Image Denoising Using AI with Entropy as Metric Analysis

Mallellu Sai Prashanth[1(✉)], Ramesh Karnati[1], Muni Sekhar Velpuru[2], and H. Venkateshwara Reddy[1]

[1] Department of Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, India
saiprashanth08@ieee.org, {ramesh.krnt, h.venkateswarareddy}@vardhaman.org

[2] Department of Information Technology, Vardhaman College of Engineering, Hyderabad, India
munisek@vardhaman.org

Abstract. This paper introduces a novel denoising approach Image denoising is one of the fundamental challenges in the field of image processing and computer vision. Our main aim of the project is to get a complete noiseless image with high accuracy and less time. So, in our project we are proposing an effective denoising technique using RNN (Recurrent neural network) for fixed pattern noisy images which may reduce the usage of number of auto encoders. Here, we are passing images into the recurrent neural networks as pixel information in the form of a 3D coordinate system. RNN doesn't migrate the information from one node to another node until it gets its basic requirements. As we are using a single auto encoder, it will reduce noise as well as time complexity. The statistical analysis is going to be observed by using the following metric considerations, namely SNR (Signal to noise ratio), PSNR (Peak signal to noise ratio), MSE (Mean square error) and Entropy. From this research work we are going to get a complete noiseless image.

Keywords: RNN · Flatten 2D layers · Relu · Sentiment · Natural Language Processing · Collective Intelligence · Entropy

1 Introduction

Noise is typically defined as a random variation in brightness or colour information. The presence of noise in an image might be additive or multiplicative. In the Additive Noise Model, an additive noise signal is added to the original signal to produce a corrupted noisy signal. Similarly, the Multiplicative Noise Model multiplies the original signal by the noise signal. There are different types of noises namely Gaussian noise, salt and pepper noise, poison noise, impulse noise, and speckle noise

- **Gaussian Noise:** It is commonly known that Gaussian noise is statistical noise with a probability density function (PDF) equal to the normal distribution. Gaussian noise has a uniform distribution throughout the signal.

O. Castillo et al. (Eds.): PerSOM 2023, LNICST 517, pp. 169–181, 2024.
https://doi.org/10.1007/978-3-031-66044-3_12

- **Salt and Pepper Noise:** A type of noise commonly seen in photographs is salt and pepper noise. It manifests as white and black pixels that appear at random intervals. Errors in data transfer because this form of noise to appear.
- **Poison Noise:** Poisson noise is produced by the image detectors and recorders nonlinear responses. This type of noise is determined by the image data
- **Speckle Noise:** Unlike Gaussian or Salt and Pepper noise, speckle noise is multiplicative noise. This type of noise can be found in a wide range of systems, including synthetic aperture radar (SAR) images, and ultrasound imaging.

The Denoising concept is classical chapter in Image processing, In Ancient, before use of Emerging Technologies in Image processing, there exists some filters to smoothen the images and increase the sharpness in the pixels of an images. As Image processing and Denoising concept became trend with emerging tools and techniques Introduction of Machine Learning in Image processing is started with minor simulations which changes the Properties and Metrics of the Images. The Machine Learning consists of three different types of Algorithms like Supervised Machine Learning Algorithm, Unsupervised Machine Learning algorithm and Reinforcement Machine Learning algorithm. Here by In Image Processing Mostly till now Achieved the observations and change in properties in Images by Super vised Machine Learning Algorithms to achieve the good Accuracy and Time Complexity with smooth and Easy Simulation. Till now the Image denoising using Machine Learning done by using Auto Encoders and CNN (Convolution Neural Networks). Neural Networks are similar and imitates the functionality of human Brain. The Neural Networks consists of nodes and links were connected to each other. In this research work the denoising concept is going to implement with RNN (Recurrent Neural Network) and Single Auto Encoder. Image denoising is always a challenging task in the field of computer vision and in image processing. Image denoising is the process of removing noise from the original image. Addition of noise will cause loss of information in an image. So, to get noiseless image, this paper is going to use effective denoising technique. Image denoising plays an important role in a wide range of applications. There are many denoising techniques existing, but failed to get an accurate output.

