CONVOLUTIONAL NEURAL NETWORK BASED IMAGE DE-NOISING

By

Usman Abdullah

204345 NS

Supervisor

Col(R) Dr. Imran Tauqir

A thesis submitted to the faculty of Electrical Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in Electrical Telecommunication Engineering

Feb 2021

Declaration

I, Usman Abdullah declare that this thesis titled "Convolutional Neural Network based Image De-noising" and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST.

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated.

3. Where I have consulted the published work of others, this is always clearly attributed.

4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

5. I have acknowledged all main sources of help.

6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Usman Abdullah,

204345 NS

DEDICATION

This thesis is dedicated to

MY FAMILY, FRIENDS AND TEACHERS

for their love, endless support and encouragement

Abstract

De-noising of an image is a cardinal pre-processing step for image scrutiny and computer vision such as visual tracking, image registration, image classification, image segmentation and image restoration. The reasons of using Convolution Neural Network (CNN) based image de-noising are CNN with very broad design is successful in improving the versatility and efficiency to leverage image resources. Second, substantial progress has been made on regularizing and practicing strategies for CNN instruction, including the Rectifier Linear Unit (ReLU), batch normalization, and residual practicing. In CNN such approaches may be implemented to simplify the training cycle and boost the de-noising efficiency.

Usually, all of those strategies suffer from two big pitfalls. First, in the testing level, such approaches usually have complicated question of optimization, which would make the procedure time consuming. Therefore, most approaches without losing numerical efficiency can hardly attain high results. Furthermore, the models are usually non-convex and require many parameters selected by hand, CNN centered simple discriminative learning approach is used to solve such issues, in which noise is isolated from a discrete picture by feed-forward convolutionary neural networks.

Discriminative learning based image de-noising has been attracting considerable attention due to its fast extrapolation and good performance. Wavelet decomposition enhance the performance and minimize the computational complexity of this process. It provides us best compression ratio without degrading the quality of image. Convolutional neural network based image de-noising use batch normalization and residual learning which help to speed up the training process and accelerate the convergence of the network. Latent clean image extracted using residual learning from hidden layers. De-noising Convolution neural network (DnCNNs) model able to handle multiple Gaussian noise level as well as blind noise level. Several experiments are performed and compared with Avant-grade de-noising method to evaluate the de-noising performance and complexity of the network. The result shows that neural network based image de-noising is effective and efficient for practical image de-noising applications.

ACKNOWLEDGEMENTS

I am very thankful to Almighty Allah who give me strength to achieve this goal.

Table of Contents

1.1 Problem statement 13
1.2 Objectives 14
1.3 Contributions 14
1.4 Applications 14
1.5 Thesis Outline 14
2.1 Machine learning 17
2.2 Neural Networks 17
2.3 Convolutional Neural Networks 19
2.3.1 Convolution layer 20
2.3.2 Pooling layer 21
2.3.3 Non Linearity (ReLU) 22
2.3.4 Fully Connected layer 22
2.3.5 Batch normalization in neural networks 23
2.5 Markov random field (MRF) 24
2.6 Block matching and 3D filtering 25
2.7 Non-local self-similarity 25
2.7.1 WNNM for image de-noising 26
2.7.2 Patch group based NSS prior Learning 27
3.1 Network Depth 30
3.2 Network Architecture 30
3.3 De-noising on sub-image 31
3.4 Boundary Artifacts 31
3.5 Residual Learning and Batch Normalization 32
4.1 Network dataset training setting 35
4.2 Training and Testing Data 35
4.3 Network Training 36

4.4 Compared Methods 36
4.5 Quantitative and Qualitative measures 39
Bibliography 51

List of Figures

figure 2. 1 Simple block diagram of Neural Network 18
figure 2. 2 two-layer Neural network 18
figure 2. 3Architecture of Convolutional Neural Network 19
figure 2. 4 feature extraction of convolution layer 20
figure 2. 5 some common filters used for features extractions 21
figure4. 1 Set of 12 images used for testing 35
figure4. 2De-noising result of image from widely used data set with noise level 25 37
figure4. 3De-noising result of image from widely used data set with noise level 50 38
figure4. 4 de-noise result of a color image 38
figure4. 5 de-noise result of a color image 41
figure4. 6 de-noised result of a color image 421
figure4. 7 de-noised result of a color image 42
figure4. 8 de-noised result of a color image 42

List of tables

Table 3.1 Effective patch size for different techniques ……………………………………………………………..……..……30

Table 4.1 PSNR(dB) results of various methods on widely used testing images…………………………….……40

Table 4.2 Average PSNR results of various methods on BSD68 dataset……………………………………..……..….41

List of Acronyms

Convolutional Neural Networks	CNN
Rectified Linear Unit	ReLU
Short Term Fourier Transform	STFT
Continuous Wavelet Transform	CWT
Discrete Wavelet Transform	DWT
Non-local Self Similarity	NSS
Weighted Nuclear Norm Minimization	WNNM
Patch Group	PG
Gaussian Mixture Model	GMM
Patch Group Prior based De-Noising	PGPD
Peak Signal to Noise Ratio	PSNR
Visual Geometry Group	VGG
Block Matching 3-dimensional Filtering	BM3D
Trainable Non-linear Reaction Diffusion	TNRD
Multi-layer Perception	MLP
De-noising Convolutional Neural Networks	DnCNN
Structural Similarity Index Measure	SSIM
Expected Patch Log Likelihood	EPLL

Chapter 1

Introduction

Image de-noising is a vintage active problem in deplorable vision since it is a consequential step in practical applications such as human vision and image analysis. The aim of image de-noising is to extract a latent clean image x from clatter observation Y follows the equation, $Y = x + v$ where v is the additive white Gaussian noise(AWGN). As we know that noise corruption is unable to avoid and it degrade the image visual quality. Second the image prior model plays an important role in image de-noising. For Bayesians perspective various model has been purposed for image prior model like Non Local self-similarity models discussed in [1] [2] [3] [4] Sparse model based image de-noising [2] [5] [6] [7] Gradient based models discussed in [8] [9] and Markov random field based algorithm explained in [10] [11] [12].

Image captured by digital camera is clean at low sensitive grade but it becomes noisy for consumer grade like mobile camera at high sensitives to address this problem simultaneous sparse coding is used in [3] for image restoration. In super resolution multiple low quality images fused into a single image with better resolution quality non-local means based algorithm discussed in [2] for super resolution. Another non-local means based algorithm discussed in [1] for image de-noising in which non-local averaging values of all pixels are used for noise removal. Sparse coding noise can be defined difference between sparse code of original image which is unknown and the sparse code of degraded image to improve the performance of sparsity we can minimize this kind of noise. This problem discussed in [6] in which sparse noise is estimated and then non-local means algorithms are used for image de-noising.

Image de-noising via sparse and redundant representation discussed in [5] in which trained dictionary is used for learning image content effectively. The training algorithm is based on single corrupted image or multiple images. To measure the sparsity of an image patch an important technique discussed in [7]. In this letter adaptive dictionary is used to measure the sparse coefficients of each group. Sparsity of an image analyzed by taking the singular values of each group. For better estimation of real singular values for each group weighted Schatten ρ-norm minimization method is used.

Markov random field is used for color image unsupervised segmentation [12]. This paper involved Euclidean distance of intensity values and spatial location information of neighboring pixels. Fuzzy C-means based algorithm is used to specify the class number. The optimal class number is chosen by minimum message length criteria for unsupervised segmentation completion to enhance the better relationship between pixel values and distance between neighboring pixels. It is hard to model the image prior because relationship between pixel values are complicated. To overcome this issue Markov random field based method discussed in [11].

Although above methods have the ability to extract a clear image from clatter observation but these methods suffer some drawbacks. First, these methods generally have a problem called complex optimization at initial training stage it makes the de-noising algorithm time consuming.

Secondly, these methods are non-convex in nature and parameters like window size, patch size and central pixel weight are not estimated automatically so it does not boost up the de-noising performance of these algorithms.

To overcome the above drawbacks many discriminative learning algorithms are used to learn the image prior model and able to get rid of manually parameters estimation. CSF [13] [14] learn an image prior by fixed number of gradient descent inference steps. Although these two methods best performed in terms of computational complexity and boost up the de-noising performance but the problem is these methods are useful only for specific prior model and limited for blind image de-noising.

Another plain discriminative learning method DnCNN [15] achieve competitive de-noising performance. This method is limited in flexibility and customized to a specific noise level. From regression stance, the focus in the said work is to learn a function $x = \mathrm{F}(y; \Theta)$ between noise y and x, whereas Θ is a trained model parameter for fixed noise level. But the said de-noising method lacks flexibility to deal with spatially invariant noise levels. Fast and flexible convolutional neural network (FFDNet) [16] is composed of $x = \mathrm{F}(y, M; \Theta)$ where M is a noise level map. In this method, noise level modeling is done at input stage where model parameter Θ is invariant to noise level. Moreover, haar wavelet compression reduces the computational complexity of the proposed de-noising model without degrading the image quality, with the ability to handle multiple noise levels in a single network.

In this thesis clean image recover from noisy image by feed-forward convolutional neural networks (CNN). CNN effectively increase the flexibility and capacity of image characteristics. This method based on batch normalization residual learning at training stage for better feature extraction and textures as well as increase the performance of algorithm.

1.1 Problem statement

Most of the method generally involve a complex optimization problem in the testing stage, making the de-noising process time consuming. Second, the models in general are non-convex and involve several parameters like window size, patch size and central pixel weight, are not estimated automatically so it does not boost up the de-noising performance of these algorithms. In this thesis deep convolutional neural networks based image de-noising method is explained in which noise is separated from noisy observation to recover a clean image. Batch normalization is used to increase the de-noising performance as well as stimulate the algorithm with the help of Residual learning. this method handles the noise with different noise levels. Wavelet decomposition is used to compress the input image so that processing speed is increased and with minimum time we get a clean image from noisy image.

1.2 Objectives

The objective of thesis is:

- capacity to handle spacious range of noise levels.
- Faster speed then many ultra-modern de-noising methods without degrading the de-noising performance.

1.3 Contributions

The contribution of this work minimize the computational complexity and handle noise with multiple noise without degrading the noise performance. These are summarizing as:

- A fast and efficient convolutional neural network is purposed for image de-noising. A single network able to handle multiple noise level map.
- Throughout trainable deep convolutional algorithm estimate a latent clear image. This network arrogates the residual learning approach to extract a latent clear image from clatter observation.
- In this thesis, the combination of residual learning and batch normalization speed up the training process and improves the de-noising efficiency in terms of both quantitative and visual consistency.

1.4 Applications

De-noising of an image is an important pre-processing step for image scrutiny and computer vision such as visual tracking, image registration, image classification, image segmentation and image restoration. Our goal is to recover clear image from clatter observation with minimal damage of information.

