Techniques in Image Denoising

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

October 11, 2018

Contents

1 Introduction

2 Measurements

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, ∞), 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

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

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 Linear Diffusion Models

3.1.1 Model Explanation

3.1.2 Results

3.2 PM model

3.2.1 Model Explanation

3.2.2 Results

3.3 CPM model

3.3.1 Model Explanation

3.3.2 Results

3.4 Edge Enhancing Diffusion

3.4.1 Model Explanation

3.4.2 Results

3.5 Coherance Enhancing Diffusion

3.5.1 Model Explanation

3.5.2 Results

4 Noise Level with cleaning performance

5 Effect of Noise distributions

5.1 Gaussian Distribution

5.2 Poisson Distribution

5.3 Uniform Distribution

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.

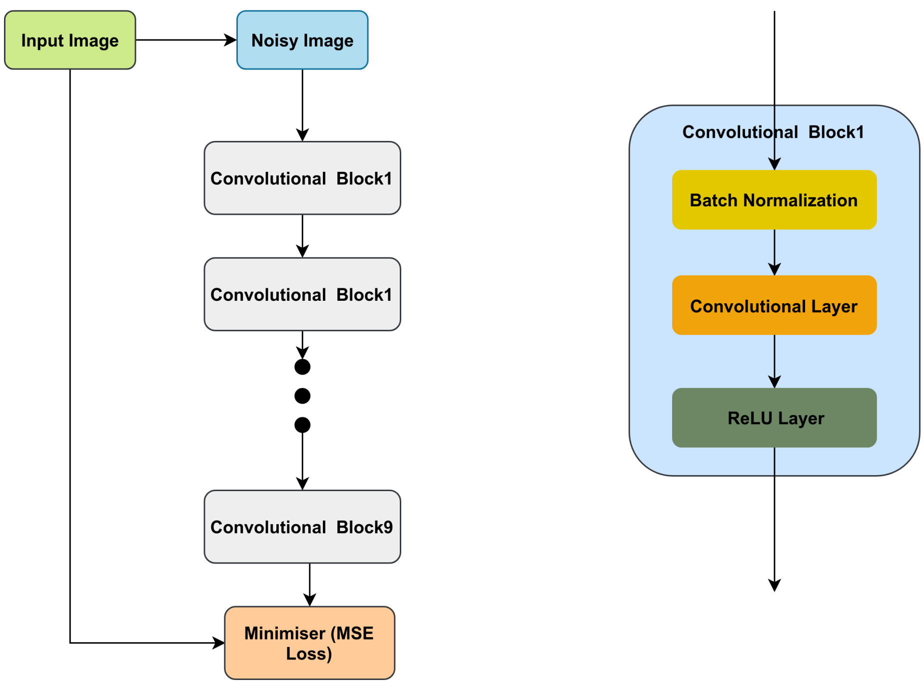
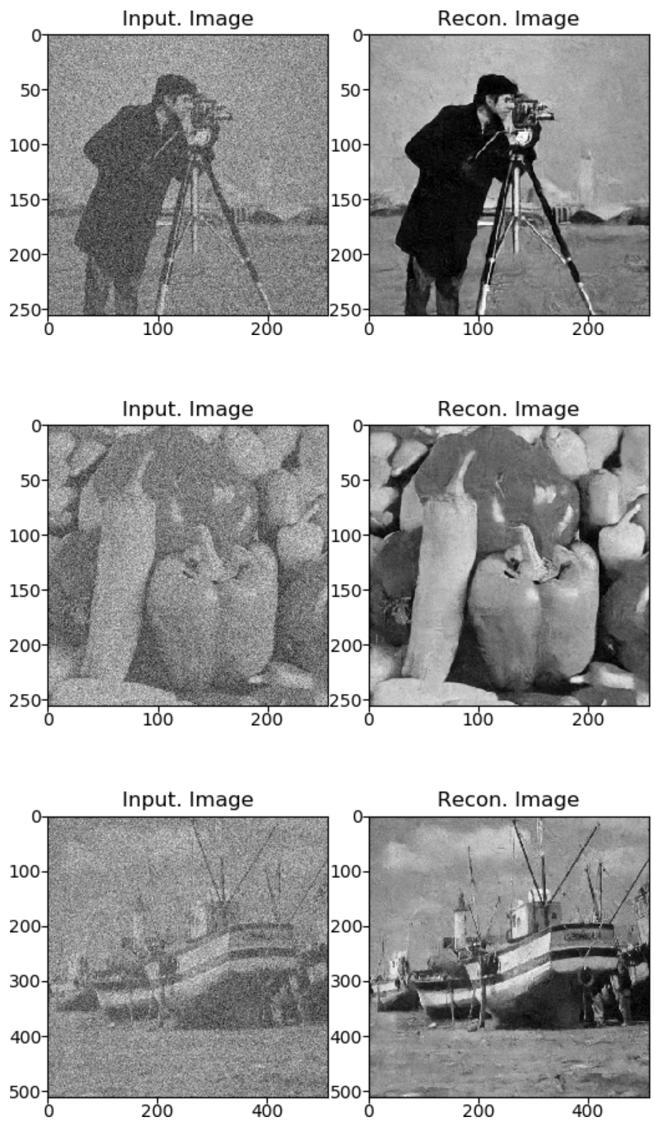
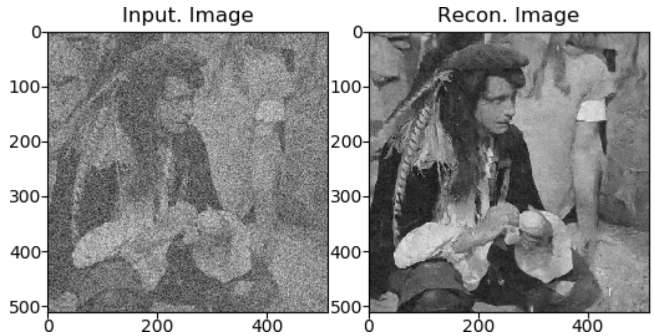
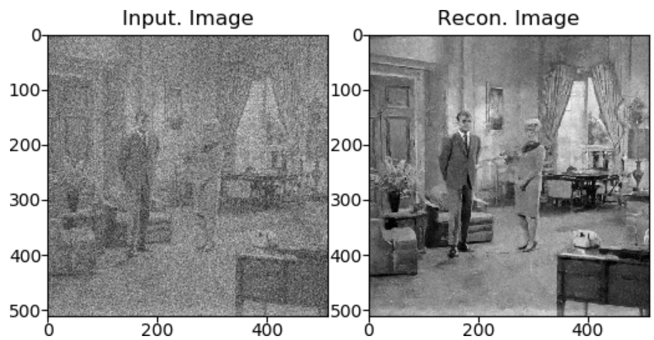


