Techniques in Image Denoising

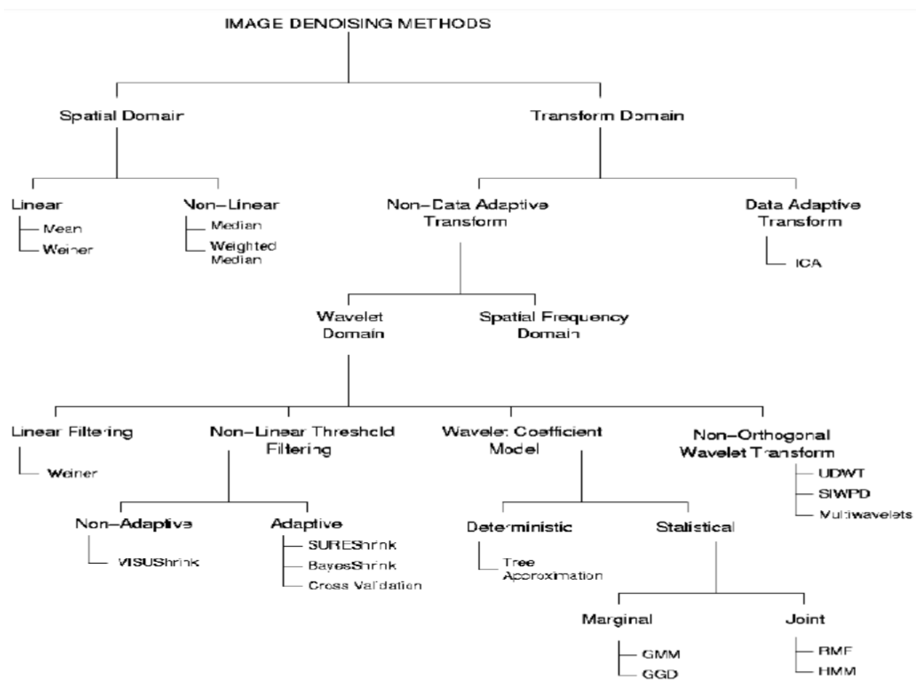
Avinash Kori —ED15B006
Engineering Design Department,
Indian Institute of Technology, Madras
koriavinash1@gmail.com

October 22, 2018

Contents

1 Introduction

The main challenge in digital image processing in research field is to remove noise from the original image. With respect to various assumptions, advantages, applications and limitations, different denoising algorithms have been proposed. In this paper, some important denoising techniques are discussed and classification of such techniques is listed. The denoising procedure which can be applied to all the types of noises is discussed at the last.Research and Technology field requires the application of Image Processing Schemes. Digital images are very important in the areas of geographical information systems and astronomy. Satellite television magnetic resonance Imaging, computer tomography is some of the daily life applications wherein digital images play a vital role. All natural phenomena and transmission errors are degrading the image quality thereby noise is introduced in the image. Hence there is a need for image denoising procedure to reduce the noise level present in the image so as to produce the denoised image closer to the original image. There are plethora of image denoising algorithms few of includes: Linear diffusion models, transformation based models, factorization based models, statistical models, variational methods and probabilistic approaches. below flow chart shows all different types of denoising methods traditionally used:



Apart from these techniques, deep convolutional network based techniques are also very popular and provide very good quantitative results. In this case study few Convolutional Neural Network based and Convolutional Variational Autoenoders based models for image denoising are explained, along with traditional diffusion models.

2 Measurements

Most frequent measurement metrics used for comparing image qualities are list below, In this report image denoising methods are qualitatively assessed based on these matrics.

2.1 MSE

Mean Squared Error (MSE) is an error metric which finds the pixel wise deviation between fixed and moving image. MSE is given by equation 1.

$$MSE = \frac{1}{H \times W} \sum_{x=0}^{W} \sum_{y=0}^{H} (NoisyImage_{x,y} - ReconstructedImage_{x,y})^2 \tag{1}$$

where $NoisyImage_{x,y}$ denotes pixel value at x,y position in the Fixed Image, similarly for $ReconstructedImage_{x,y}$ denotes pixel value at x,y position in the moving image. MSE $\in [0, \infty)$, as lesser the MSE as better the overlap.

2.2 PSNR

PSNR is the ratio often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image. PSNR is proportional to negative log of MSE error, which is described in equation 2

$$PSNR = 10log_{10}(\frac{R^2}{MSE \quad error}) \tag{2}$$

2.3 Mutual Information

Mutual information (MI) is one of the quantities which measures the amount of correlation between two different random variables. In case of registration as higher the MI score as good the performed registration. MI between two variables is given by equation 3.

$$MI(N, R) = \mathbf{E}(P_{NR}(N, R)) \times log(\frac{P_{NR}(N, R)}{P_N(N)P_R(R)}) \tag{3}$$

where $P_{FM}(N, R)$ denotes joint distribution, $\mathbf{E}$ denotes expectation value and $P_N(N), P_R(R)$ denotes marginals. MI $\in$ [0, 1.0], where 0 being least and 1 being maximum score, as higher the score as better the similarity.

2.4 SSMI: Structural Similarity

Structural similarity index (SSIM) is measure of similarity between two images. SSIM considers local pixel information for score calculation. This means it carries an idea of spatial positioning of pixels in an images. SSIM if given by equation 4.

