International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 10 | Oct 2023 www.irjet.net p-ISSN: 2395-0072

IMPROVEMENT IN IMAGE DENOISING OF HANDWRITTEN DIGITS USING AUTOENCODERS IN DEEP LEARNING

DHARANI D[1], S.KRITHIGA [2], Dr. J.SUNDARAVANAN [3]

[1]PG student, Department of Electronics and Communication Engineering
[2] Assistant Professor, Department of Electronics and Communication Engineering
[3]Head of The Department, Electronics and Communication Engineering
Thanthai Periyar Government Institute Of Technology, Vellore, Tamil Nadu, India

---

Abstract – *Compressed sensing is a signal processing algorithm that reconstructs a signal based on random measurements matrices. But designing a measurement matrix is a tedious process as they are random in nature. Also, the available CS reconstruction algorithms are time consuming which are unsuitable to real time applications. So, a deep learning model based on neural networks is proposed to solve the problems of CS. This model namely Stacked Sparse Denoising Autoencoder(SSDAE) model is discussed. It is a Encoder and Decoder network architecture that learns representation from the image which stores the essential information and tries to reconstruct at the decoder end. The combination of Sparse Autoencoder and denoising Autoencoder is used to perform CS reconstruction with fewer non linear measurements. The robustness of the model to various levels of noises in the image is studied and tries to reconstruct from the original input image. It is found from simulations that the model has good denoising ability, stable reconstruction and shows better performance interms of its reconstruction quality and attains higher Peak to Signal Noise Ratio compared to other algorithms.*

***Key Words*: Compressed sensing, Denoising Autoencoder, Sparse Autoencoder, reconstruction algorithms.**

1.INTRODUCTION

Compressed Sensing also called as Compressive Sensing or Sparse sampling is a technique used to reconstruct a signal in a compressed form. It considers only few number of measurements to reconstruct a signal obtained by efficient design of sensing matrices. These sensing matrix is necessary for obtaining the required compressed signal at the encoder and its exact reconstruction at the decoder part. It imposes two conditions namely sparsity constraint which makes the signal to be sparse. Another condition is incoherence which involves isometric property.

Compressed sensing reconstructs a signal by finding solution to underdetermined linear systems. This type of systems has more unknowns and generate infinite number of solutions. The system is defined by equation y = Dx, where solution for x must be found. All underdetermined system doesn't have a sparse solution. In CS, the weighted linear combination of samples known as sparse will be small but contains all the useful information. So, for the process of converting the image back into desired image is done by solving underdetermined matrix equation since the measurements are small than the number of pixels in the full image. So adding sparsity constraint enables to solve this underdetermined system of linear equations. By enforcing sparsity, one can reduce the number of nonzero components of the solution. So, the function of counting the number of nonzero components of the vector is known as L^0 norm. So, Compressed sampling combines sampling and compression at a single process which helps to reduce the storage cost, size and processing time. It finds applications in areas such as Facial Recognition, Holography, Magnetic Resonance Imaging, Aperture synthesis astronomy, shortwave infrared cameras and many more.

1.1 Limitations of Compressed Sensing:

But, compressed sensing has shortcomings that must be taken into account. One problem is due to the random nature of the measurement matrix and another problem is to construct a stable reconstruction algorithm with low computational complexity which considers only the recovery process while the connection with compressed sampling process is ignored. It is difficult to built a appropriate measurement matrix for compressed sampling. Basically, these measurement matrix is divided into random matrix and definite matrix. Examples of random matrices are Gaussian and Bernoulli marices. These matrices are unsuitable for the design because they occupy more storage, uncertain reconstruction quality and other problems such as computation time. On the other hand, definite matrices such as Toeplitz polynomial and Circulant matrix has been used. Here, the reconstruction quality of the image isn't better than random matrix. So, both these matrices has some disadvantages and doesn't suit well for the design of signals.