1.5 Thesis Outline

The thesis accommodates five chapters:

- Chapter 1: In this chapter introduction and objectives are discussed along with the major contribution and application of this technique.
- Chapter 2: Background and literature review Neural network and deep convolutional neural networks discussed in this chapter along with previous techniques used for image de-noising.

- Chapter 3: This chapter is about convolutional neural network based method for image de-noising is discussed in which integration of batch normalization and residual learning are used for enhancing the performance and speed up the training process.

- Chapter 4: Confers experimentation and analysis of results in order to evaluate our methodology over previous proposals for image de-noising.

- Chapter 5: This chapter conclude the given method for image de-noising.

Chapter 2

Background

The theory related to machine learning and deep convolutional neural networks discussed in this chapter. Later on some previous techniques which are using for image de-noising for further image analysis and computer vision such as visual tracking, image registration, image classification, image segmentation and image restoration are explained. This chapter will help to understand the convolutional neural networks and wavelet based decomposition.

2.1 Machine learning

Machine learning is an important subfield of artificial intelligence and become widely used in various applications. it is started when Alan Turing raised a question ‘’can machines think’’ [17]. Mitchell in his book defines learning as: " A computer program is said to benefit from experience E about any task class T and performance evaluate P if its success at tasks in T, as calculated by P, increases with experience E " [18]. Machine learning is the application of artificial intelligence in which system automatically learn from previous experiments without any program. In machine learning popular areas of interests are recommended system if you like this you also like this classification of different peoples based on some attributes. In social network predict the peoples who interest in specific domains of websites, movies videos and articles. In weather prediction based on previous weather condition using machine learning we estimate crop growth, stock market estimation and many more in real life related estimations.

2.2 Neural Networks

A beautiful programming framework influenced by biology that helps one machine to learn from empirical results. A neural network comprises of hidden layers of the input layer and hidden layers of the output layer can be more than one weight, so limits are added to hidden layers to achieve a particular form of decision taking. Neural networks information process scenario is same like human brain. These networks learn from example and cannot be trained to perform a specific task. The model should be trained very carefully otherwise result may be worse or incorrect and unable to solve the problems and operations. Fig 2.1 shows the simple block diagram of neural networks.

Conventional computer uses an awareness approach to solve a specific problem. We instruct the computer through some algorithm and translate this algorithm into machine language and then machine can solve the problem in a different way. Neural networks and conventional computers algorithms are complement of each other. Some task is performing well through computers and some task outperforms through neural networks. Sometimes both neural networks and computers perform the task in which computer supervise the neural network to achieve a maximum efficiency. An example of three layers’ neural networks can be shown in figure 2.2.

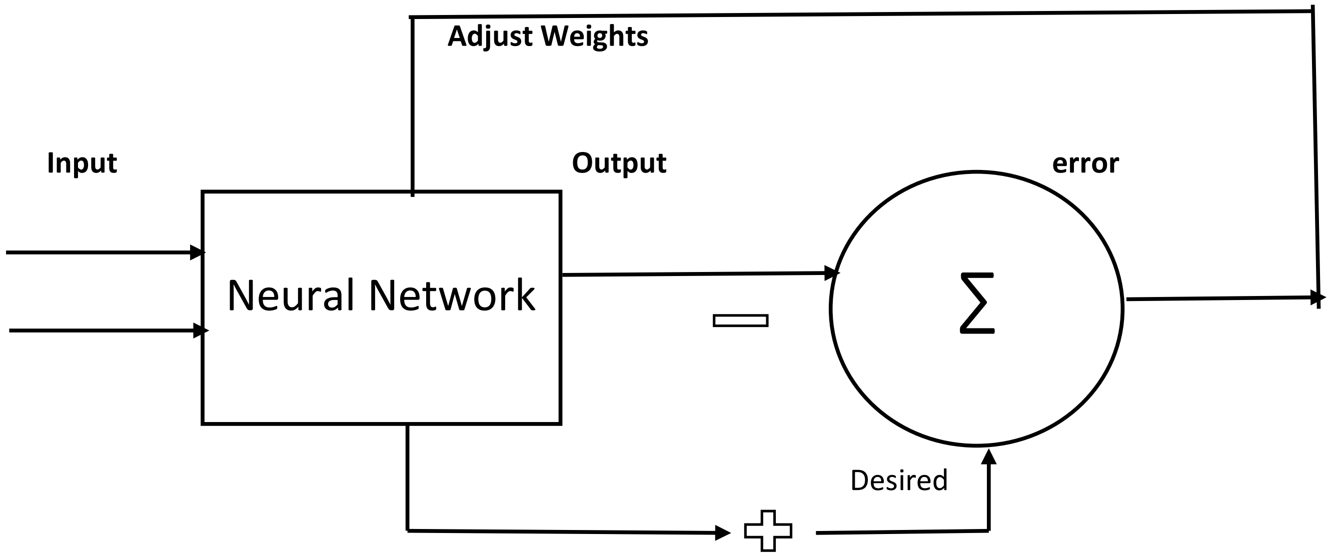


figure 2. 1 Simple block diagram of Neural Network

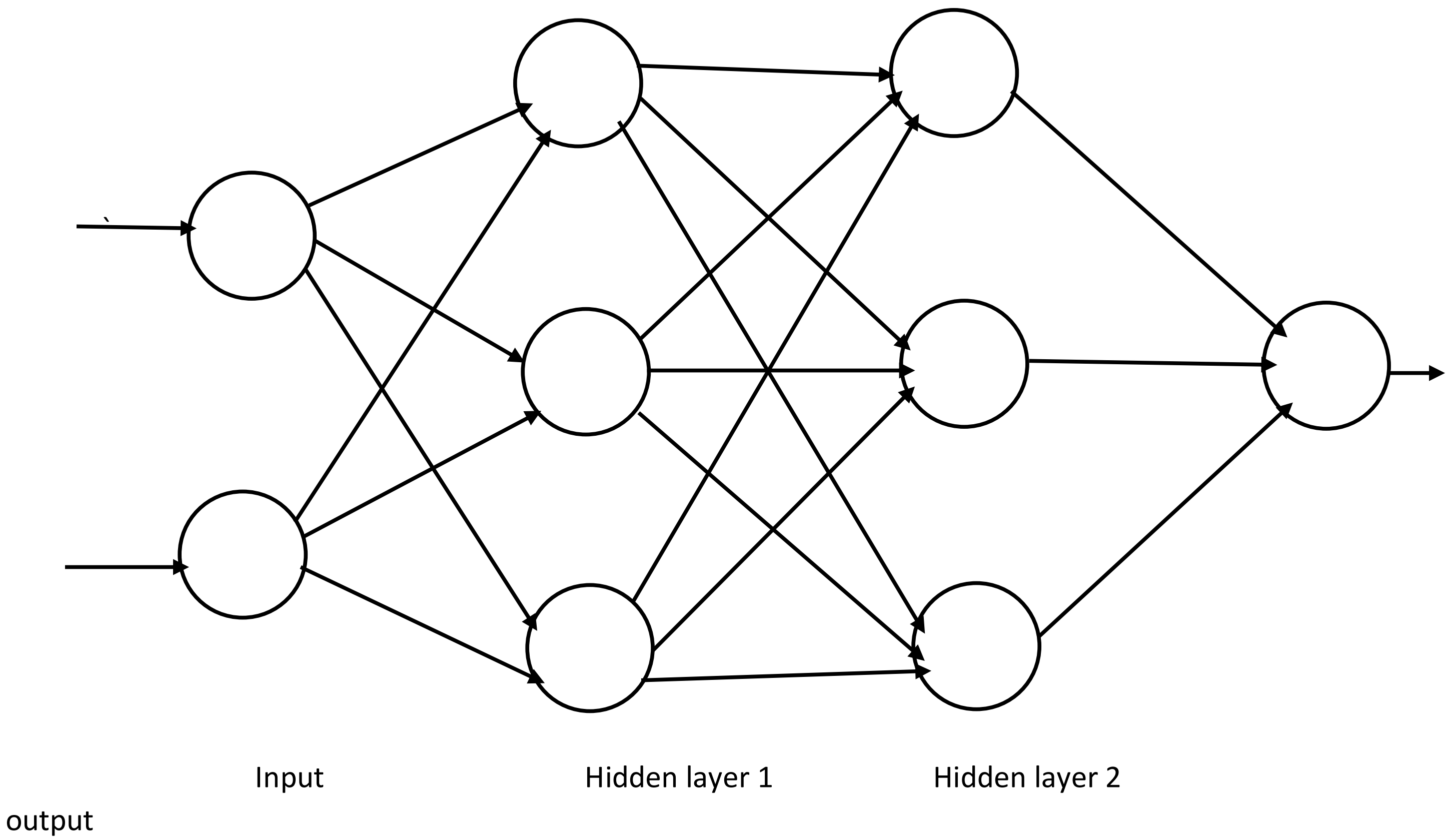


figure 2. 2 two-layer Neural network

2.3 Convolutional Neural Networks

Convolutional generally called a CNN is a feed forward neural networks which is popularly used to analyze an image as well as in other classification and data analysis problems. CNN picks out and detect a pattern and make sense of them for next processing steps this pattern detection makes it more useful for image analysis. Convolutional layer receives an input image in the form of matrix array and transform it this layer output becomes next layer input. In this layer all operations are based on convolution. In hidden or convolution layer filters are used to detect the patterns filters are used to extract edges, textures, shapes, circles and irregular changes and in deep layers some filters are able to detect specific object eyes, hairs, feathers ears etc. in more deep layers some filters are able to detect more specific objects like birds, lizards and cats etc. filters used in this layers are in matrix form normally these filters are 3×3, values of these filters are initialize randomly. Input of an image convolve with these filters and pass the output to next layers some weights and threshold defines at each node these weight and threshold are multiplied and added respectively with output values before process to next layers and at output stage value is compared if there is some error then feedback system is used to set new weights and threshold on the bases of error function. CNN model trained for different set of image in which all possible shapes, textures and edges are under consideration.

