Astronomical image reconstruction with deep convolutional neural networks

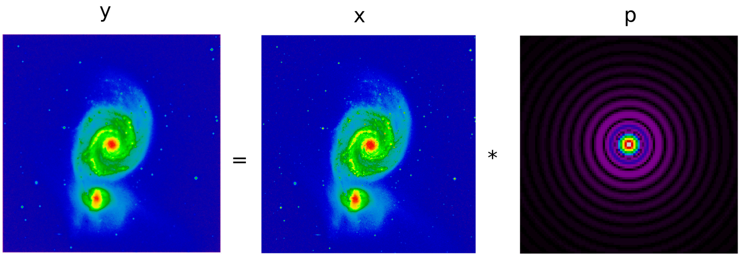
Rémi Flamary - Côte d'Azur University, Lagrange Laboratory, OCA
Collaboration with : M. Moscu, R. Ammanouil, A. Ferrari, C. Richard

January 26, 2018, Cosmostat, CEA Saclay

Summary

Introduction

Astronomical image reconstruction



Astronomical image observation

- Convolutional model : $y = x * p$
  - y is the observed image (dirty).
  - x is the true image.
  - p is the Point Spread Function (PSF)
- Geometry of the telescope gives the Point Spread Function (PSF).
- Some noise due to the observation is also present (Gaussian,Poisson).
- On wide field of view the PSF can be space variant (Fredholm's integral).

Inverse problem

Image reconstruction

$$\min_{x} \quad L(y, x * p) \tag{1}$$

where L is a data fitting loss.

- We want to inverse the observation process.
- Reconstruct an estimation of the true image x from y.
- For every new observation one needs to solve the problem.
- Linear PSF interpolation for fast fft convolution [Denis et al., 2015].

Common approaches and algorithms

- Wiener filtering (inverse filtering+noise attenuation).
- [Richardson, 1972, Lucy, 1974], CLEAN [Högbom, 1974].
- Sparsity promoting regularization (iterative algo. with proximal gradient descent).

Proximal gradient methods

Principle of proximal gradient descent

- Minimization of convex non-smooth objective (sparsity promoting regularization).
- Use proximity operator for non-differentiable functions.
- Can be accelerated and solved with Primal-Dual algorithm.

Optimization problem for Muffin [Deguignet et al., 2016]

$$\min_{x} \quad \frac{1}{2}\|y - x * p\|^2 + \mathcal{I}_{\mathbb{R}+}(x) + \mu_s \cdot \| W_s x \|_1 \tag{2}$$

- Vu [Vũ, 2013] Condat [Condat, 2014] algorithm.
- Iterative approach where every iteration is $O(n)$.
- Convergence can be slow (very sensitive to the initialization).
- Regularization parameter selected automatically [Ammanouil et al., 2017]

Supervised deep learning

Deep neural network [LeCun et al., 2015]

$$f(x) = f_K(f_{K-1}(...f_1(x)...)) \tag{3}$$

- f is a composition of basis functions f_k of the form :

$$f_k(x) = g_k(W_k x + b_k) \tag{4}$$

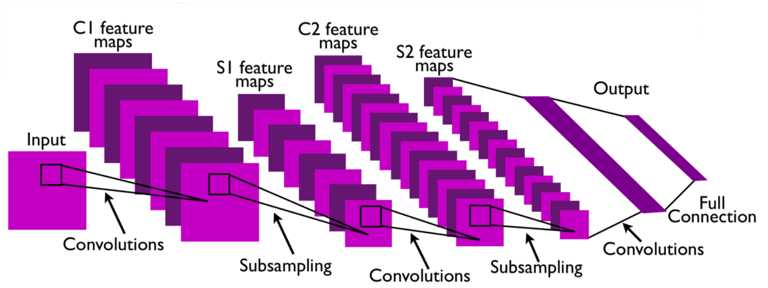
- W_k is a linear operator and b_k is a bias for layer k.
- g_k is a non-linear activation function for layer k.