Another shortcoming is to construct a stable reconstruction algorithm with low computational complexity and with less number of measurements. These CS reconstruction algorithms are divided into two types namely hand-designed recovery methods and data driven recovery methods. The hand-designed methods have three directions: convex optimization, greedy iterative, and Bayesian. This Greedy

iterative algorithms approach the original signal by selecting a local optimal solution in iteration. Examples are orthogonal matching pursuit (OMP) and compressive sampling matching pursuit (CoSaMP). The second recovery method is data-driven method that builds deep learning framework to solve the CS recovery problem by learning the structure within the training data. For example, the SDA model was proposed to recover structured signal by capturing statistical dependencies between the different elements of certain signals. Some previous works show that hand driven recovery algorithm works better than hand designed algorithm when performance is considered. So, both these recovery algorithms consumes more time and data for training the network and its very slow when incorporated in real time applications. So, these recovery algorithms is not suitable for the stable reconstruction of signals.

2. PROPOSED METHOD

From the above shortcomings, A deep learning model is suggested namely Stacked Sparse Denoising Autoencoder compressed sensing (SSDAE_CS) model which is a combination of Denoising AutoEncoder (DAE) and Sparse AutoEncoder (SAE). The model comprises of Encoder network, Decoder network and code layer (information bottleneck). Instead of conventional linear measurements, the encoder subnetwork takes many non linear measurements for training and the decoder network tries to reconstruct the input image with few number of measurements. This model achieves greater accuracy, good denoising ability, excellent signal reconstruction.

2.1Overall Architectue of The Autoencoder

Autoencoders comes under unsupervised learning model. These are neural networks that is used to compress and reconstruct data. This model mainly consists of an encoder subnetwork ,a code layer and decoder subnetwork.

Encoder: This layer compresses the data into smallest possible representation. It extracts the prominent features from the data and stores in a encoded version. The input image is compressed down, and dimensions are reduced in the encoder layer, which leads to a bottleneck.

Code layer: The Code layer represents the compressed input fed to the decoder layer. The layer between the encoder and decoder network. that is, the code is also known as information Bottleneck. This layer stores all the important features of the input data, thus reducing the computation time.

Decoder: The decoder layer decodes the encoded image back to the original image. The decoded image is reconstructed from latent space representation. It aims to minimize the loss while reconstruction. In an inverse fashion, the number of neurons progressively increases all the way back to the original image.

2.2 Architecture of the Model:

The proposed model deals with two types of Autoencoders namely Sparse Autoencoders and Denoising Autoencoders to solve the problem of CS recovery. In Denoising Autoencoders, the input image is corrupted with noise and used to produce a clean image as output. In Sparse Autoencoders, it introduces sparsity parameter which penalizes the activation of the neurons. As discussed above, earlier methods of CS uses linear measurement for sampling. Instead the model uses a multiple nonlinear measurement encoder sub-network for obtaining the measurements during the process. Then, at the recovery stage of the data, traditional recovery algorithms is replaced with a decoder network for obtaining the original image.

Use of Denoising Autoencoders: Denoising Autoencoders are an important and crucial tool for feature extraction. The purpose of this autoencoder is to remove noise. The network is trained using corrupted version of the input and tries to reconstruct a clean image through the use of denoising techniques. Usually, these autoencoders can also be stacked on each other to extract more features and to ensure best image reconstruction. A Denoising Autoencoder is a modified version of the basic Autoencoder. Specifically, if the autoencoder is too big, it just learns the data, and produces the output same as input from which they doesn't learn any useful information and the purpose of feature extraction is nullified. So, Denoising autoencoders solve this problem by corrupting the input data by adding noise or masking some of the input values. The proposed model reconstructs excellent images by adding various values of noise parameter in the image while training.

Use of Sparse Autoencoders: A sparse autoencoder is an another common type of Autoencoder which deploys sparsity to achieve an information bottleneck. In SAE, sparse regularization inhibits the activation of neurons to enhance the overall performance of the model.Here, the loss function is imposed by penalizing activations of hidden layers so that only a few nodes are encouraged to activate when a image is fed into the network in training phase. There are two kinds of loss functions that are used namely, L1 regularization and KL-divergence. The difference between two loss functions is that L1 regularization make penalty coefficient to zero while KL divergence is the measure of the relative difference between two probability distributions for a given random variable or set of events. In this work, L1 regularization is used because it is often used for feature extraction. Due to the sparsity of L1 regularization, sparse autoencoder learns better representations than basic Autoencoder. Let,

$$\hat{\rho}_j = \frac{1}{M}\sum_{i=1}^{m}[x^i a_j^2] \quad (1)$$

be the average activation of hidden unit j. To enforce the constraint for the activation

$$\hat{\rho}_j = \rho \quad (2)$$

where ρ is a sparsity parameter, typically a small value close to zero. To satisfy this constraint, the hidden unit's activations must mostly be near 0.

