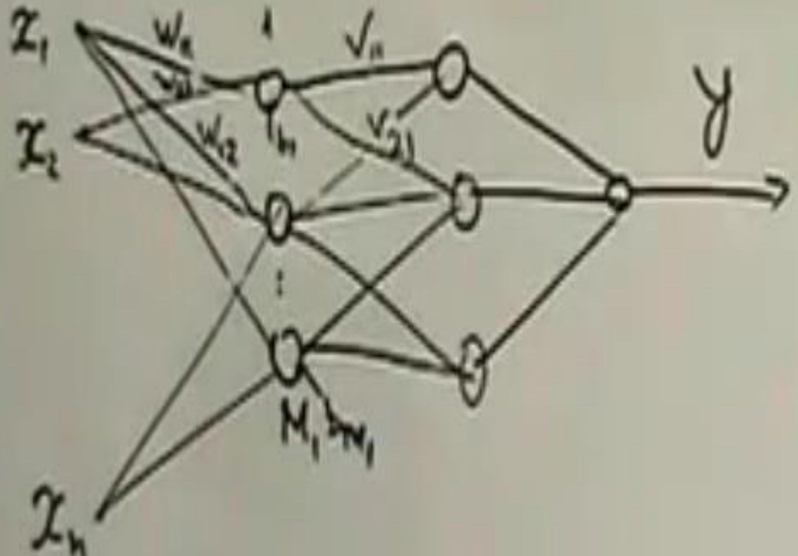
Compressed Sensing and Computed Tomography with Deep Neural Networks

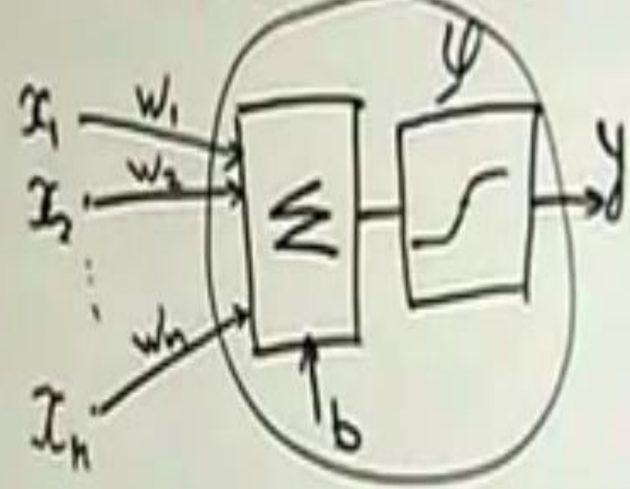
David Boublil, Michael Zibulevsky and Michael Elad

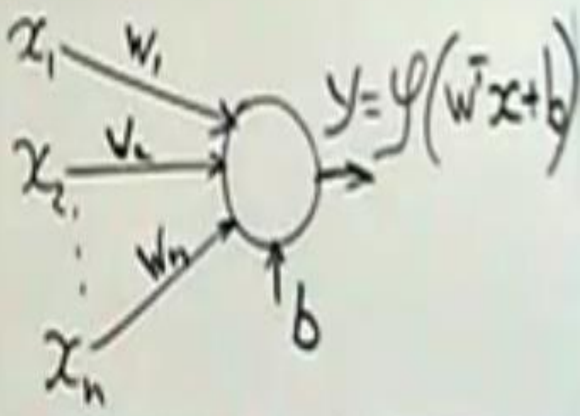
Technion, Computer Science Department

Neural Networks



Neuron model



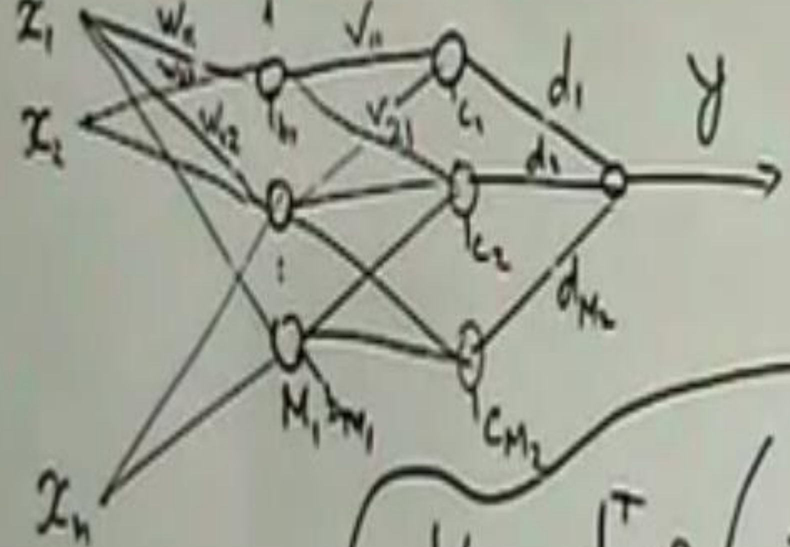


Neural Networks

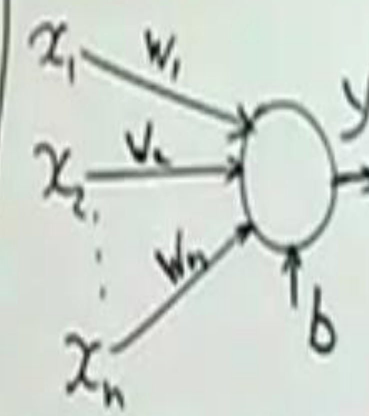
$$u = w^T x$$

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_{M_1} \end{bmatrix}$$

$$\varphi(u) \triangleq \begin{bmatrix} \varphi(u_1) \\ \vdots \\ \varphi(u_{M_1}) \end{bmatrix}$$



$$\boxed{y = d^T \varphi\left(v^T \varphi(w^T x + b) + c\right)}$$



$$y = \varphi(w^T x + b)$$

$$y = f(x; w, b, v, c, d) = d^T \varphi\left(v^T \varphi(w^T x + b) + c\right)$$

Training examples $\left\{x^{(k)}, y^{(k)}\right\}_{k=1}^{K}$

$$\min_{w,b,v,c,d} \left\{ E = \sum_{k=1}^{K} \left(f(x^{(k)}; w, b, v, c, d) - y^{(k)} \right)^2 \right\}$$

Photon detection in positron emission tomography using neural network [12]

