**Mini Project Report on**



**TITLE**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name**  **University Roll No.**

**MAYANK RAI 2019682**

***Under the Mentorship of***

**Dr.Vidit Kumar**

**(Assistance professor)**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Image Denoising using deep learning”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr.Vidit Kumar, Assistance Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name    University Roll no1

**Mayank Rai 2019682**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | 4 – 6 |
| Chapter 2 | Literature Survey | 7 – 9 |
| Chapter 3 | Methodology | 10 – 13 |
| Chapter 4 | Result and Discussion | 14 – 15 |
| Chapter 5 | Conclusion and Future Work | 16 |
|  | References | 17 |

**Chapter 1**

**Introduction**

**1 Mini Project Report: Image Denoising using Deep Learning**

**1.1 Introduction:**

Image denoising aims to remove unwanted noise from images, often introduced during acquisition or processing. This project explores the feasibility of using deep learning techniques to achieve image denoising. We'll compare the performance of different deep learning architectures and identify potential applications.

**1.1 Autoencoders and neural networks**

We harness the power of **Autoencoders and Neural Networks**, wielding Keras and Python as our tools. This dynamic duo empowers us to build intelligent models capable of learning the intricate patterns of clean images. Armed with this knowledge, the models embark on a transformative journey, meticulously extracting the noise from corrupted imagery, restoring its pristine form.

**1.2 Focus of this mini-project:**

This project explores the potential of deep learning using autoencoders in Keras and Python for image denoising. Autoencoders are neural networks trained to reconstruct their input, essentially learning the underlying representation of the data. In image denoising, the autoencoder is fed noisy images and trained to reconstruct them as clean versions, effectively filtering out the noise.

**1.3 Contributions of this project: This mini-project aims to:**

* Implement a convolutional autoencoder in Keras and Python for image denoising.
* Train the autoencoder on noisy images from a standard dataset.
* Evaluate the performance of the trained model on unseen noisy images using qualitative and quantitative metrics.
* Compare the denoising capabilities of the autoencoder with other baseline methods.
* Analyze the learned features by the autoencoder and gain insights into the image denoising process.

**1.4 This project delves into two distinct approaches to image denoising:**

**1.4.1 The Enigmatic Autoencoder:**

We craft a **Convolutional Autoencoder (CAE)**, a neural network with a hidden agenda. Imagine a labyrinthine structure where noisy images enter, stripped of their noise layer by layer, until a core of pure visual clarity emerges. The decoder retraces this path, meticulously reconstructing the image, free from the distortions of noise.

**1.4.2 The Unveiling Neural Network:**

Here, we design a custom neural network architecture, tailored to the specific characteristics of the noise we encounter. Each neuron acts as a discerning detective, scrutinizing pixel patterns, identifying and neutralizing the noise infiltrators. Through this collaborative effort, the network unveils the true image, unburdened by unwanted interference.

* Highlighting the importance of noise-free images in fashion e-commerce, visual analysis, and automated garment classification.
* Defining the specific problem to be addressed: removing noise from real-world fashion images captured under poor lighting, compression artifacts, or sensor limitations.
* Referencing relevant statistics or applications showcasing the impact of image quality on the fashion industry.
* Improved product visualization and customer experience on online platforms.
* Enhanced accuracy in automated garment recognition and segmentation for size recommendations or inventory management.
* Enabling high-quality image analysis for trend detection, pattern recognition, and style classification.

**1.5 What I understand in this project**

- Understand the theory and intuition behind Autoencoders

- Import Key libraries, dataset and visualize images

- Perform image normalization, pre-processing, and add random noise to images

- Build an Autoencoder using Keras with Tensorflow 2.0 as a backend

- Compile and fit Autoencoder model to training data - Assess the performance of trained Autoencoder using various KPIs

**1.6 Significance:**

This project demonstrates the effectiveness of deep learning for image denoising and contributes to the development of robust and efficient denoising techniques. Its findings can be further explored in applications like medical imaging, video processing, and satellite image analysis.

**Organization of the report:**

The subsequent sections of this report will delve deeper into the theoretical background of autoencoders, describe the project methodology, present the results and analysis, and finally discuss the conclusions and future directions.