Collective Intelligence is implemented with Machine Learning and Deep Learning Algorithms. But in order to achieve good Accuracy and to overcome above all the drawbacks this research is initiating with Advanced supervised neural network RNN with advanced optimiser Adam and Relu Filters.

2 Block Diagram

(See Fig. 2).

2.1 Methodology

In this research work unsupervised machine learning algorithms is developed with the help of powerful python libraries like NumPy, keras and matplotlib. Here the denoising concept is applied for the large number of datasets at a time. Till now the literature review just shown about the denoising concepts for single images with supervised Machine Learning algorithm. From Keras datasets Minist Dataset is extracted in order to perform

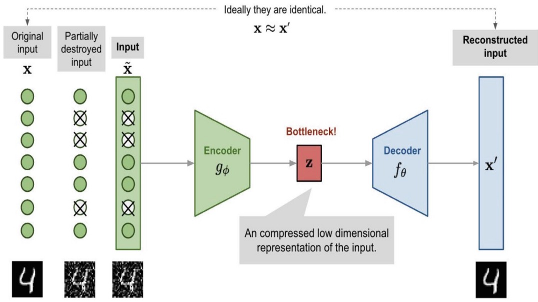


Fig. 1. Block Diagram of Auto Encoder

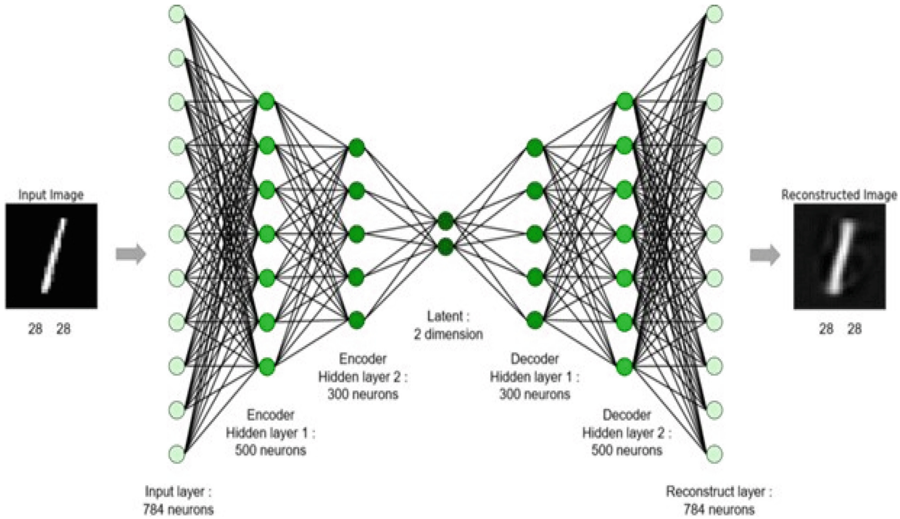


Fig. 2. Internal Architecture of RNN in Auto Encoder

the experimental analysis of denoising concept. The Minist Dataset consists of the Images of Hand written digit recognitions where it consists of 70,000 of images. The all images of hand written digits will be available in gray scale. Among those 70,000 images only 10,000 Images will be considered for testing and remaining 60,000 Images will be considered under training. Each image in the MNIST dataset is 28 pixels by 28 pixels. Let us look Fig. 3 that describes the samples of the Minist Dataset.

Fig. 3. Samples of Minist Dataset

Testing Dataset: Test data is data specifically designed for use in experiments, usually in a computer program. Some data can be used in a valid way, usually to ensure that a given set of inputs for a particular task produces the expected result. In this research work nearly 10,000 images are taken as testing Dataset.