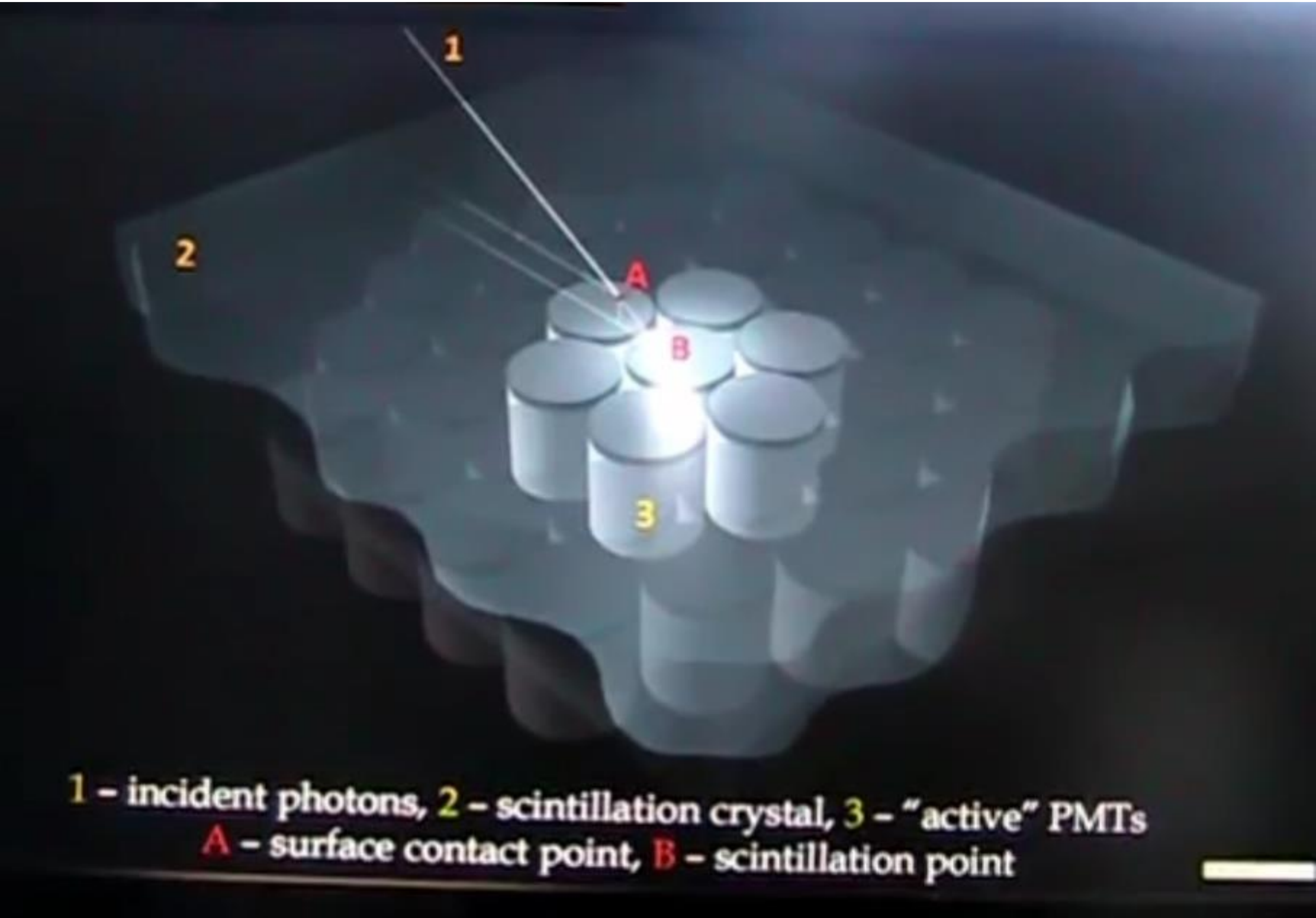
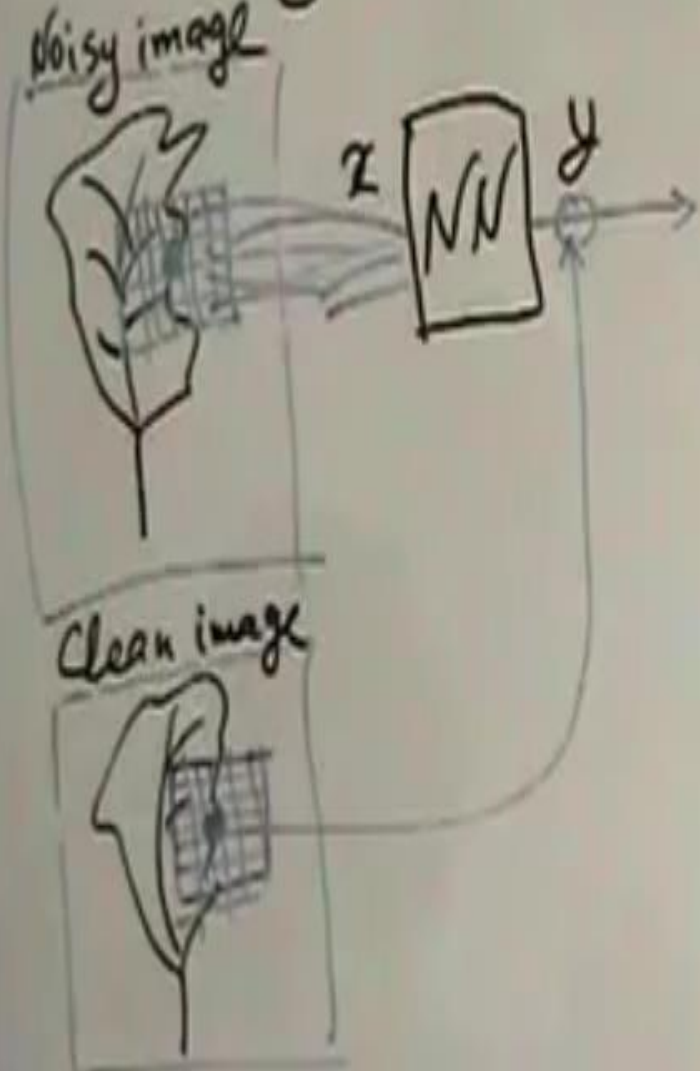
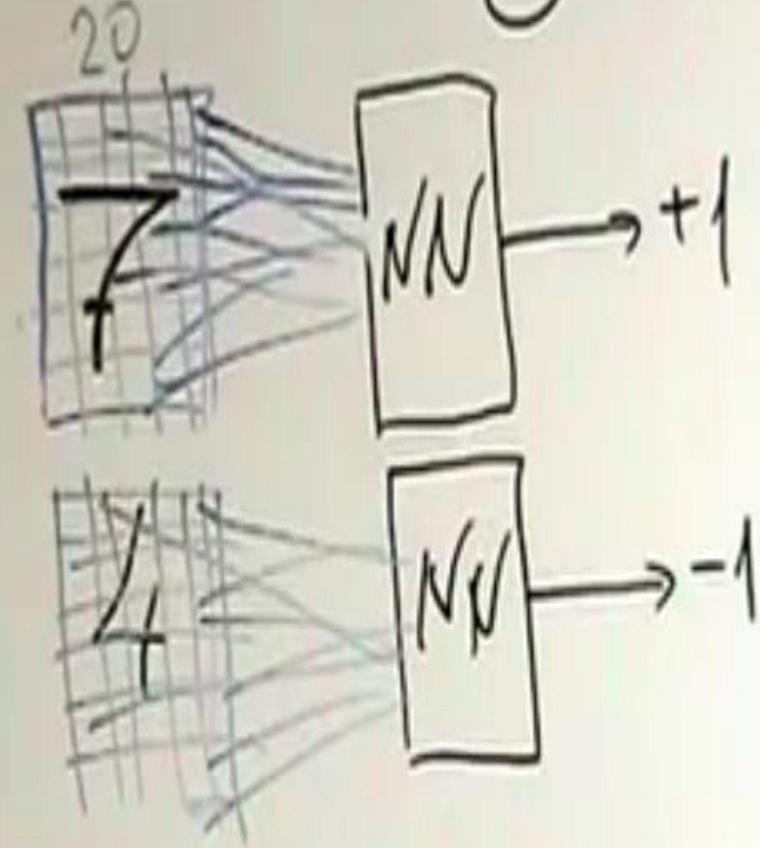