Supervised training of deep neural networks

$$\min_f \quad \sum_i L(p_i, f(x_i)) \tag{5}$$

- L is the prediction error.
- $\{p_i, x_i\}_i$ is the training dataset.
- Function parameters $\{W_k, b_k\}_k$ learned with stochastic gradients.

Convolutional neural network



- Replace the linear operator by a convolution [LeCun et al., 2010].
- Reduce image dimensionality with sub-sampling or max pooling.
- Number of parameters depends on the size fo the filter, not the image.
- Recent deep CNN use Relu activation [Glorot et al., 2011] : $g(x) = \max(0, x)$

Image reconstruction with deep learning

Training for image reconstruction

Deep learning for inverse problem [McCann et al., 2017]

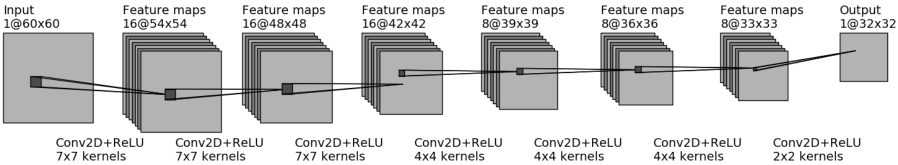
- Train a function f that solves approximately the inverse problem.
- Move computational complexity to the training step.

Deep network for image reconstruction [Xu et al., 2014, Flamary, 2017]

$$\min_f \quad \frac{1}{2N}\sum_i^N \|x_i - f(y_i)\|^2$$

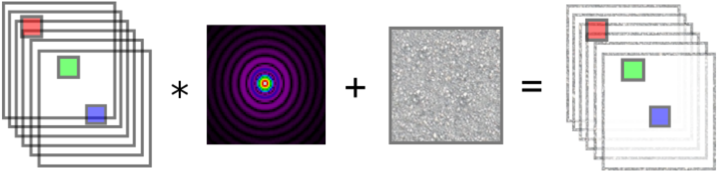
- f is the deep network with architecture tailored for image reconstruction.
- $\{x_i, y_i\}_{i=1\ldots N}$ are the training dataset.
- Optimization of f is done once.
- Reconstruction for new image is $f(y)$.

Network architecture



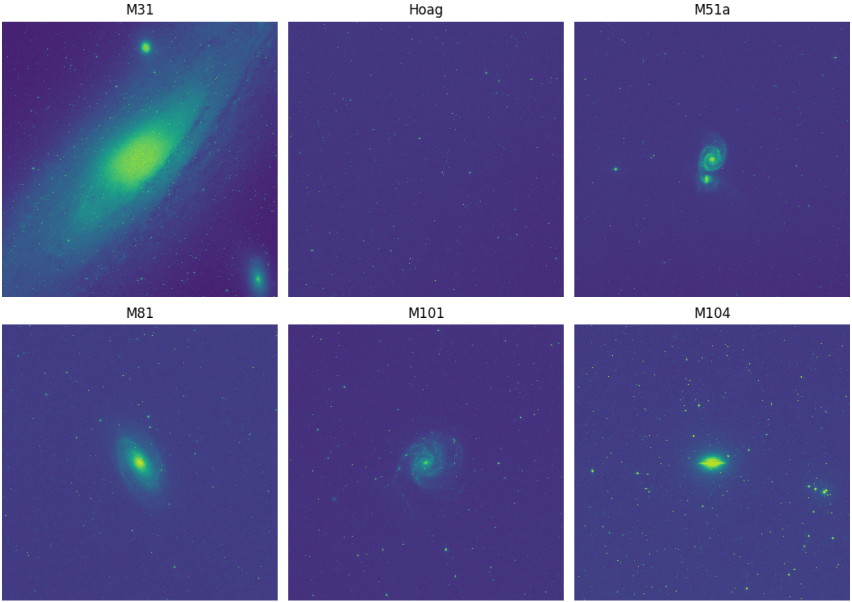
- Architecture is a classical 6 layers CNN.
- Each Layers consists in
  - a convolutional layer with small 2D filters,
  - a Relu activation of the form $g(x) = \max(0, x)$ [Glorot et al., 2011] .
- Exact convolution leads to an output smaller than the input (60$\rightarrow$32).
- The network is stationary and can be adapted to any image size.
- Reconstruction can be done on patches or one large image.
- Relu is good for deep learning because it has no vanishing gradients.

