Deep Convolutional Autoencoders for Deblurring and Denoising Low-Resolution Images

Michael Fernando Mendez Jimenez
Department of Applied Mathematics
University of California, Merced
Merced, USA
mmendezjimenez@ucmerced.edu

Omar DeGuchy
Department of Applied Mathematics
University of California, Merced
Merced, USA
odeguchy@ucmerced.edu

Roummel F. Marcia
Department of Applied Mathematics
University of California, Merced
Merced, USA
rmarcia@ucmerced.edu

***Abstract*—In this paper, we implement machine learning methods to recover higher-dimensional signals from lower-dimensional, noisy, and blurry measurements. In particular, rather than utilizing optimization-based reconstruction methods, we use fully-connected multilayer perceptron (MLP) architectures and convolutional neural networks (CNN). In addition, we consider two different loss functions based on mean squared error and a Huber potential to train our models. Numerical experiments on the Street View House Numbers dataset show that while fully-connected MLPs are faster to train, reconstructions using CNNs are much more accurate.**

***Index Terms*—Machine learning, Autoencoders, Deblurring, Denoising, Upsampling**

I. Introduction

With the number of digital images being taken everyday, image processing techniques used to improve the quality of these signals become increasingly more important. Two of the main sources of deterioration during image acquisition are blurring and noise. Blurring occurs when information seeps among neighboring pixels in an image. This phenomenon can be the result of mechanical failure in the imaging system, such as an out of focus lens. It can also be the result of the physical conditions when the image is recorded such as image obfuscation by fog or the turbulent atmosphere present in astronomical imaging applications [1].

In most imaging applications, the presence of noise is modeled by additive white Gaussian noise and can be a result of temperature or electrical fluctuations within the imaging system [2]. This model assumes adequate lighting conditions when the observations are recorded. However, in applications such as medical imaging and night vision, the number of photons recorded at the detector is relatively low. Under this photon-limited regime, the measurements at the photon-detector are corrupted by noise that is modeled more appropriately using the Poisson distribution [3]. In addition to being noisy, these observations are often undersampled linear measurements, further complicating the recovery process. Under the process of blurry photon-limited imaging, we seek to reconstruct images from noisy, low-dimensional, and blurry observations. In this paper we look towards various deep learning architectures as a technique to recover the images associated with these degraded observations.

This work was supported by National Science Foundation Grant IIS-1741490.

II. Problem Formulation

In this section we explain the model that we use to describe the low-resolution blurred noisy observations. Because we are operating in the regime of photon-limited imaging, we model the arrival of photons at the detector by the following inhomogenous Poisson process:

$$\mathbf{y} \sim \text{Poisson}(\mathbf{D}\mathbf{G}\mathbf{f}^*),$$

where $\mathbf{y} \in \mathbb{Z}_+^m$ is the observation vector whose entries consists of photon counts, $\mathbf{f}^* \in \mathbb{R}_+^n$ is the true signal, $\mathbf{D} \in \mathbb{R}^{m\times n}$ is a downsampling operator with $m < n$, and $\mathbf{G} \in \mathbb{R}^{n\times n}$ is a Gaussian blur operator. The two-dimensional blurring operation takes a two-dimensional Gaussian distribution of the form

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \tag{1}$$

and creates a normalized pixel masked that is convolved with the image of interest in order to perform the blur operation [4]–[6]. Here, σ^2 corresponds to the variance of the Gaussian distribution. Our interest is in recovering the true high-dimensional signal $\mathbf{f}^*$ from the lower dimensional observation vector $\mathbf{y}$. Traditional signal recovery methods rely on statistical methods in order to maximize the probability of observing the vector $\mathbf{y}$. This can include using optimization techniques in conjunction with the maximum likelihood principle as well as Baysian-based approaches. The sparse nature of the true signal of interest invites the use of a sparsity promoting penalty term in the optimization process. This type of algorithm is iterative and requires the tuning of parameters which affects the quality of the reconstruction [7]–[11].

Related work. This work seeks an alternative to traditional optimization methods by solving the low-quality Poisson blurred reconstruction problem using deep learning architecture as an approximate mapping from the downsampled observations $\mathbf{y}$ to the space of the true signal $\mathbf{f}^*$. While previous implementations of deep learning techniques have been used to address Poisson denoising problems, downsampling problems, and image deblurring [12]–[16] separately, the novelty of our

approach is that we implement architecture which solves all three problems simultaneously.