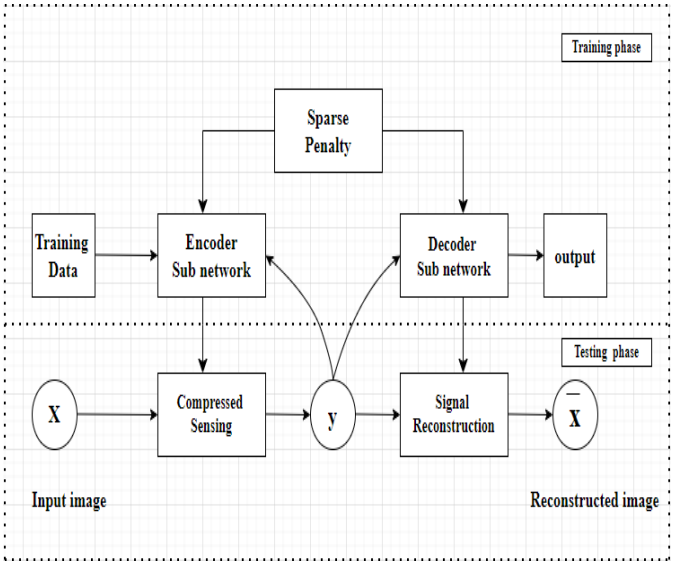

Fig -1: Proposed Model

Encoder and Decoder Sub Network: To build an Autoencoder of SSDAE_CS model, we require three functions: an encoding function, a decoding function and a loss function to determine the amount of information loss between the input and the original image. The SSDAE_CS model is a deep neural network consisting of multiple layers of basic SAE and DAE, in which the outputs of each layer are connected to the inputs of each successive layer. In the training phase, the input samples are corrupted with noise factor which is a hyperparameter. This noise factor is multiplied with random matrix that has mean zero and variance one. This matrix will draw samples from normal (Gaussian) distribution. The proposed model extracts robust features by sparse penalty term, which punishes and inhibits the larger change in the hidden layer. By this way, the proposed model is robust to the input because it reconstructs the original signals from the corrupted input. Next, the encoder sub-network compresses the signal to M measurements by utilizing multiple nonlinear measurement method. The decoder sub-network reconstructs the original signals from measurements by minimizing the reconstruction error between input and output. Then, the two sub-networks are integrated into SSDAE_CS model by jointly optimizing parameters to improve the overall performance of CS.

The encoder sub-network can be represented as a deterministic mapping $T_e(\bullet)$, which transforms an input x ∈ R^{dx} into hidden representation space y ∈ R^{dy}. The conventional CS method uses linear measurements which are unsuitable. In the SSDAE_CS model, multiple nonlinear measurements are applied to obtain measurements because non linear measurements preserve effective information when compared to linear meaurements. In this model, the encoder consist of three layers an input layer with N nodes, the first hidden layer with K nodes and the second hidden layer with M nodes, where N > K > M. The first hidden feature vector is the value of the first hidden layer represented by eqn (3).The final measurement vector y is the value of the second hidden layer, which receives the first hidden feature vector as its input which is represented by eqn (4).

$$a^{(1)} = f(z^{(1)}) = f(W^{(1)}\hat{x} + b^{(1)}) \quad (3)$$

$$y = f(z^{(2)}) = f(W^{(2)}a^{(1)} + b^{(2)})$$

$$y = f(z^{(2)}) = f(W^{(2)} f(W^{(1)}x + b^{(1)}) + b^{(2)}) \quad (4)$$

where W represents for matrices, b represents bias vector and X denotes the vector. Measurement vector y can also be written as:

$$y = T_e(\hat{x}, \Omega_e) \quad (5)$$

where $\Omega_e = \{W^{(1)}, W^{(2)}, b^{(1)}, b^{(2)}\}$ denotes the set of encoded parameters and $T_e(\bullet)$ denotes the encoding nonlinear mapping function. Now, The decoder sub-network maps the measurement vector y back to input space x ∈ R^{dx} by capturing the feature representation of the signal.

