# Medical image denoising using convolutional denoising autoencoders

Lovedeep Gondara
Department of Computer Science
Simon Fraser University
lgondara@sfu.ca

*Abstract*—**Image denoising is an important pre-processing step in medical image analysis. Different algorithms have been proposed in past three decades with varying denoising performances. More recently, having outperformed all conventional methods, deep learning based models have shown a great promise. These methods are however limited for requirement of large training sample size and high computational costs. In this paper we show that using small sample size, denoising autoencoders constructed using convolutional layers can be used for efficient denoising of medical images. Heterogeneous images can be combined to boost sample size for increased denoising performance. Simplest of networks can reconstruct images with corruption levels so high that noise and signal are not differentiable to human eye.**

*Keywords*—*Image denoising, denoising autoencoder, convolutional autoencoder*

## I. INTRODUCTION

Medical imaging including X-rays, Magnetic Resonance Imaging (MRI), Computer Tomography (CT), ultrasound etc. are susceptible to noise [21]. Reasons vary from use of different image acquisition techniques to attempts at decreasing patients exposure to radiation. As the amount of radiation is decreased, noise increases [1]. Denoising is often required for proper image analysis, both by humans and machines.

Image denoising, being a classical problem in computer vision has been studied in detail. Various methods exist, ranging from models based on partial differential equations (PDEs) [18], [20], [22], domain transformations such as wavelets [6], DCT [29], BLS-GSM [19] etc., non local techniques including NL-means [30], [3], combination of non local means and domain transformations such as BM3D [7] and a family of models exploiting sparse coding techniques [17], [9], [15]. All methods share a common goal, expressed as

$$z = x + y \qquad (1)$$

Where $z$ is the noisy image produced as a sum of original image $x$ and some noise $y$. Most methods try to approximate $x$ using $z$ as close as possible. In most cases, $y$ is assumed to be generated from a well defined process.

With recent developments in deep learning [14], [11], [23], [2], [10], results from models based on deep architectures have been promising. Autoencoders have been used for image denoising [24], [25], [28], [5]. They easily outperform conventional denoising methods and are less restrictive for specification of noise generative processes. Denoising autoencoders constructed using convolutional layers have better image denoising performance for their ability to exploit strong spatial correlations.

In this paper we present empirical evidence that stacked denoising autoencoders built using convolutional layers work well for small sample sizes, which are typical of medical image databases. It is in contrary to the belief that for optimal performance, very large training datasets are needed for deep architecture based models. We also show that these methods can recover signal even when noise levels are extremely high, at the point where most other denoising methods would fail.

Rest of this paper is organized as following, next section discusses related work in image denoising using deep architectures. Section III introduces autoencoders and their variants. Section IV explains our experimental set-up and details our empirical evaluation and section V presents our conclusions and directions for future work.

## II. RELATED WORK

Although BM3D [7] is considered state-of-the-art in image denoising and is a very well engineered method, Burger et al. [4] showed that a plain multi layer perceptron (MLP) can achieve similar denoising performance.

Denoising autoencoders are a recent addition to image denoising literature. Used as a building block for deep networks, they were introduced by Vincent et al. [24] as an extension to classic autoencoders. It was shown that denoising autoencoders can be stacked [25] to form a deep network by feeding the output of one denoising autoencoder to the one below it.

Jain et al. [12] proposed image denoising using convolutional neural networks. It was observed that using a small sample of training images, performance at par or better than state-of-the-art based on wavelets and Markov random fields can be achieved. Xie et al. [28] used stacked sparse autoencoders for image denoising and inpainting, which performed at par with K-SVD. Agostenelli et al. [1] experimented with adaptive multi column deep neural networks for image denoising. Built using a combination of stacked sparse autoencoders, this system was shown to be robust for different noise types.

## III. PRELIMINARIES

### A. Autoencoders

An autoencoder is a type of neural network that tries to learn an approximation to identity function using backpropagation, i.e. given a set of unlabeled training inputs $x^{(1)}, x^{(2)}, ..., x^{(n)}$, it uses

IEEE
computer
society

$$z^{(i)} = x^{(i)} \qquad (2)$$

An autoencoder first takes an input $x \in [0,1]^d$ and maps(encode) it to a hidden representation $y \in [0,1]^{d'}$ using deterministic mapping, such as

$$y = s(Wx + b) \qquad (3)$$

where $s$ can be any non linear function. Latent representation $y$ is then mapped back(decode) into a reconstruction $z$, which is of same shape as $x$ using similar mapping.

$$z = s(W'y + b') \qquad (4)$$

In (4), prime symbol is not a matrix transpose. Model parameters $(W, W', b, b')$ are optimized to minimize reconstruction error, which can be assessed using different loss functions such as squared error or cross-entropy.