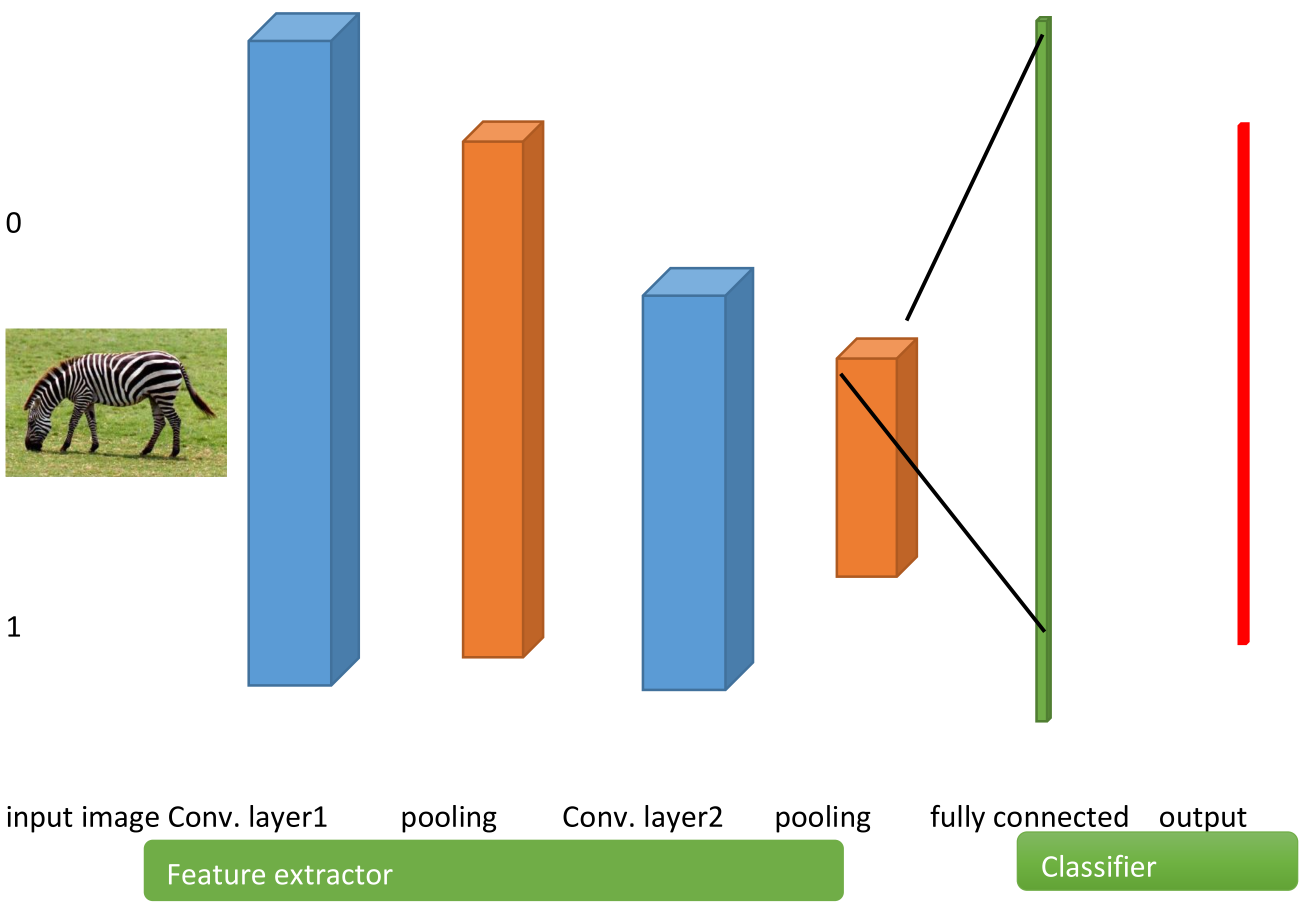


figure 2. 3Architecture of Convolutional Neural Network

2.3.1 Convolution layer

It is the first layer of CNN in which input image in the form of matrix convolve with kernel. The primary objective of this layer is to extract the features of input images. It takes two input one is image matrix and second is filter in matrix form. Image features learn by convolution and work with pixel coordination of input image by using small square of input data. Consider a 5×5 matrix whose image pixels values are 0 and 1 and a kernel of size 3×3 as shown below.

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 4 & 2 \\ 2 & 3 & 3 \\ 3 & 3 & 5 \end{bmatrix}$$

Image matrix Filter matrix Convolved feature

figure 2. 4 feature extraction of convolution layer

When image matrix multiplied with kernel the resulting matrix is called feature map output matrix shown in fig 2.4. The above computation is done when filter matrix slid over input image matrix on single matrix, known as stride. Element wise multiplication is performed in above operation then sum up these values the resulting single value put in output convolved matrix. When kernel slid over input matrix the resulting matrix is called feature map or activation map or convolved feature. When we have different filters the resulting feature map has different values even on same image. Some filters extract horizontal, vertical, diagonal feature it only depends on filter. In figure 2.5 different types of filter are applied on same image to detect horizontal, vertical, diagonal edges and also sharpen and box filters are applied on same image. As various filter forms move over the height and width of an input picture it creates the receptive field's image representation. The kernel is spatially smaller than an image, yet the kernel depth is marginally greater than the matrix of the input picture. It implies that using different kinds of filters convolve with input image matrix results different features maps. The values of these filters sets at training stage of neural networks using greater filters convolve with input matrix help us to extract more feature like edges and different types of shapes. It helps to get more feature and our network is more efficient and identify pattern in new image.

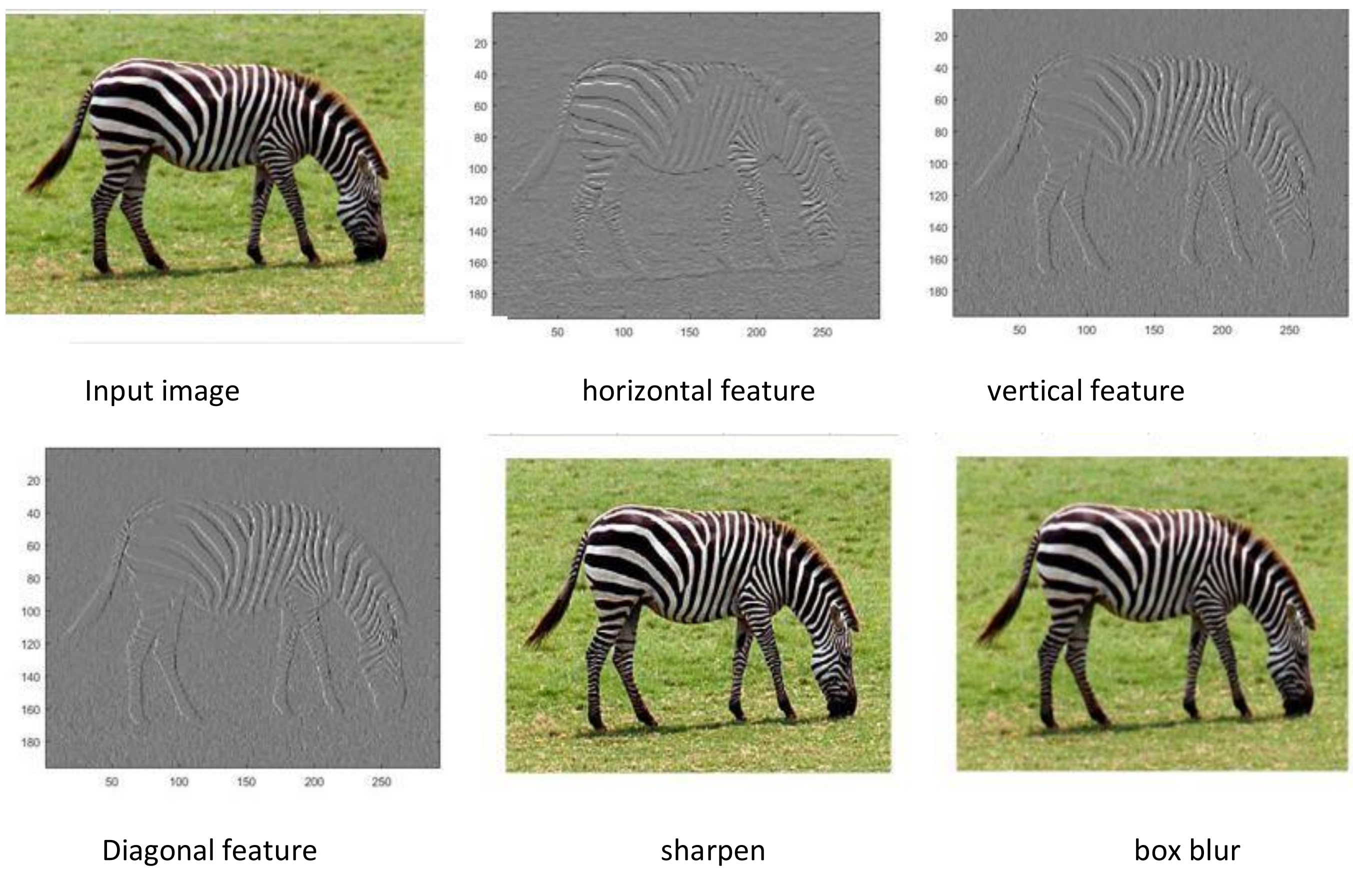


figure 2. 5 some common filters used for features extractions

Filter is slide over pixels of input image when filter slide over one pixel at a time we say that stride is 1 and when stride is two filter moves 2 pixels at a time of input image. Sometimes filter does not cover all pixels of input image and we drop that pixels and consider only valid part of image this may cause loss of information to retain maximum information we pad zero at input image so that size of filter and image fit together and we extract maximum useful information.

2.3.2 Pooling layer

Pooling layer uses to limit the parameters where the picture scale becomes very broad spatial pooling used to subsample or decrease the dimension of each map and preserves the initial image pooling knowledge that might be max. Pooling in percentages and quantities. First we consider the window of pixels e.g. 2×2 in max pooling takes the largest element of the rectified future map in that window in average pooling average pixels value in that particular window and in sum pooling we take sum of all the elements in a feature map of given window. In pooling layer, we extract the features dimension which are very small so it is used to manage the feature extraction of small dimensions. To control the overfitting and reduce the computation and parameters it also helps to enhance the efficiency of the network. When there is a very small object placed in our image and filter does not recognize the object and placed zero value against this small pooling layer shows the small object inside the image irrespective of the fact where this small object is placed.

2.3.3 Non Linearity (ReLU)

Real world data wants our model to learn non-negative values of the data so we introduce non linearity in convolutional model. Convolution is a linear process matrix multiplication and pixel wise multiplication but in real life data is not linear so we introduce some non-linearity. There is a rectified linear unit after every convolution layer which applied on every pixel and set negative values of pixels to 0. The rectified linear unit is the most of the common used activation function in deep learning models. This function returns the zero value if the output of function is negative value and return the same value if the output is positive value so we can define the $f(x) = \max(0, x)$. A function is said to be non-linear if the slop isn't constant so rectified linear unit have zero value in negative interval of x and it remains constant when the value of x is positive. There are many non-linear function used in deep learning such as tanh and sigmoid but rectified linear unit is better than other two so this non-linear function we used after every convolution step.