The previous signal reconstruction algorithms are replaced by decoder sub network. The decoder whose nodes are symmetric with the encoder consists of three layers: input layer with M nodes, the first hidden layer with K nodes, and the second hidden layer with N nodes. The decode function Eqn (6) and (7) are used to recover the reconstruction signals $\hat{x}$ from measurement vector y.

$$a^{(3)} = f(z^{(3)}) = f(W^{(3)}y + b^{(3)}) \quad (6)$$

$$\hat{x} = f(z^{(4)}) = f(W^{(4)}a^{(3)} + b^{(4)})$$

$$= f(W^{(4)} f(W^{(3)}y + b^{(3)}) + b^{(4)}) \quad (7)$$

The reconstructed signals $\hat{X}$ can also be represented as:

$$\hat{x} = T_d(y, d) \quad (8)$$

where $\Omega_d = \{W^{(3)}, W^{(4)}, b^{(3)}, b^{(4)}\}$ denotes the set of decoded parameters and $T_d(\bullet)$ denotes the decoding nonlinear mapping function.

Training the Network:

The main objective of the training phase is to learn the structural features from the samples. The input samples are fed into the network and trained for several epochs. To be specific, the encoder and decoder sub-networks are integrated into SSDAE_CS model through end-to-end training for strengthening the connection between the two processes. Since, the AutoEncoders are unsupervised model, the input samples in the training set as N signals whose label is same as the sample ie) D_{train} = {(x1, x1),(x2, x2), ⋯ ,(xn, xn)}. A trained nonlinear mapping $T_e(\bullet)$ acting as the measurement matrix obtains the measurements y from the original signals x, and a trained inverse nonlinear mapping $T_d(\bullet)$ acts as the reconstruction algorithm to recover the reconstruction signals $\hat{x}$ from y in the proposed model.

To ensure excellent reconstruction of image, the loss function $J_{SDAE}(W,b)$ is computed by using the formula:

$$J_{SDAE}(W,b) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}\left(\|\hat{x} - x_i\|^2\right)\right)$$

$$+\frac{1}{2}\,\alpha\sum\|W^2\| + \beta\sum_{j=1} KL(\rho||\hat{\rho_j}). \qquad (9)$$

Where the whole second term limits the weight parameters W with L1 norm as weight decay term. α denotes the strength of penalty. ρ is the sparsity parameter. $\hat{\rho_j}$ denotes the average activation of j-th neuron. β controls the activation of sparsity penalty. The main goal is to minimize $J_{SDAE}(W,b)$ and updates weights W and biases b. The batch gradient descent algorithm is performed to compute the gradients and updates Ω_e and Ω_d.Each iteration of the gradient descent method updates the parameters W and b by Eqn. (7) and (8) respectively

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha\frac{\partial J_{SDAE}}{\partial W_{ij}^{(l)}}(W,b) \qquad (10)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha\frac{\partial J_{SDAE}}{\partial b_i^{(l)}}(W,b) \qquad (11)$$

where α is the learning rate needed for training.

2.3 Dataset Description:

The dataset considered in this paper is standard MNIST dataset, which contains 70,000 grayscale images of handwritten digits of size N = 28 × 28, is used for training. The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. This database is widely used for training and testing in the field of machine learning. The dataset is divided into 55,000 samples for training, 5000 samples for validation, and 10,000 samples for testing. MNIST is a grey-scale image and the range is 0-255 which indicates the brightness and the darkness of that particular pixel.

Handwritten Digits Recognition turns out to be significant nowadays because of its actual implementation in our every day life. Programmed preparing of bank cheques, the postal mail, form data entry is a general utilization of hand-written digit recognition. In this project, Convolutional neural network model is built to reconstruct hand digits from 0 to 9. Handwritten digit recognition is the process of providing the ability to machines to recognize human handwritten digits. But it's not that easy for the machine because handwritten digits are not perfect, vary from person-to-person. One of the major application of handwritten digits is computer vision digit recognition.

2.4 Simulation Platform: The simulation tool used in this project is google colab. Commonly known as Colaboratory or "Colab" is a product from Google Research. It integrates other libraries such as PyTorch, TensorFlow, Keras, OpenCV. Since a Colab notebook can be accessed remotely from any machine through a browser, it's well suited for commercial purposes as well.