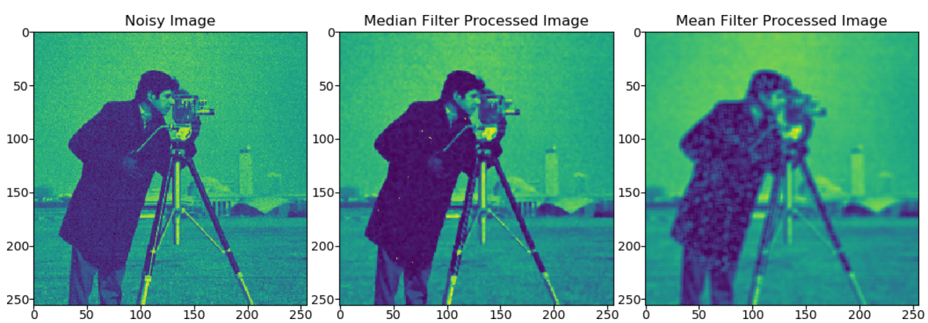
$$\mathrm{SSIM}(N, R) = \frac{(2\mu_N \mu_R + c_1)(2\sigma_{NR} + c_2)}{(\mu_F^2 + \mu_R^2 + c_1)(\sigma_N^2 + \sigma_R^2 + c_2)} \tag{4}$$

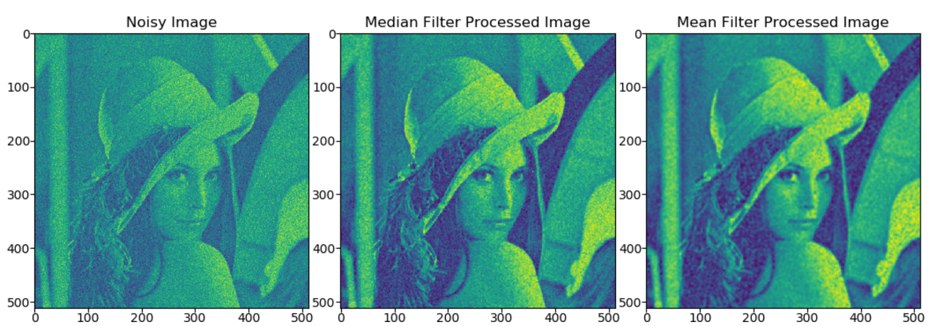
In the above equation N corresponds to Noisy image and R corresponds to Reconstructed image. c_1, c_2 are two variables to stabilize week denominators. SSIM $\in$ [0, 1.0], where 0 being least and 1 being maximum score, as higher the score as better the overlap.

3 White box based methods

3.1 Simple filters for noise filtration

Filtering using mean and median filters, these filtering techniques involves the convolution operation of an image with filter kernels. Mean filter is a linear operation while Median is non linear filter. The effect of these linear and nonlinear filters on noisy images are shown below:





Based on the above images it's clear that normal mean and median filter's removes noise to certain extent but can't be removing entire noise. Based of visual effects it seems that median filter has larger effect in removing high frequency noise. In the subsequent sections we see various other techniques for denoising.

3.2 Edge Enhancing Diffusion

3.2.1 Model Explanation

Edge enhancing diffusion is helps in enhancing the edges in an image and then applying PM diffusion model. EED is also known as edge perserving smoothness. This is achieved by using Deffusion tensor as introduced by Weickert.

D after spectral factorization can be written as

$$D = [V_1, \quad V_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} [V_1, \quad V_2]^T \tag{5}$$

where V_1 and V_2 ortho-normal tensors which are given by:

$$V_1 = [U_x, \quad U_y]^T \tag{6}$$

$$V_2 = [-U_y, \quad U_x]^T \tag{7}$$

where U_x and U_y denotes gradient of an image in x & y direction respectively.

In the above equation λ is an gaussian smoothened image (smoothening is application of gaussion filter on noisey image). Sometimes these λ are considered as conductivity parameters, λ_1 conductivity along the edges and λ_2 to be conductivity across the edges.

$$\lambda_1 = e^{-\frac{||\nabla U||^2}{\sigma^2}} \tag{8}$$

$$\lambda_2 = c\lambda_1 \qquad c \to 0 \tag{9}$$

by all the above equations we get D:

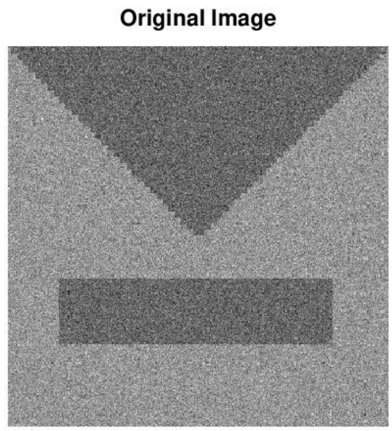
$$D = \begin{bmatrix} U_x & -U_y \\ U_y & U_x \end{bmatrix} \begin{bmatrix} e^{-\frac{||\nabla U||^2}{\sigma^2}} & 0 \\ 0 & ce^{-\frac{||\nabla U||^2}{\sigma^2}} \end{bmatrix} \begin{bmatrix} U_x & U_y \\ -U_y & U_x \end{bmatrix} \tag{10}$$

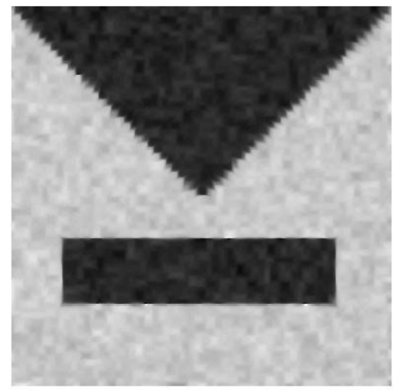
Only parameters in the above equations are σ and c. U_x, U_y can be numerically determined using image. $\lambda's$ can be determined by convolving gaussian kernel with image, with σ as hyper-parameter.