2.3.4 Fully Connected layer

That neuron in this layer is completely associated with previous neurons in the layer. Function map matrix transformed to a vector from previous layers. Such feature maps help the layer build a layout at the end where we provide an activation function such as Softmax or sigmoid for classification of pictures. The fully connected layers decide on the bases of training dataset and classify each and every image. Sum of all the possible outputs of the Fully Connected Layer comes as 1. The Softmax or sigmoid activation function work on arbitrary value vector and transform these vector value between zero and one. Convolution neural networks used to predict the right values of the data and put them into classification summarize as:

- Input image provide to convolution layer.
- Parameters adjustments, kernel with stride, padding, convolution on image and rectified linear unit after every convolution step.
- Pooling help to reduce dimensions.
- smoothen the output and feed into a fully connected layer (FC Layer)
- Output the image using an activation function and perform image classifications.

2.3.5 Batch normalization in neural networks

During neural network training some preprocessing steps is taken for normalization of input data so that all data should be in same range. For example, if we have some data in 0 to 1 range and other data in 1 to 1000 range we normalize these data in same distribution. The problem is arising when all layers work on same distribution data and at activation layer distribution is constantly changing during training. This slow down the training process because each layer learns themselves at training process to new distribution. When the training and testing data set come from difference sources both have different distribution the behavior of machine learning algorithm changes when the input distribution is different the problem is known as covariate shift.

In neural network main concern is change of distribution of input layer to inner layers during training process due to activation function the weights of each layer changes according to requirement so the distribution also changes at each layer. We know that output activation of every layer is input to the next layer, every layer in neural network faced the issue where distribution is changes at every step. Basic idea behind batch normalization is that activation function of every layer changes into a normalized value with 0 mean and unity variance this will also help to reduce covariate shift. This also help each layer to learn on most stable distribution and increase the speed of training process.

Batch normalization is used for the stability of neural network in which output of previous activation function subtracting from batch mean and divide by batch standard deviation. During training process following steps are involved in batch normalization:

1. Calculate the input layer batch mean and batch variance

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma^2{}_B = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2$$

2. Normalize each layer input using previously statics

$$\bar{x}_i = \frac{x_i - u_B}{\sqrt{\sigma^2{}_B + \epsilon}}$$

3. Scale and shift for obtaining output of layer

$$y_i = \gamma \bar{x}_i + \beta$$

γ and β learn during training model with original parameters of network mean and variance are fixed they are estimated through each training batch. In short we can say that batch normalization help the network to learn batch parameters γ and β and convert the value of mean and variance in our desired values.

2.5 Markov random field (MRF)

The success of MRF is based on stochastic image model because it provides good and flexible image prior model. MRF define image model not only on intensities values but also used different hidden attributes like textures, edges and region labels. Bayesian estimation used in MRF it consists of two ingredients prior and degradation model. The prior model is based on the image attributes let say X. for example in edge preservation we take intensities values of image and set of binary values of edge label and in case of textures X include set of intensities values and texture label at each location. In degradation model observation are generated usually we are facing noisy and incomplete observation.

MAP estimation minimizing the posterior energy function in this minimization different kind of attributes are under consideration. In edge preservation the process X includes both continuous values of image intensities and binary values of edge variables. It’s very difficult to find the global minima due to non-trivial local minima to find the global minima a technique named as Simulated Annealing is used but its computational complexity is very intensive. The Maximum Likelihood method aren’t work on prior model. We concluded that the model with a higher temperature has significant performance over the temperature estimated by Maximum Likelihood method.

According to experiments, we can achieve better result if we set temperature coefficient double by MLP. The model parameters are learned by PCA and MLE. On the basis of this model, image de-noising by Bayesian analysis. This method applied on de-noising algorithm to fMRI and ultrasound images and attain very good de-noising results. But the problem is several parameters are chosen manually and size of the training set also have an impact on de-noising algorithm [12].

2.6 Block matching and 3D filtering

It's a 3D block-matching algorithm used for image de-noising. image has a sparse representation when transformation done from time domain to frequency domain this sparsity can be optimized by group of 2D image similar patches into 3Dgroups. Collaborative filter is used for grouping and filtering purpose firstly, similar image patches are determined to given image and transform into 3D blocks secondly, 3D linear transform of this 3D block thirdly, shrinkage of the transform spectrum coefficients and at the last inverse 3D transformations. By using collaborative filter, it reveals the finest details by the group patches.

Grouping can be done by K-means clustering to form a group first find the set of coordinates patches or blocks similar to original reference patch in below equation set S and Z_{X_R} is the location of reference patch Z_X is the patch at location x and some threshold γ where d is the dissimilarity we can write the equation of set as follows:

$$S_X = \{x \epsilon X : d(Z_{X_R}, Z_{X_R})| \leq \gamma$$

$$d\left(t_{X_R}, Z_X\right) = ||J^{-1}\Gamma J_{2D}\left(Z_{X_R}\right) - J_{2D}{}^{-1}\Gamma_{2D}(Z_X)||^2$$

The above equation perform the square Euclidean distance between the de-noised patches and Γ is hard threshold operator. In group de-noising collaborative filter is used in which each grouped fragment conspire for the filtering of all other and vice versa for this the equation of d+1 dimensional array holding the de-noised group can be written as:

$$\hat{Y}_{S_{X_R}} = J_{3D}{}^{-1}\Gamma J_{3D}(Z_{S_{X_R}})$$

This de-noised group is the collection of all patches. In accumulation each pixel has apparently seen member of many different groups. By this conspirator filter the patch around this pixel has modified in different way in every group. For reconciliation of all the estimation the weight assign to the inverse of noise because the noisy weighs are small so normalization is required at the end of stage. This method is applied on gray scale and color image in the specter of additive white Gaussian clatter result is impressive but several parameters are change when image scale is changed and noise level changed. This method design only for specific noise level and when spatial variant noise present in the image it does not perform well and information lost in the resultant output image.

2.7 Non-local self-similarity

Nonlocal self-similarity prior it broaches to certitude that local patch may have similar patch beyond the image. it enhances the de-noising performance many method has been used for this purpose like patch group based NSS prior learning in which patch group are extricating from training images which contain non local images and then patch grouped based Gaussian mixture model is used for non-local image prior and then simple weighted sparse coding is used for image de-noising. Weighted nuclear norms minimization for non-local self-similarity in image de-noising in which singular values are assign different weights. The solution of the WNNM are peruse under different weights conditions.

2.7.1 WNNM for image de-noising

The aim of image de-noising is to reconstruct an original clean image from clattered observation. In WNNM estimation of local patch is very basic step for this purpose similar patches extracted from original image. From these local patches build a patch matrix denoted by Y_J we can say that $Y_J = X_J + N_J$ where X_J is the original image matrix and N_J is the noisy matrix. Noisy matrix have low rank matrix which is extracted from original image matrix we can write the energy function as:

$$\hat{X} = argminX_J \frac{1}{{\sigma_n}^2} ||Y_J - X_J||_F{}^2 ||X_J||_{Wi^*}$$

The key issue in this equation is to find the weight matrix in natural image we already know the image prior model. The larger the singular values the lesser the shrunk so the idea to assign the weight of ιth singular value of original image matrix is inversely proportional to $\sigma_i(X_J)$

$$w_i = c\sqrt{n}/(\sigma_i(X_J) + \epsilon)$$

We suspect that noise energy is steadily distributed so that estimation of singular values is denoted by following equation:

$$\hat{x}_i(X_J) = \sqrt{\max({\sigma_i}^2(Y_i) - n{\sigma_n}^2, 0)}$$

At the end aggregate X_j to form the clean image. This method applied on every patch together to revamp the original image x.

2.7.2 Patch group based NSS prior Learning

A huge deal has been accomplished in low-level vision such as pixel de-noising image processing like patch focused. Exceptionally, use image non-local self-similarity (NSS) prior to, a local patch can have equivalent non-local patches in it in the picture such patches improve the de-noising results. Whereas only feedback degraded picture NSS is included in most current systems.

First figure out the non-local related patches from clean picture in this process and measure the mean set, then a patch set dependent Gaussian mixture model is used to know a prior sample image. At the training stage we have N patch groups each of these group has M patch so need to find out the covariance matrix. For image de-noising weighted sparse coding is used we know that patch group represent the variations of same patches in a group. The clean patch from patch can be extracted using below equation:

$$\hat{X}_m = D\ \hat{\propto} + \mu_y$$

Where D is the orthonormal dictionary $\hat{\propto}$ is the maximum a-posterior estimation μ_y is the mean of that patch and clean image $\hat{X}_m$ aggregating from all the estimated patch group. This method can be improving by several iterations to recover the clean image without losing the information.

While methods above have the potential to retrieve a coherent picture from the detection of clatters, such methods suffer several disadvantages. First, these approaches typically entail a complex problem of optimization at the initial stage of training which makes the de-noising algorithm time consuming. Second these methods are non-convex and lots of parameters are estimates manually so it does not boost up the de-noising performance of these algorithms. To overcome the above drawbacks many discriminative learning algorithms are used to learn the image prior model and able to get rid of manually parameters estimation. In next chapter discriminative learning method is explained in which clean image recover from clatter image by CNN. CNN effectively increase the flexibility and capacity of image characteristics. This method based on batch normalization residual learning at training stage for better feature extraction of almost every types of edges and textures as well as increase the performance of algorithm.