3. RESULTS:

The major goal is to achieve stable reconstruction with fewer measurements and to improve its denoising ability using Denoising Autoencoders (DAE) and Sparse Autoencoders (SAE) stacked upon each other. The model is trained and tested with two layer network with number of nodes in input and output layer as 784. The model is tested with adding Gaussian noise with mean zero and variance one and sparse penalty parameter ρ. Both these parameters are added at the encoder network of the Autoencoder. The noise level is increased by varying the levels of noise factor λ and changing the value of sparse penalty factor ρ. The performance of the model is good at smaller values of λ and ρ. The performance of the model degrades at higher levels of values. Optimal result is obtained when noise factor λ = 0.3 and ρ = 0.1. Keeping one of the value smaller and the other value higher produces better results. Image reconstruction is degraded when both the values are very high. Figure 2 shows the reconstruction of the images at various noise levels of value λ.

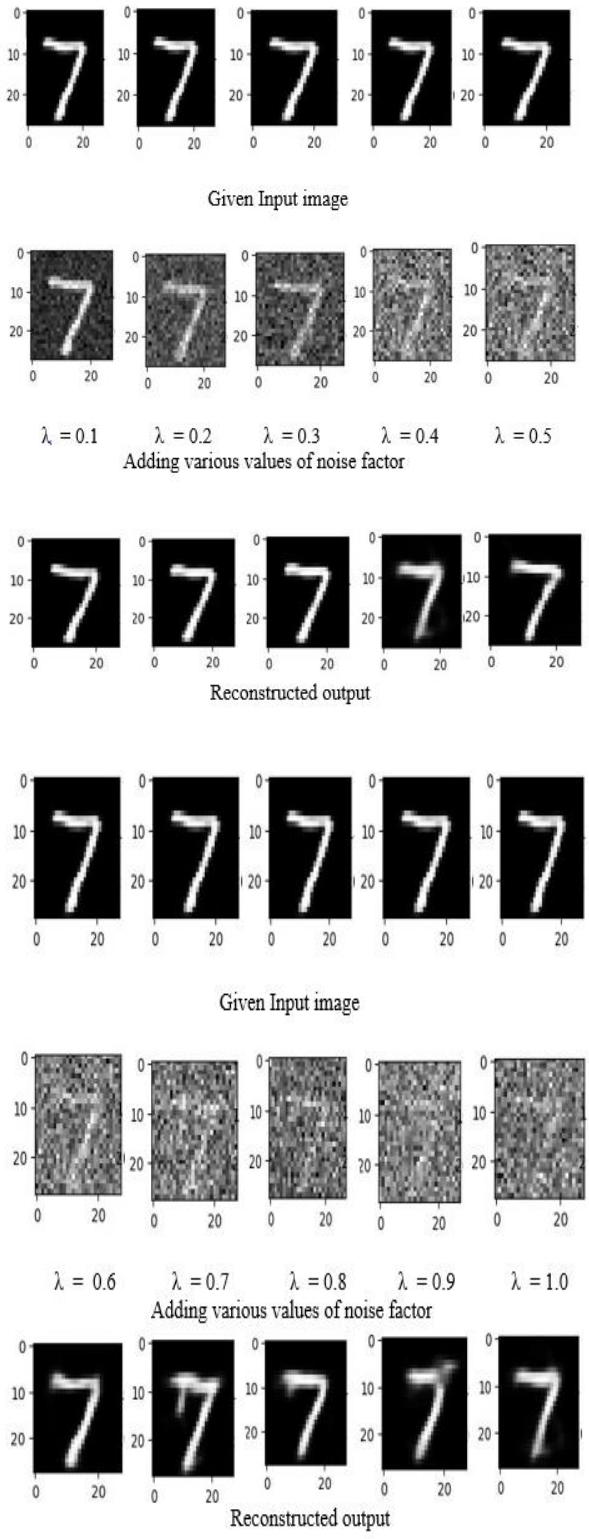

Fig -2: Image Reconstruction for various noise factor value λ

The figure 3 shows the reconstruction of the model when trained by adding noise and sparsity parameter. Best result is achieved when λ = 0.3 and sparse parameter ρ = 0.1