Training dataset

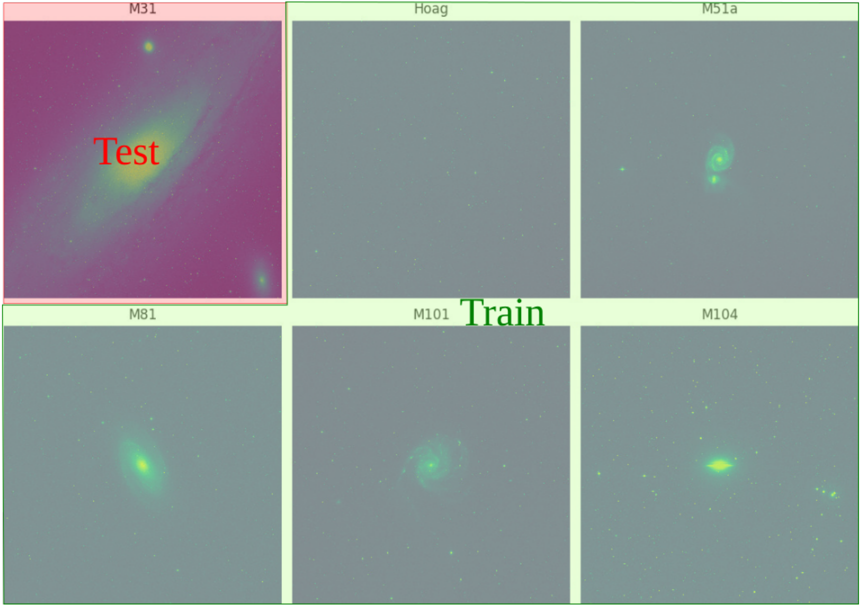


- Dataset is generated online from true/observed images.
- We randomly draw patches from training images and add random noise.
- Generated noise ensure that a sample is never seen twice by the network.
- We use 6 large images of size 3564x3564 from STScIDigitized Sky Survey, HST Phase 2 dataset.
- Performance is evaluated with One-VS-All approach (train on 5 images, test on the 6th).

Training dataset



Training dataset



Model estimation

Estimation problem

$$\min_f \quad \frac{1}{2N} \sum_i^N \|x_i - f(y_i)\|^2$$

- The full model has ≈ 30000 parameters.
- Use a generator to draw randomly training samples.
- Optimization with stochastic gradient with minibatch.
- Two kind of minibatch for gradient computation :
  - Local due to the size of the patch.
  - Global due to the number of patch.
- Use Nesterov-type acceleration.
- Stop learning when the average loss do not decrease anymore.

Implementation

Python implementation

- Implementation using Theano/Keras.
- Train and predict using NVIDIA Titan X GPU.
- One epoch takes $\approx$ 45 seconds.

Training parameters (tricks of the trade)

- Parameter initialization with normalised Gaussian [Glorot and Bengio, 2010].
- Learning rate=0.01, momentum=0.9.
- Minibatch of size 50 patches.
- Epochs of 300 000 samples.
- Restart initialization if no change in loss after one epoch.