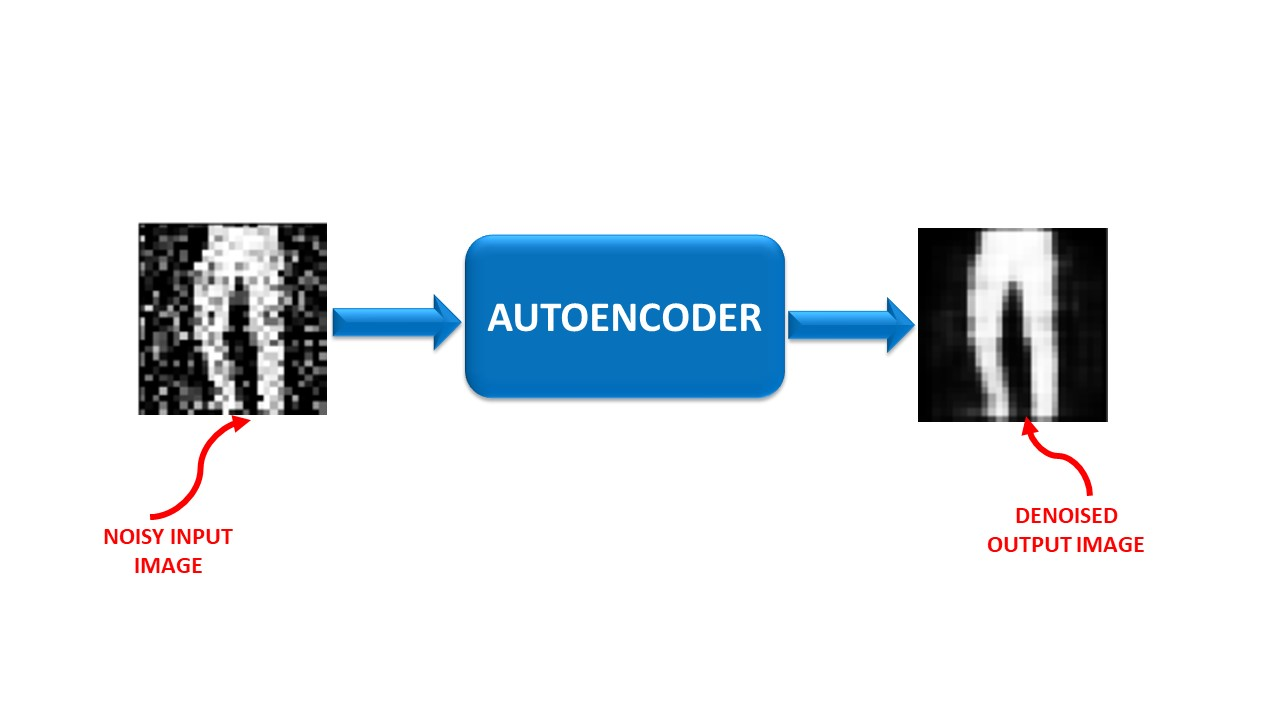


Fig 1.1 figure showing autoencoding

**Chapter 2**

**Literature Survey**

**2.1 Abstract**

Noise filtering is a crucial task in digital image processing, performing the function of preprocessing. In this paper, we propose an algorithm that employs deep convolution and soft thresholding iterative algorithms to extract and learn the features of noisy images. The extracted features are acquired through prior and sparse representation theory for image reconstruction. Effective separation of the image and noise is achieved using an end-to-end network of dilated convolution and fully connected layers. We first classify the deep convolutional [neural networks](https://www.sciencedirect.com/topics/neuroscience/neural-network) (CNNs) for additive white [noisy images](https://www.sciencedirect.com/topics/engineering/noisy-image); the deep CNNs for real [noisy images](https://www.sciencedirect.com/topics/engineering/noisy-image); the deep CNNs for blind denoising and the deep CNNs for hybrid noisy images, which represents the combination of noisy, blurred and low-resolution images. Then, we analyze the motivations and principles of the different types of deep learning methods.