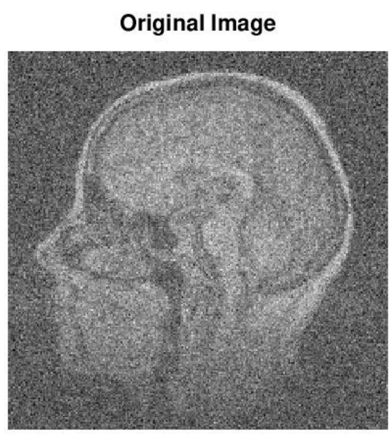
Partial differential equation for EED is given by:

$$\frac{\partial U}{\partial t} = \nabla . D(\nabla U_\sigma) \nabla U \tag{11}$$

3.2.2 Results







3.3 Edge Enhancing With Unsharp masking

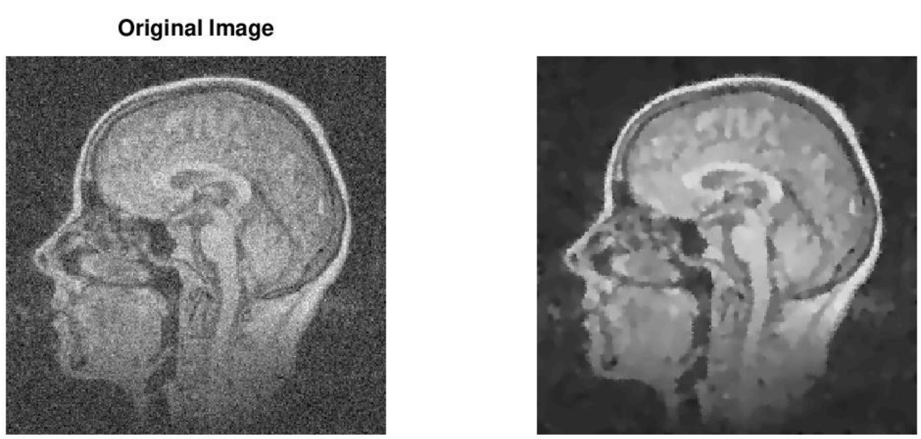
3.3.1 Model Explanation

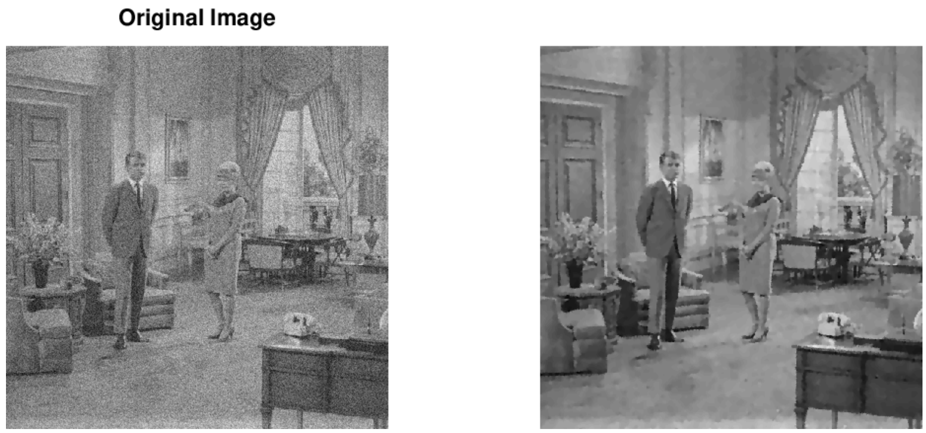
In this technique the edges in an image are enhanced using unsharped masking before denoising. Unsharped masking is obtained by superposition of image with it's laplacian. (source: Gonzalis and woods, Digital Image Processing)

This enhanced image is then passed through anisotropic diffusion model like PM model for noise removal.