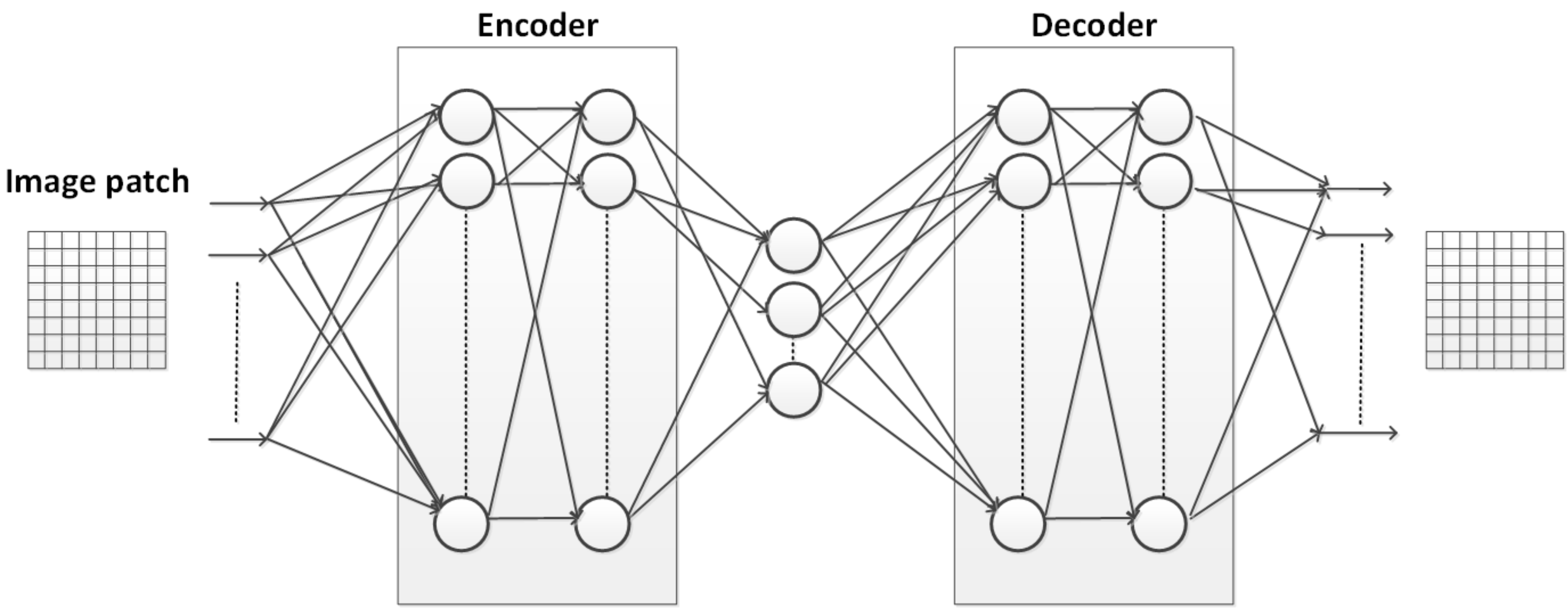
Image denoising



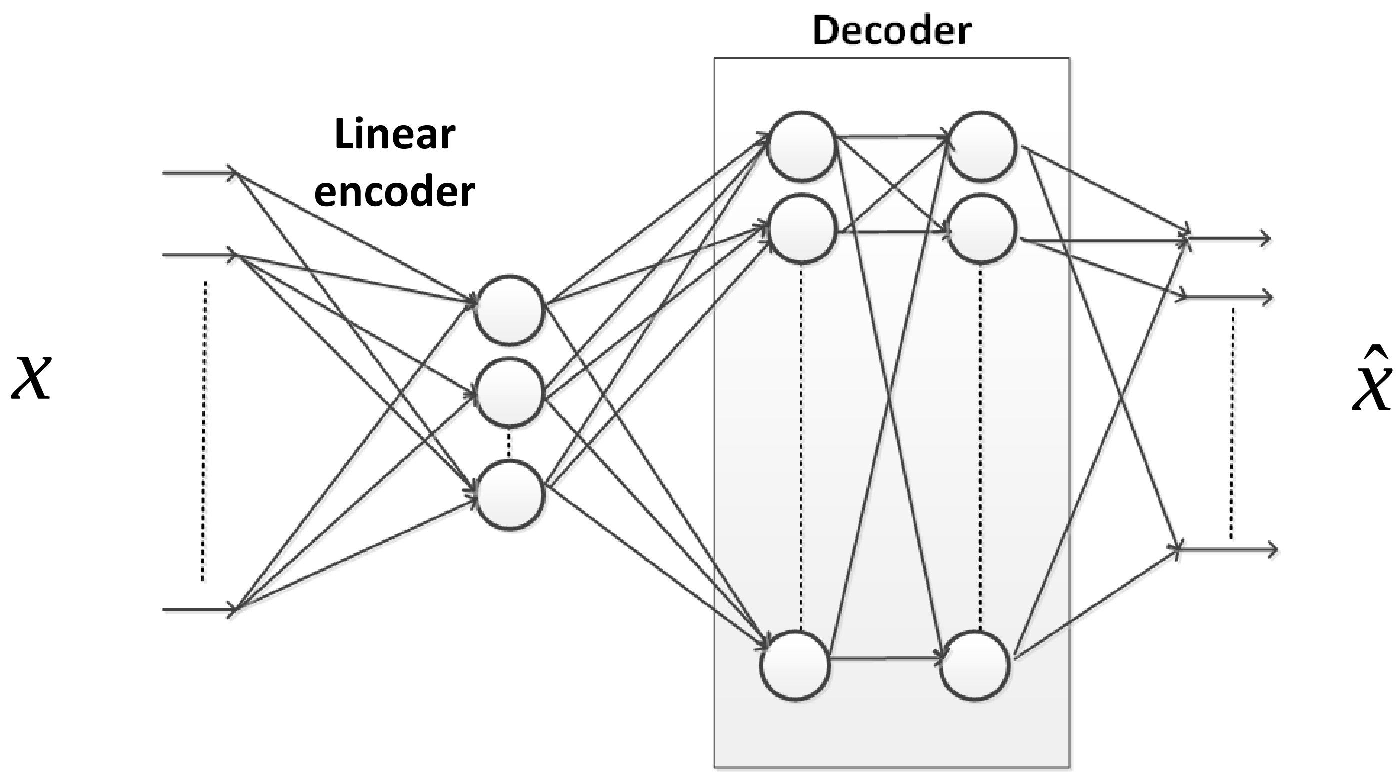
Pattern recognition



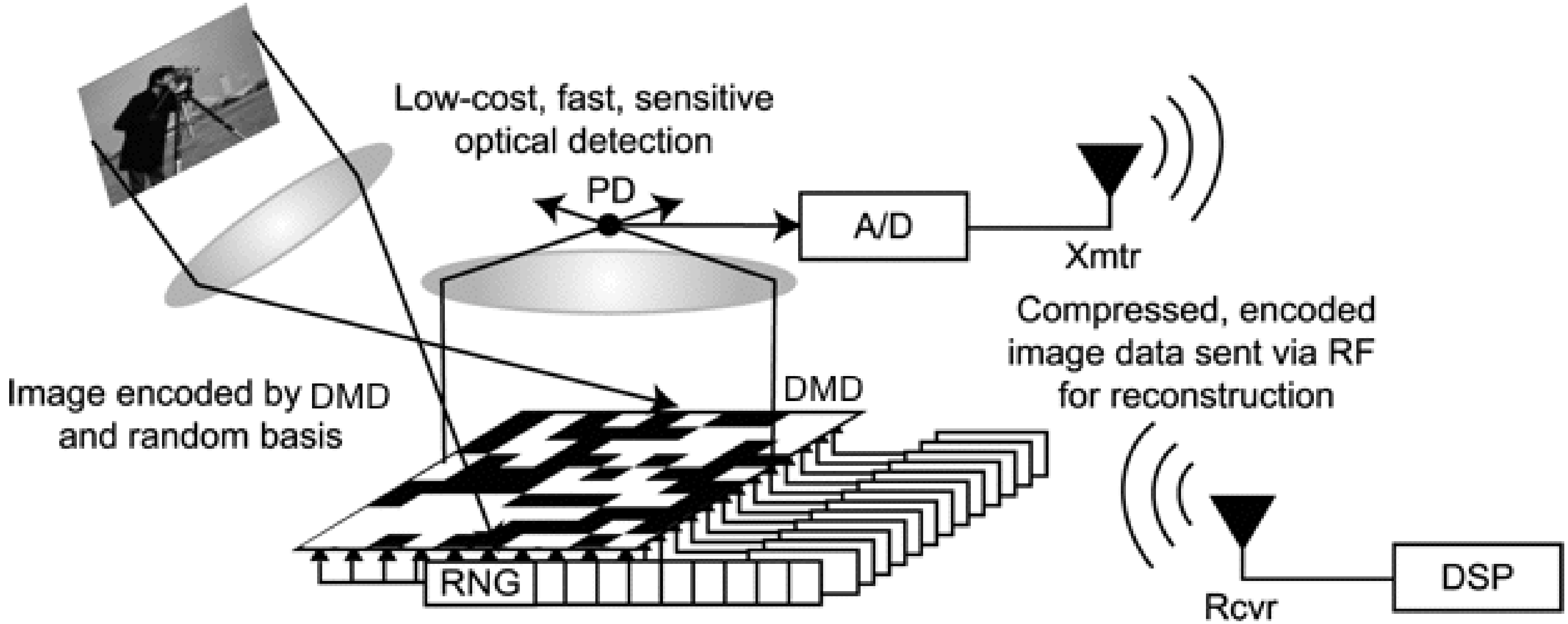
Nonlinear compression



Linear compression (Compressed sensing)



Linear compression (Compressed sensing)



A New Single-Pixel Camera, http://dsp.rice.edu/cscamera

CS using Neural Network: Training Set

- Some image examples from the training set:

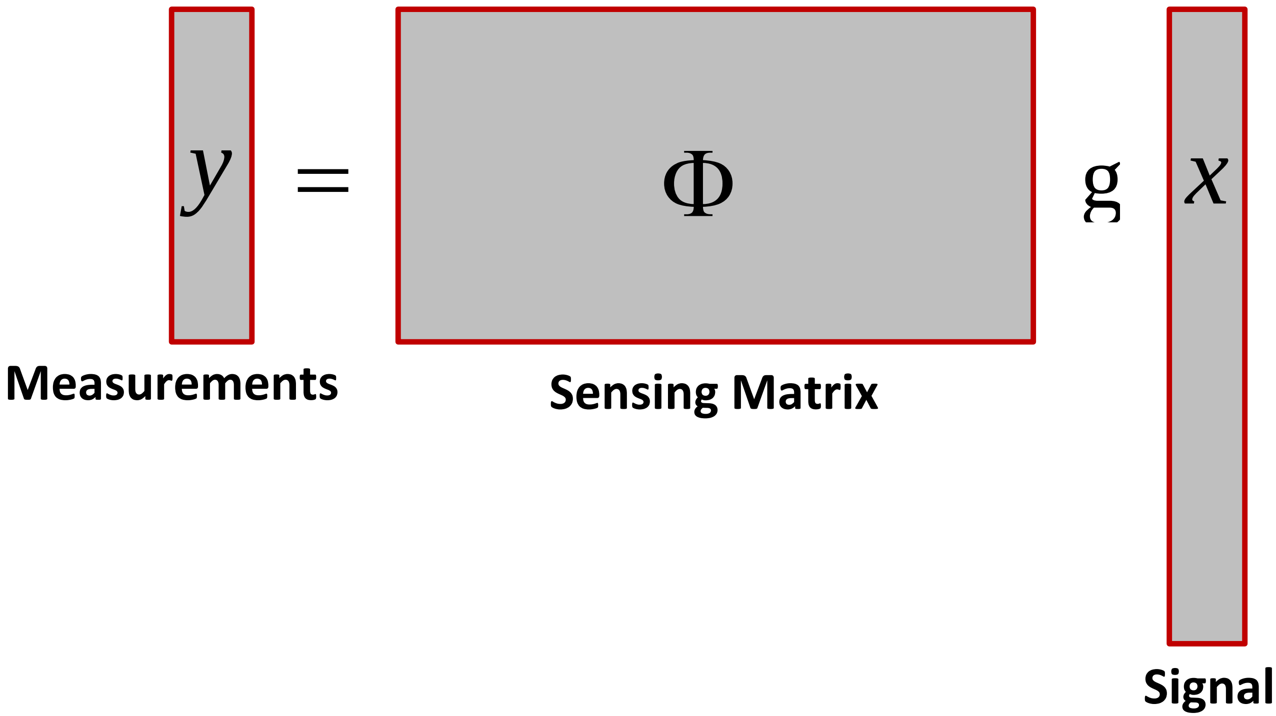
CS using Neural Network – Results

- Hidden layers: 12
- Activation: Tanh
- Redundancy factor: 2
- Block size: 16x16.
- Training examples: 500k

CS using Neural Network – Results

CS using Neural Network – Results

Classical Compressed sensing



- **Recovery:**

$$\min_{x \in X} \|\Phi x - y\|_2^2$$

Set X reflects prior knowledge about the signal

Classical Compressed sensing

- **Recovery:** $$\min_{x \in X} \left\| \Phi x - y \right\|_2^2$$

- **Set X reflects prior knowledge about the signal:**

  1. **Limited Total Variation (TV):**

  $$\int \left\| \nabla x \right\| dx \le a$$

  2. **Sparse representability:**

  $$x = \sum_i c_i \psi_i \ , \ \left\| c \right\|_0 \le b$$

Constraints may be put as a penalty term

Block Compressed Sensing using Smoothed projected Landweber (SPL-BCS)