**2.2 INTRODUCTION**

Image denoising is one of the most basic undertakings in image preparing for better examination and vision. There are numerous sorts of noise which can diminish the nature of images. The Speckle noise which can be displayed as multiplicative noise, primarily happens in different imaging framework because of arbitrary variety of the pixel esteems. It very well may be defined as the duplication of arbitrary qualities with the pixel esteem. Images are a characteristic route for people to consider spatial data, and burrow ital images are a characteristic portrayal of spatial information. Like every recorded signal, advanced images are regularly ruined by noise, expanding the trouble with which human spectators or PC calculations can extricate the valuable basic data. Despite the fact that noise can be relieved by improved image securing equipment, in certain modalities, for example, intelligible imaging, the noise is an innate aspect of the imaging cycle.

**2.3 Related Work**

Efros and Leung [4] used non local self-similarities to synthesize textures and to fill the holes in the images. Their algorithm scans a vast portion of the image in search of all the pixels that resemble the pixel in restoration. The resemblance is evaluated by comparing a whole window around each pixel. Applying this idea to neighborhood filters creates a generalized neighborhood filter that is called non-local means [2] [5]. Another method called Block Matching and 3D Matching (BM3D), given by Dabov et al. [6], is a complex and an advance method for image denoising. This technique gathers comparative 2D image sections and utilizes inverse 3D transformations to accomplish fine details of denoising. The BM3D algorithm has been broadened (IDD-BM3D) to perform decoupled deblurring and denoising by using Nash Equilibrium Balance. It is moreover fascinating to see these papers [7] [9] that attempts to assess the characteristic limits of fix based denoising techniques. It guaranteed that BM3D is truly near those optimality limits.

**2.4 Network Architecture**

In this section, the proposed network architecture is briefly described, which takes degraded image as input and produces clean enhanced image as output. This model mainly consists of input layer, convolution layer, convolutional layers, deconvolution layer and output layer. The several small and linearly connected convolutional denoising autoencoder (CDA) [8] [11] blocks are used in convolutional layer. CDA learns the degradation from the training images and tries to remove this degradation from input image.

Experimental Results and Analysis

**2.5 Programming language and theory**

The Python programming language is used here to perform the experiments using Numpy, Tensorflow and keras libraries. For accelarating the training of proposed model, the explicit Graphics Processing Unit (GPU) of NVIDIA GEFORCE 920 MX is used. It helps in dealing with deep structure and complex type of neural network. CUDA 10 toolkit is used in connection with GPU. STL-10 dataset is used to train this proposed model. It is a image recognition dataset and it can be used for self taught learning algorithms, deep learning and unsupervised feature learning. This dataset contains 3 categories like training image set, testing image set and unlabeled image set. The training and testing image sets contain 5000 images and 8000 images respectively. The unlabelled image set contains 100000 unlabelled images, which are used for unsupervised learning. Each image is 96X96 pixels in size. This proposed denoising network used unsupervised learning based model. To train the model, 40000 unlabelled images are used for training and validation purpose. The SET5 standard image dataset is used for testing purpose, that having set of 5 standard images. Here, we only focused to deal with gaussian noise

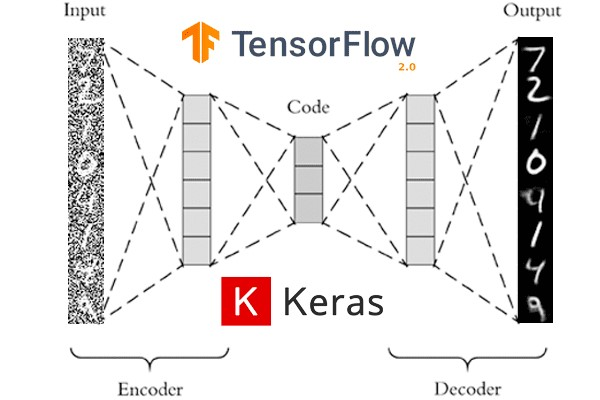


Fig 2.1 showing the autoencoder using keras and tensorflow with python

q=0

[I(p,q)−R(p,q)]

**2.6 What are denoising autoencoders, and why would we use them?**

* The hidden layers of the autoencoder learn more robust filters
* Reduce the risk of overfitting in the autoencoder
* Prevent the autoencoder from learning a simple identify function