$$Image = Image + \nabla^2 Image \tag{12}$$

3.3.2 Results





3.4 Edge Enhancing With Canny edges

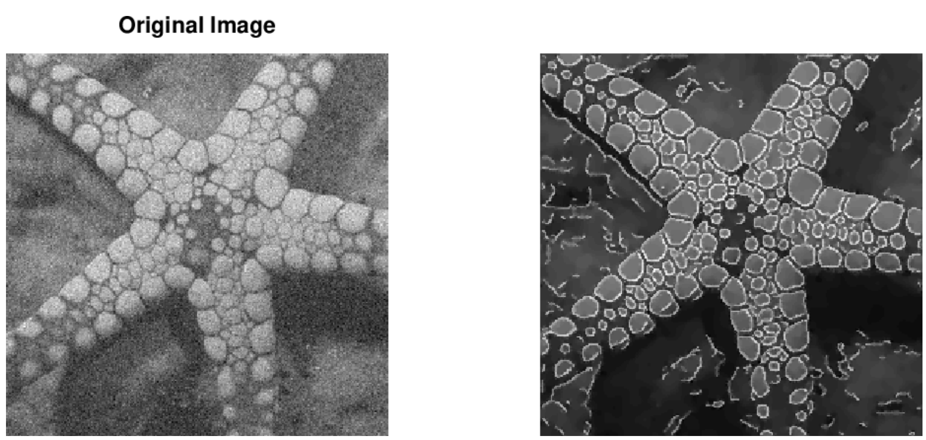
3.4.1 Model Explanation

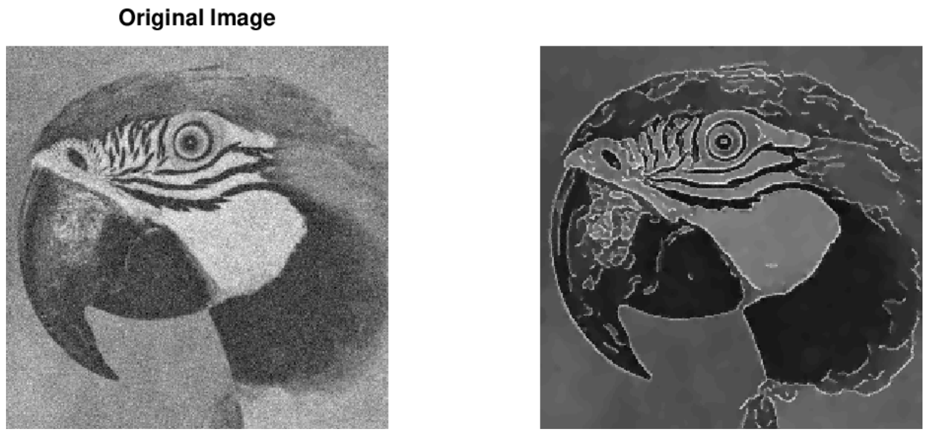
In this technique the edges in an image are enhanced using unsharped masking before denoising. Canny edges are obtained by superposition of image with it's gradient after doing non-maximal supression. (source: Gonzalis and woods, Digital Image Processing)

$$Image = Image + cannyedges(Image) \tag{13}$$

This enhanced image is then passed through anisotropic diffusion model like PM model for noise removal.

3.4.2 Results





3.5 Coherance Enhancing Diffusion

3.5.1 Model Explanation

In the situations which envolves estimating the local orientation as the direction of the gradient vector is not possible to determine using EED. Consider the finger print image or any biomedical images the details are lost by EED. The local orientation estimation is based on the structure tensor.

$$J_\rho(\nabla u_\sigma) = k_\rho * (\nabla u_\sigma \nabla u_\sigma^T) \tag{14}$$

$$J_\rho(\nabla u_\sigma) = k_\rho * [U_x, \quad U_y]^T [U_x, \quad U_y] \tag{15}$$

$$J_\rho(\nabla u_\sigma) = \begin{bmatrix} U_x U_x * k_\rho & U_x U_y * k_\rho \\ U_y U_x * k_\rho & U_y U_y * k_\rho \end{bmatrix} \tag{16}$$

Component wise convolution with the Gaussian k_ρ averages orientation information over an integration scale ρ.

Diffusion tensor for CED is given by:

$$D = [V_1, \quad V_2] \begin{bmatrix} c1 & 0 \\ 0 & c2 \end{bmatrix} [V_1, \quad V_2]^T \tag{17}$$

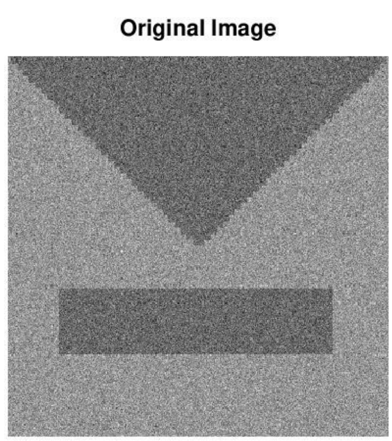
where V_1 and V_2 are eigen vectors of matrix J_ρ. c1 and c2 are the conductivity coefficients along the principal directions. c1 and c2 are determined using the eigenvalues of J_ρ matrix. Let λ_1 and λ_2 be eigenvalues of J_ρ matrix.

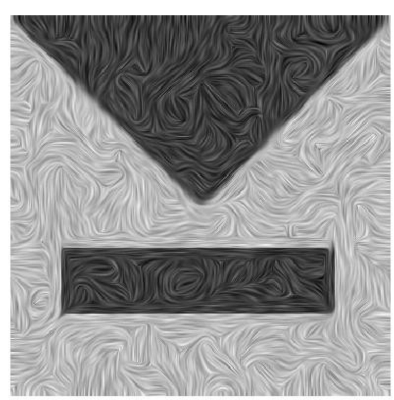
$$c1 = max(\alpha, e^{\frac{(\lambda_1 - \lambda_2)^2}{\sigma^2}}) \tag{18}$$

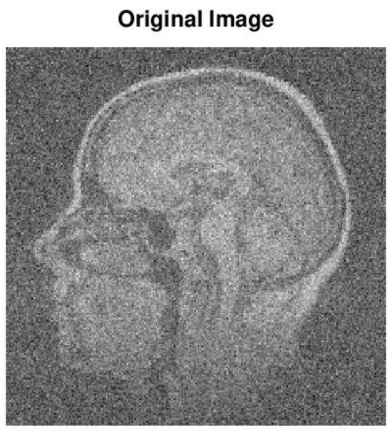
$$c2 = \alpha \tag{19}$$

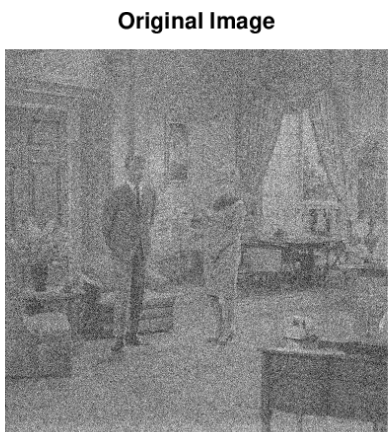
All the parameters used in $J_\rho(\nabla u_\sigma)$ can be estimated numerically using image, Which means V_1, V_2, λ_1 and λ_2 are known, c1 and c2 are also known. only hyper-parameters involved are α & σ standard deviation of gaussian used for image smoothening in J_ρ.