Training Dataset: Training data (or set of training data) is the first data used to train machine learning models. Training data sets are provided with machine learning algorithms to teach them how to guess or perform the required task. In this project work nearly 60,000 Images are considered as Training dataset

After Segregation of training and testing of dataset in Machine Learning the dataset will be loaded to a model in order to perform the experimental analysis or task of denoising concept. In Machine Learning Model the dataset is loaded using powerful python library named NumPy. NumPy is a very popular Python library for working with large arrays and multidimensional arrays using large collections of high-level mathematical functions. Very useful for basic scientific calculations in machine learning. The Minist dataset images will load under 3D cartesian system. It describes like (Fig. 4):

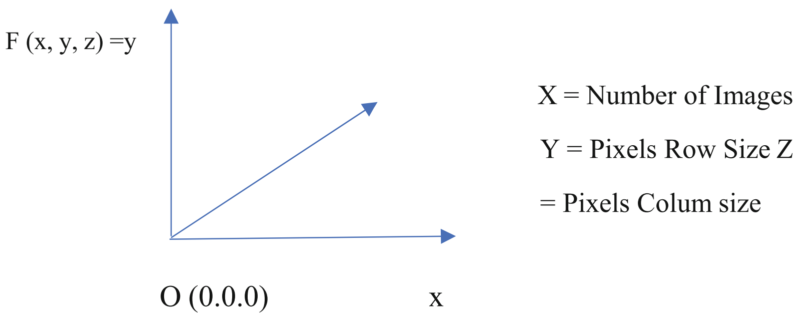


Fig. 4. Representation of 3d-Cartesian System

Example: In this research work 60,000 images are taken under training dataset for ML Model so if the NumPy function loaded training data then it represents as:

- ***F (x, y, z) = (60,000, 28, 28)***.

3 Formatting Data for Keras

This research project in the process of denoising concept it will flatten the two-D array of snap shots into a vector of $28 \times 28 = 784$ numbers. It is no matter how we flatten the array, so long as we're regular between images. From this perspective, the MNIST pixels are just a bunch of factors in a vector space of 784-dimensional. However, the records need to constantly be of the format "(range of records points, information factor size)". In this case, the training information may be of format $60{,}000 \times 784$. For the above process this project uses the 2D Flatten Layers in Neural Network. Then the 3D cartesian function converts into the 2D cartesian function. Let us say like **F (x, y, z) = G (a, b)**. After formatting the Minist dataset to keras the function which was obtained by 2D flatten layers results as **G (a, b) = (10,000, 784)** for the testing dataset of images in ML model and similarly, for the training dataset the format function G (a, b) results in **G (a, b) = (60,000, 784)**

Note: The 2D Flatten layers are used for only uniform size of pixels in images for the purpose to achieve better accuracy in less time complexity

4 Noise Factor

While solving the problem statement, our project goal is to make a model that is capable of performing noise removal on images. To be able to do this, the research use existing images and add them to random noise. Here we will feed the original images as input and we get the noisy images as output and our model (i.e., autoencoder) will learn the relationship between a clean image and a noisy image and learn how to clean a noisy image. So, let's create a noisy version of our MNIST dataset and give it as input to the decoder network.

We start with defining a noise factor which is a hyperparameter. The noise factor is multiplied with a random matrix that has a mean of 0.0 and a standard deviation of 1.0. This matrix will draw samples from a normal (Gaussian) distribution. While adding the noise, we have to remember that the shape of the random normal array will be similar to the shape of the data you will be adding the noise.

$$\textbf{Noise Factor} = \textbf{RMS Noise} = \boldsymbol{\sigma}(\mathbf{S}),\ \text{where } \sigma \text{ denotes the standard deviation.}$$
$$\text{RMS is used because, } \textbf{Noise Power} = (\textbf{RMS Noise})^{\mathbf{2}}.$$

To ensure that our final images array item values are within the range of 0 to 1, we may use np. Clip method. The clip is a NumPy function that clips the values outside of the Min-Max range and replaces them with the designated min or max value. The Fig. 5. Represents the noisy images of Minist Dataset