Numerical experiments

Numerical comparisons

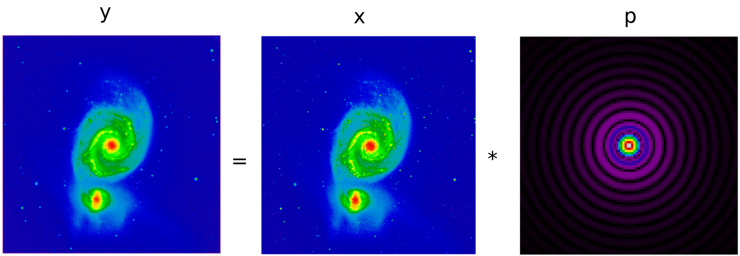
Constant PSF

- Wiener filtering with Laplacian regularization [Orieux et al., 2010].
- Richardson Lucy [Richardson, 1972, Lucy, 1974].
- Proximal gradient descent with sparse wavelet regularization and automatic regularization estimation [Ammanouil et al., 2017].
- Shallow CNN with 1 linear Layer, supervised Wiener (CNN0).
- Proposed Deep CNN (DCNN).

Space variant PSF

- Approximate variation with linear interpolation [Denis et al., 2015].
- Adaptation of Richardson-Lucy and Proximal gradient descent using FFT.
- Comparison of DCNN learned on fixed center PSF (DCNN C) and on variant PSF (DCNN SV).

Constant PSF : data and protocol



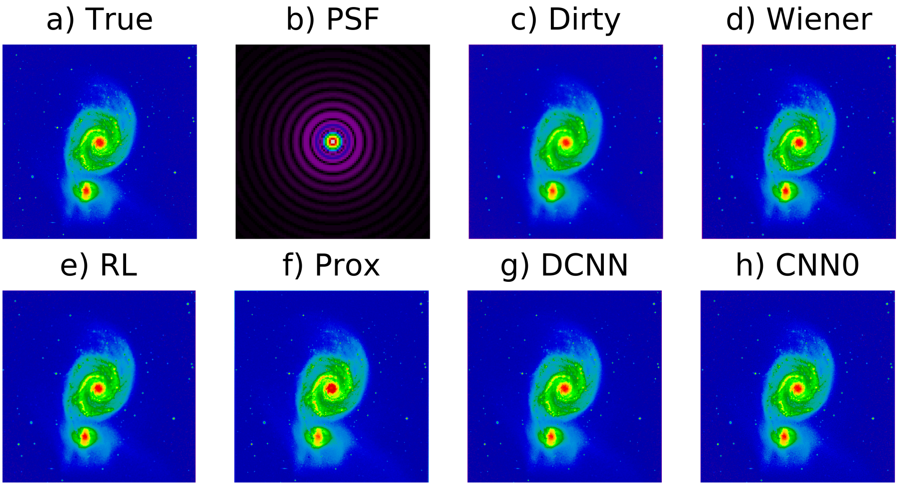
- We use the central 1024*x*1024 pixels images for comparison.
- Data normalized to a maximum value of 1.
- PSF for a circular apperture : $p(r) = I_0(J_1(r)/r)^2$
- Radius of PSF *r* scaled so that we have 100 rebounds in the image.
- Gaussian noise of standard deviation $\sigma = 0.01$.

Constant PSF : numerical performances

Image \ Method	Wiener	RL	Prox	DCNN	CNN0
M31 : 31.83	31.88	31.17	**31.98**	31.26	31.44
Hoag : 35.39	36.70	36.77	36.76	**40.04**	37.98
M51a : 35.81	37.29	37.16	38.39	**39.89**	38.16
M81 : 34.23	35.05	34.82	35.91	**36.79**	36.02
M101 : 34.71	35.97	36.28	36.63	**39.75**	37.78
M104 : 33.49	33.97	33.27	34.52	**35.39**	35.07
Avg. PSNR (dB)	35.14	34.91	35.70	**37.18**	36.11
Avg. time (s)	**0.22**	4.94	593.42	1.65	0.44