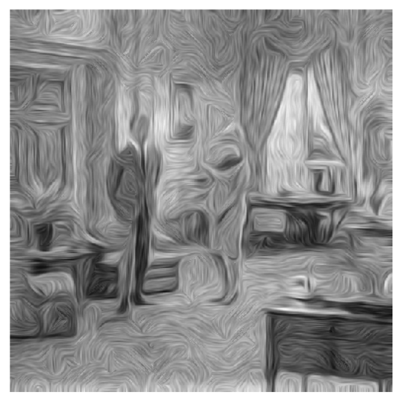
3.5.2 Results











4 Noise Level with cleaning performance

The Values tabulated below are the average of 15 different images. The expirement was conducted on 15 different images and the corresponding mean is considered as final score.

PM model: noise to cleaning performance

Noise Level	**MSE**	**SSIM**	**MI**	**PSNR**
0.0, 0.01 Gaussian	**241.75**	**0.999**	**0.475**	**28.37**
0.0, 0.05 Gaussian	**1231.0**	**0.999**	**0.041**	**20.44**
0.6, 0.01 Gaussian	**5.06e+4**	**0.954**	**0.467**	**4.415**
0.6, 0.05 Gaussian	**5.16e+4**	**0.954**	**0.029**	**4.321**

EED model: noise to cleaning performance

Noise Level	MSE	SSIM	MI	PSNR
0.0, 0.01 Gaussian	**534.92**	**0.999**	**0.406**	**24.292**
0.0, 0.05 Gaussian	**574.81**	**0.999**	**0.357**	**23.876**
0.6, 0.01 Gaussian	**5.09e+4**	**0.954**	**0.413**	**4.385**
0.6, 0.05 Gaussian	**5.09e+4**	**0.954**	**0.349**	**4.386**

CED model: noise to cleaning performance

Noise Level	MSE	SSIM	MI	PSNR
0.0, 0.01 Gaussian	**405.11**	**0.999**	**0.466**	**25.081**
0.0, 0.05 Gaussian	**649.65**	**0.999**	**0.426**	**23.876**
0.6, 0.01 Gaussian	**5.07e+4**	**0.954**	**0.482**	**4.397**
0.6, 0.05 Gaussian	**5.15e+4**	**0.954**	**0.440**	**4.7**

5 Effect of Noise distributions

Effect of distribution was tested using PM model, Below table shows the effect of each noise distribution with various similarity/error metrics. The similarity/error scores in the table are the average scores of 15 different images which are shared along with github repository.

Noise Type	MSE	SSIM	MI	PSNR
0.0, 0.01 Gaussian	**240.9**	**0.999**	**0.423**	**28.39**
Poisson	**178.61**	**0.999**	**0.588**	**30.871**
salt & pepper	**371.42**	**0.998**	**0.617**	**25.87**
speckle	**269.12**	**0.999**	**0.596**	**27.50**

6 Black box based methods

In the recent year's data driven models are performing state-of-the-art results in all vision, audio based tasks. Convolutional neural networks (CNN) have proved human level performance in object detection, classification, localization. These experiments were conducted to test the performance of CNN's in image Denoising.

6.1 Data

All the netowrks were trained using open source standard computer vision dataset with about 480 gray scaled images of multiple shapes. Few data samples are shown in figure 1. Data was split into training, validation and testing with 400, 68, 12 images respectively. About 100,000 patches were extracted from training data, and the network was trained using these patches with random noise (normal distribution) level ranging from [20/255-60/255] (variance of white noise).

Figure 1: Sample training data

6.2 Deterministic approach: Convolutional Neural Network

All data driven models behave as deterministic quantty once trained. The only randomness present is during training process, picking random samples from entire dataset during each step of stochastic gradient descent.

In this experiment, 9 layered deep convolutional network, Network architecture is described in figure 2, where each block involves CNN layer, Non-linearity layer which in this case is ReLU layer, and Batch normalization layer. Batch normalization behaves as feature regularization layer, which prevents model from overfitting and helps model to learn rich features from the data. Network was initialized using Xavier initializer, and was trained with sthocastig gradient descent using Adam optimizer. Initial learning rate of 0.001 and decay of 0.1

was used. Pixel wise **Mean Squared Error** was used as Cost function for training the model.