III. Proposed Approach

In this section we describe the deep learning techniques implemented to recover data from noisy, blurred, low-dimensional observations. In particular, we present three different architectures labeled Methods I, II, and III. Each method utilizes a different configuration of different types of layers. Specifically, Method I relies on fully connected layers and is modeled to closely resemble a stacked denoising autoencoder [17]. Methods II and III rely on convolutional layers typically found in convolutional neural networks (CNNs) [18]. In all methods, the architectures are trained using back-propagation and the resultant features from every layer are activated by the rectified linear unite (ReLU) activation function. The types of loss or cost functions used during training will be addressed in Section IV. We now describe each method in further detail.

Method I (MLP-AE). The first architecture is composed entirely of fully connected layers. As stated before, the feed-forward network was modeled after a stacked denoising autoencoder (SDA). This type of architecture has been successful in addressing applications where denoising is required [12], [17], [19], [20]. The motivation for this technique is that by compressing the signal into a latent space, the encoding becomes more robust to noise. The proposed method distinguishes itself from a traditional SDA in that the dimensions of the input do not match those of the recovered signal. Instead, the first layer of the architecture performs an upsampling of the observational input, implicitly learning the inverse projection from the observation space. For a 16×16 input of one channel, the input is reshaped into a vector of length 256. The fully connected layer brings the dimension of the features to 1024. The architecture then begins to behave like a traditional encoder by compressing the feature space back to a length of 256 and finally back up to 1024 which is reshaped to a 32×32 reconstruction of the original signal (see Fig. 1).

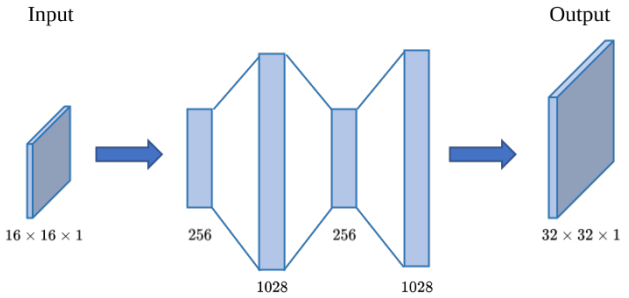


Fig. 1. Method I: Multilayer perceptron autoencoder-type (MLP-AE) network composed of fully-connected layers.

Method II (CNN). The second method relies on the power of convolutions and draws inspiration from the successful application of CNNs to a variety of problems in inverse problems and compressed sensing [16], [21]–[23]. The network is required to address the same initial upsampling problem as the architecture in Method I. In order to accommodate the discrepancy between the input size and the recovered signal size, we use an initial fully connected layer to boost the features from a vector of length 256 to 1028. The vector is then reshaped into a tensor with dimensions of $32 \times 32 \times 1$. The convolutional layers begin after the reshaping and attempt to maintain feature output sizes of $32 \times 32 \times N$ where $N \in \{64, 32, 1\}$. Each element N corresponds to the number of channels in the feature map produced by the previous layer. The final layer produces the reconstructed signal with a single channel. The first two dimensions of the feature tensors are maintained through the careful consideration of padding, filter size and stride (see Fig. 2).

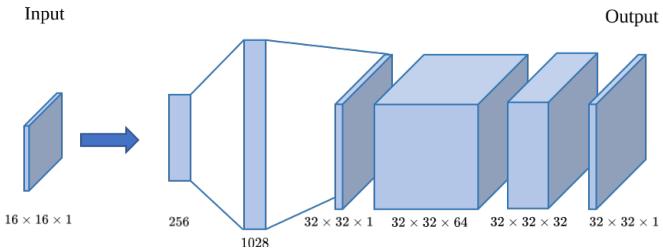


Fig. 2. Method II: Convolutional neural network (CNN).

Method III (CNN-AE). The final architecture draws on the structures of Methods I and II. In a similar fashion to Method II, we use an initial fully connected layer to upsample the feature space and rely on convolutions for the remaining layers. The convolutional layers are arranged to resemble an autoencoder taking inspiration from the fully connected layers of method I. The number of channels is initially increased in the feature map after the first layer while maintaining the dimensions of the height and width. The "encoder" portion of the convolutional layers compress the feature maps by half in all directions until a latent tensor of dimension $8 \times 8 \times 16$ is acheived. We then use 2-D transpose convolution operators in order to increase the dimensions while reducing the number of channels until we achieve dimensions of the original signal (see Fig. 3).

IV. Numerical Experiments

The three architectures, Methods I (MLP-AE), Method II (CNN), and Method III (CNN-AE), were all implemented using Pytorch, the open source machine learning language for python. Training and testing was performed using an NVIDIA GTX 960 on a local PC with 16 GB of RAM. The networks were trained using the stochastic gradient descent (SGD) method.