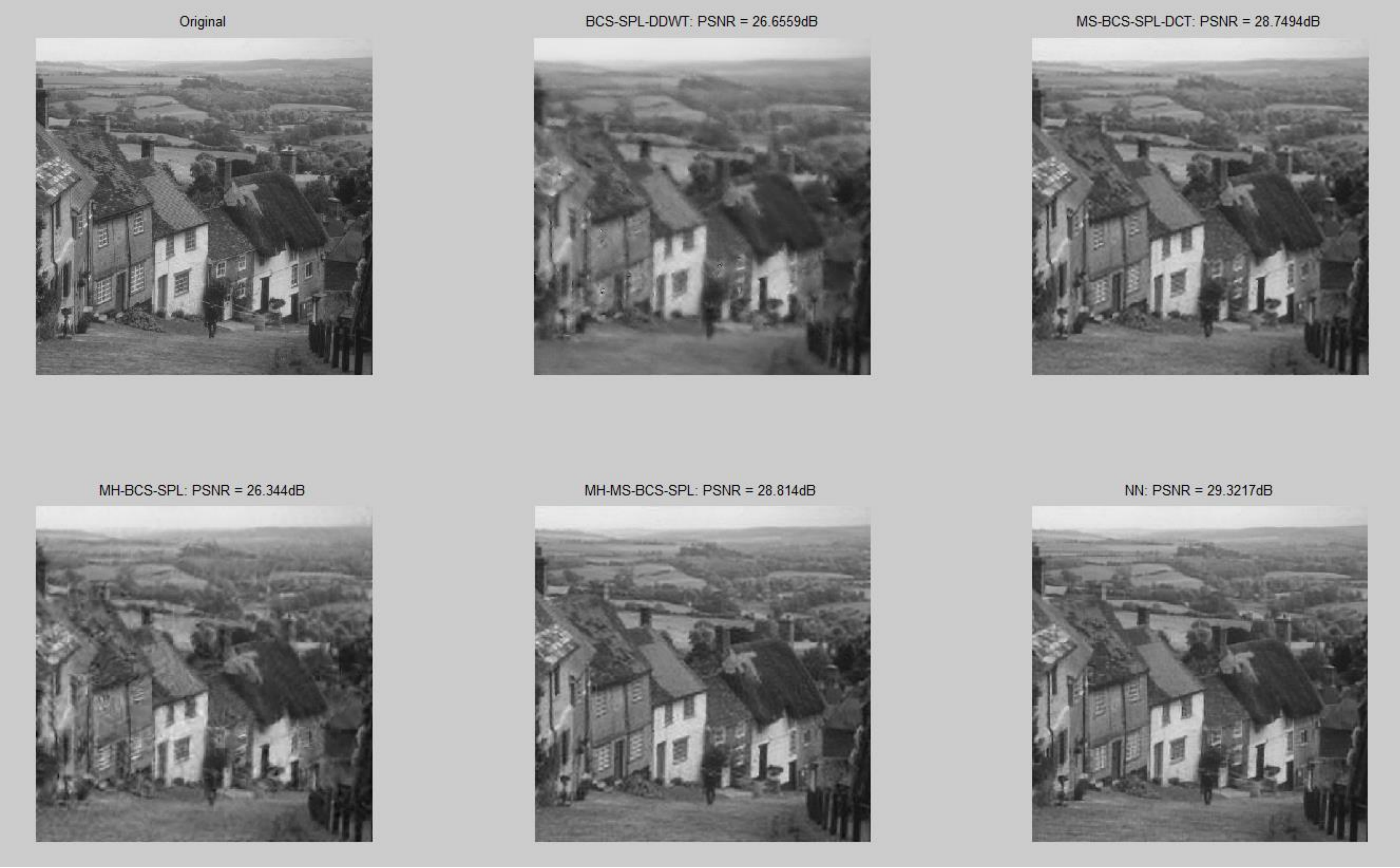
L. Gan , J. E. Fowler and others…

- **Transforms: DCT, Wavelet (DWT), Contourlet (CT), Dual tree wavelet (DDWT)…**
- **Initialization of each block:** $y_i = \Phi_B x_i$
- **The algorithm :**

$$
\begin{aligned}
&\texttt{function}\ \mathbf{x}^{(i+1)} = \mathrm{SPL}(\mathbf{x}^{(i)}, \mathbf{y}, \mathbf{\Phi}_B, \mathbf{\Psi}, \lambda) \\
&\quad \hat{\mathbf{x}}^{(i)} = \mathrm{Wiener}(\mathbf{x}^{(i)}) \\
&\quad \texttt{for each block}\ j \\
&\quad\quad \hat{\hat{\mathbf{x}}}_j^{(i)} = \hat{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_B^T(\mathbf{y} - \mathbf{\Phi}_B \hat{\mathbf{x}}_j^{(i)}) \\
&\quad \check{\check{\mathbf{x}}}^{(i)} = \mathbf{\Psi} \hat{\hat{\mathbf{x}}}^{(i)} \\
&\quad \check{\mathbf{x}}^{(i)} = \mathrm{Threshold}(\check{\check{\mathbf{x}}}^{(i)}, \lambda) \\
&\quad \bar{\mathbf{x}}^{(i)} = \mathbf{\Psi}^{-1} \check{\mathbf{x}}^{(i)} \\
&\quad \texttt{for each block}\ j \\
&\quad\quad \mathbf{x}_j^{(i+1)} = \bar{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_B^T(\mathbf{y} - \mathbf{\Phi}_B \bar{\mathbf{x}}_j^{(i)})
\end{aligned}
$$

- **Further improvement by multi-scale (MS-BCS-SPL) and multi-hypothesis (MH-BCS-SPL) extensions.**

Comparison with other Block-CS algorithms



Comparison with other Block-CS algorithms

	Compression rate				
	8%	10%	20%	30%	40%
Methods	Lena				
BCS-SPL-DDWT	26.45	27.94	30.94	33.39	35.38
MS-BCS-SPL	**31.01**	31.83	35.19	37.58	39.08
MH-BCS-SPL	28.04	28.73	33.05	35.44	37.14
MH-MS-BCS-SPL	30.87	**31.99**	**35.58**	**37.86**	40.06
BCS with NN	30.88	30.93	34.73	37.54	**40.27**

	Compression rate				
	8%	10%	20%	30%	40%
Methods	Mandrill				
BCS-SPL-DDWT	20.2	20.63	21.68	22.7	23.93
MS-BCS-SPL	21.2	21.37	22.9	24.43	25.36
MH-BCS-SPL	19.86	20.01	21.93	23.57	25.02
MH-MS-BCS-SPL	21.33	21.66	23.09	24.63	25.67
BCS with NN	**21.65**	**21.89**	**23.7**	**25.21**	**26.7**

	Compression rate				
	8%	10%	20%	30%	40%
Methods	Goldhill				
BCS-SPL-DDWT	26.65	27.12	29	30.73	32.04
MS-BCS-SPL	28.74	29.25	31.42	33.32	34.34
MH-BCS-SPL	26.34	27.1	30.12	32.41	34.09
MH-MS-BCS-SPL	28.81	29.3	31.75	33.63	35.57
BCS with NN	**29.32**	**29.82**	**32.5**	**34.58**	**37.12**

	Compression rate				
	8%	10%	20%	30%	40%
Methods	Boat				
BCS-SPL-DDWT	24.24	25.3	27.96	30	31.57
MS-BCS-SPL	26.91	27.53	30.47	32.52	33.9
MH-BCS-SPL	24.25	25.53	29.11	31.72	33.45
MH-MS-BCS-SPL	26.89	27.57	30.69	32.73	34.88
BCS with NN	**27.6**	**27.8**	**31**	**33.23**	**35.46**

Comparison with other Block-CS algorithms

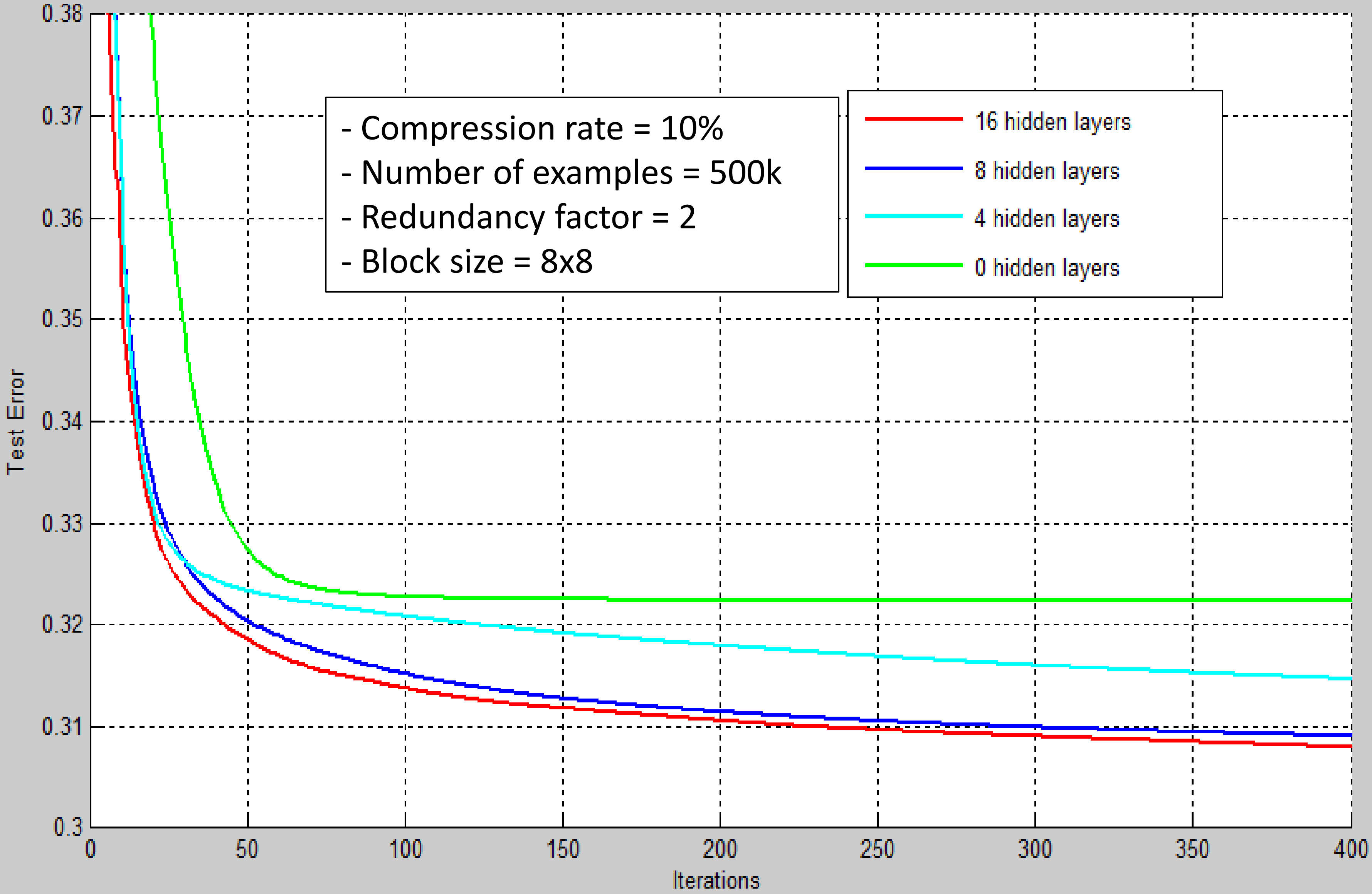
	Compression rate				
	8%	10%	20%	30%	40%
Methods	Average on 10 images				
BCS-SPL-DDWT	24.01	24.95	27.16	29.08	30.7
MS-BCS-SPL	26.75	27.35	30.05	32.04	33.4
MH-BCS-SPL	24.56	25.9	29.53	31.72	33.32
MH-MS-BCS-SPL	26.95	**27.69**	**30.75**	32.65	35.01
BCS with NN	**27.27**	27.43	30.44	**32.81**	**35.22**