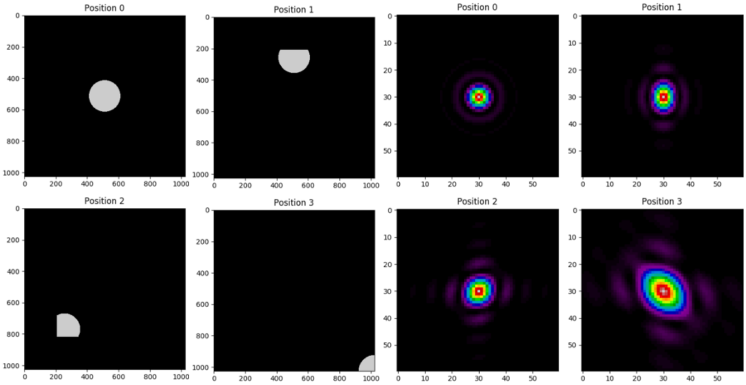
- DCNN has best PSNR on all images except M31.
- Importance of representative dataset.
- Prox works best of all other methods but important numerical cost.
- 1024*x*1024 image reconstructed in 1.65 seconds.

Constant PSF : Visual comparison



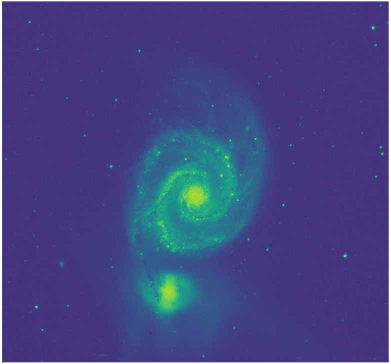
- Visual comparison for different methods.
- PSF is zoomed and represented with its square root.

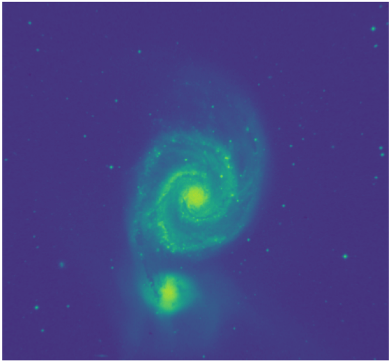
Varying PSF : data



- PSF for circular aperture at the center of the image.
- Varying PSF corresponding to box occultation in a wide field.
- Pre-compute exact Fredholm's integral on the images.

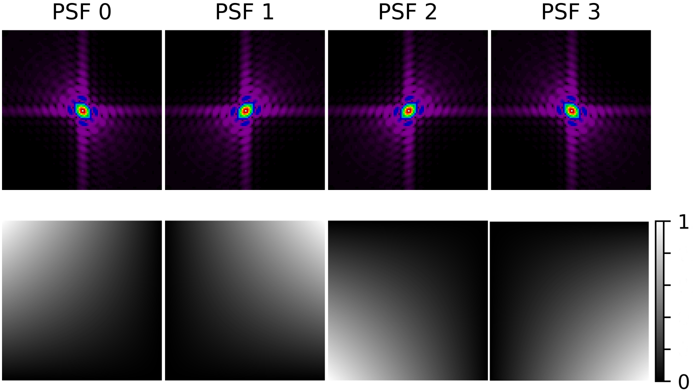
Varying PSF : data





- PSF for circular aperture at the center of the image.
- Varying PSF corresponding to box occultation in a wide field.
- Pre-compute exact Fredholm's integral on the images.