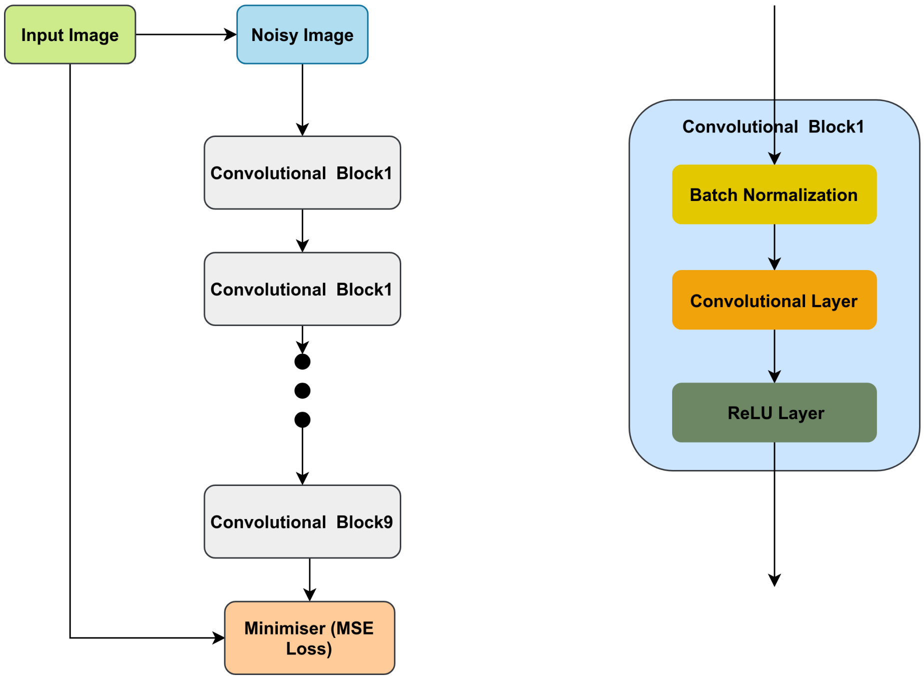
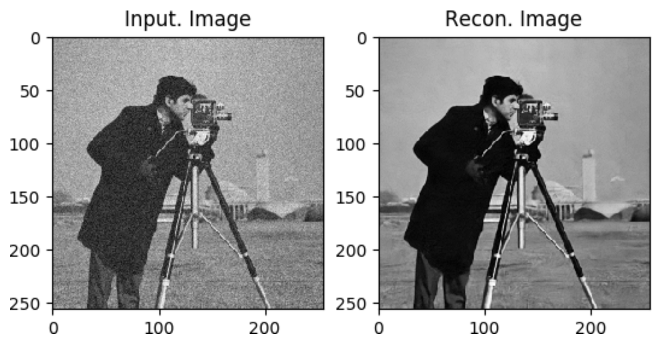
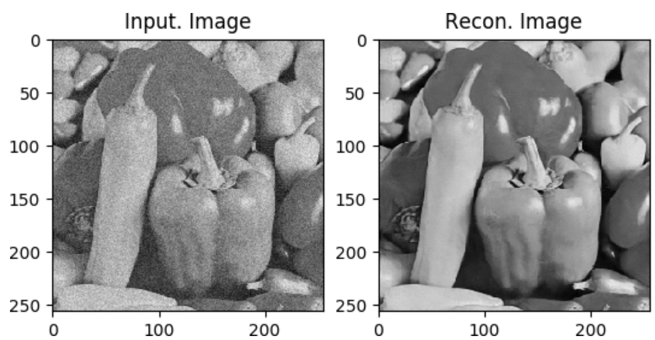
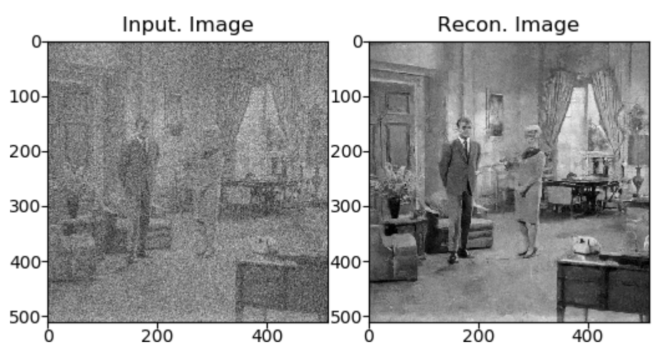


Figure 2: Convolutional neural netowrk architecture used

Results obtained

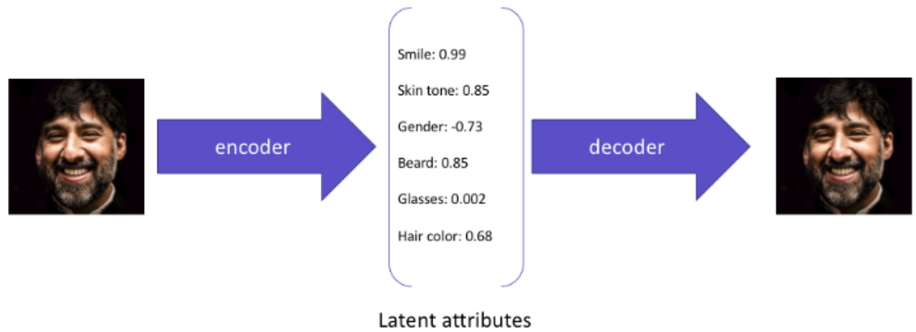






6.3 Variational approach: Convolutional Variational Autoencoders

A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space. In VAE input convert from input space to lower dimensional latent space (encoder part of VAE), Encoder provides us control over input data by reducing higher dimensional data to very few tractable latent space variables. Encoders can formulated to describe a probability distribution for each latent variables. For example, An ideal autoencoder will learn descriptive attributes of faces such as skin color, whether or not the person is wearing glasses, etc. in an attempt to describe an observation in some compressed representation. which is described in figure **??** [1]. These latent variables are processed and fed to decoder, which involves upsampling pathway (in this case bi-linear upsampling was used) to convert from low dimensional space to original image space. Reconstructed image from decoder and input image are used in cost calculation, cost funciton in case of VAE is linear combination of MSE and KL-Divergence.



In our case encoder helps in identifying noise properties in an input data, which is further processed and image is reconstructed back using decoder architecture which involves multiple CNN layers along with bilateral upsampling layers which helps to reconstrcting noise free image, with same as input dimension. Network architecture used in this expiriment is described in figure 3. Network was initialized using Xavier initializer, and was trained with sthocastig gradient descent using Adam optimizer. Initial learning rate of 0.001 and decay of 0.1 was used. Pixel wise **Mean Squared Error + KL Divergence** was used as Cost function for training the model.