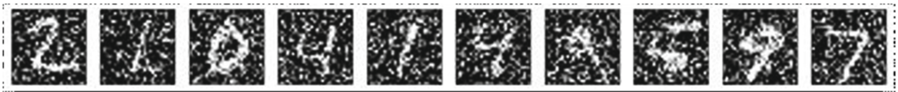


Fig. 5. Noisy Images of Minist Dataset. *Note: Noise Factor varies from 0 to 1 always and as, the noise factor increases the noise in the images will be increases*

Hence, till now the exploratory of data analysis and data building for the machine learning model is done with the help of keras to format the huge data for the machine learning model to compile and experiment the denoising concept. Now the neural network and autoencoder building part will be started.

4.1 Recurrent Neural Networks

In this Project RNN (Recurrent Neural Networks) are built to perform the metric analysis of Denoising concept. Recurrent neural networks (RNN) are state-of-the-art data sequencing algorithm and are used by Apple Siri and Google voice search. The first algorithm that remembers its input, due to internal memory, makes it perfectly suited for machine learning problems involving sequential data. From Keras the project is build and brought up by sequential model.

5 Brief About Auto Encoder

Autoencoder is a type of neural network that can be used to learn a compressed representation of raw data. An autoencoder is composed of an encoder and a decoder sub-model. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder

The Auto Encoder is shown in Fig. 1 block diagram, as from the reference of Fig. 1 the auto encoder consists of three parts namely Encoder, Code Block (Bottle Neck) and Decoder

5.1 Role of Encoder

An encoder is a neural network that transmits, fully connected and presses the input into an image of the hidden space and encodes the input image as a compressed representation at a reduced size.

- **Role of Code Block (Bottle Neck):**
  The code is a compact "summary" or "compression" of the input, also called the latent-space representation.
- Role of Decoder:
  The decoder then reconstructs the input only using this code (Fig. 6).

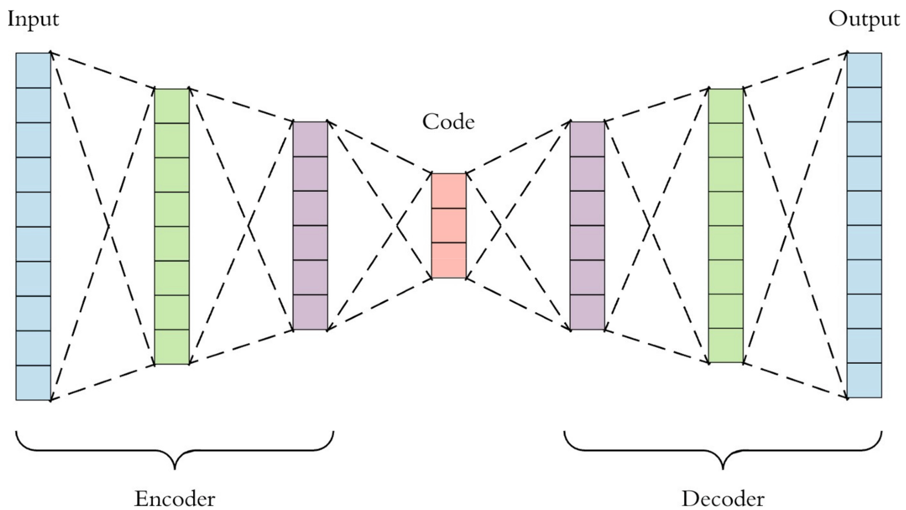


Fig. 6. Visualization of Auto Encoder

There are 4 hyperparameters that we need to set before training an autoencoder:

- **Code size:** number of nodes in the middle layer. Smaller size results in more compression.
- **Number of layers:** the autoencoder can be as deep as we like. In the figure above we have 2 layers in both the encoder and decoder, without considering the input and output.
- **Number of nodes per layer:** the autoencoder architecture we're working on is called a *stacked autoencoder* since the layers are stacked one after another. Usually stacked autoencoders look like a "sandwich". The number of nodes per layer decreases with each subsequent layer of the encoder, and increases back in the decoder. Also, the decoder is symmetric to the encoder in terms of layer structure. As noted above this is not necessary and we have total control over these parameters.
- **Loss function:** we either use *mean squared error (mse)* or *entropy*. If the input values are in the range [0, 1] then we typically use entropy, otherwise we use the mean squared error.