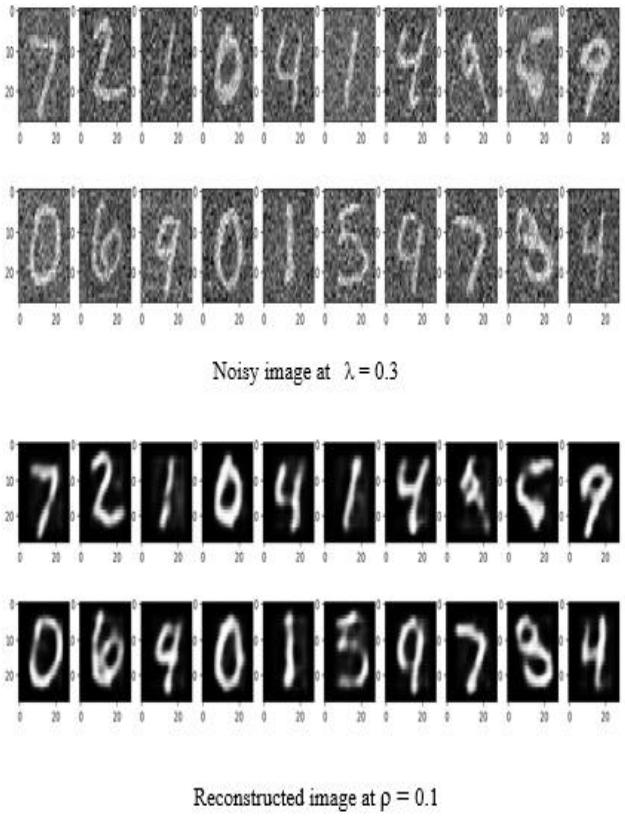

Fig -3: Image Reconstruction when λ = 0.3 and ρ = 0.1

3.1 Performance Metrics:

For checking the similarity between the original and the reconstructed image Peak Signal to Noise Ratio (PSNR) to applied to model. PSNR is mostly used to measure the quality of reconstruction. PSNR is calculated by following equation (12).

$$\text{PSNR (dB)} = 10 log_{10} \frac{peak\ value^2}{MSE} \qquad (12)$$

where peakvalue is either specified by the user or taken from the range of the image data. Mean square error (MSE) is calculated by equation (13)

$$\text{MSE} = \frac{1}{M} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2 \qquad (13)$$

Thus, the figure 4 shows the PSNR (in dB) for various levels of noise factor showing the quality reduces when noise factor increases.

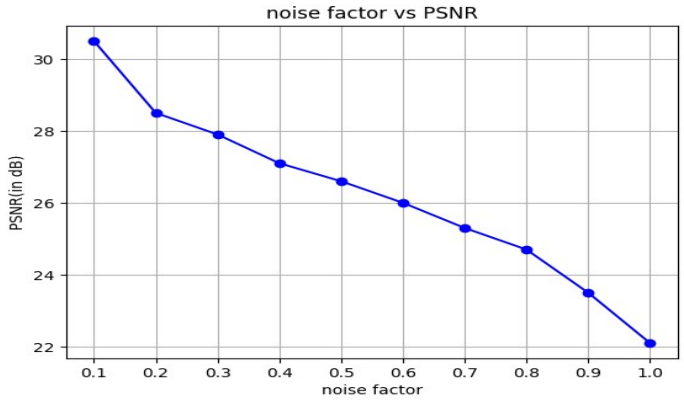

Fig -4: Mean PSNR value of the images for various levels of noise factor λ

The below graph shows the PSNR of the reconstructed images for various values of sparsity parameter ρ.