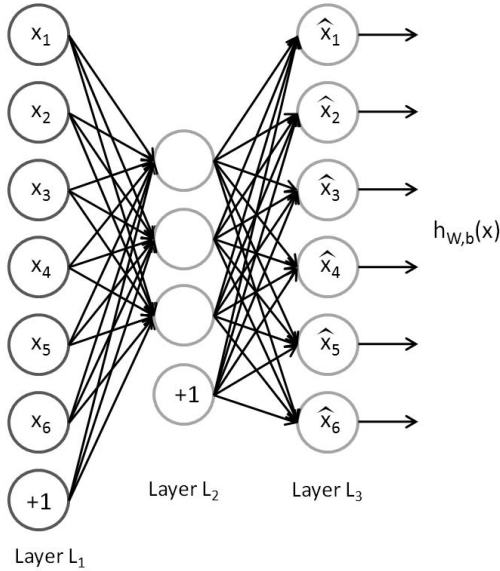Basic architecture of an autoencoder is shown in Fig. 1 [32]



Fig. 1.   A basic autoencoder

Here layer $L_1$ is input layer which is encoded in layer $L_2$ using latent representation and input is reconstructed at $L_3$.

Using number of hidden units lower than inputs forces autoencoder to learn a compressed approximation. Mostly an autoencoder learns low dimensional representation very similar to Principal Component Analysis (PCA). Having hidden units larger than number of inputs can still discover useful insights by imposing certain sparsity constraints.

*1) Denoising Autoencoders:* Denoising autoencoder is a stochastic extension to classic autoencoder [24], that is we force the model to learn reconstruction of input given its noisy version. A stochastic corruption process randomly sets some of the inputs to zero, forcing denoising autoencoder to predict missing(corrupted) values for randomly selected subsets of missing patterns.

Basic architecture of a denoising autoencoder is shown in Fig. 2
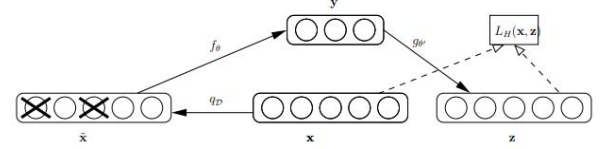


Fig. 2.   Denoising autoencoder, some inputs are set to missing

Denoising autoencoders can be stacked to create a deep network (stacked denoising autoencoder) [25] shown in Fig. 3 [33].
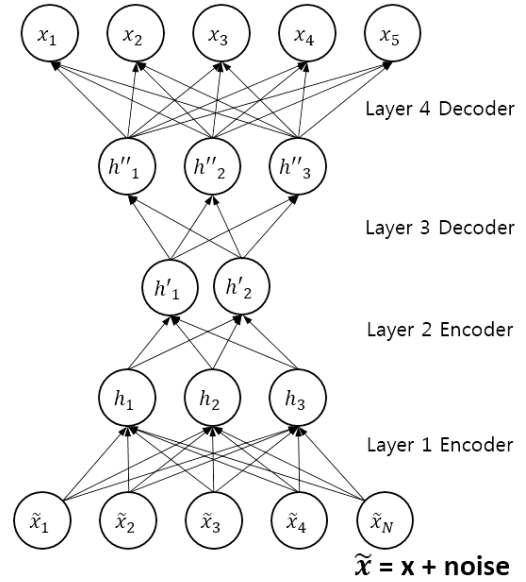


Fig. 3.   A stacked denoising autoencoder

Output from the layer below is fed to the current layer and training is done layer wise.

*2) Convolutional autoencoder:* Convolutional autoencoders [16] are based on standard autoencoder architecture with convolutional *encoding* and *decoding* layers. Compared to classic autoencoders, convolutional autoencoders are better suited for image processing as they utilize full capability of convolutional neural networks to exploit image structure.