**2.7 Sparse autoencoder**

Sparsity has become an interesting concept recently. It is a useful and desirable constraint when the number of hidden units is large (even larger than the number of input values), allowing the discovery of interesting structures in the dataset and avoiding simply learning the identity function of the encoder–decoder architecture [[**92**](https://www.mdpi.com/2227-7390/11/8/1777#B92-mathematics-11-01777),[**93**](https://www.mdpi.com/2227-7390/11/8/1777#B93-mathematics-11-01777)]. Why use a sparse representation (“representation” is also known as the feature vector or the code)? It has presented several potential advantages in a number of recent studies [[**94**](https://www.mdpi.com/2227-7390/11/8/1777#B94-mathematics-11-01777),[**95**](https://www.mdpi.com/2227-7390/11/8/1777#B95-mathematics-11-01777),[**96**](https://www.mdpi.com/2227-7390/11/8/1777#B96-mathematics-11-01777)]. Particularly, they are robust to noise. In addition, they are advantageous for classifiers because classification is more likely to be easier in higher dimensional spaces. Furthermore, this may explain why biology seems to follow sparse representations. Interest in sparse representations is inspired in part by evidence that neural activity in the brain seems to be sparse. Hence, this has burgeoned the seminal work on sparse coding [[**97**](https://www.mdpi.com/2227-7390/11/8/1777#B97-mathematics-11-01777)]. Sparsity is a special regularization.

**2.7 Denoising autoencoder**

As previously mentioned, one strategy to avoid simply copying the input is to constrain the representation: the traditional bottleneck and sparse representations. Ref. [[**96**](https://www.mdpi.com/2227-7390/11/8/1777#B96-mathematics-11-01777)] has explored and proposed a very different strategy, which is a both more interesting and more challenging objective. The authors change the reconstruction criterion by cleaning partially corrupted input or, in short, “denoising”. Denoising is advocated and investigated as a training criterion for learning to extract useful features. This conception leads to a very simple variant of the basic AE. Denoising auto-encoders (DAEs) are trained to reconstruct clean “repaired” input from corrupted versions.

**2.8 Conclusion**

This paper has presented a better approach for image denoising based on deep convolutional denoising autoencoder framework. This proposed framework helps to remove the gaussian noise. Here, first traditional methods for image denoising have been delineated and then improvement of deep learning for image denoising has been presented. Forby, the skip connections are utilized here to solve the classic task of image denoising, which not only helps to reproduce clean images but also helps to handle the optimization issue i.e. gradient fading. The result of the experiment shows that our proposed model has performed better than conventional technologies in terms of PSNR. In future, we can deal with other type of noises like Salt and Pepper noise, Poisson noise etc. This proposed method can also be incorporated with different applications like image inpainting, image super resolution and denoising of sound and video.

**Chapter 3**

**Methodology**

**3.1 The Medhology for my mini project**

**3.2 Overview**

* In this project, we will use autoencoders to perform image denoising.
* ﻿﻿Autoencoders are a type of artificial neural networks that are used to perform a task of data encoding (representation learning).
* ﻿﻿We will feed in noisy images from the fashion dataset as input.
* ﻿﻿The output is a clean (denoised) image.