6 Metric Consideration During Denoising of an Image

In this research work mainly focused on two metric values namely PSNR and Entropy. Generally, an image consists of several metrics like Contrast, Brightness, Mean Square error, PSNR and Entropy etc., Among that PSNR and Entropy are very related to the denoising concept of Image

Entropy: Entropy is generally defined as the average bit rate of Information. In denoising concept is generally discussed as the in-image processing, discrete entropy is a measure of the number of bits required to encode image data. The higher the value of the entropy, the more detailed the image will be. Formula for Entropy is shown below

$$H = -\sum_{i=0}^{255} p_i \ \log_2 p_i$$

Peak Signal to Noise Ratio (PSNR): Peak signal-to-noise ratio (PSNR) is the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of its representation. In this denosing Concept by the help of Mean Square error the PSNR is extracted. The obtained formula for PSNR is shown below

$$PSNR = 10\ log\frac{(255)^2}{MSE}$$

Note: *PSNR is inversely proportional to the Mean Square Error value of an image. As, MSE increases PSNR decreases. As, MSE decreases then PSNR increases.*

7 Denoising Concept on Single Image

This research is implemented on both heavy large type of datasets as well as single individual images. Initially the image is loaded to the Collab, then after with the help of different types of noise filters like Gaussian noise filter, Poison Noise filter, Speckle Noise filter, salt noise filter, pepper noise filter and Salt and Pepper Noise filter the image is submerged with noise in its pixels. The level of noise is determined by the noise factor. As, noise factor increases a greater number of noises is added in the images and its pixel values.

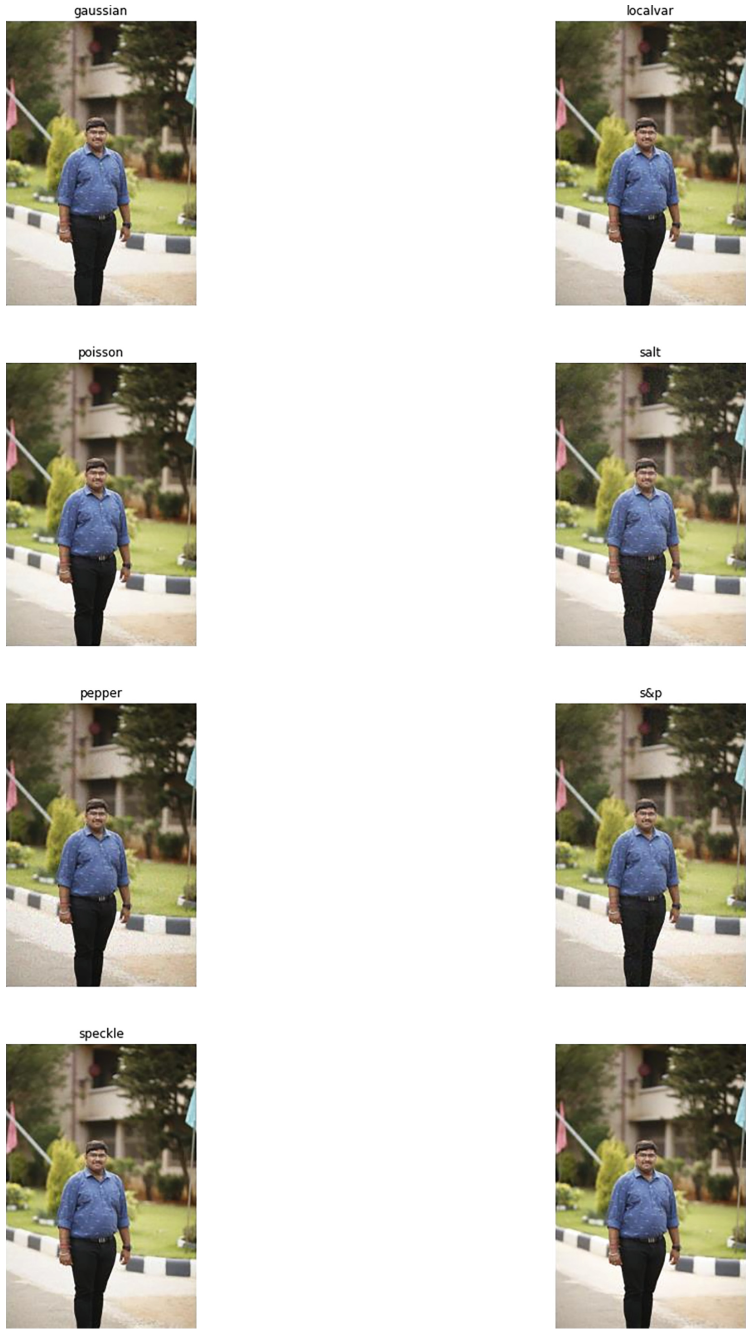
About Noise Factor:

$$\text{Noise Factor (F)} = \frac{\text{SNR at input}}{\text{SNR at output}}$$

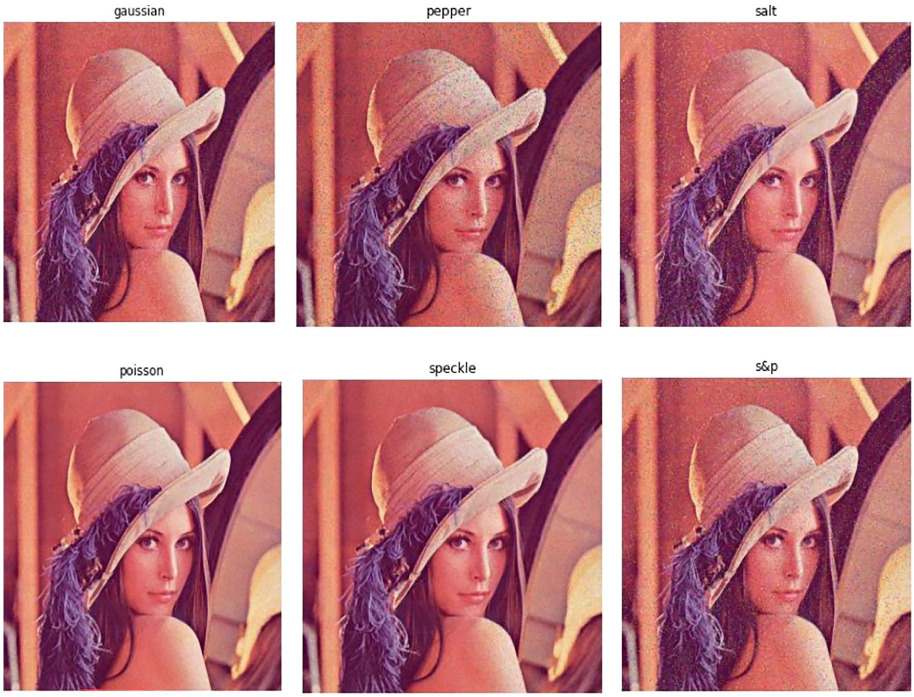
Image noise is random variation of brightness or colour information in the images captured. The Noise factor has its individual formula where it decides the level of noise should apply on the image and its pixels.