Methods	Run Time (sec) (Compression rate 10%)
BCS-SPL-DDWT	15.59
MS-BCS-SPL	8.26
MH-BCS-SPL	49.93
MH-MS-BCS-SPL	24.49
BCS with NN	0.34

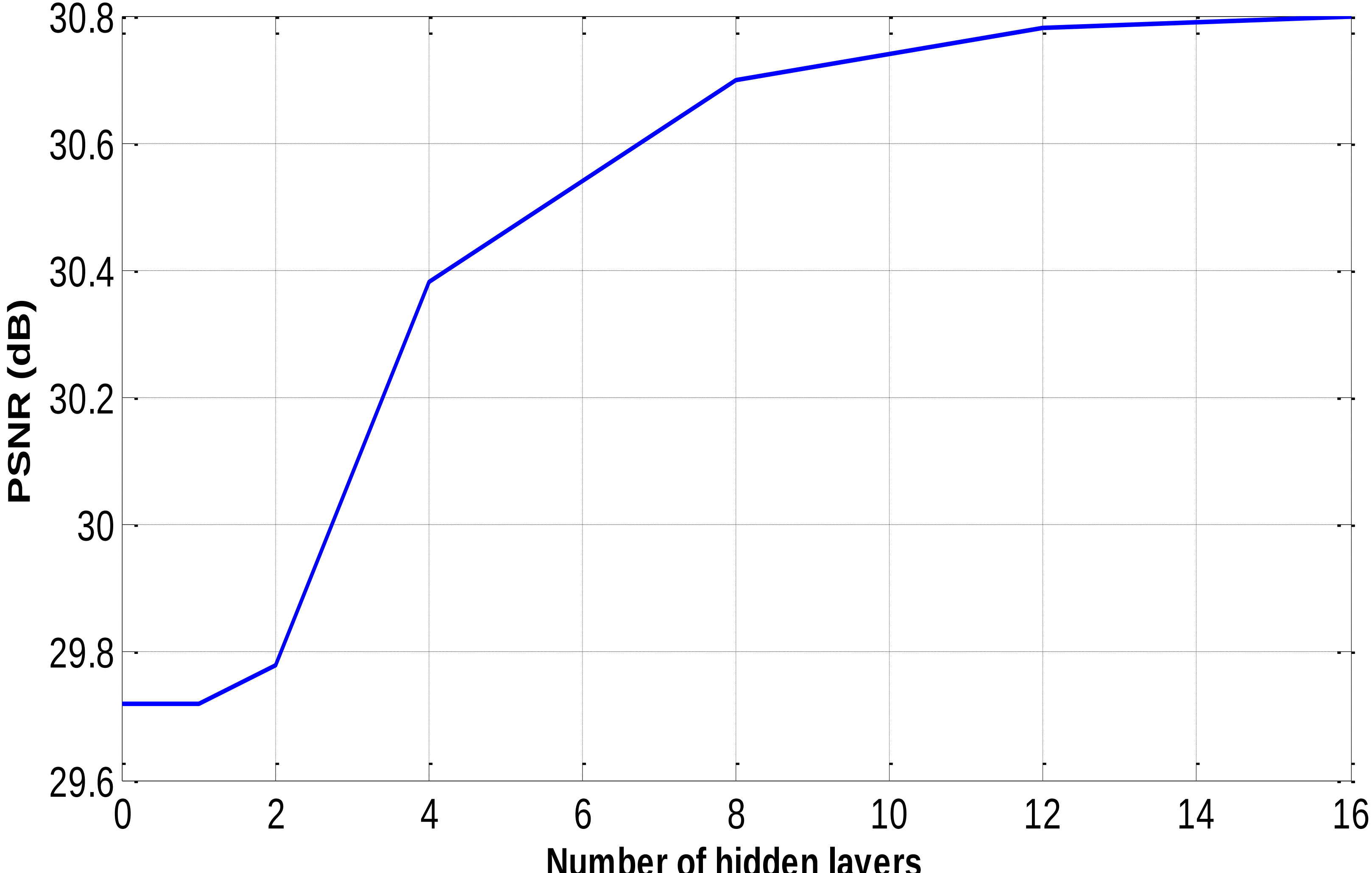
Comparison with Romberg Algorithm [11]

- It is generally applied on the whole image, for this comparison we have used it on each block to get a fair comparison.
- Compression Rate = 5%