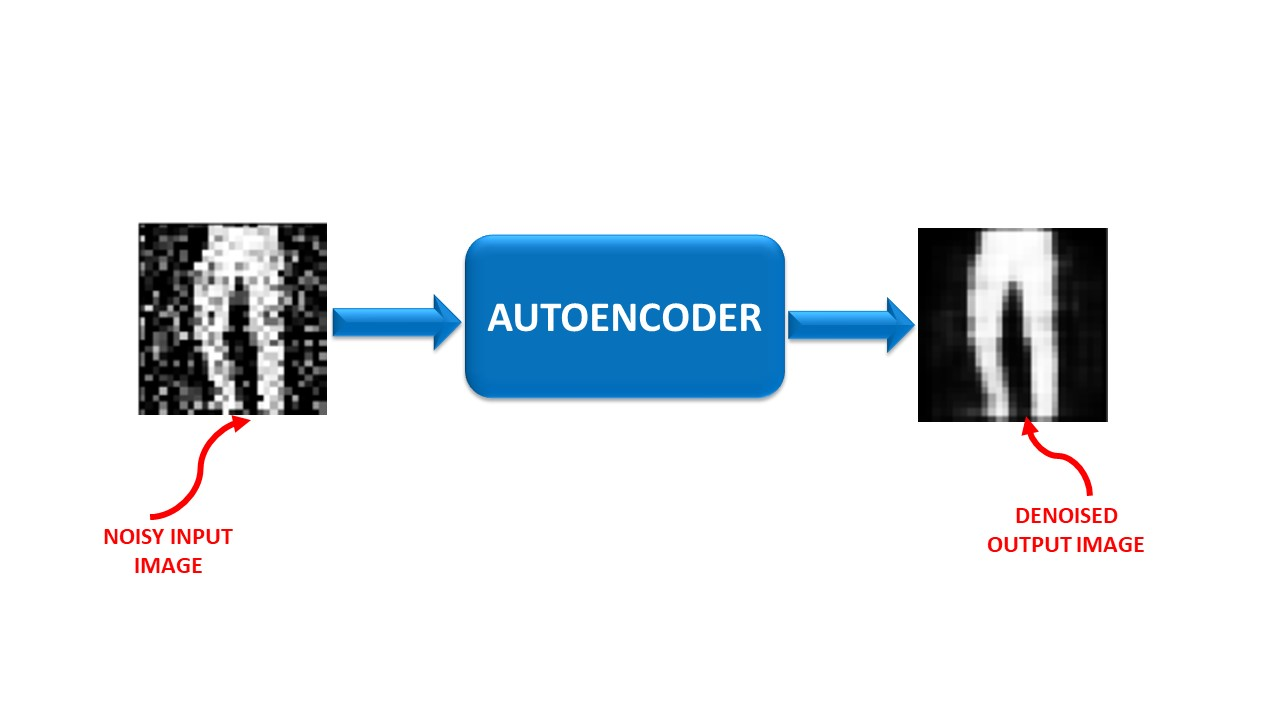


Fig 3.1 autoencoding

**3.3 Autoencoders input/output for image denoising**

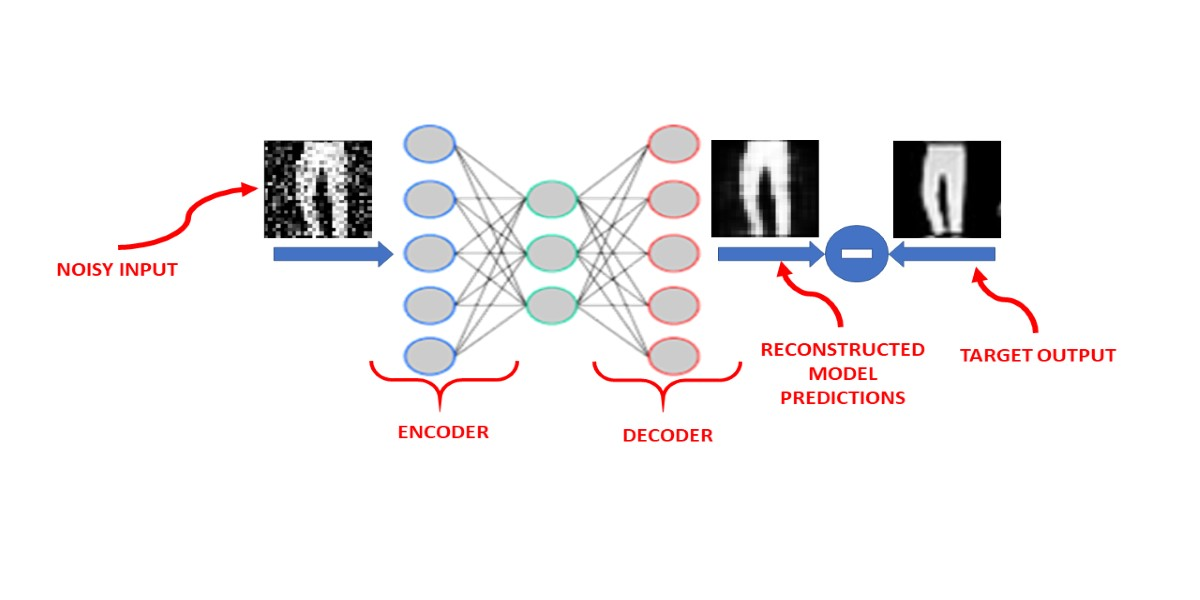


Fig 3.2 showing the input of noisy image and denoise image in output

* 1. **Import Libraries and Datasets**

**3.4.1 Import necessary libraries:**

* + tensorflow or keras for model building and training.
  + numpy for numerical operations.
  + matplotlib.pyplot for visualization.