The existing method proposed an encoder decoder model with direct attention, which is capable of denoising and reconstruct highly corrupted images. This model consists of encoder as well as decoder where encoder is a convolutional neural network and decoder is a multilayer Long short-term memory network. Encoder reads image and catches the abstraction of that image in a vector, where decoder takes the vector as well as corrupted image to reconstruct a clean image. In the existing technique using Deep Convolution neural network, even though applying a number of epoxies, it failed to get an accurate output and, in the end, we have observed that there is some noise present which is called SPARSE noise. So, to remove sparse noise as well, here, in this research work, the input image will be denoised completely and produce the required accurate output. To avoid that noise and improve accuracy of the output, this project introduces a new technique called RECURRENT NEURAL NETWORK. After passing an input image to a recurrent neural network, it will not migrate from one node to another node until the entire noise will be removed and it also requires a lower number of epoxies and will be getting high accurate output.

The below shown different types of noises applied on single Picture:



Let us take another individual Picture Lena to add different types of Noises:



8 Observations

The below observations are caluculated for three categorical pictures like Test image, Different types of noisy image and Denoised Images Entropy's and PSNR is caluculated. Note: As PSNR increases MSE decreases, if Mean Square error decreases that indicates noisless output. For Denoised Image PSNR should be greater or similar to the value of Test Image

- Observations for MINIST Dataset+:

Type of Image	Entropy Value
MINIST TEST IMAGE	2.820261
MINIST NOISY IMAGE	4.950857
MINIST DENOISED IMAGE	3.401394

- **Observations of Individual Images of Lena.JPG: (PNG IMAGE READINGS)**

Type of Image	Entropy Value
Lena Test Image	12.353033
Lena Gaussian Noise Image	12.029046

(*continued*)

(*continued*)

Type of Image	Entropy Value
Lena Poisson Noise Image	11.623464
Lena Speckle Noise Image	11.689871
Lena Salt Noise Image	12.028935
Lena Pepper Noise Image	11.996971
Lena Salt and Pepper Noise Image	12.049296
Lena Gaussian Denoised Image	12.895645
Lena Poisson Denoised Image	12.035689
Lena Speckle Denoised Image	12.032112
Lena Salt Denoised Image	12.569789
Lena Pepper Denoised Image	12.325654
Lena Salt and Pepper Denoised Image	12.289699

- Observations of Individual Image Prashanth.png: (JPEG IMAGE READINGS)

Type of Image	Entropy Value
Anees Test Image	10.227338
Anees Gaussian Noise Image	10.744131
Anees Poisson Noise Image	10.385127
Anees Speckle Noise Image	10.553204
Anees Salt Noise Image	10.296273
Anees Pepper Noise Image	11.360876
Anees Salt and Pepper Noise Image	11.130279
Anees Gaussian Denoised Image	11.23564
Anees Poisson Denoised Image	10.96259
Anees Speckle Denoised Image	11.02365
Anees Salt Denoised Image	10.86954
Anees Pepper Denoised Image	11.65423
Anees Salt and Pepper Denoised Image	11.54869

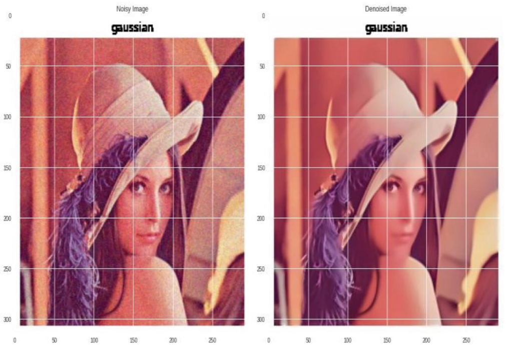
Note: After Applying Denoising Concept the average rate of Information of an image is increased

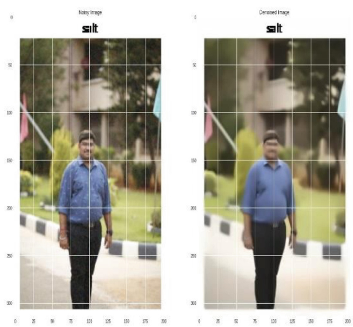
There are total 5 optimisers present from Keras Model like RMSprop, Adam, Adadelta, Adagrad, Adamax. The denoising concept is used to calculate for each and every optimiser and Adam is declared as best optimiser based upon the accuracy acquired. Let us have a look in below table about the Accuracies of different optimisers.