In convolutional autoencoders, weights are shared among all input locations which helps preserve local spatiality. For a mono channel input, representation of $i$th feature map is given as

$$h^i = s(x * W^i + b^i). \qquad (5)$$

242

where bias is broadcasted to whole map, $*$ denotes convolution (2D) and $s$ is an activation. Single bias per latent map is used and reconstruction is obtained as

$$y = s(\sum_{i \in H} h^i * \tilde{W}^i + c) \qquad (6)$$

where $c$ is bias per input channel, $H$ is group of latent feature maps, $\tilde{W}$ is flip operation over both weight dimensions.

Backpropogation is used for computation of gradient of the error function with respect to the parameters.

## IV. EVALUATION

### A. Data

We used two datasets, mini-MIAS database of mammograms(MMM) [13] and a dental radiography database(DX) [26]. MMM has 322 images of $1024 \times 1024$ resolution and DX has 400 cephalometric X-ray images collected from 400 patients with a resolution of $1935 \times 2400$. Random images from both datasets are shown in Fig. 4.
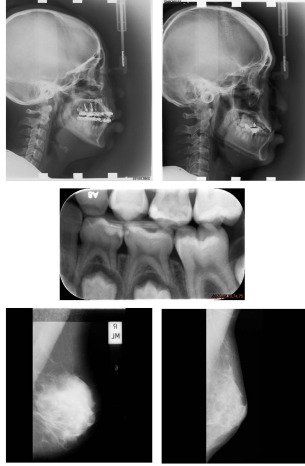


Fig. 4. Random sample of medical images from datasets MMM and DX, rows 1 and 2 show X-ray images from DX and row 3 shows mammograms from MMM

### B. Experimental setup

All images were processed prior to modelling. Preprocessing consisted of resizing all images to $64 \times 64$ for computational resource reasons. Different parameters detailed in Table I were used for corruption.

TABLE I. DATASET PERTURBATIONS

| Noise type | corruption parameters |
|---|---|
| Gaussian | $p=0.1$, $\mu = 0$, $\sigma = 1$ |
| Gaussian | $p=0.5$, $\mu = 0$, $\sigma = 1$ |
| Gaussian | $p=0.2$, $\mu = 0$, $\sigma = 2$ |
| Gaussian | $p=0.2$, $\mu = 0$, $\sigma = 5$ |
| Poisson | $p=0.2$, $\lambda = 1$ |
| Poisson | $p=0.2$, $\lambda = 5$ |

$p$ is proportion of noise introduced, $\sigma$ and $\mu$ are standard deviation and mean of normal distribution and $\lambda$ is mean of Poisson distribution

Instead of corrupting a single image at a time, flattened dataset with each row representing an image was corrupted, hence simultaneously perturbing all images. Corrupted datasets were then used for modelling. Relatively simple architecture was used for convolutional denoising autoencoder (CNN DAE), shown in Fig. 5.
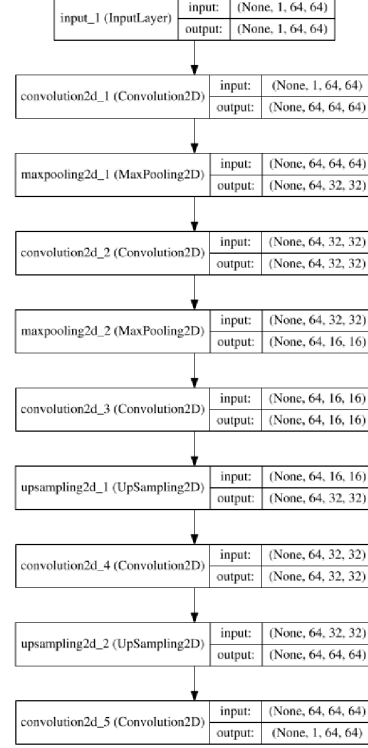