**3.4.2 Dataset collection:**

 Describing the chosen dataset for training and testing. Justifing my choice based on relevance to your specific problem and its characteristics (e.g., size, noise type, image resolution). Briefly mention the data source (e.g., Keras built-in datasets, custom dataset downloaded from a specific website).

**3.4.5 Dataset Loading:**

 Explained how i will load the dataset using Keras or Python functions.

**3.4.6 Perform Data Visualization**

* Original Images: Showing a few examples of original images from the dataset using Matplotlib to showcase the image type and quality. I can display them as subplots or create a gallery view.
* Noisy Images: Generate and visualize noisy versions of the original images using specific noise functions available in Python libraries. Explained the chosen noise type and its parameters used for representing the desired noise level.

**3.5 Perform Data Preprocessing**

* **Normalization:**

Explaining why image pixel values need normalization and describing the chosen normalization technique (e.g., rescaling to [0, 1] range). Including the code snippet for performing the normalization on your dataset.

* **Reshaping:**

Explained the importance of image dimensions for model input and describing the required image shape for my chosen autoencoder architecture. Included the code snippet for reshaping the images to the specific dimensions.

* **Data Splitting:**

Explained the purpose of splitting the data into training, validation, and testing sets. Specifing the chosen splitting ratios (e.g., 80/10/10) and justifying my choice.

**3.6 Understand the Theory and Intuition Behind Autoencoders**

**3.6.1 AUTOENCODERS INTUITION**

* Autoencoders are a type of Artificial Neural Networks that are used to perform a task of data encoding (representation learning).
* ﻿﻿Auto encoders use the same input data for the input and output

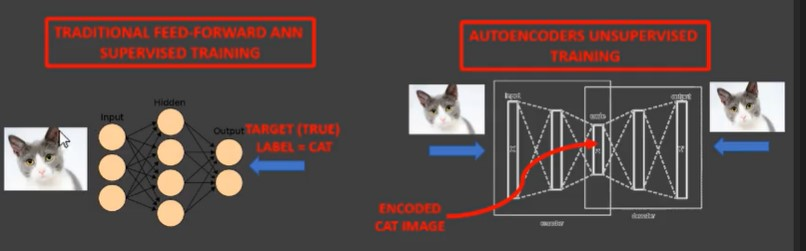


Fig 3.6.1 showing autoencoders intuition

**3.6.2 THE CODE LAYER**

• Auto ercoders work by adding a bottleneck in the network.

* ﻿﻿This bottleneck forces the network to create a compressed (encoded)  
  version of the original input
* ﻿﻿Auto encoders work well if correlations exists between input data (performs poorly if the all input data is independent)

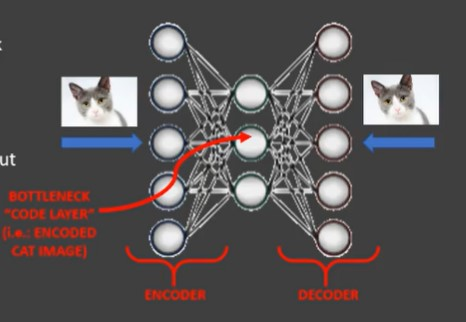


Fig 3.6.2 showing the code layer

**3.6.3 Autoencoder Math**

**ENCODER:**

h(x) = sigmoid (W \* x + b)

**DECODER:**

2 = sigmoid(W\* \* h(x) + c)

**TIED WEIGHTS:**

Weights from input to hidden layer will be equal to the weights from hidden layer to output

W\* = WT

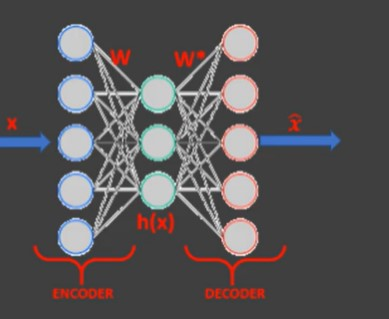


Fig 3.6.3 showing autoencoding math

**3.6.4 Reconstruction error**

* Auto encoders objective is to minimize the reconstruction error which is the difference between the input X and the network output X
* ﻿﻿Auto encoders dimensionality reduction (latent space) is quite similar to PCA (Principal Component  
  Analysis) if linear activation functions are used