Dataset. The dataset used to test and train the proposed architectures is based on the Street View House Numbers Dataset (SVHS) dataset [24]. This dataset consists of 95,000 32×32 images of street view house numbers from 0-9. The data is then partitioned into 73,257 training examples, 26,032 testing examples, and 5,000 images used as a validation

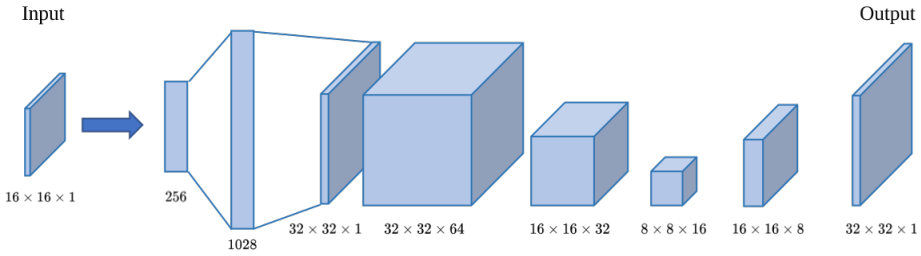


Fig. 3. Method III: Convolutional-type autoencoder architecture (CNN-AE).

set. To create our dataset, we converted the original color training and testing samples to gray scale, applied a Gaussian filter to create the blur, downsampled through an average 2D pooling to reduce the size of the images, and finally imposed Poisson noise on the downsampled blurry images. The labels were discarded because the focus of this work is not classification. The images were normalized so that the pixel intensities ranged between 0 to 1. The training set was created by pairing the clean images with their corresponding noisy images. Using this structure the neural network is expected to train on a set of noisy images with a fixed 16×16 dimension and reconstruct the full 32×32 image.

Training. Training was performed via stochastic gradient descent (SGD) with a learning rate of 0.01 and a batch size of 32 images. Two different experiments were implemented comparing the reconstructed images ($p(x_i)$) of inputs x_i with their targets (y_i). The first experiment uses the common choice of the Mean Squared Error (MSE) as a loss function given by

$$\text{MSE}(x,y) = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\|p(x_i)-y_i\|_2^2, \tag{2}$$

where $\mathcal{S}$ is the dataset and $|\mathcal{S}|$ is its cardinality. The second experiment makes use of a modified ℓ_1 loss known as the Huber loss, or smooth ℓ_1 loss [25], which is given by

$$H(x,y) = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\sum_{k=1}^{n} z_{ik} \tag{3}$$

where

$$z_{ik} = \begin{cases} 0.5((p(x_i))_k - y_{ik})^2, & \text{if } |(p(x_i))_k - y_{ik}| < 1 \\ |(p(x_i)_k - y_{ik}| - 0.5, & \text{otherwise.} \end{cases}$$

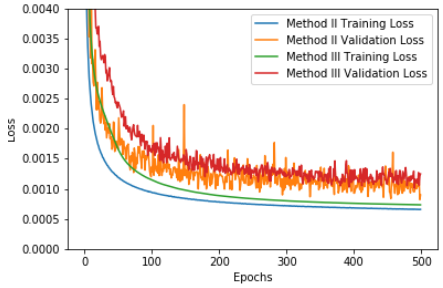
The implementation of the loss in Pytorch is restricted to a threshhold value of 1 and is less sensitive to outliers as opposed to the MSE loss.

Results and Analysis. Initially we compare the difference in loss functions across all methods. In either choice of function, the losses resulting from the convolutional based architectures (Methods II and III) were significantly lower than the method using fully connected layers (Method I). Figs. 4(a) and 4(b) illustrate the evolution of MSE and smooth ℓ_1 losses for Methods II and III – the losses for Method I were significantly higher and are thus not presented. Methods II and III were both able to reduce the loss functions in a similar manner. We also note that both methods had relatively similar behavior between the training loss and the validation loss, exhibiting that the networks were not over trained. In either choice of loss function, Method II was able to achieve lower training and validation losses much sooner and maintained than Method III. Method II was also able to maintain an improved training loss for the duration of training.