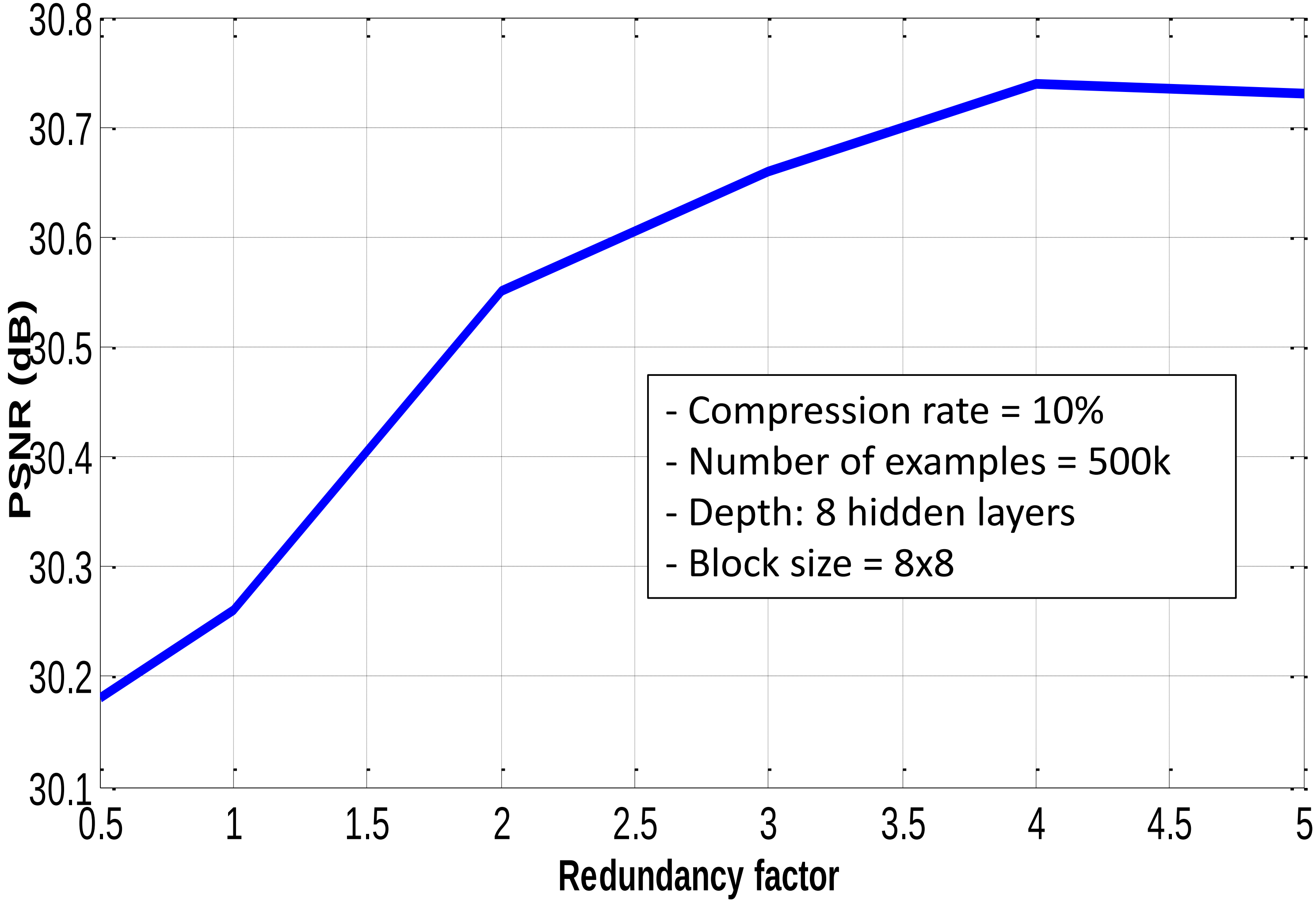
Influence of Network depth



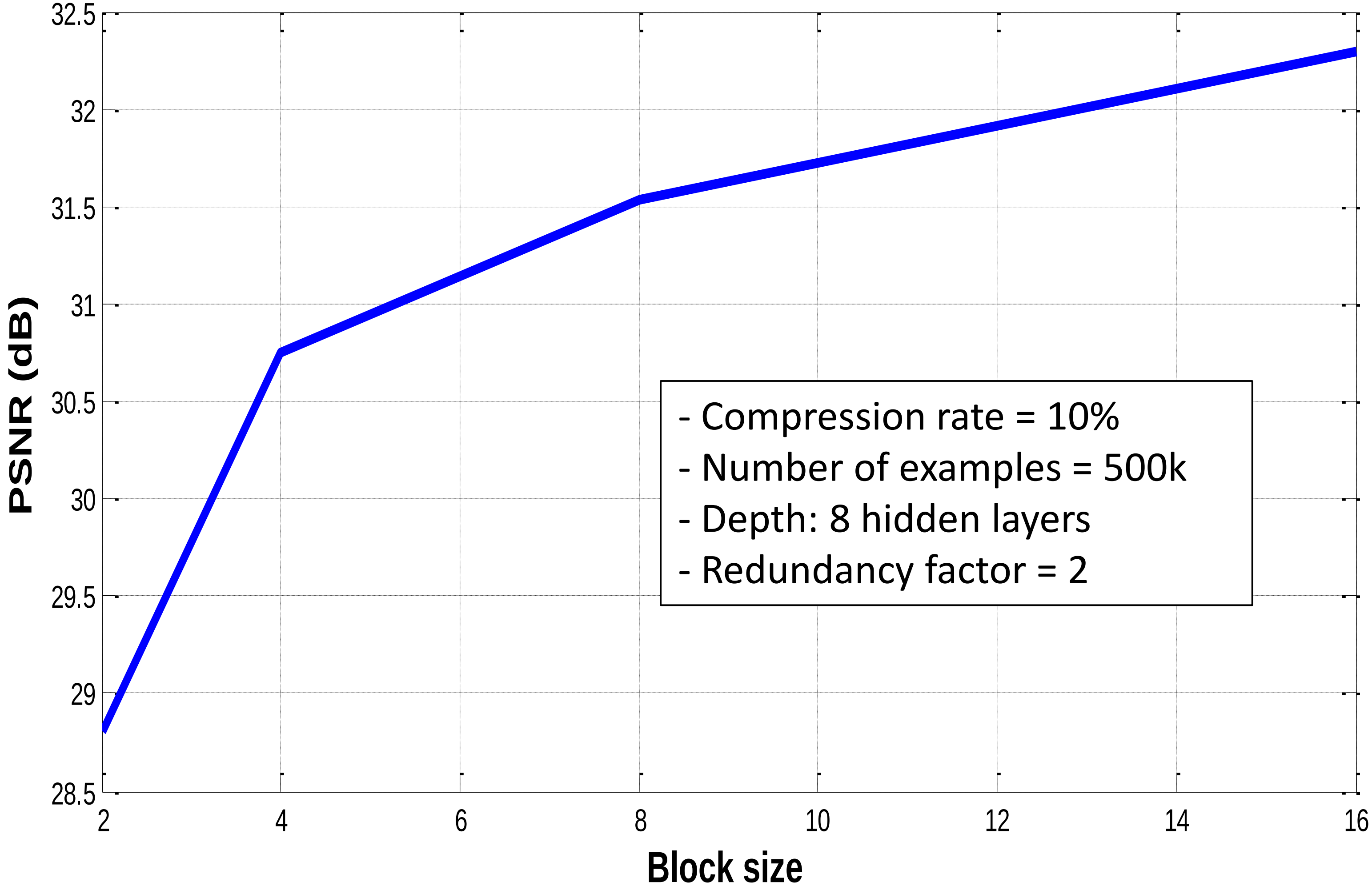
Influence of Network depth: Peppers



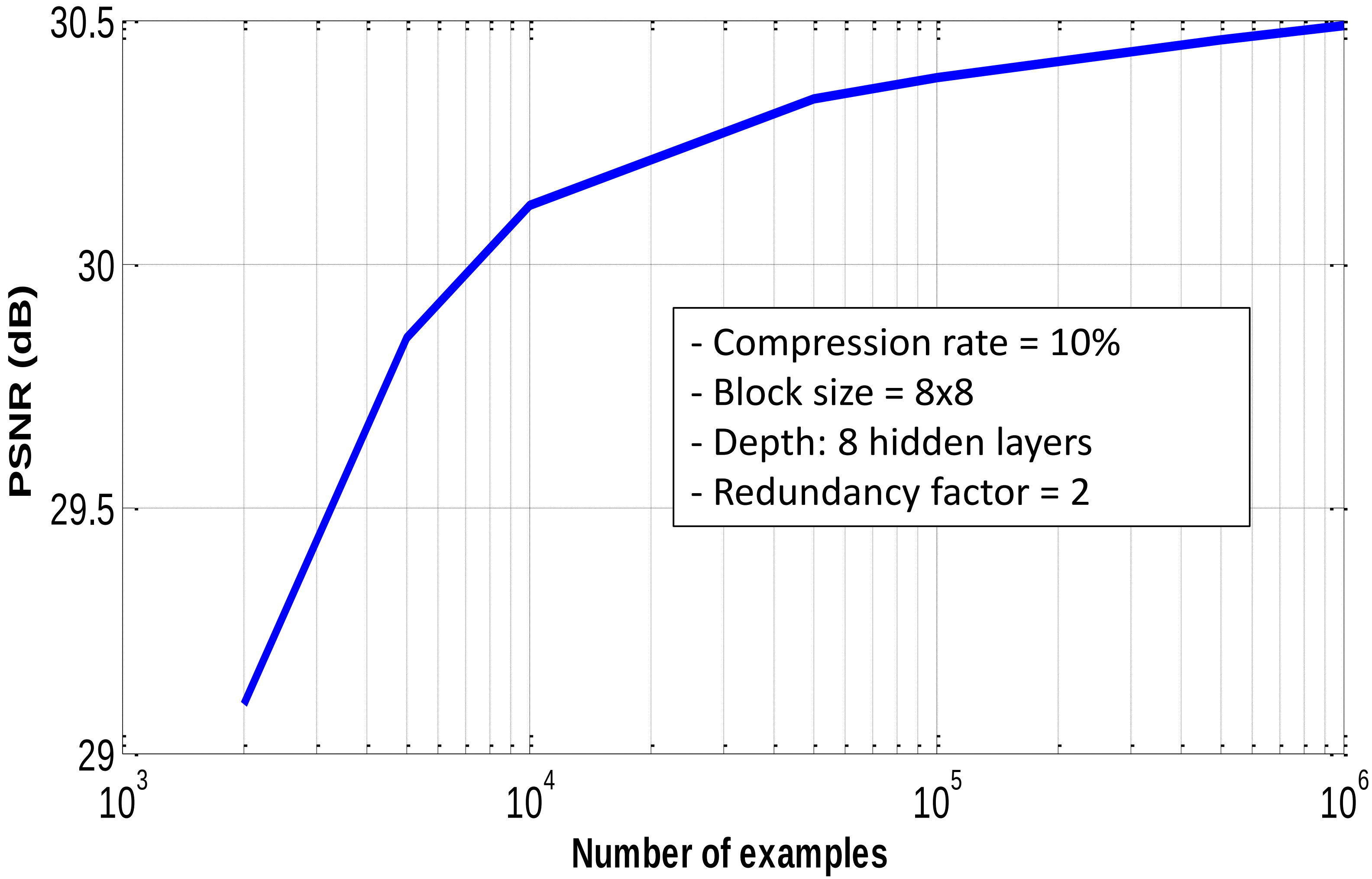
Influence of redundancy factor (layer width): Lena



Influence of block size: Boat



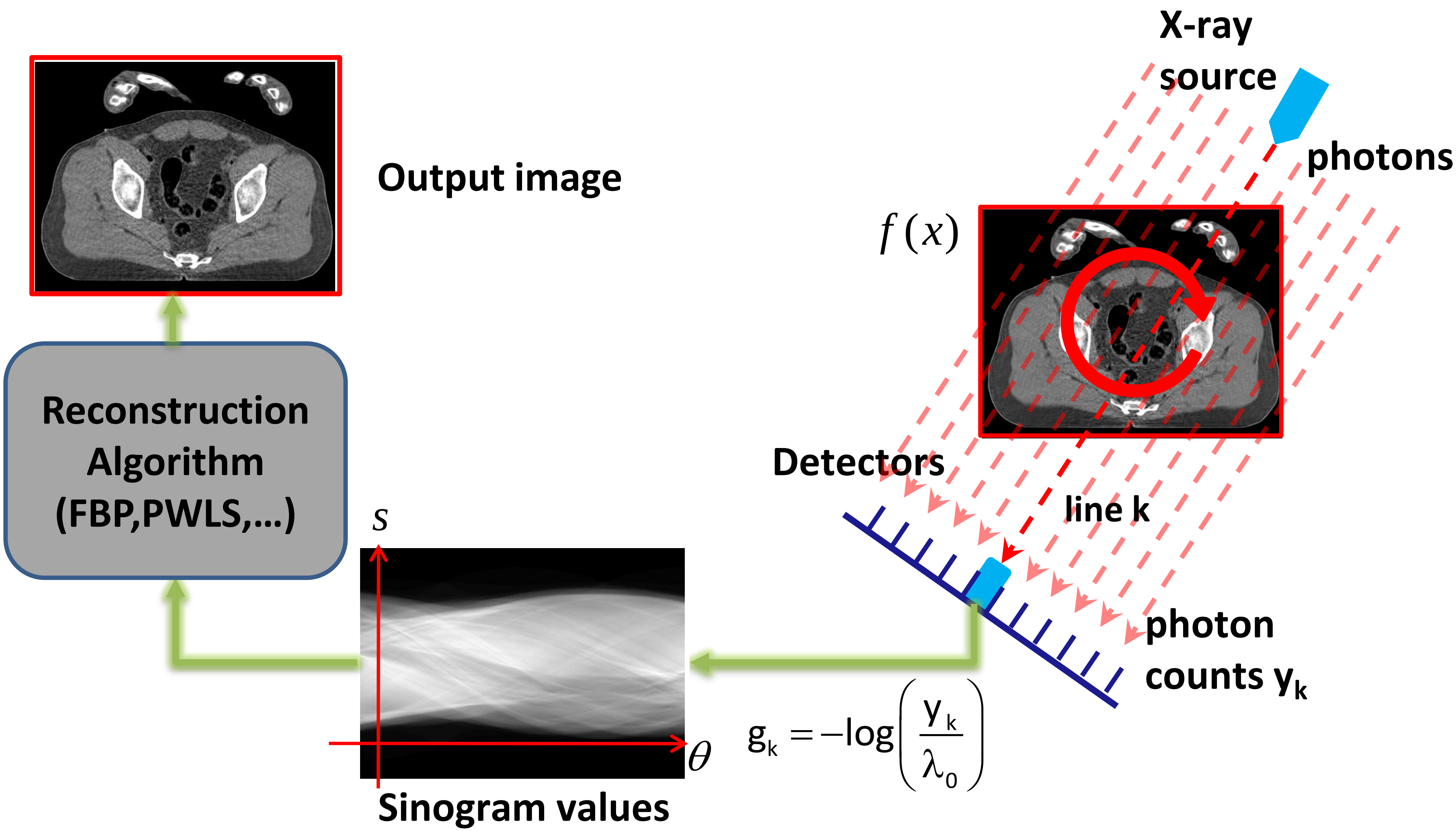
Influence of number of training examples