Fig. 5. Architecture of CNN DAE used

Keras [31] was used for modelling on an Acer Aspire M5 notebook (Intel Core i5-4200U, 10 GB RAM, no GPU). Images were compared using structural similarity index measure(SSIM) instead of peak signal to noise ratio (PSNR) for its consistency and accuracy [27]. A composite index of three measures, SSIM estimates the visual effects of shifts in image luminance, contrast and other remaining errors, collectively called structural changes. For original and coded signals $x$ and $y$, SSIM is given as

$$SSIM(x,y) = [l(x,y)]^\alpha [c(x,y)]^\beta [s(x,y)]^\gamma \qquad (7)$$

where $\alpha, \beta$ and $\gamma > 0$ control the relative significance of each of three terms in SSIM and $l$, $c$ and $s$ are luminance, contrast and structural components calculated as

$$l(x,y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \qquad (8)$$

$$c(x,y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \qquad (9)$$

$$s(x,y) = \frac{2\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \qquad (10)$$

where $\mu_x$ and $\mu_y$ represents the mean of original and coded image, $\sigma_x$ and $\sigma_y$ are standard deviation and $\sigma_{xy}$ is the covariance of two images.

Basic settings were kept constant with 100 epochs and a batch size of 10. No fine-tuning was performed to get comparison results on a basic architecture, that should be easy to implement even by a naive user. Mean of SSIM scores over the set of test images is reported for comparison.

## C. Empirical evaluation

For baseline comparison, images corrupted with lowest noise level ($\mu = 0, \sigma = 1, p = 0.1$) were used. To keep similar sample size for training, we used 300 images from each of the datasets, leaving us with 22 for testing in MMM and 100 in DX.

Using a batch size of 10 and 100 epochs, denoising results are presented in Fig. 6 and Table II.
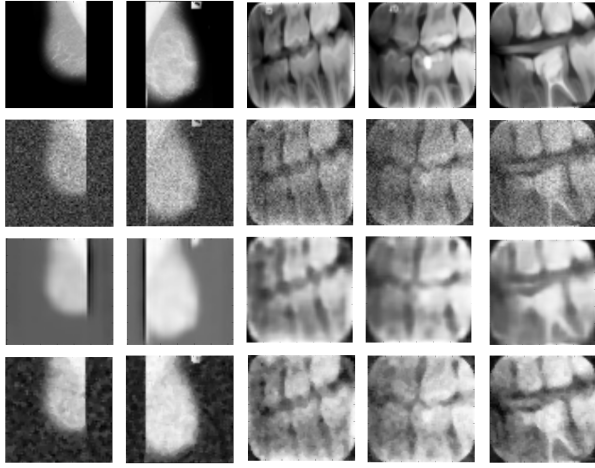


Fig. 6. Denoising results on both datasets, top row shows real images with second row showing the noisier version ($\mu = 0, \sigma = 1, p = 0.1$), third row shows images denoised using CNN DAE and fourth row shows results of applying a median filter

TABLE II. MEAN SSIM SCORES FOR TEST IMAGES FROM MMM AND DX DATASETS

| Image type | MMM | DX |
|---|---|---|
| Noisy | 0.45 | 0.62 |
| CNN DAE | 0.81 | 0.88 |
| Median filter | 0.73 | 0.86 |

Results show an increased denoising performance using this simple architecture on small datasets over the use of median filter, which is most often used for this type of noise.

Model converged nicely for the given noise levels and sample size, shown in Fig. 7. It can bee seen that even using 50 epochs, reducing training time in half, we would have got similar results.

To test if increased sample size by combining heterogeneous data sources would have an impact on denoising performance, we combined both datasets with 721 images for training and 100 for testing.
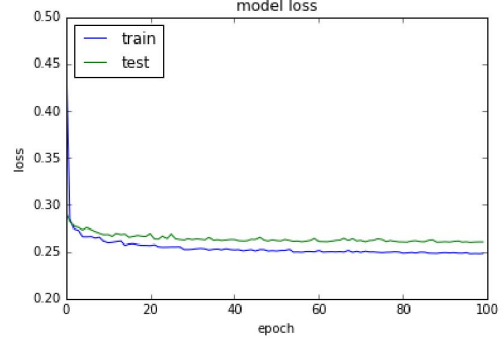