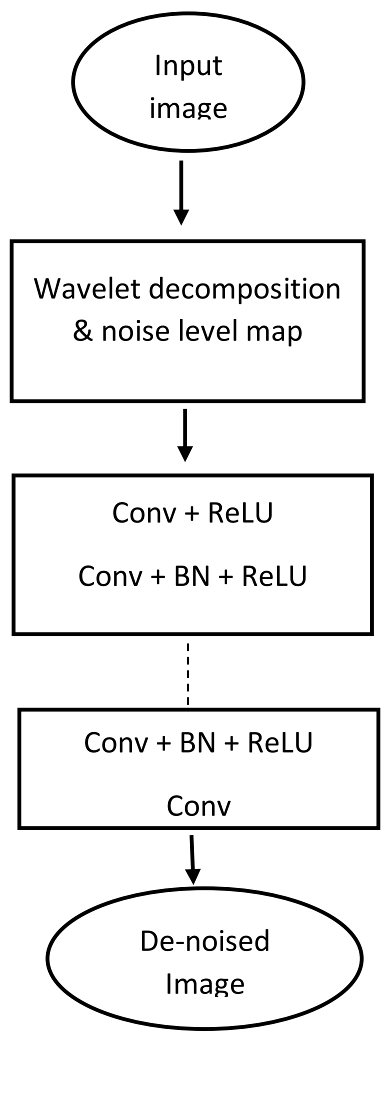
Chapter 3

Convolutional Neural Network based Image De-noising

In this chapter CNN based image de-noising method is explained. In general, deep CNN model training involved two steps: (a) Network architecture design (b) Model learning. For network architecture and effective patch size we modify the method given in VGG [19] pre-owned in most of the ultra-modern de-noising methods. Batch normalization and residual learning used for model training both of these two improve the de-noising performance as well as boost up the algorithm speed. The proposed method achieves the following objectives;

- Fast speed: Highly efficient without degrading the de-noising performance.
- Flexibility: algorithm is flexible for different noise levels and image characteristic.
- JPEG de-blocking: reduce the effects of compression artifacts.

figure3. 1 Block diagram of proposed algorithm



3.1 Network Depth

The principle explained in [14], convolutional filter size to be 3×3 and there is no pooling layer in CNN. Therefore, with depth d the DnCNN receptive field should be $(2d + 1) \times (2d + 1)$ Properly chosen for better trade-off between productivity and machine complexity results. There is an interrelation between receptive field size and constructive patch size in a de-noising neural networks. Moreover, exorbitant noise level requires enormous patch size to extract more information about edges, surfaces for image restoration. The effective patch size for different de-noising method can be determined by fixing the noise level $\sigma = 25$. In BM3D [20] effective patch size window is 25×25 and use two time iteratively so the final constructive patch size is 49×49 [19]. Similar to BM3D, WNNM [21] uses a large scale of window with iteration so the effective patch size for this algorithm is 361×361 . In MLP [22] 39×39 window size is used to generate the constructive patch size then 9×9 filter is used to average the output patch thus the overall constructive patch size become 47×47. Both CSF [13] and TNRD [14] use a patch size of 61×61 and uses 10 convolutional layers in five stages.

Table 3.1 shows the constructive patch size for various techniques with noise level $\sigma = 25$ table shows that the effective patch size for EPLL [23] is 36×36 is the smallest. In our DnCNN method the effective patch size is set to 35×35 with corresponding network depth 17 for better performance.

Technique	BM3D. [20]	WNNM. [21]	EPLL. [23]	MLP. [22]	CSF. [13]	TNRD. [14]
constructive patch size	49×49	361×361	36×36	47×47	61×61	61×61

Table 3.1

Effective patch size for different techniques with noise level $\sigma = 25$

3.2 Network Architecture

Block diagram shows the network architecture. Approximate image of input image x extract using wavelet decomposition and also noise is added this noise image passes through the neural network. $y = x' + v$ where Y is noisy image x' represent the approximate coefficient extracted from wavelet decomposition v is the noise. Aim of discriminative de-noising method such as CSF [13] and MLP [22] is to predict the latent clear image from clattered one by mapping function $F(y) = x'$. in this method residual learning is used for prediction of latent residual image so residual mapping $\mathcal{R}(y) = v$ and then we have $x' = y - \mathcal{R}(y)$ the average means square error betwixt the estimated and residual image.

$$l(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} ||\mathcal{R}(Y_i : \Theta) - (Y_i - x_i)||^2$$

Above equation is for learning a trainable parameter Θ in above equation N represent the noisy clean patch of input training images. In block diagram of network architecture residual learning use to recover a latent clean image.

By considering the complexity and balance property we set the convolutional layers for gray scale and color scale images. For gray scale image 15 layers and 12 layers for color scale image. For very first layer Conv. plus ReLU 64 filters of size 3×3×c is used for feature maps c represents the image channel for gray scale image c=1 and for color image c=3 are used. For second layer 64 filters of size 3×3×64 are utilized and batch normalization is used between convolutional and rectifier linear unit. The reason for different setting for color and gray scale image is twofold. In color image number of input channels is greater than gray scale image so more filters are used for feature extraction.

3.3 De-noising on sub-image

Efficiency of Neural network based image de-noising is an important issue one idea is, simply reduce the depth of architecture by reducing number of filters in hidden layers which causes receptive field problem in network. The traditional way of subsampling makes the image blurry so we should have kept the edge information for this purpose an important technique is used in this algorithm is wavelet based image compression for this purpose Haar wavelet is used. It provides us best compression ratio while quality of image is remaining same. We use the approximate coefficient of input image for further de-noising process. It helps us to reduce the computational complexity and memory burden. De-noising on compressed image can also expand the receptive field.

3.4 Boundary Artifacts

Convolutional generally called a CNN is a feed forward neural networks which is popularly used to analyze an image as well as in other classification and data analysis problems. CNN picks out and detect a pattern and make sense of them for next processing steps this pattern detection makes it more useful for image analysis. Convolutional layer receives an input image in the form of matrix array and transform it this layer output becomes next layer input. In this layer all operations are based on convolution. In secret or convolution layer filters are used to detect patterns filters are used to remove lines, curves, forms, circles and unusual shifts and in deep layers other filters may detect particular entity heads, fur, feather ears etc. in deeper layers certain filters may detect more common artifacts such as dogs, lizards and cats etc. filters used in this layers are in matrix form normally these filters are 3×3, values of these filters are initialize randomly. Input of an image convolve with these filters and pass the output to next layers some weights and threshold defines at each node these weight and threshold are multiplied and added respectively with output values before process to next layers and at output stage value is compared if there is some error then feedback system is used to set new weights and threshold on the bases of error function. CNN model trained for different set of image in which all possible shapes, textures and edges are under consideration.

In pooling layer, we extract the features dimension which are very small so it is used to manage the feature extraction of small dimensions. To control the overfitting and reduce the computation and parameters it also helps to enhance the efficiency of the network. When there is a very small object placed in our image and filter does not recognize the object and placed zero value against this small pooling layer shows the small object inside the image irrespective of the fact where this small object is placed.

In most of the image applications it requires that input and resultant image size should be equal otherwise boundary artifacts occurred which causes a loss of boundary information of an image. In MLP [22] boundary of clattered image symmetrically padded in processing phase this same approach is used in CSF [13] and TNDR [14]. But in this method we pad zero after each stage to maintain the size of input image in experiments we see that simply zero padding reduce the boundary artifacts. Another approach pooling layers are used for reducing boundary artifacts but adding pooling layers increase the complexity of network architecture and it also time consuming. So we simply use zero padding to reduce boundary artifacts.

3.5 Residual Learning and Batch Normalization

It has been indicated that amalgamation of residual learning along with batch normalization is helpful for noise removal because it used in neural networks for better performance. The architecture used in this method is trained to obtained both original mapped function $F(y)$ to divine x or residual mapping $\mathcal{R}(y)$ to predict v. According to [24] when residual mapping and identity are likely to be same then it’s better to optimize residual mapping.

During neural network training some preprocessing steps is taken for normalization of input data so that all data should be in same range. For example, if we have some data in 0 to 1 range and other data in 1 to 1000 range we normalize these data in same distribution. The problem is arising when all layers work on same distribution data and at activation layer distribution is constantly changing during training. This slow down the training process because each layer learns themselves at training process to new distribution. When the training and testing data set come from difference sources both have different distribution the behavior of machine learning algorithm changes when the input distribution is different the problem is known as covariate shift.

In neural network main concern is change of distribution of input layer to inner layers during training process due to activation function the weights of each layer changes according to requirement so the distribution also changes at each layer. We know that output activation of every layer is input to the next layer, every layer in neural network faced the issue where distribution is changes at every step. Basic idea behind batch normalization is that activation function of every layer changes into a normalized value with 0 mean and unity variance this will also help to reduce covariate shift. This also help each layer to learn on most stable distribution and increase the speed of training process. Batch normalization is used for the stability of neural network in which output of previous activation function subtracting from batch mean and divide by batch standard deviation. During training process following steps are involved in batch normalization:

During network training internal covariant shift causes change in network parameters when non residual learning is used it effect the convergence property of network. In stochastic gradient descent and Adam algorithm [25] consolidation of residual learning along with normalization is used for best throughput of network. In Gaussian de-noising both batch normalization and residual learning associated with Gaussian distribution.

- Batch normalization offers some useful characteristics such as improve internal covariant shift for convolutional neural networks.
- Batch normalization with Residual learning boost the performance and good effect towards convergence. Without residual learning intensities of input and convolutional features correlate with neighbors.

With the help of residual learning Neural network completely remove the latent clear image from concealed layers. In this manner input layers are less correlated and make each layer are Gaussian like distribution. In other words, we can say that consolidation of residual learning along batch normalization increase the speed of network and boost up the de-noising performance.

Chapter 4

Experiment and Result

4.1 Network dataset training setting

Deep convolutional neural network training for image de-noising several parameters are under consideration including patch size, number of patches and filter dimensions. For training the model several methods discussed previously for multiple noise levels. When we consider large number of images for training purpose de-noising performance of model increase rapidly.

4.2 Training and Testing Data

For Gaussian de-noising training method discussed in [14] in which about 400 images of dimensions 180×180 are used for training purpose with experiments we conclude that taking large number of training set can improve the de-noising performance. Three different noise level are taken to train the DnCNN model for Gaussian de-noising with known clatter level, i.e., σ=15, 25 and 50. We fix the patch size as 40×40, and shear 128×1,600 patches for training the model. We broach to our DnCNN model for Gaussian de-noising with known particular clatter level as DnCNN-S. We set the clatter levels range as $\sigma \in [0,55]$, and the patch dimension as 50×50 to train a single DnCNN model for blind Gaussian de-noising 128×3,000 patches are sheared to train the model.

For an obtuse Gaussian de-noising function as DnCNN-B we broach our single DnCNN pattern. Referring to two commonly utilized databases, we set up the test images to take all relevant approaches into account. Berkeley segmentation dataset (BSD68) [26] contains 68 images and other test set have 12 images as shown in fig 4.1 it is noted that these test images are not used for training the model its only for experiment purposes for evolution of Gaussian image de-noising.