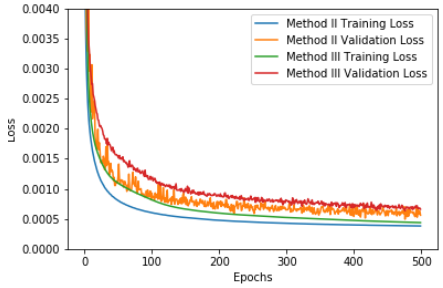
The results on the testing set seem to confirm those provided in the validation set. When considering the approximately 26,000 images in the testing set, the convolutional neural networks improve on the perceptron based architectures (by approximately an order of magnitude). Performance is evaluated using the MSE by comparing the reconstruction with the ground truth image (see Fig. 5). The higher MSE values associated with Method I can be attributed to the "dead pixels" present in its output (see Fig. 6). These pixels are consistently being set to zero by the network across all reconstructions. Upon further inspection of the weights being learned, we note that the outputs are actually being set to negative values while the ReLU activation at the end of the network transforms these values to zero. Method II and Method III perform comparably in terms of MSE. Qualitatively, Method III using the MSE as a loss function seems to reconstruct slightly sharper details such as the thin roof of the "9" (Image 3) in Fig. 6. Using the smooth ℓ_1 loss function particularly in Method II seems to produce blurred sections in the reconstructions as well as lower pixel intensities than reconstructions from Method III.

V. Conclusions

In this paper we implement three different deep learning architectures in order to solve the photon-limited deblurring problem. The first method (MLP-AE) involves using fully connected layers in an autoencoder type configuration. The second method (CNN) uses convolutional layers in a classical feed forward network. Finally, the third method (CNN-AE) uses convolutional layers in an autoencoder type structure which is analogous to the first method. Although all three methods produce discernible reconstructions, the advantage clearly goes to the use of convolutional layers. The output from the MLP-AE contains many artifacts which can be attributed to negative values being mapped to zero pixel intensities by the ReLU activation function. We further extend the experiments



(a) MSE loss



(b) Smooth ℓ_1 loss

Fig. 4. Losses for Methods II and III. (a) MSE loss. (b) Smooth ℓ_1 loss. Both loss functions for both methods behave similarly and achieve small values.

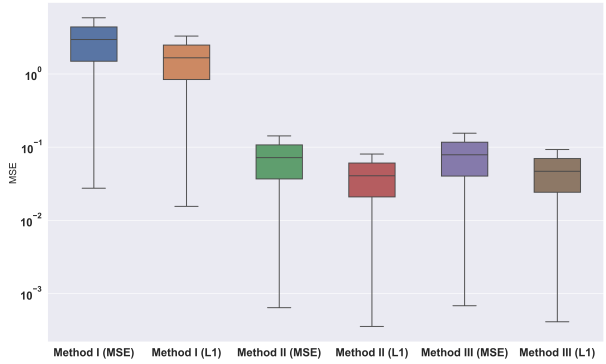


Fig. 5. Boxplots comparing the mean squared error (MSE) computed using the SVHS dataset (26,032 images). The MSE is computed comparing the reconstructions from Method I, Method II and Method III with the ground truth image.

by using Mean Squared Error (MSE) or the smooth ℓ_1 loss as cost functions during training. The experiments indicate that using the ℓ_1 loss during training improves the reconstruction in terms of MSE. Not being captured by the MSE as a performance metric is the ability of the CNN-AE to reconstruct certain details when the MSE is used as a loss function. For future work we hope to extend these techniques to different types of noise and image modalities.

REFERENCES

[1] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring images: matrices, spectra, and filtering*. SIAM, 2006, vol. 3.

[2] F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed Poisson-Gaussian noise," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, 2010.

[3] D. L. Snyder and M. I. Miller, *Random point processes in time and space*. Springer Science & Business Media, 2012.

[4] R. A. Haddad, A. N. Akansu *et al.*, "A class of fast Gaussian binomial filters for speech and image processing," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 723–727, 1991.

[5] A. R. Weeks, *Fundamentals of electronic image processing*. SPIE Optical Engineering Press, 1996.

[6] J. C. Russ, "Image Processing," *Materials Science Engineering Department North Carolina State University Raleigh, North Carolina*, 2002.

[7] M. Carlavan and L. Blanc-Féraud, "Sparse Poisson noisy image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1834–1846, 2011.

[8] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini, "Image deblurring with Poisson data: from cells to galaxies," *Inverse Problems*, vol. 25, no. 12, p. 123006, 2009.

[9] S. Setzer, G. Steidl, and T. Teuber, "Deblurring Poissonian images by split Bregman techniques," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 193–199, 2010.

[10] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "This is SPIRAL-TAP: Sparse Poisson intensity reconstruction algorithms; theory and practice," *IEEE Trans. on Image Processing*, vol. 21, no. 3, pp. 1084–1096, 2012.

[11] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3133–3145, Dec 2010.

[12] O. DeGuchy, F. Santiago, M. Banuelos, and R. F. Marcia, "Deep neural networks for low-resolution photon-limited imaging," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3247–3251.

[13] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891.

[14] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1439–1451, 2015.