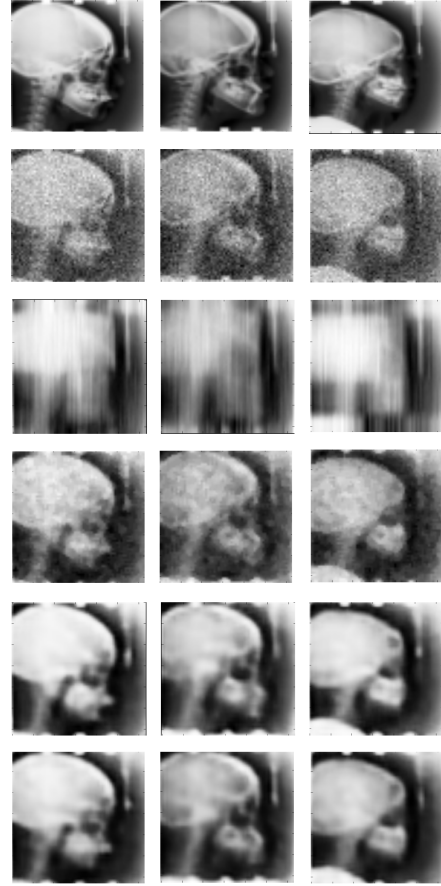Fig. 7. Training and validation loss from 100 epochs using a batchsize of 10



Fig. 8. Denoising performance of CNN DAE on combined dataset, top row shows real images, second row is noisier version with minimal noise, third row is denoising result of NL means, fourth rows shows results of median filter, fifth row is results of using smaller dataset (300 training samples) with CNN DAE, sixth row is the results of CNN DAE on larger combined dataset.

Denoising results on three randomly chosen test images from combined dataset are shown in Fig. 8 and Table III.

Table III shows that CNN DAE performs better than NL means and median filter. Increasing sample size marginally enhanced the denoising performance.

To test the limits of CNN DAEs denoising performance, we

244

| Image type | SSIM |
|---|---|
| Noisy | 0.63 |
| NL means | 0.62 |
| Median filter | 0.80 |
| CNN DAE(a) | 0.89 |
| CNN DAE(b) | 0.90 |

CNN DAE(a) is denoising performance using smaller dataset and CNN DAE(b) is denoising performance on same images using the combined dataset.

used rest of the noisy datasets with varying noise generative patterns and noise levels. Images with high corruption levels are barely visible to human eye, so denoising performance on these is of interest. Denoising results along with noisy and noiseless images on varying levels of Gaussian noise are shown in Fig. 9.
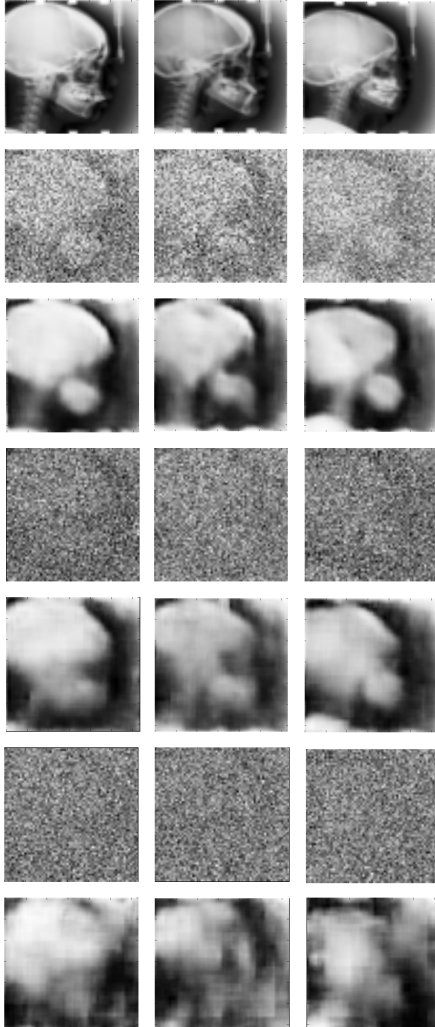


Fig. 9.    Denoising performance of CNN DAE on different Gaussian noise patterns. Top row shows original images, second row is noisy images with noise levels of $\mu = 0, \sigma = 1, p = 0.5$, third row shows denoising results, fourth row shows corruption with $p = 0.2, \sigma = 5$, fifth row is denoised images using CNN DAE, sixth and seventh rows shows noisy and denoised images corrupted with $p = 0.2, \sigma = 10$.

It can be seen that as noise level increases, this simple network has trouble reconstructing original signal. However, even when the image is not visible to human eye, this network is successful in partial generation of real images. Using a more complex deeper model, or by increasing number of training samples and number of epochs might help.