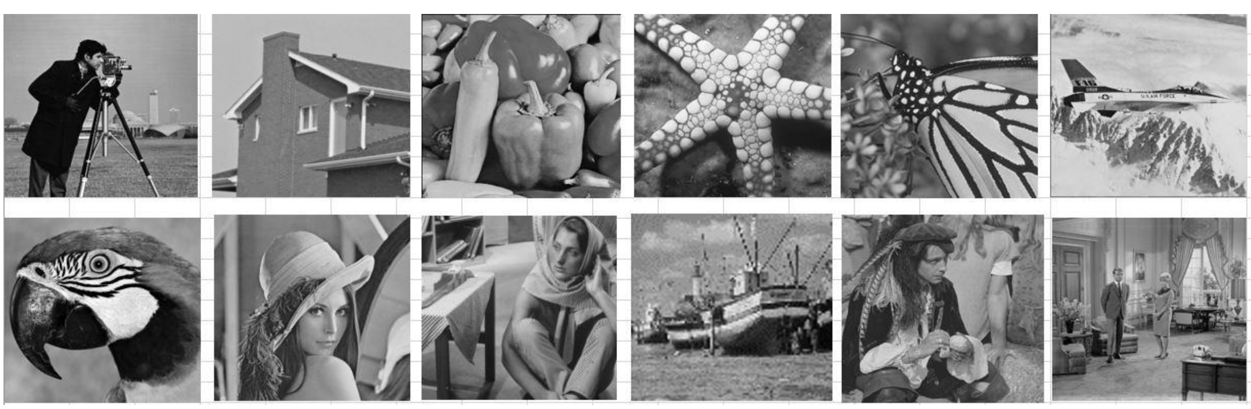


figure4. 1 Set of 12 images used for testing

4.3 Network Training

Network depth set to be 17 for getting better spatial information for de-noising. The loss is used to in residual mapping $\mathcal{R}(y)$ for envisaging residual v. For training purpose of model MatConvNet package discussed in [27] other all experiments perform on MATLAB (R2015b). The de-noising result of images compared with different methods discussed below.

4.4 Compared Methods

Deep convolutional neural network based method discussed in this thesis compared with various ultra-modern de-noising method including BM3D [20] and WNNM [21] also with EPLL [23] and other three discriminative training based method like MLP [22] ,CSF [13] and TNRD [14]. It has been noted that CSF and TNDR are highly efficient and provide a better de-noised image but our method results are comparable with minimum computational complexity.

Fig 4.2 shows the de-noising result of one image taking from data set with clatter level 25 (a) BM3D with PSNR 32.0 (b) EPLL with PSNR 31.55 (c) and at (d) our algorithm result with PSNR 32.31 with SSIM 0.8573. the noise level in below de-noised image is same i.e 25. Figure shows that our method performs well with ultra-modern de-noising methods with relatively greater PSNR. Another image from same set is shown in figure 4.3 with noise level 50 [28]. In figure 4.4 a color image of an airplane reconstructed from noisy observation with noise level 25. Three sub images in one fig from right to left noisy image with noise level 25, original image and final reconstructed image respectively we see that PSNR of color image is 33.66db and SSIM is 0.9259.

(a)

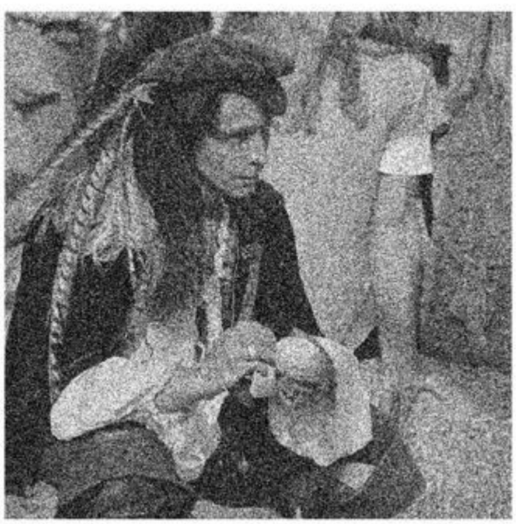
(b)

(c)

(d)

figure4. 2De-noising result of image from widely used data set with noise level 25

(a) bluster image (b) BM3D / 32.0 (c) EPLL / 31.55 (d) DnCNN-W / 32.31 with SSIM 0.8573.



(a)

(b)

(c) (d)

figure4. 3De-noising result of image from widely used data set with noise level 50

(a) bluster image (b) BM3D / 26.81 (c) EPLL / 26.66 (d) DnCNN-W / 27.15 with SSIM 0.7055.

figure4. 4 de-noising result of a color image

From left to right noisy image with clatter level 25, original image and reconstructed image with PSNR 33.66db and SSIM 0.9259.

4.5 Quantitative and Qualitative measures

Table 4.1shows the PSNR values of various methods on widely used test set twelve images with best PSNR value highlighted in bold format. From table we can see that the purposed algorithm concedes the best PSNR values on most of the images it performs well by 0.2dB to 0.6dB. this method fails to achieve only on two images ''House'' and ''Barbara'' because these two images are dominated with repetitive structures and textures. The reason is that non-local similarity based method outperforms with efficient tedious structures although discriminative training based method broadly produces better results on lopsided shapes and textures. From results shows in table it is observed that BM3D, WNNM, EPLL and MLP lean to produce over flat edges and textures. While conserve sharp edges and fine details, TNRD is likely to conjure artifacts in flat region.

This method not only extract flat edges but also provide visual good result in smooth regions and remove boundary artifacts which is due to compression of image on input stage. In addition, neural network based image de-noising method work well on real noisy image. Table 4.2 shows the average PSNR result of different methods used for image de-noising it can be observed that our method achieve the best PSNR result over stat of the art de-noising methods on twelve widely used images in image processing. Our method recovers the output image with best details with minimum loss of information even on colored images.

Noise Level	15											
Images	Cameraman	House	Peppers	Starfish	Monar	Aero plan	Parrot	Lena	Barbara	Boat	Man	Couple	Average
BM3D	31.9	34.9	32.6	31.1	31.8	31.0	31.3	34.2	33.1	32.1	31.9	32.1	32.3
WNNM	32.1	35.1	32.9	31.8	32.7	31.3	31.6	34.2	33.6	32.2	32.1	32.1	32.6
EPLL	31.8	34.1	32.6	31.1	32.1	31.1	31.4	33.9	31.3	31.9	31.9	31.9	32.1
CSF	31.95	34.39	32.85	31.55	32.33	31.33	31.37	34.06	31.92	32.01	31.98	31.98	32.31
TNRD	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.11	32.11	32.49
DnCNN	32.21	34.45	33.2	31.95	33.01	31.5	31.71	34.52	32.32	32.32	32.37	32.35	32.66

Noise Level	25												
Images	Cameraman.	House.	Peppers.	Starfish.	Monar.	Aero plan.	Parrot.	Lena.	Barbara.	Boat.	Man.	Couple.	Average.
BM3D	29.4	32.8	30.1	28.5	29.2	28.4	28.9	32.0	30.7	29.9	29.6	29.7	29.96
WNNM	29.6	33.3	30.4	29.0	29.8	28.6	29.1	32.2	31.2	30.0	29.7	29.8	30.26
EPLL	29.2	32.1	30.1	28.5	29.3	28.6	28.9	31.7	28.6	29.7	29.6	29.5	29.69
MLP	29.6	32.5	30.3	28.8	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.02
CSF	29.48	32.39	30.32	28.8	28.82	28.72	28.9	31.79	29.03	29.76	29.71	29.53	29.78
TNRD	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32	29.41	29.91	29.87	29.71	30.06
DnCNN	30.01	32.85	30.65	29.23	30.12	29.01	29.34	32.31	30	30.04	30.01	30.02	30.31

Noise Level	50												
Images	Cameraman.	House.	Peppers.	Starfish.	Monar.	Aero plan.	Parrot.	Lena.	Barbara.	Boat.	Man.	Couple.	Average.
BM3D	26.1	29.6	26.6	25.0	25.8	25.1	25.9	29.0	27.2	26.7	26.8	26.4	26.72
WNNM	26.4	30.3	26.9	25.4	26.3	25.4	26.1	29.2	27.7	26.9	26.9	26.6	27.05
EPLL	26.1	2912	26.8	25.12	25.9	25.31	25.95	28.68	24.8	26.7	26.7	26.3	26.67
MLP	26.3	29.6	26.6	25.4	26.2	25.5	26.1	29.3	25.2	27.0	27.0	26.6	26.78
TNRD	26.62	29.4	27.1	25.42	26.3	25.59	26.16	28.9	25.7	26.98	26.98	26.5	26.81
DnCNN	26.95	30	27.15	25.55	26.57	25.76	26.33	29.33	26.02	27.14	27.15	26.83	27.07

Table 4.1

PSNR(dB) results of various methods on widely used testing images

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN
$\sigma = 15$	32.37	32.62	32.13	--	32.31	32.49	**32.66**
$\sigma = 25$	29.96	30.26	29.69	30.02	29.78	30.06	**30.31**
$\sigma = 50$	26.72	27.05	26.67	26.78	--	26.81	**27.07**

Table 4.2

Average PSNR results of various methods on BSD68 dataset with highlighted best results