Future Directions

- Use large networks and training sets to achieve patch size of 32x32 and bigger
- Global sensing / reconstruction using multiresolution neural networks
- Get back to nonlinear compression

Computed Tomography



Reconstruction Algorithms

❑ Filtered Back-Projection (FBP) => Linear Operator

$$T_{FBP} = \mathrm{R}^{*} F_{low} F_{Ram-Lak}$$

Adjoint of Radon Transform (Back-projection)

LPF –prevents noise amplification at high frequencies

1D convolution filter - Applied to each projection

❑ Penalized Weighted Least Square (PWLS) => Iterative algorithm

$$\tilde{f} = \arg\min_{f} \left\| \log(y) - Af \right\|_{D} + \beta R(f)$$

Measured counts – projections

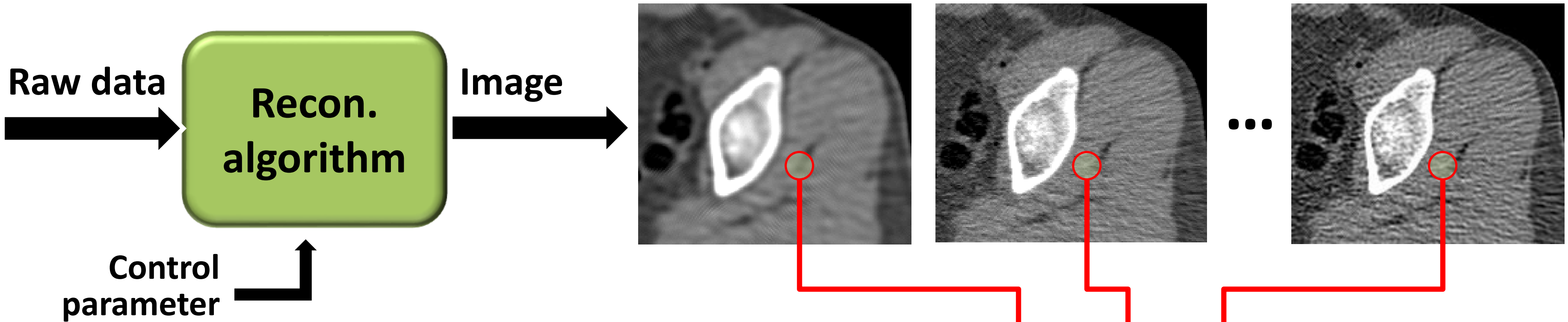
Radon transform approximation- Models scan process

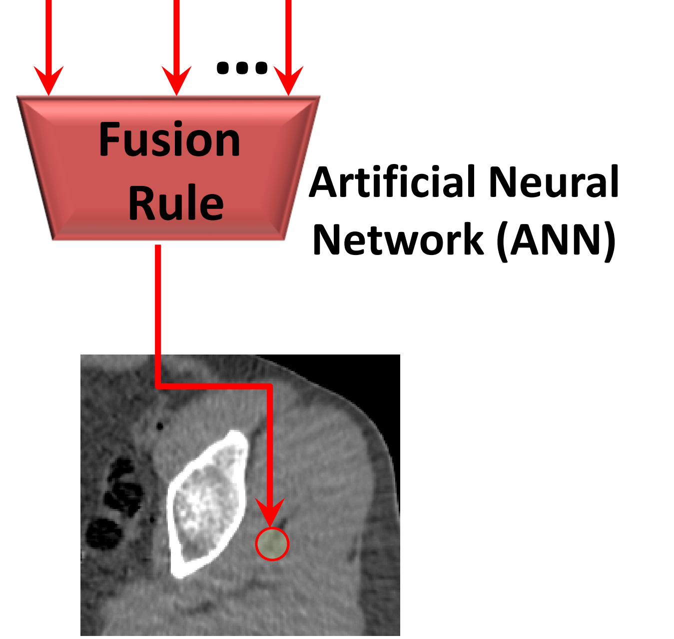
Prior on clean CT image

Main Themes of Our Work [13]

Reducing Radiation Dose By Learning to Fuse Several Output Images

Fusion Over a Smoothing Parameter





The Algorithm:

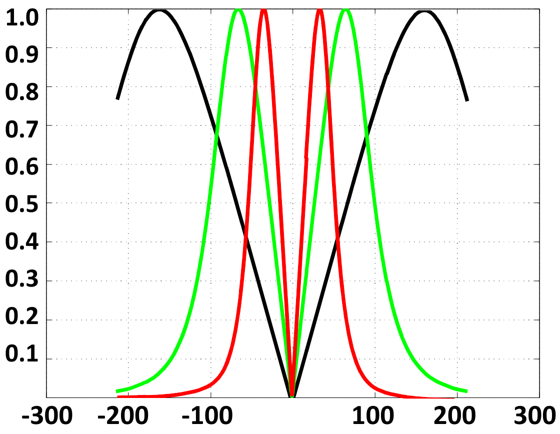
- ❑ Sweep the variance-resolution tradeoff
- ❑ Extract pixel neighborhood from each version
- ❑ Build a decision rule to perform the local fusion

Fusion Rule:

Use a regression to build one automatically, with an Artificial Neural Network (ANN)

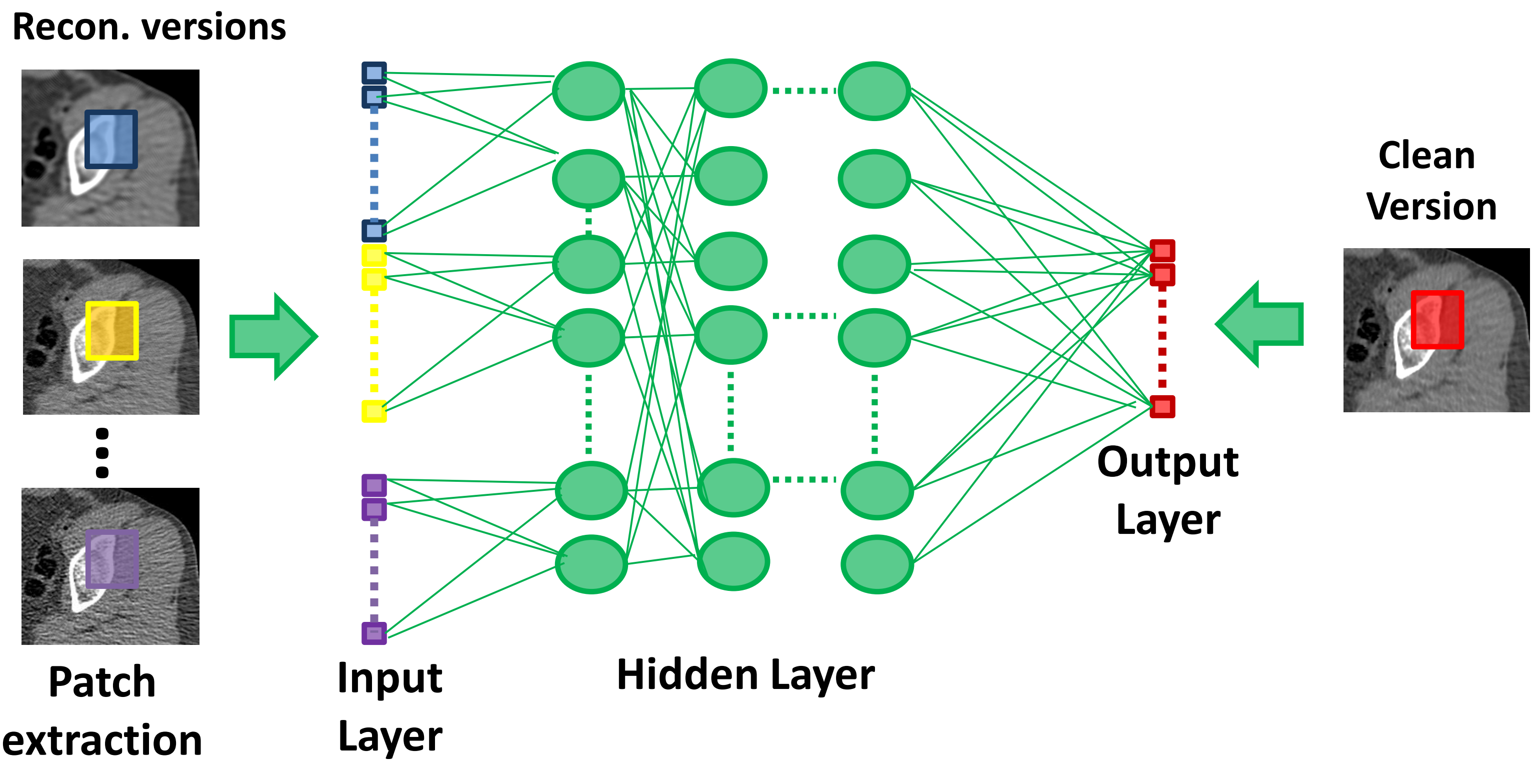
Fusion: Which Images to Use?