Performance of CNN DAE was tested on images corrupted using Poisson noise with $p = 0.2$, $\lambda = 1$ and $\lambda = 5$. Denoising results are shown in Fig. 10.
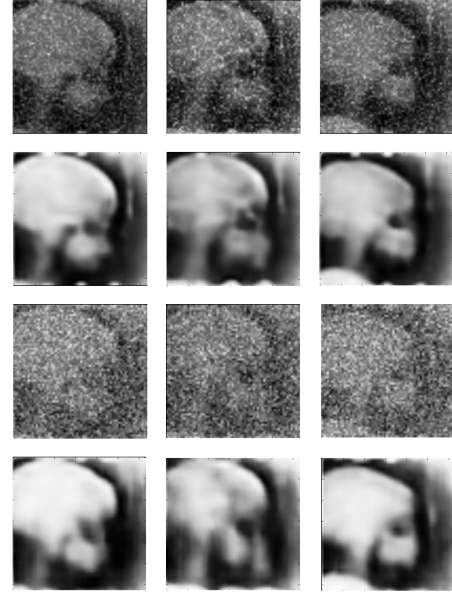


Fig. 10.    CNN DAE performance on Poisson corrupted images. Top row shows images corrupted with $p = 0.2, \lambda = 1$ with second row showing denoised results using CNN DAE. Third and fourth rows show noisy and denoised images corrupted with $p = 0.2, \lambda = 5$.

Table IV shows comparison of CNN DAE with median filter and NL means for denoising performance on varying noise levels and types. It is clear that CNN DAE outperforms both denoising methods by a wide margin, which increases as noise level increases.

| Image type | $p = 0.5$ | $sd = 5$ | $sd = 10$ | $Poisson, \lambda = 5$ |
|---|---|---|---|---|
| Noisy | 0.10 | 0.03 | 0.01 | 0.33 |
| NL means | 0.25 | 0.03 | 0.01 | 0.15 |
| Median filter | 0.28 | 0.11 | 0.03 | 0.17 |
| CNN DAE | 0.70 | 0.55 | 0.39 | 0.85 |

$p = 0.5$ represents 50% corrupted images with $\mu = 0, \sigma = 1$, $sd = 5$ are images corrupted with $p = 0.2, \mu = 0, \sigma = 5$, $sd = 10$ are corrupted with $p = 0.2, \mu = 0, \sigma = 10$ and $Poisson, \lambda = 5$ are corrupted with a Poisson noise using $\lambda = 5$

Also, as the noise level is increased the network has trouble converging. Fig. 11 shows the loss curves for Gaussian noise with $\mu = 0, p = 0.2, \sigma = 10$. Even using 100 epochs, model has not converged.
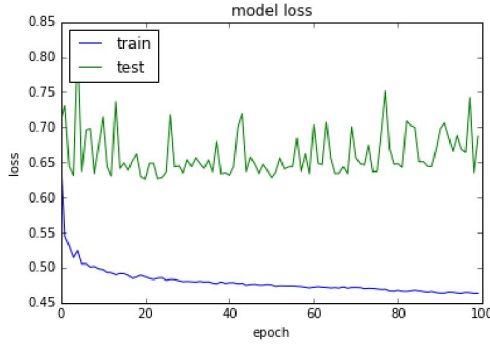
Fig. 11. Model having trouble converging at higher noise levels, no decrease in validation errors can be seen with increasing number of epochs.

## V. CONCLUSION

We have shown that denoising autoencoder constructed using convolutional layers can be used for efficient denoising of medical images. In contrary to the belief, we have shown that good denoising performance can be achieved using small training datasets, training samples as few as 300 are enough for good performance.

Our future work would focus on finding an optimal architecture for small sample denoising. We would like to investigate similar architectures on high resolution images and the use of other image denoising methods such as singular value decomposition (SVD) and median filters for image pre-processing before using CNN DAE. It would also be of interest, if given only a few images, we can combine them with other readily available datasets such as ImageNet [8] for better denoising performance by increasing training sample size.

## REFERENCES

[1] Agostinelli, Forest, Michael R. Anderson, and Honglak Lee. "Adaptive multi-column deep neural networks with application to robust image denoising." *Advances in Neural Information Processing Systems*. 2013.

[2] Bengio, Yoshua, et al. "Greedy layer-wise training of deep networks." *Advances in neural information processing systems* 19 (2007): 153.