figure4. 5 de-noising result of a color image

figure4. 6 de-noised result of a color image

figure4. 7 de-noised result of a color image

figure4. 8 de-noised result of a color image

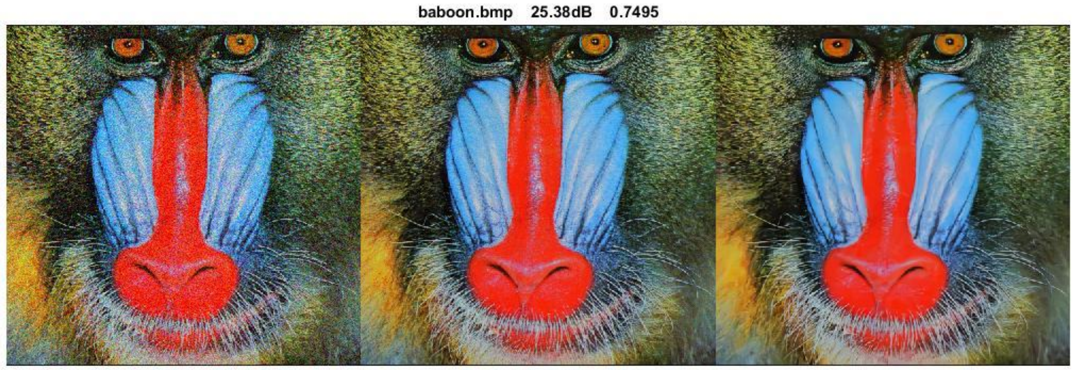


figure4. 9 de-noised result of a color image

figure4. 10 de-noised result of a color image

figure4. 11 de-noised result of a color image

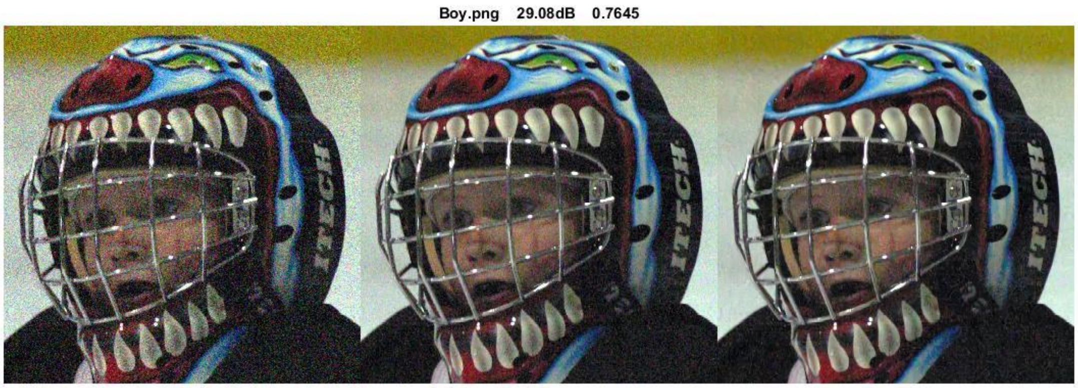


figure4. 12 de-noised result of a color image

figure4. 13 de-noised result of a Grayscale image

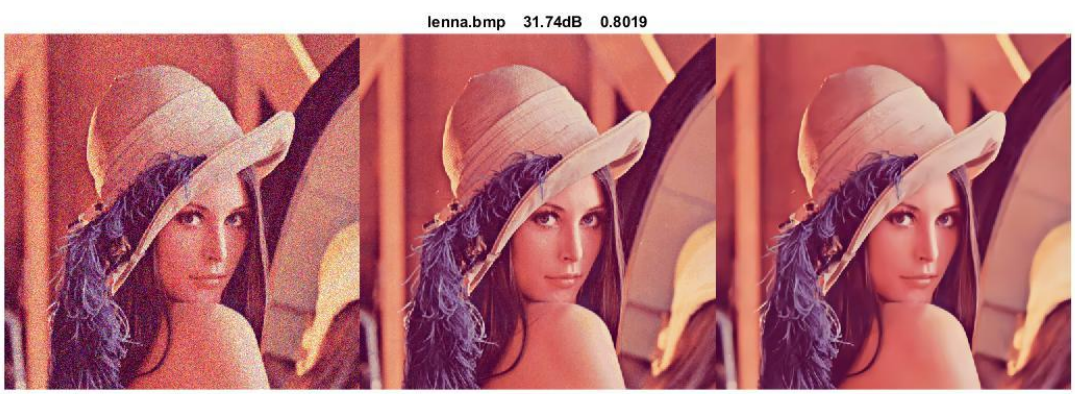


figure4. 14 de-noised result of a color image

figure4. 15 de-noised result of a color image

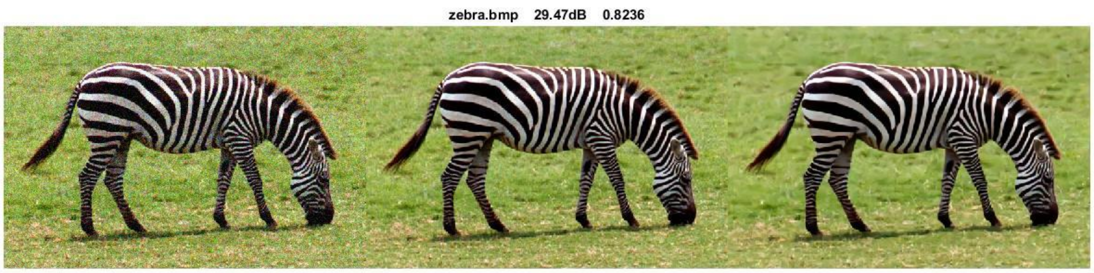


figure4. 16 de-noised result of a color image

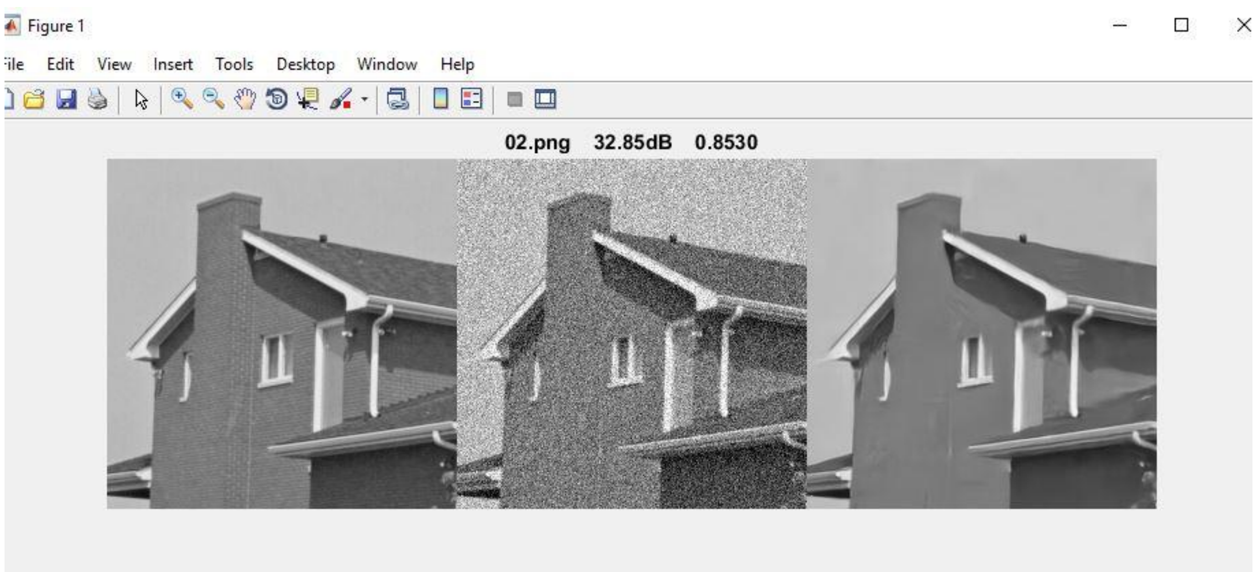


figure4. 17 de-noised result of a grayscale image

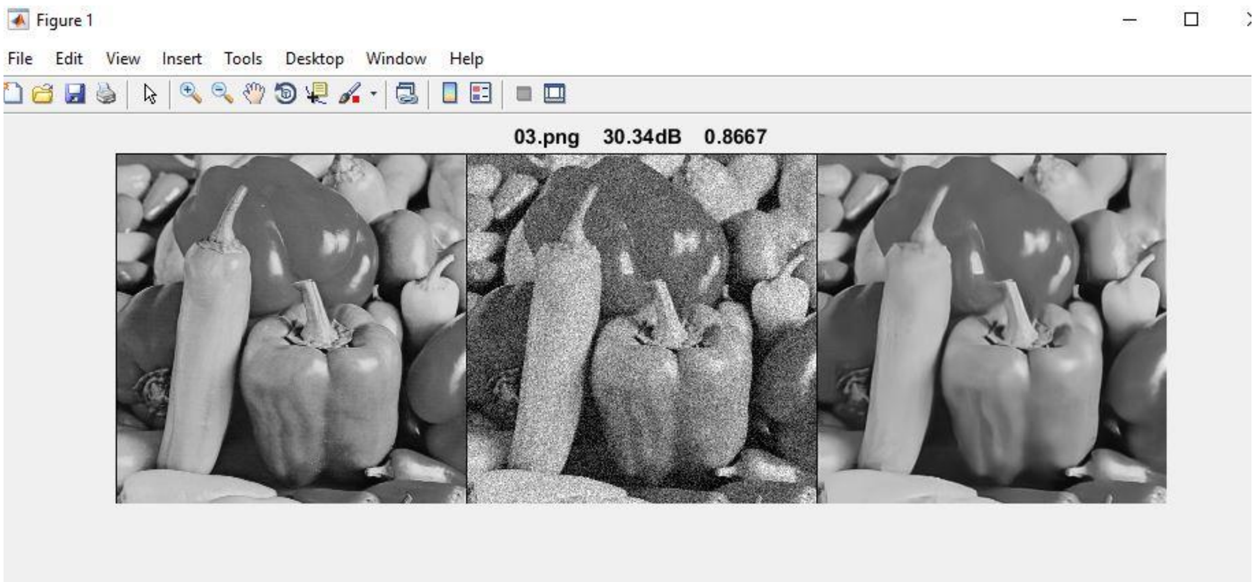


figure4. 18 de-noised result of a grayscale image

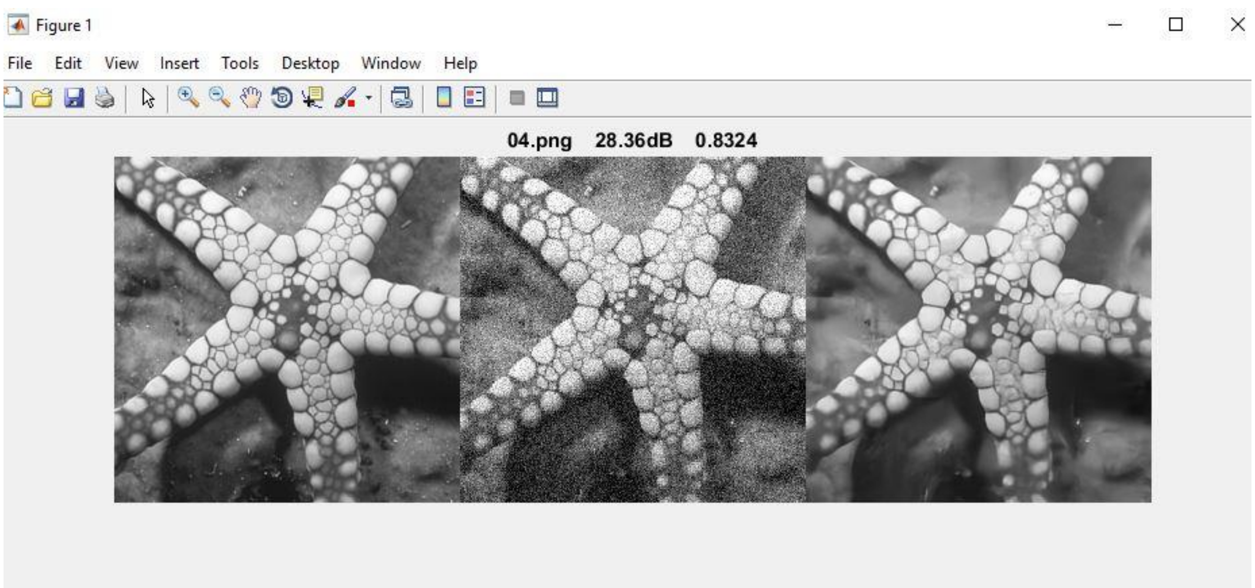


figure4. 19 de-noised result of a grayscale image

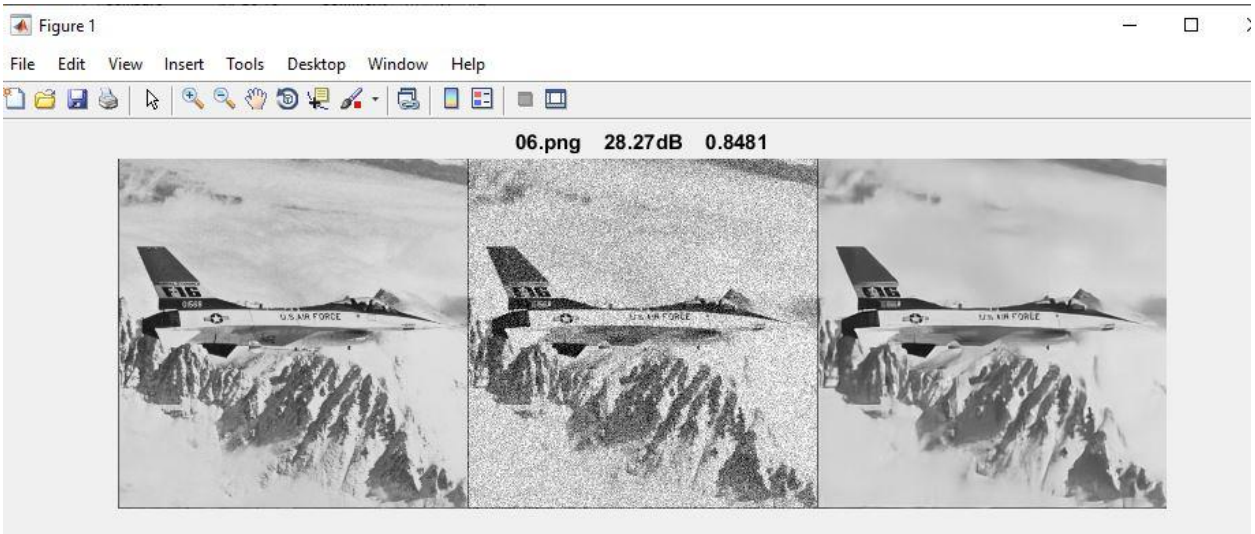


figure4. 20 de-noised result of a grayscale image

figure4. 21 de-noised result of a grayscale image

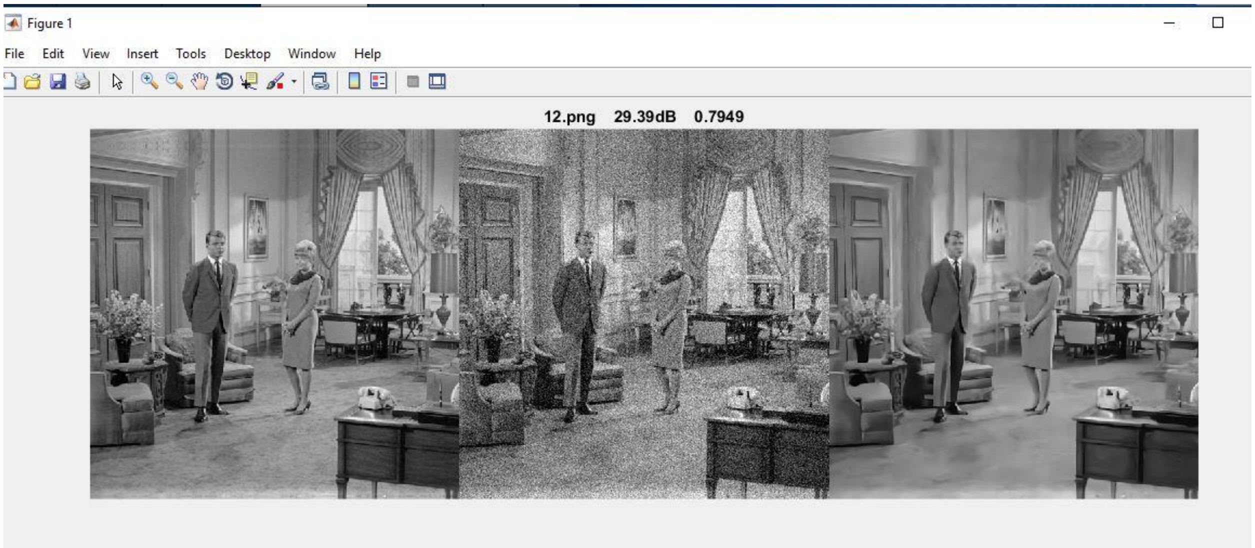


figure4. 22 de-noised result of a grayscale image

figure4. 23 de-noised result of a grayscale image

Chapter 5

Conclusion

In this thesis deep convolutional neural network based image de-noising method is discussed in which input image is compressed using wavelet decomposition so that we minimize the computational complexity. The traditional way of subsampling make the input image blurred so we use approximate coefficient extracting using wavelet decomposition for further operation of image de-noising. The advantage of using wavelet is that it compresses the image without degrading the quality of an image and provide a best compression ratio. It also reduces the memory burden and expand the receptive field.

Neural network algorithm processes the compressed image in which batch normalization and residual learning is used to retrieve the latent clear image from clattered observation and we see from experiments our CNN based method outperform with stat of the art de-noising methods it also except the several noise level with single model, unlike other methods no need to estimate the parameters manually. We also observe that integration of batch normalization and residual learning not only speed up the training affair but also stimulate the de-noising performance. From result we conclude that it provides a best result for gray scale image as well as colored image. It also handles the different noise levels with single model and provide a better result with minimum computational complexity without degrading the image quality. in other words, CNN based method is flexible, efficient and effective over other de-noising methods.

Bibliography

[1] B. C. a. J. M. M. A. Buades, "Image denoising by non-local averaging," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, 2005.

[2] M. E. H. T. a. P. M. M. Protter, "Generalizing the Nonlocal-Means to Super-Resolution Reconstruction," *IEEE Transactions on Image Processing,* vol. 18, no. 1, pp. 36-51, Jan 2009.

[3] F. B. J. P. G. S. a. A. Z. J. Mairal, "Non-local sparse models for image restoration," in *IEEE 12th International Conference on Computer Vision*, Kyoto, 2009.

[4] L. Z. W. Z. D. Z. a. X. F. J. Xu, "Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising," in *IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015.

[5] M. E. a. M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Transactions on Image Processing,* vol. 15, no. 12, pp. 3736-3745, Dec 2006.

[6] L. Z. G. S. a. X. L. W. Dong, "Nonlocally Centralized Sparse Representation for Image Restoration," *IEEE Transactions on Image Processing,* vol. 22, no. 4, p. April 2013, 1620-1630.

[7] Z. Z. e. al, "Analyzing the group sparsity based on the rank minimization methods," in *IEEE International Conference on Multimedia and Expo (ICME)*, Hong Kong, 2017.

[8] M. B. D. G. J. X. a. W. Y. S. Osher, "An iterative regularization method for total variation-based image restoration," in *Multiscale Model.*, abc, 2005.