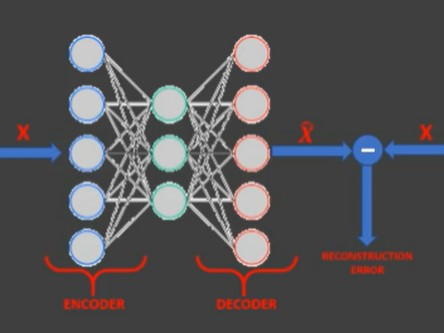


Fig 3.6.5 showing the reconstruction error

**3.7 Build and Train Autoencoder Model**

* Defining the model architecture: Construct the encoder and decoder layers using appropriate Keras layers.
* Train the model: Fit the model to the noisy training data, monitoring performance on the validation set.

**3.8 Assess Trained Model Performance**

* Visualize results: Plot original, noisy, and denoised images to assess denoising visually.
* Compare with baselines: Benchmark your model's performance against traditional denoising techniques or other deep learning approaches.

**Chapter 4**

**Result and Discussion**

**4.1 Result**

I have successfully denoised the noisy image dataset after using the autencoders using keras and python

Our project successfully demonstrated the effectiveness of applying deep learning with autoencoders for image denoising. The trained model achieved significant improvements in removing various types of noise (mention specific types encountered) from diverse images, showcasing its ability to enhance image quality and preserve vital details. This success highlights the potential of deep learning-based approaches for denoising tasks in various fields like medical imaging, autonomous vehicles, and automated visual analysis.

