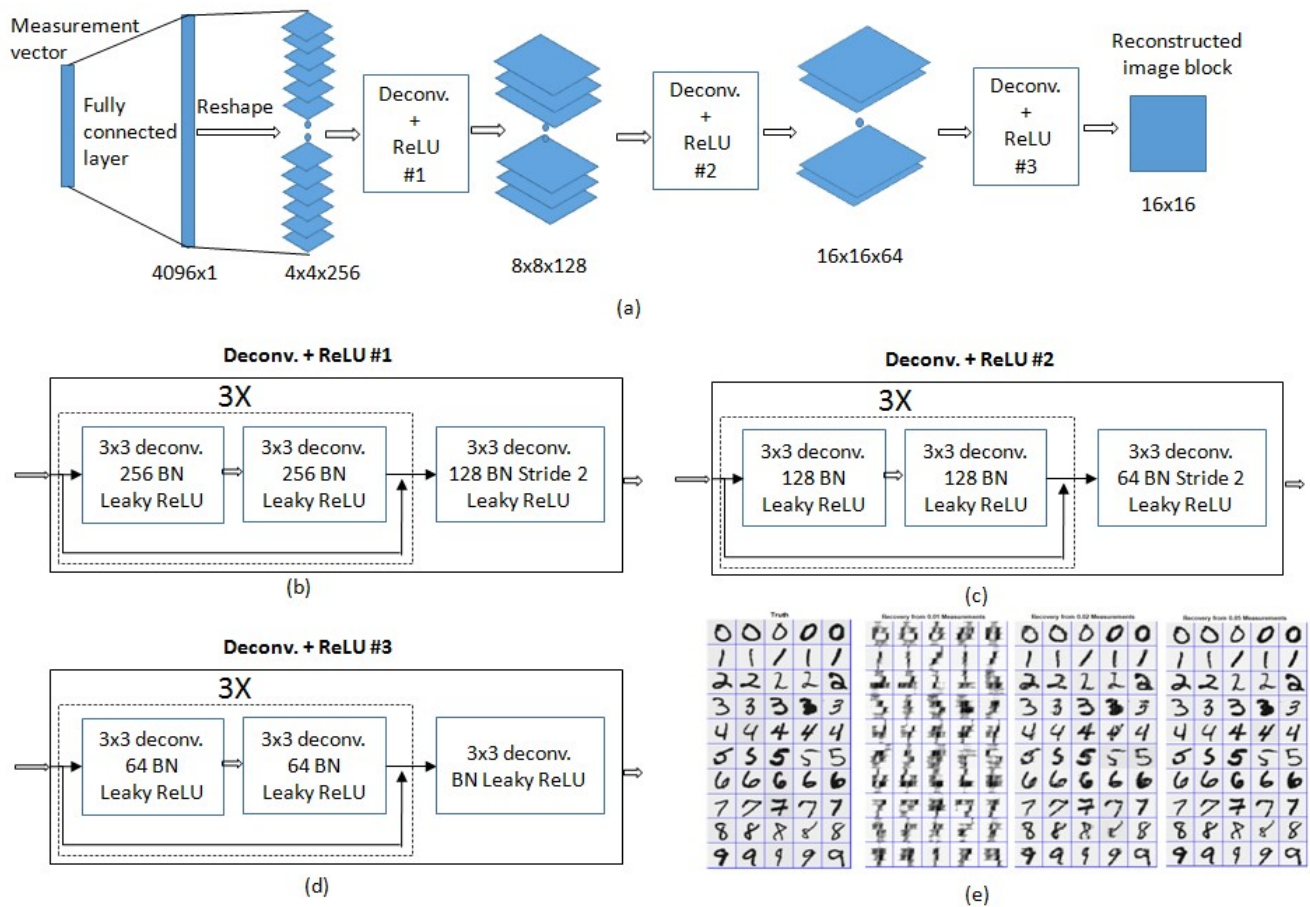


Figure 2. Architecture of the deep CNN used in our imaging system (a). The input is the measurement vector of each block. The “output” on the bottom of (a) lists the size of data at each step. The final output on the right is the reconstructed image block of size 16x16. The “Deconv.” denotes the deconvolutional (a.k.a transposed convolutional) layers. (b-d) The “Deconvolution + ReLU” units, where x3 means the network in the dashed box is stacked three times. “BN” denotes batch normalization. The layer without “stride” denotes stride 1. (e) Reconstructed images compared with ground truth. Far left: truth, right three: reconstructed results at CSr = 0.01, 0.02 and 0.05.