TYPE OF OPTIMISER	ACCURACY GOT IN EACH EPOCH		TOTAL ACCURACY OF MODEL
RMS Prop	1st Epoch	0.0524	96.58%
	2nd Epoch	0.0342	
ADAM	1st Epoch	0.0239	98.07%
	2nd Epoch	0.0193	
ADADELTA	1st Epoch	0.294	97.12%
	2nd Epoch	0.293	
ADAGARD	1st Epoch	0.03	97.02%
	2nd Epoch	0.0296	
ADAMAX	1st Epoch	0.0465	95.89%
	2nd Epoch	0.0389	

9 Results and Analysis

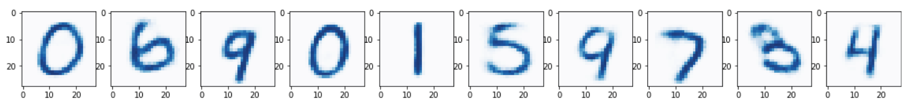
By using denoising concept several denoised pictures extracted from individual and Dataset Images.





Lena Gaussian Denoised Image

Prashanth Salt and Pepper Denoised



Minist Denoised Images

10 Conclusion

This work states the new approach of denoising model using RNN and single auto encoder where at a less time complexity with better accuracy the denoising concept can be applied for both individual and large number of datasets. From this work the Minist dataset is applied the Gaussian noise by the help of noise factor and for individual images the noise is applied with the help of different types of noise filters and seen the observations by the help of image metrics. The entropy and PSNR are considered as main metrics during this research. The average bit rate of information is calculated for three types of images like test image, noise image and denoised image. From the research observations the entropy of denoised image is greater than or equal to test image entropy. Noise image entropy is independent and it totally varies up and down based upon the noise factor. Hence by denoising concept the average bit rate of information in denoised image is going to be increased.

References

1. Jiang, X., Zhang, Y., Zhang, W., Xiao, X.: A novel sparse auto-encoder for deep unsupervised learning. In: 2013 Sixth International Conference on Advanced Computational Intelligence (ICACI), pp. 256–261. IEEE (2013)
2. Xing, C., Ma, L., Yang, X.: Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. J. Sens. **2016**, 1–10 (2016)
3. Gondara, L.: Medical image denoising using convolutional denoising autoencoders. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 12–15 December 2016, pp. 241–246. IEEE (2016)
4. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: Advances in Neural Information Processing Systems, pp. 769–776 (2009)
5. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in Neural Information Processing Systems, pp. 341–349 (2012)
6. Agostinelli, F., Anderson, M.R., Lee, H.: Adaptive multicolumn deep neural networks with application to robust image denoising. In: Advances in Neural Information Processing Systems, pp. 1493–1501 (2013)
7. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103. ACM (2008)
8. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
9. https://www.cs.toronto.edu/~kriz/cifar.html
10. https://blog.keras.io/building-autoencoders-in-keras.html
11. Convolutional auto-encoder for image denoising. https://doi.org/10.1016/j.heliyon.2017.e00393
12. LLNet: A deep autoencoder approach to natural low-light image enhancement. https://doi.org/10.1016/j.patcog.2016.06.008
13. Image denoising using deep CNN. https://doi.org/10.1016/j.neunet.2019.08.022
14. A review on Image Denoising Algorithms. https://doi.org/10.1137/040616024
15. Convolution neural network-based Image denoising for Better Quality of Images. https://doi.org/10.14419/ijet.v7i3.27.17972
16. Overview of Image Denoising Based on Deep Learning. https://doi.org/10.1088/1742-6596/1176/2/022010