[3] Buades, Antoni, Bartomeu Coll, and Jean-Michel Morel. "A review of image denoising algorithms, with a new one." *Multiscale Modeling and Simulation* 4.2 (2005): 490-530.

[4] Burger, Harold C., Christian J. Schuler, and Stefan Harmeling. "Image denoising: Can plain neural networks compete with BM3D?." *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on. IEEE, 2012.

[5] Cho, Kyunghyun. "Boltzmann machines and denoising autoencoders for image denoising." *arXiv preprint* arXiv:1301.3468 (2013).

[6] Coifman, Ronald R., and David L. Donoho. Translation-invariant denoising. *Springer* New York, 1995.

[7] Dabov, Kostadin, et al. "Image denoising by sparse 3-D transform-domain collaborative filtering." *IEEE Transactions on image processing* 16.8 (2007): 2080-2095.

[8] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[9] Elad, Michael, and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries." *IEEE Transactions on Image processing* 15.12 (2006): 3736-3745.

[10] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks." *Aistats*. Vol. 15. No. 106. 2011.

[11] Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal Processing Magazine* 29.6 (2012): 82-97.

[12] Jain, Viren, and Sebastian Seung. "Natural image denoising with convolutional networks." *Advances in Neural Information Processing Systems*. 2009.

[13] J Suckling et al (1994): The Mammographic Image Analysis Society Digital Mammogram Database *Exerpta Medica*. International Congress Series 1069 pp375-378.

[14] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[15] Mairal, Julien, et al. "Online dictionary learning for sparse coding." *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009.

[16] Masci, Jonathan, et al. "Stacked convolutional auto-encoders for hierarchical feature extraction." *International Conference on Artificial Neural Networks*. Springer Berlin Heidelberg, 2011.

[17] Olshausen, Bruno A., and David J. Field. "Sparse coding with an overcomplete basis set: A strategy employed by V1?." *Vision research* 37.23 (1997): 3311-3325.

[18] Perona, Pietro, and Jitendra Malik. "Scale-space and edge detection using anisotropic diffusion." *IEEE Transactions on pattern analysis and machine intelligence* 12.7 (1990): 629-639.

[19] Portilla, Javier, et al. "Image denoising using scale mixtures of Gaussians in the wavelet domain." *IEEE Transactions on Image processing* 12.11 (2003): 1338-1351.

[20] Rudin, Leonid I., and Stanley Osher. "Total variation based image restoration with free local constraints." Image Processing, 1994. Proceedings. *ICIP-94.*, IEEE International Conference. Vol. 1. IEEE, 1994.

[21] Sanches, Joo M., Jacinto C. Nascimento, and Jorge S. Marques. "Medical image noise reduction using the SylvesterLyapunov equation." *IEEE transactions on image processing* 17.9 (2008): 1522-1539.

[22] Subakan, Ozlem, et al. "Feature preserving image smoothing using a continuous mixture of tensors." 2007 *IEEE 11th International Conference on Computer Vision*. IEEE, 2007.

[23] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.

[24] Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.

[25] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of Machine Learning Research* 11.Dec (2010): 3371-3408.

[26] Wang, Ching-Wei, et al. "A benchmark for comparison of dental radiography analysis algorithms." *Medical image analysis* 31 (2016): 63-76.

[27] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." *IEEE transactions on image processing* 13.4 (2004): 600-612.

[28] Xie, Junyuan, Linli Xu, and Enhong Chen. "Image denoising and inpainting with deep neural networks." *Advances in Neural Information Processing Systems*. 2012.

[29] Yaroslavsky, Leonid P., Karen O. Egiazarian, and Jaakko T. Astola. "Transform domain image restoration methods: review, comparison, and interpretation." *Photonics West 2001-Electronic Imaging*. International Society for Optics and Photonics, 2001.

[30] Zhang, Dapeng, and Zhou Wang. "Image information restoration based on long-range correlation." *IEEE Transactions on Circuits and Systems for Video Technology* 12.5 (2002): 331-341.

[31] Franois Chollet, keras, (2015), GitHub repository, https://github.com/fchollet/keras

[32] Deep learning tutorial, Stanford University.Autoencoders. Available: *http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/*

[33] Introduction Auto-Encoder, wikidocs.Stacked Denoising Auto-Encoder (SdA). Available: *https://wikidocs.net/3413*