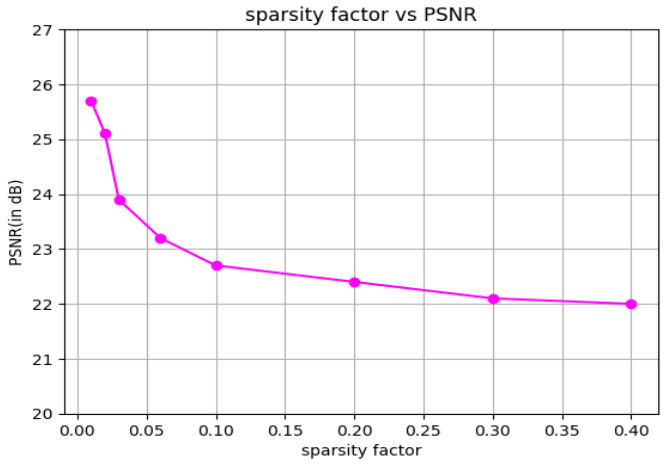

Fig -5: PSNR of the reconstructed images for various sparse penalty ρ

3.2 Comparison of the proposed model with Dictionary Learning:

To evaluate the performance of the proposed model, the SSDAE model is compared with Dictionary Learning Algorithm. dictionary learning can be represented in form of vectors that contains sparse representation of data. The signal can be represented as a linear combination of atoms in an overcomplete dictionary. Various dictionary learning algorithms such as matching pursuit, orthogonal matching pursuit, method of optimal direction (MOD) or k-singular value decomposition (k-SVD) are used in many applications such as signal separation, denoising, classification, image inpainting and reconstruction. Here, the size of the dictionary depends upon the dimensionality of the vectors of the data which ensures stable reconstruction. But computational time increases when dictionary size and vector size increases. To address this issue, the noisy images is break into patches and treat the vectorized version of each patch as signals, thereby restricting the dimensionality of each atom in the dictionary. However, the size of the patch has to be chosen such that it covers all the details of the signal. For a given image Y containing patches Z is represented as

$$Y = Z + \eta. \quad (14)$$

where η is assumed to be the noise which corrupts the patches. The noise over the entire image is assumed to be zero mean Gaussian noise. The patches obtained for image '0' from mnist dataset is shown in fiqure (6)

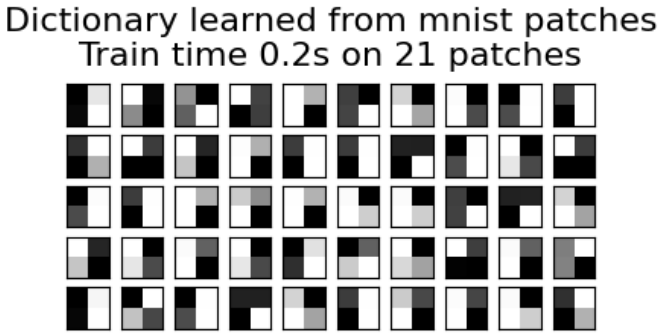

Fig -6: Patches learned using Dictionary Learning

The simulation results shows the corrupted or distorted image and its corresponding denoised image. The simulations results were conducted for various levels of noise variance σ. Fiqure 7 shows the distorted image of the image '0' from mnist dataset.

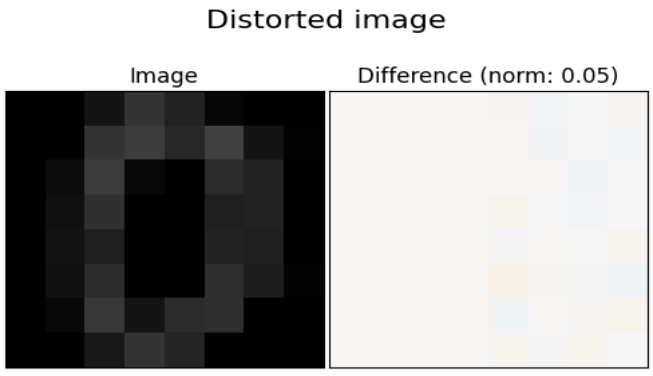

Fig -7: Distorted image when σ =0.2

Orthogonal matching pursuit is used as the reconstruction algorithm in dictionary learning which is a greedy type of algorithm that tries to find the sparse representation of a signal given a specific dictionary. Figure 8 shows its corresponding denoised image when σ =0.2