FBP algorithm: sweep the cut-off frequency of the low-pass sinogram filter, and collect few images with different resolution-variance trade-off



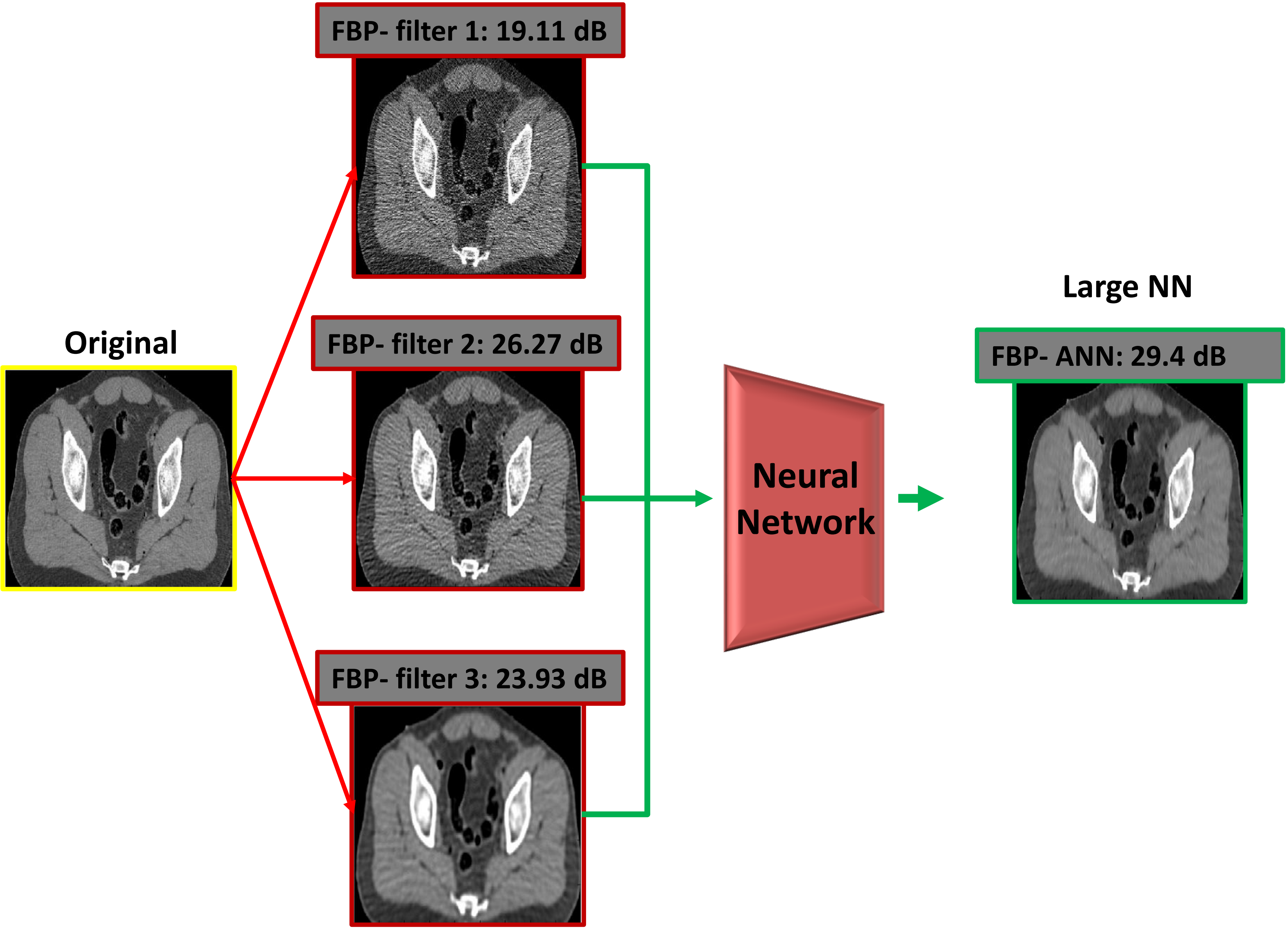
PWLS algorithm: perform the regular reconstruction while collecting versions along the iterations or sweeping different penalized weights β .

Neural Network (NN) Architecture

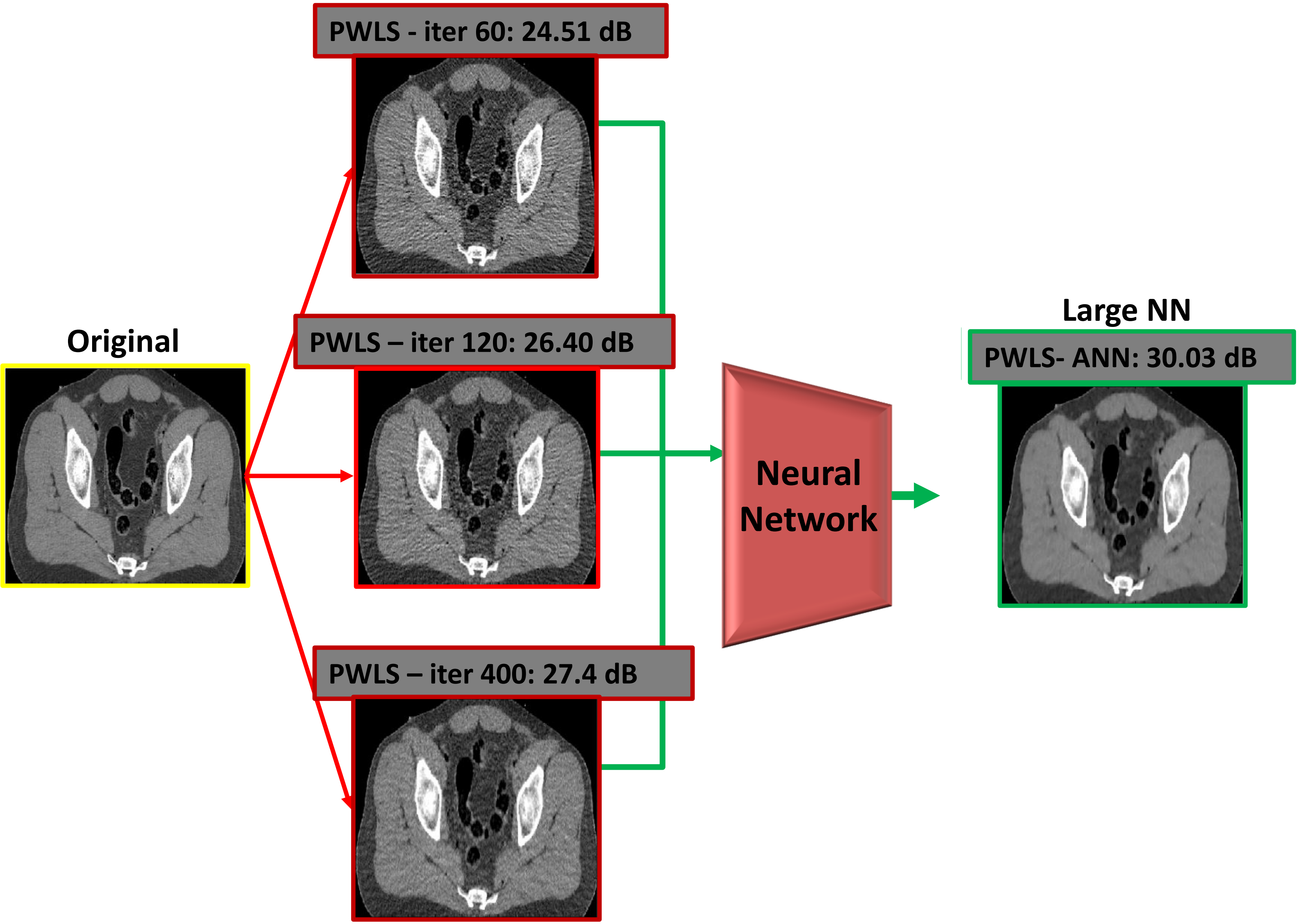


- Simple fully connected Neural Network
- Activation function is an hyperbolic tangent
- Caffe software was used to train the network
- Training data (CT images) is taken from Visible Human Project

Empirical results- ANN FBP



Empirical results- ANN PWLS



Thank You !!!

Bibliography (1)

[1]M. Mougeot, R. Azencott, B. Angeniol, Image compression with back propagation: improvement of the visual restoration using different cost functions, Neural Networks 4 (4) (1991) 467-476.

[1a] Elad, M. (2007). Optimized projections for compressed sensing. *Signal Processing, IEEE Transactions on*, *55*(12), 5695-5702.

[1b] Chang, H. S., Weiss, Y., & Freeman, W. T. (2009). Informative sensing of natural images. In *Image Processing (ICIP), 2009 16th IEEE International Conference on* (pp. 3025-3028). IEEE.

[2] Duarte-Carvajalino, J. M., & Sapiro, G. (2009). Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization.*Image Processing, IEEE Transactions on*, *18*(7), 1395-1408.

[3] Xu, J., Pi, Y., & Cao, Z. (2010). Optimized projection matrix for compressive sensing. *EURASIP Journal on Advances in Signal Processing*, *2010*, 43.

[4]Chen, W., Rodrigues, M. R., & Wassell, I. J. (2012). On the use of unit-norm tight frames to improve the average mse performance in compressive sensing applications. *Signal Processing Letters, IEEE*, *19*(1), 8-11.

[5] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.

[6]Gan, L. (2007, July). Block compressed sensing of natural images. In *Digital Signal Processing, 2007 15th International Conference on* (pp. 403-406). IEEE.

Bibliography (2)

[7]Mun, S., & Fowler, J. E. (2009, November). Block compressed sensing of images using directional transforms. In Image Processing (ICIP), 2009 16th IEEE International Conference on (pp. 3021-3024). IEEE.

[8] Fowler, J. E., Mun, S., & Tramel, E. W. (2011, August). Multiscale block compressed sensing with smoother projected Landweber reconstruction. InProceedings of the European signal processing conference (Vol. 564, p. 568).

[9] Chen, C., Tramel, E. W., & Fowler, J. E. (2011, November). Compressed-sensing recovery of images and video using multihypothesis predictions. InSignals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on (pp. 1193-1198). IEEE.

[10] Fowler, J. E., Mun, S., & Tramel, E. W. (2012). Block-based compressed sensing of images and video. Foundations and Trends in Signal Processing,4(4), 297-416.

[11] Romberg, J. (2008). Imaging via compressive sampling [introduction to compressive sampling and recovery via convex programming]. IEEE Signal Processing Magazine, 25(2), 14-20.

[12] Bronstein, A. M., Bronstein, M. M., Zibulevsky, M., & Zeevi, Y. Y. (2003). Optimal nonlinear line-of-flight estimation in positron emission tomography. *Nuclear Science, IEEE Transactions on*, *50*(3), 421-426.

[13] D Boublil, M Elad, J Shtok and M Zibulevsky (2015) Spatially-Adaptive Reconstruction in Computed Tomography using Neural Networks, IEEE Trans. on Biomedical Imaging, accepted

[14] Y Pfeffer and M Zibulevsky (2010) A Micro-Mirror Array based System for Compressive Sensing of Hyperspectral Data