Figure 2: Convolutional neural netowrk architecture used

Results obtained



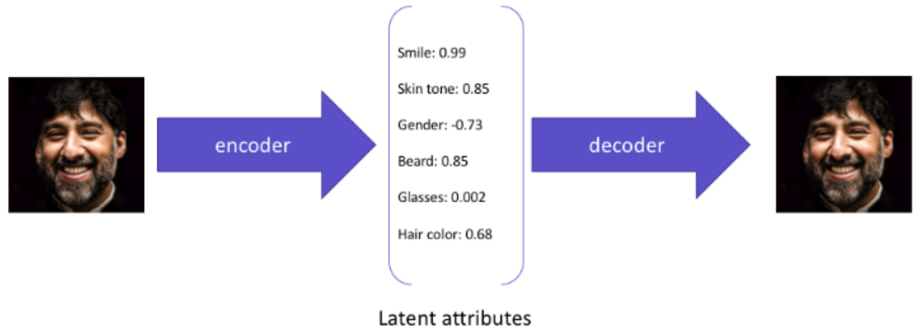




6.3 Variational approach: Convolutional Variational Autoencoders

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.

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



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.

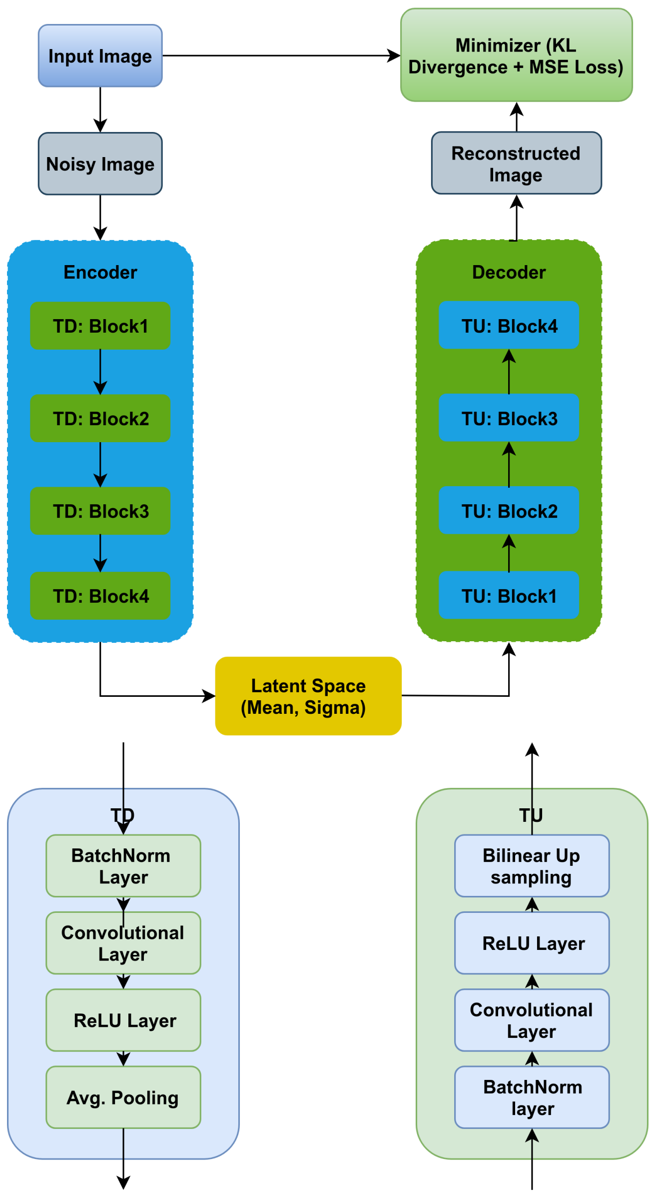