[1]Image taken from: https://www.jeremyjordan.me/variational-autoencoders/

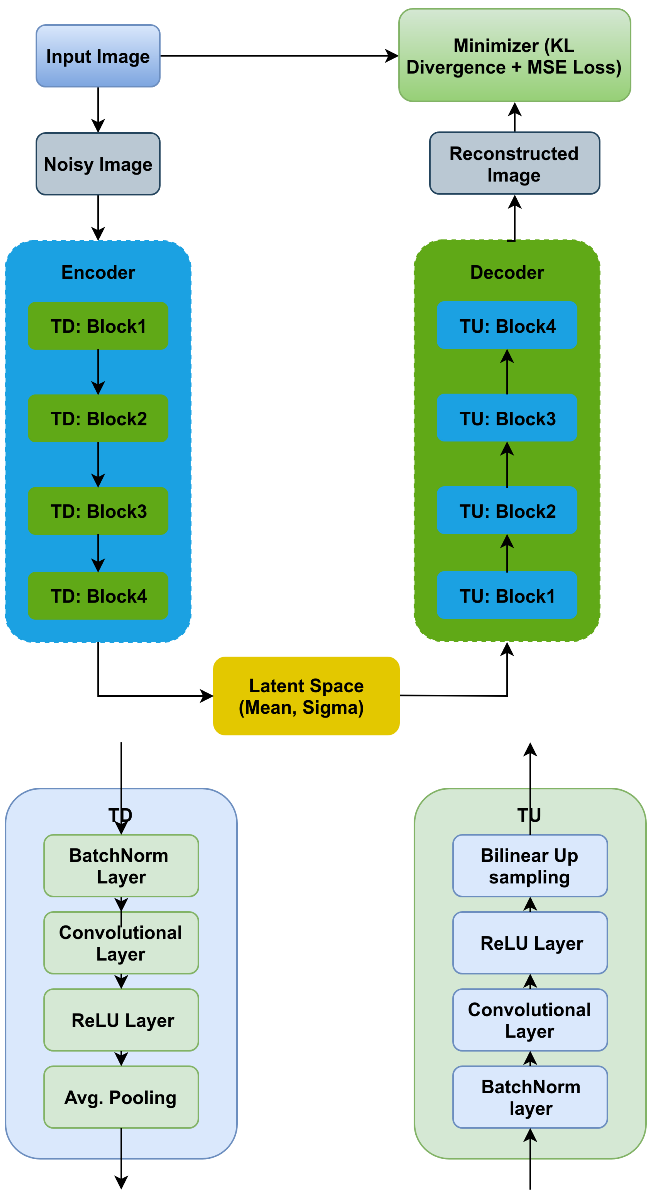
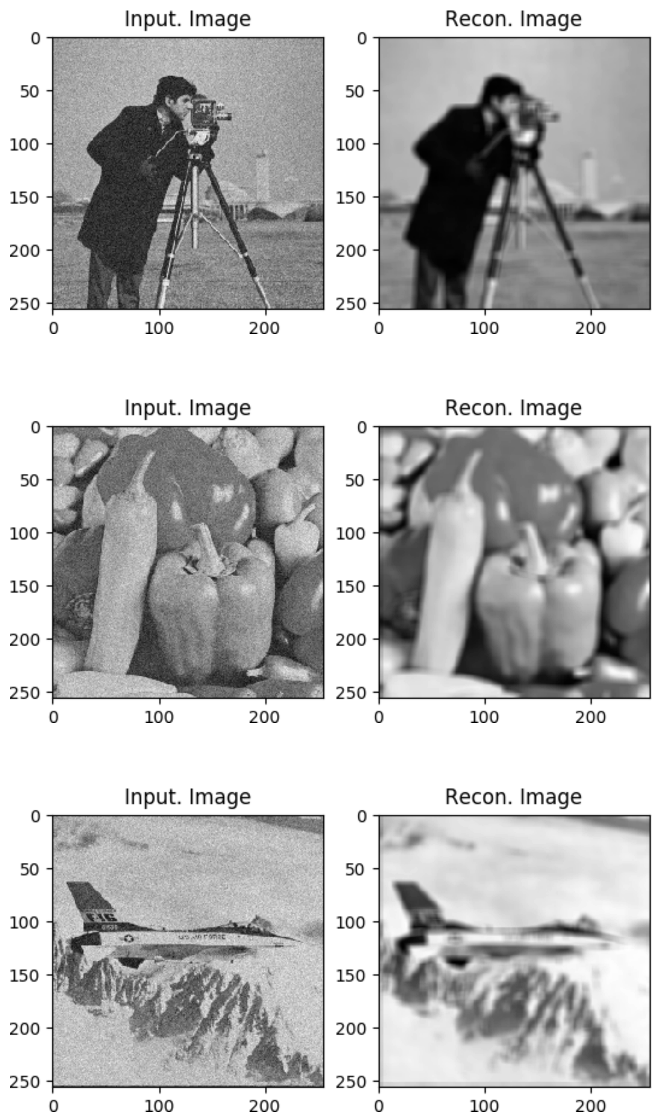


Figure 3: Convolutional Variational Autoencoder netowrk architecture used

Results obtained



7 Sparse coding for image denoising

7.1 Methodology

Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data efficiently. The aim of sparse coding is to find a set of basis vectors ϕ_i such that we can represent an input vector $\mathbf{x}$ as a linear combination of these basis vectors:

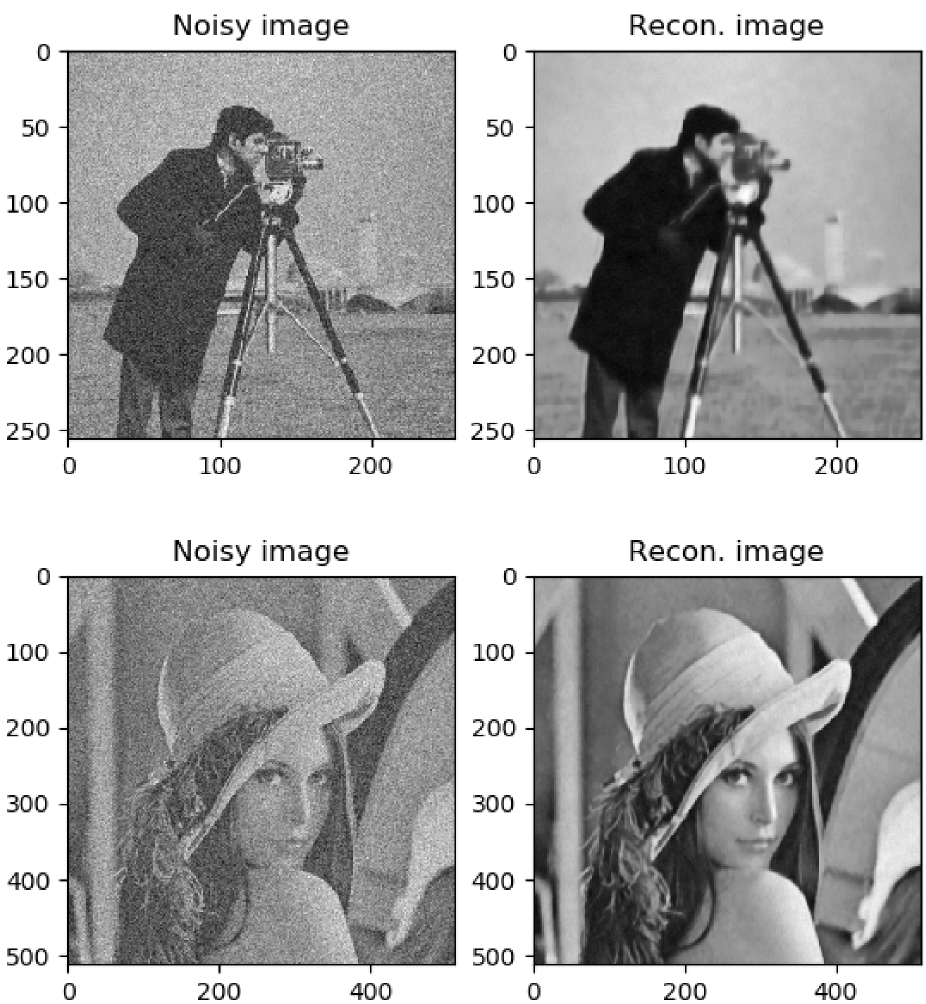
$$\hat{X} = \Sigma_i^k a_i \phi_i$$

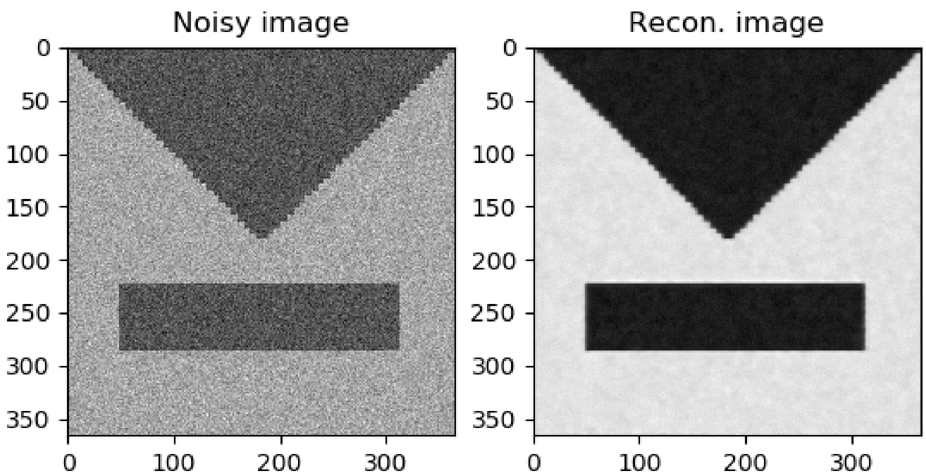
Objective function used :

$$minimize||X - \Sigma_i^k a_i \phi_i|| + |\Sigma_i^k a_i|$$

Due to the presence of L1 regularizer sparse a vectors are learnt.

7.2 Results





8 Comparison between white box and black box models

- All the white box models are iterative in nature while Blackbox models once trained, behaves as direct method.
- Time taken for processing of each image in whitebox model is about 300 sec (in CED), while case of blackbox models we get results in 0.62 sec
- Quality of reconstructed images in black box models are similar to that of white box models visually

9 Code availability and structure

This Report comes with a dedicated GitHub repository where all codes, animations and pre-trained models will be uploaded.
(https://github.com/koriavinash1/ImageDenoising)

Folder Structure of Code:

- ImageDenoising-master
  - BlackBoxModel
    - DeterministicMethod
    - VariationalInferMethod
  - WhiteBoxModel
    - Perona-Malik program
  - Reports

10 Conclusions

Various techniques involved in image denoising were analysed and implemented in this case study. It turnsout that black box modelling also performs equivalently good when compared with some white box models. Numerical approximations involved in EED, CED were studied in case of whitebox models, While in case of blackbox models Convolutional Neural Networks, Convolutional Variational Autoencoders, Spare Encoding and few simple image filters like mean/ median filters were analysed and implemented as part of this case study.

11 Acknowledgements

- White box models were implemented in Matlab using existing base code
- Blackbox models were implemented using Python, following libraries were used:
  - Pytorch for deep network construction
  - matplotlib for plot visualization
  - cv2 for image processing
  - sklearn for D matrix construction in Sparse Coding

12 References

References used in addition with papers and class notes:

- http://ufldl.stanford.edu/wiki/index.php/Sparse$_{C}oding$
- http://web.ipac.caltech.edu/staff/fmasci/home/astro$_{r}efs$ $/Digital_{I}mage_{P}rocessing_{2}ndEd.pdf$
- https://jaan.io/what-is-variational-autoencoder-vae-tutorial/
- https://www.mia.uni-saarland.de/weickert/Papers/book.pdf
- https://staff.fnwi.uva.nl/r.vandenboomgaard/nldiffusionweb/nldiffusioncode.pdf