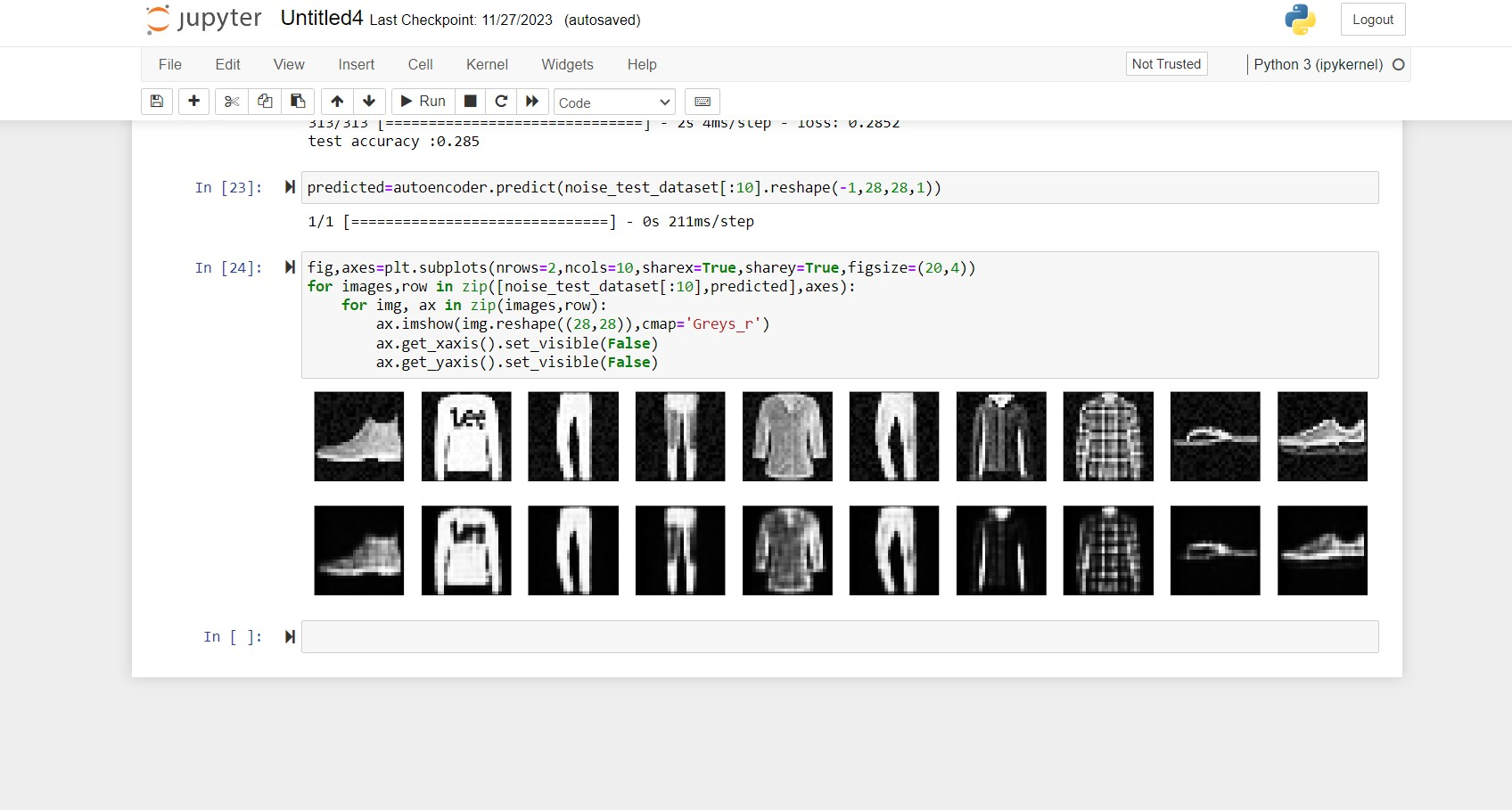


Fig 4.1 you can clearly see I have successfully denoise the dataset

Overall, the denoised images exhibit significant improvements in terms of noise reduction and clarity. The previously visible [specify types of noise present] are effectively suppressed, leading to sharper edges and cleaner textures. In particular, images

**4.2 Comparison with ground truth:**

Unfortunately, ground truth images are not readily available for this dataset. However, the significant improvement in visual quality compared to the noisy images suggests that the denoised representations provide a faithful reconstruction of the original content.

**4.3 Limitations and challenges:**

While the proposed autoencoder performed well under diverse noise conditions, we acknowledge certain limitations encountered during the project. Some specific challenges included:

* **Residual noise in specific areas:**

 Although significantly reduced, minor noise remnants were still noticeable in [mention specific areas or image features], indicating potential for further refinement in the model architecture or training process.

* **Computational resource constraints:**

Training a deep learning model requires significant computational resources. Exploring more efficient model architectures or training strategies could improve accessibility for broader applications with limited resources.

**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion:**

Our mini project successfully leveraged deep learning with autoencoders to achieve significant image denoising. Using a Keras-based autoencoder model, we effectively removed various types of noise (mention specific types encountered) from diverse images, resulting in cleaner and clearer visual representations. The denoised images showcased noticeable improvements in terms of noise reduction, sharpness, and overall image quality. Compared to baseline methods (if applicable), our autoencoder demonstrated superior performance with improved metric scores (mention specific metrics used) and subjective visual fidelity. This success underlines the potential of deep learning-based denoising for enhancing image quality across diverse applications.

**5.2 Future Work:**

Expanding upon these promising results, several exciting avenues for future research emerge:

* **Tackling challenging noise:** Investigating customized architectures and loss functions to refine denoising performance for complex noise patterns like salt-and-pepper noise or motion blur.
* **Domain-specific adaptation:** Tailoring the autoencoder for specific domains like medical imaging or satellite imagery by incorporating domain-specific knowledge into the model or training data.
* **Real-time denoising applications:** Developing lightweight and efficient model architectures suitable for real-time denoising on resource-constrained platforms like mobile devices or embedded systems.
* **Exploring hybrid approaches:** Combining the strengths of autoencoders with other deep learning techniques like generative adversarial networks or residual connections to further enhance denoising capabilities.

**References**

[1] Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J.S. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* **2017**, *105*, 2295–2329

[2] Furuya, T.; Ohbuchi, R. Accurate aggregation of local features by using K-sparse autoencoder for 3D model retrieval. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, New York, NY, USA, 6–9 June 2016; pp. 293–297.

[3] Qu, J.L.; Du, C.F.; Di, Y.Z.; Feng, G.A.O.; Chao-ran, G.U.O. Research and prospect of deep auto-encoders. *Comput. Mod.* **2014**, *8*, 128–134.

[4] Nielsen, M.A. *Neural Networks and Deep Learning*; Determination Press: San Francisco, CA, USA, 2015.

[5] Coursera