[15] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Deep convolutional denoising of low-light images," *arXiv preprint arXiv:1701.01687*, 2017.

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

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[19] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," *arXiv preprint arXiv:1206.4683*, 2012.

[20] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 1336–1343.

[21] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4293–4308, 2017.

[22] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-based deep learning architecture for inverse problems," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 394–405, 2018.

[23] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Advances in Neural Information Processing Systems*, 2017, pp. 1772–1783.

[24] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[25] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

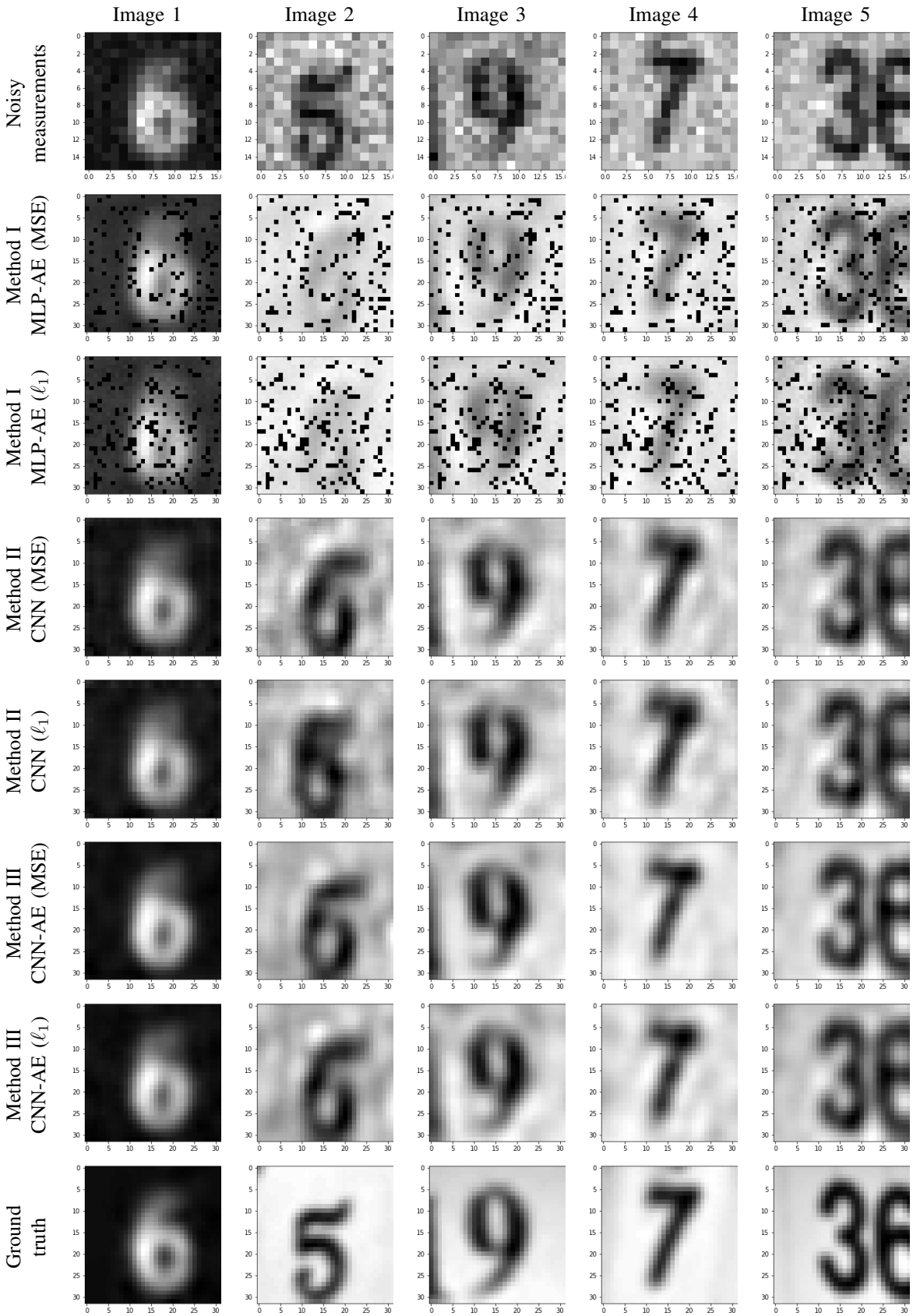


Fig. 6. Representative images of the low-dimensional, blurry, and noisy measurements (Row 1), reconstructions from Methods I, II, and III using the MSE and smooth ℓ_1 loss functions (Rows 2-7), and ground truth from the dataset (Row 8).