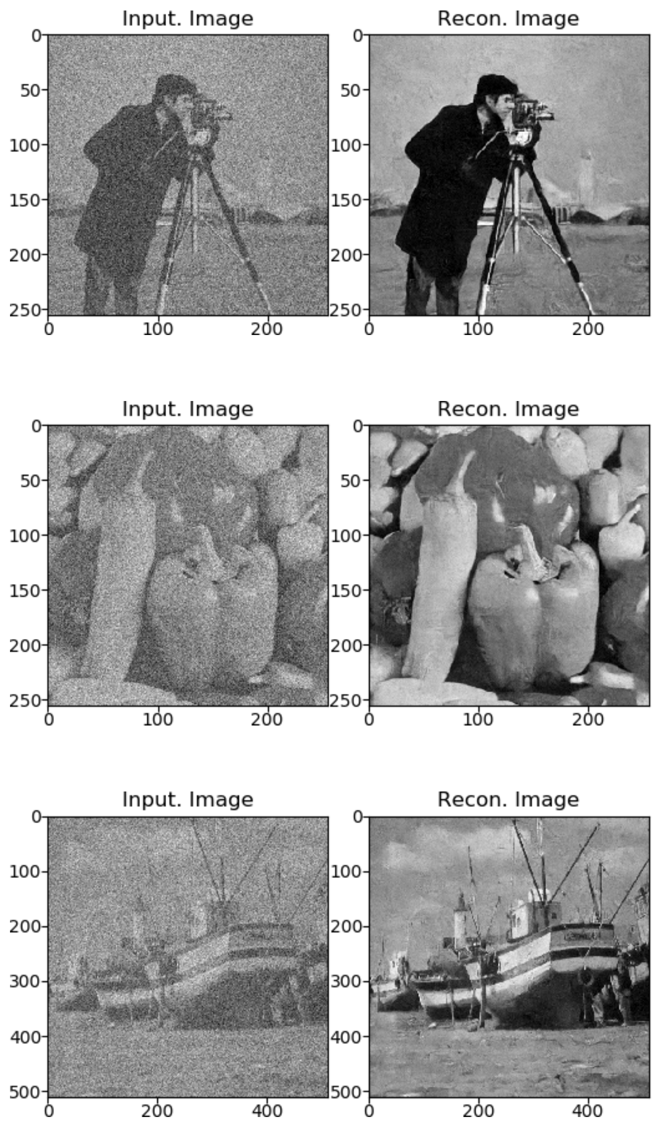
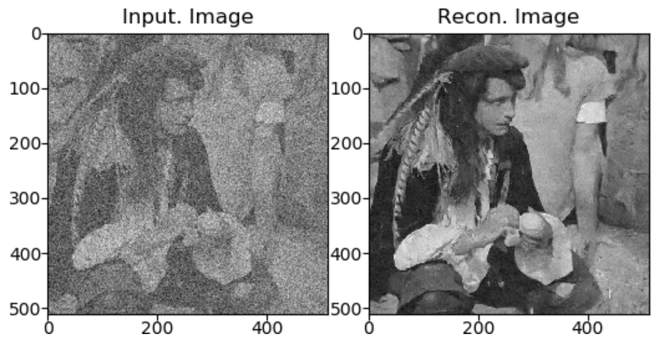
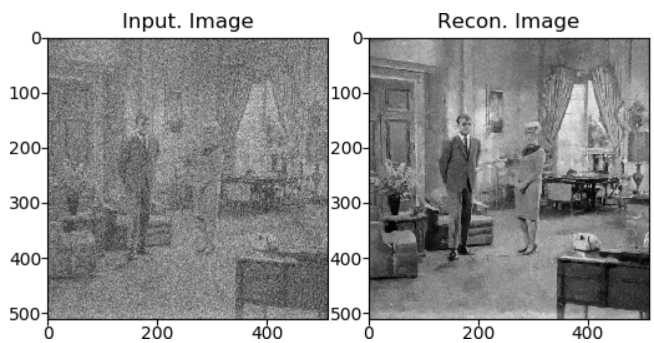


Figure 3: Convolutional Variational Autoencoder netowrk architecture used

Results obtained







7 Comparison between white box and black box models

8 Lambda-Sigma relation

9 Code availability and structure

This paper 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

References