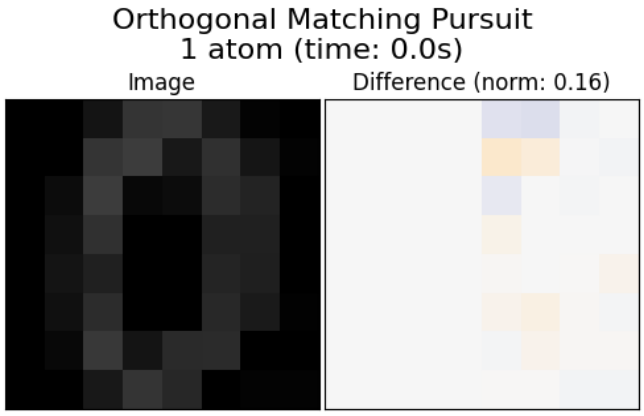

Fig -8: Denoised image when σ =0.2

The above figure implies that difference norm between the distorted and the denoised image is less which indicates a stable reconstruction but these algoorithms are time consuming in real time applications. Figure 9 compares the PSNR between the proposed model and dictionary learning showing the propsed model has attained higher PSNR when compared to Dictionary learning. The average PSNR is 3dB higher than Dictionary learning.

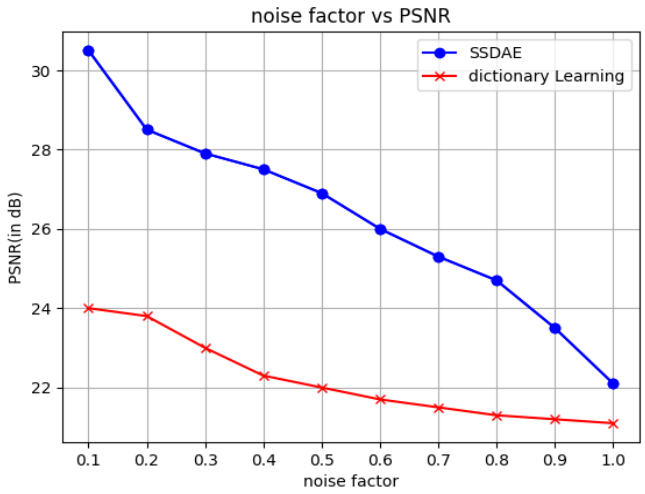

Fig -9: Comparison of PSNR between SSDAE model and dictionary Learning

4. CONCLUSIONS

In this paper, a Deep Neural Network named Stacked Sparse Denoising Autoencoder (SSDAE) model is propsed which is a encoder and decoder architecture that learns the representations and extracts essential features from the images and reconstructs at the decoder. It is found from simulations that proposed model is robust to noises achieving a stable reconstruction, denoising ability is also robust and the reconstruction quality is also better than other denoising algorithms. The experimental results shows reconstruction performance, time cost, and denoising ability, the proposed model is better than existing algorithms.

REFERENCES

[1] Zhang, Zufan, et al. "The optimally designed autoencoder network for compressed sensing." *EURASIP Journal on Image and Video Processing* 2019.1 (2019): 1-12.

[2] Kumar, Abhay, and Saurabh Kataria. "Dictionary learning based applications in image processing using convex optimisation." *Int. Conf. on Signal Processing and Integrated Networks (SPIN), Noida, India*. 2017.

[3] Metzler, C. A., Maleki, A., & Baraniuk, R. G. (2016). From denoising to compressed sensing. IEEE Transactions on Information Theory, 62(9), 5117–5144.

[4] D. Beohar and A. Rasool, "Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021, pp. 542-548

[5] Pan, Zhimeng, Brian Rodriguez, and Rajesh Menon. "Machine-learning enables Image Reconstruction and Classification in a "see-through" camera." *OSA Continuum* 3.3 (2020): 401-409

[6] D. L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory. 52(4), 1289–1306 (2006)

[7] A. Mousavi, R. G. Baraniuk, in IEEE International Conference on Acoustics, Speech and Signal Processing. Learning to invert: signal recovery via deep convolutional networks, (IEEE, New Orleans, 2017), pp. 2272–2276

[8] P. Xiong, H. Wang, M. Liu, et al., A stacked contractive denoising auto-encoder for ECG signal denoising. Physiol. Meas. 37(12), 2214–2230 (2016)

[9] Candès, E. J., & Wakin, M. B. (2008). An introduction to compressive sampling. IEEE Signal Processing Magazine, 25(2), 21–30.

[10] J. Xu, L. Xiang, Q. Liu, et al., Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE Trans. Med. Imaging. 35(1), 119–130 (2016)