[9] Y. W. a. W. T. Freeman, "What makes a good model of natural images," in *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[10] S. R. D. H. a. M. J. B. X. Lan, "Efficient belief propagation with learned higher-order Markov random fields," in *Processing Euopean Conference Computer vision*, 2006.

[11] T. Chen, "A Markov Random Field Model for Medical Image Denoising," in *2nd International Conference on Biomedical Engineering and Informatics*, Tianjin, 2009.

[12] X. L. W. M. T. L. a. X. S. Y. Hou, "Unsupervised Segmentation Method for Color Image Based on MRF," in *International Conference on Computational Intelligence and Natural Computing*, Wuhan, 2009.

[13] U. S. a. S. Roth, "Shrinkage fields for effective image restoration," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, june 2014.

[14] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern,* 2016.

[15] K. Z. W. G. S. &. Z. L. Zhang, "Learning deep CNN de-noiser prior for image restoration," *IEEE conference on computer vision and pattern recognition ,* pp. 3929-3938, 2017.

[16] K. Z. W. &. Z. L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image de-noising," *IEEE Transactions on Image Processing, 27(9), ,* pp. 4608-4622, 2018.

[17] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind,* vol. LIX, no. 236, pp. 433-460, October 1950.

[18] T. M. Mitchell, Machine learning, 1 ed., New York, USA: McGraw-Hill Science/Engineering/Math, 1997.

[19] K. S. a. A. Zisserman, "Very deep convolutional networks for," in *Proc. International Conference Learn*, 2015.

[20] A. F. V. K. a. K. E. K. Dabov, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Transactions on Image Processing,,* vol. 16, no. 8, pp. 2080-2095, 2007.

[21] L. Z. W. Z. a. X. F. S. Gu, "Weighted nuclear norm minimization with application to image denoising,," in *IEEE Conf.Comput. Vis. Pattern Recognit*, june 2014.

[22] C. J. S. a. S. H. H. C. Burger, "Image denoising: plain neural networks compete with BM3D," in *in Proc. IEEE Conf.computer vision, pattern recoginition*, june 2012.

[23] D. Z. a. Y. Weiss, "From learning models of natural image patches to whole image restoration,," in *IEEE Int. Conf. Comput. Vis*, 479–486, Nov 2011.

[24] X. Z. S. R. a. J. S. K. He, "Deep residual learning for image recognition," in *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, june 2016.

[25] D. P. K. a. J. L. Ba, "Adam: A method for stochastic optimization," in *in Proc. Int. Conf. Learn. Represent.*, 2015.

[26] S. a. M. J. Black, "Fields of expert," *international journal on computer vision ,* vol. 82, no. 2, pp. 205-229, April 2009.

[27] A. V. a. K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *in Processing 23rd Annual. ACM Conference. Multimedia Conf*, 2015.

[28] S. J. Z. Z. W. &. L. X. Zhuo, "RIDNet: Recursive Information Distillation Network for Color Image De-noising," *International Conference on Computer Vision Workshops,* 2019.

[29] V. J. a. S. Seung, "Natural image denoising with convolutional networks," in *in Proc. Adv. Neural Inf. Process. Syst.*, 2009.

[30] A. L. a. B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, june 2011.

[33] M. A K Peters Wellesley, B.B. Hubbard, The World According to Wavelets, 1995.

[34] R. Polikar., "the Wavelet tutorial," [Online]. Available: http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html.

[35] T. D. B. a. G. Y. C. D. Cho, "Image denoising based on wavelet shrinkage using neighbor and level dependency," *International Journal of Wavelets, Multiresolution and Information Processing,* vol. 7, no. 3, pp. 299-311, 2009.

[36] U. Meyer-Baese, Digital Signal Processing with Field Programmable Gate Arrays, Springer-Verlag, 2001.

[37] G. S. a. T. Nguyen, "Wavelets and Filter Banks," Wellesley-Cambridge press, 1997.

[38] X. Z. S. R. a. J. S. K. He, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015.

[39] S. J. Z. Z. W. &. L. X. Zhuo, "Recursive Information Distillation Network for Color Image De-noising," *International Conference on Computer Vision Workshops.*