Varying PSF : fast PSF Interpolation



$$z_{conv} = \sum_{m=0}^{M-1} \omega_m \odot (x * p_m) \quad (6)$$

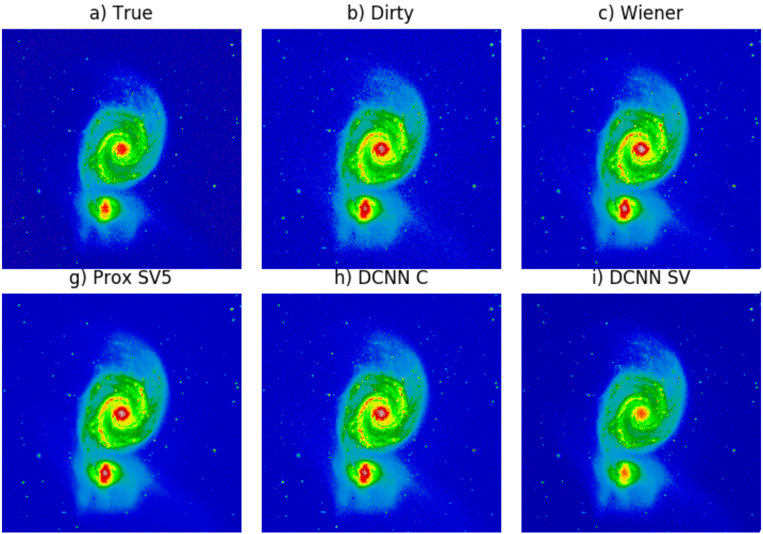
-
- Bilinear PSF interpolation for a simple 2 by 2 grid.
- FFT can still be used for fast convolution of each base PSF.

Varying PSF : numerical performances

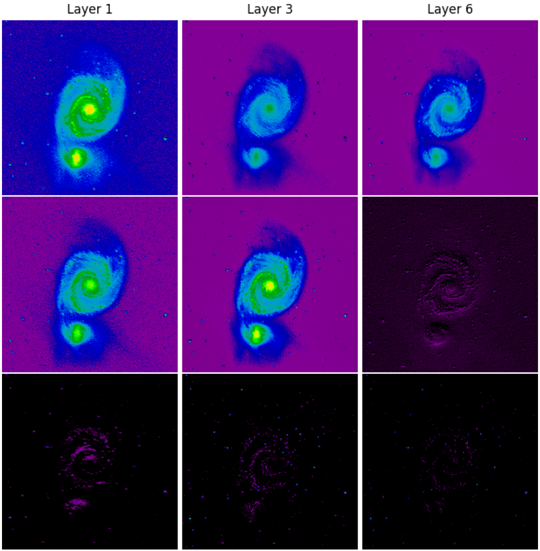
Image \ Method	Wiener	RL	RL SV9	Prox	Prox SV3	Prox SV5	DCNN C	DCNN SV
M31 : 18.60	18.61	18.45	18.59	18.74	18.74	18.75	18.28	**23.40**
Hoag : 32.66	33.37	32.61	32.91	33.62	33.58	33.62	32.26	**40.45**
M51a : 29.32	29.52	29.43	29.43	29.75	29.75	29.76	29.03	**39.02**
M81 : 33.50	34.42	33.27	33.79	34.42	34.38	34.44	32.83	**35.82**
M101 : 32.52	33.21	32.46	32.71	33.48	33.46	33.50	31.91	**39.35**
M104 : 32.30	33.01	31.38	32.45	33.16	33.12	33.17	31.16	**35.15**
Avg. PSNR (dB)	30.35	29.60	29.98	30.53	30.50	30.54	29.25	**35.53**
Avg. time (s)	**0.36**	1.41	133.20	1510.87	11381.24	24054.04	1.64	1.60

- Best PSNR for DCNN methods, same complexity as constant PSF.
- Only slight advantage to the PSF interpolation because of limited sampling.
- DCNN SV learn to simultaneously estimate the PSF and reconstruct a patch.
- Other kind of invariance can be incoded in dataset (misalignement,wavefront,...).

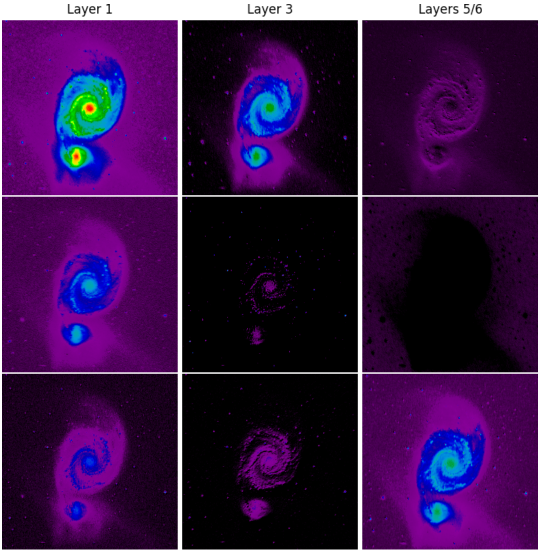
Varying PSF : Visual comparison



Constant PSF : Model Interpretation



Varying PSF : Model Interpretation



Conclusion

Conclusion

Astronomical image reconstruction with DCNN [Flamary, 2017]

- Relatively low processing time.
- Linear complexity w.r.t. number of pixels.
- Filter interpretability.
- One-time solving of an optimization problem.
- Robustness to different PSF (if learned).

What next ?

- Residual nets for a more multiscale reconstruction.
- Fast image reconstruction for adaptive optics.
- Reconstructing hyperspectral images.

References

References I

Ammanouil, R., Ferrari, A., Flamary, R., Ferrari, C., and Mary, D. (2017). **Multi-frequency image reconstruction for radio-interferometry with self-tuned regularization parameters.** *CoRR*, abs/1703.03608.

Condat, L. (2014). **A generic proximal algorithm for convex optimization—application to total variation minimization.** *IEEE Signal Processing Letters*, 21(8) :985–989.

Deguignet, J., Ferrari, A., Mary, D., and Ferrari, C. (2016). **Distributed multi-frequency image reconstruction for radio-interferometry.** In *Signal Processing Conference (EUSIPCO), 2016 24th European*, pages 1483–1487. IEEE.

Denis, L., Thiébaut, É., Soulez, F., Becker, J.-M., and Mourya, R. (2015). **Fast Approximations of Shift-Variant Blur.** *International Journal of Computer Vision*, 115(3) :pp 253–278.

References II

Flamary, R. (2017). **Astronomical image reconstruction with convolutional neural networks.**

Glorot, X. and Bengio, Y. (2010). **Understanding the difficulty of training deep feedforward neural networks.** In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Glorot, X., Bordes, A., and Bengio, Y. (2011). **Deep sparse rectifier neural networks.** In *Aistats*, volume 15, page 275.

Högbom, J. A. (1974). **Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines.** *Astronomy and Astrophysics Supplement*, 15 :417.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). **Deep learning.** *Nature*, 521(7553) :436–444.

LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). ***Convolutional networks and applications in vision*, pages 253–256.**

Lucy, L. B. (1974). **An iterative technique for the rectification of observed distributions.** *The astronomical journal*, 79 :745.

McCann, M. T., Jin, K. H., and Unser, M. (2017). **A review of convolutional neural networks for inverse problems in imaging.** *arXiv preprint arXiv :1710.04011*.

Orieux, F., Giovannelli, J.-F., and Rodet, T. (2010). **Bayesian estimation of regularization and point spread function parameters for wiener–hunt deconvolution.** *JOSA A*, 27(7) :1593–1607.

Richardson, W. H. (1972). **Bayesian-based iterative method of image restoration∗.** *J. Opt. Soc. Am.*, 62(1) :55–59.

Vũ, B. C. (2013). **A splitting algorithm for dual monotone inclusions involving cocoercive operators.** *Advances in Computational Mathematics*, 38(3) :667–681.

Xu, L., Ren, J. S., Liu, C., and Jia, J. (2014). **Deep convolutional neural network for image deconvolution.** In *Advances in Neural Information Processing Systems*, pages 1790–1798.