Denoising Images
D203 Course Project, IIT Bombay

K. P. Annirudh
210070009
Electrical Engineering
IIT Bombay
Email: 210070009@iitb.ac.in

Madhav Gupta
21d070043
Electrical Engineering
IIT Bombay
Email: 21d070043@iitb.ac.in

Devesh Soni
21d070025
Electrical Engineering
IIT Bombay
Email: 21d070025@iitb.ac.in

***Abstract*— Image denoising is a applicable issue found in diverse image processing and computer vision problems. There are various existing methods to denoise image. The important property of a good image denoising model is that it should completely remove noise as far as possible as well as preserve edges. Through our study we want to present a review of some major work in area of image denoising. There have been numerous published algorithms and each approach has its assumptions, advantages and limitations.**

***Index Terms*—Denoising, Auto-Encoders, Convolution, Spatial Domain Filtering, BM3D, Wavelet Transform**

I. Introduction

The images that are captured in the real world come with noises. These noises can appear due to many reasons such as electric signal instabilities, malfunctioning of camera sensors, poor lighting conditions, errors in data transmission over long distances, etc. This can degrade the captured image's quality and can cause loss of information as the original pixel values are replaced by random values due to noise. So, there is a need to remove these noises from images when it comes to low-level vision tasks and image processing. The process of removing such noises from images is known as Image Denoising.

Why Deep Learning?

The task of image denoising has been an interesting area of research for decades. Over the years many techniques and ideas have been introduced for image denoising. Most of these techniques assumed these noises in images to be Gaussian noise or impulse noise.

So, to tackle this issue of denoising real-world noisy images, there is a need of using more advanced techniques. This is where deep learning comes into the picture and experiments have proved that training a convolutional blind denoising deep learning network outperforms other conventional image denoising techniques by a large margin. This is why we use deep learning for image denoising tasks.

II. Noise Models

A. Gaussian Noise

Gaussian noise is statistical noise having a probability density function (PDF) equal to that of the normal distribution. This noise is presented as the following equation:

B. Salt and pepper noise

Salt and Pepper noise is one of the most popular noises and a sparsely happening white and black pixels on images. Image pixel values are replaced by corrupted pixel values either 255 'or' 0. The probability density function (PDF) of this noise

III. Denoising using AutoEncoders

An autoencoder is a type of unsupervised machine learning algorithms that tries to encode the input images and decode it back using some number of bits from the latent space representation. It is used for the purpose of data compression or denoising in most applications. In this paper, we used autoencoder for image noise reduction purpose. In general, it is composed of two main parts: an encoder and a decoder. The encoder maps the input images into some kind of hidden representation and the decoder reconstructs the original input images from the codes generated by the encoder. The typical workflow of convolutional autoencoder network

A. About the Auto-Encoder

- The example demonstrates how to implement a deep convolutional autoencoder for image denoising, mapping noisy digits images from the MNIST dataset to clean digits images.
- The working of autoencoder includes two main components-:
  1. Encoder
  2. Decoder
- Encoder and decoder are nothing but a neural network, input is fed to an neural network that extracts useful features from the input, but the point here is that an autoencoder doesn't just need every information that neural network offers, it need precisely the features which will help him regenerate the input.
- The architecture of a neural network with one input layer, two hidden layers, and at the last we have an output layer. Input layer gets all the input data or raw data, hidden layer is extremely important as it is the layer which is responsible for extracting out some useful information and features from the input.
- One interesting thing about autoencoders is that although we mainly use it for the regeneration purpose, it is also an

excellent dimensionality reduction technique as it uses a neural network. Autoencoder is fully capable of learning non-linear surfaces, which makes it even better than principal component analysis as far as dimensionality reduction is concerned.

B. Implementating AutoEncoders using Keras

- First we import various libraries such as numpy & matplotlib to perform basic operations such as numerical operation & data visualization respectively.
- The ”mnist” dataset is downloaded and stored in training and testing datasets in the following step. As the last part of data pre-processing, we need to reshape our training and testing data.
- After this, we add some gaussian and uniform noise to the training and testing datasets.Then we clipped it between 0 to 7 to get only that part of the data.
- We also need to import some major packages from keras to perform the regeneration of image, these packages are Conv2D(to add convolutional layer), maxpool2D(to calculate max value from the convolutional spot), UpSampling2D, Input.
- We are using relu, sigmoid, softmax, selu activation functions for convolutional layers at various times during encoding and decoding. We are also using UpSampling and DownSampling of the images to process the final decoded image.
- Now we need to optimize it using adam and adadelta optimizer. After this we fit our model taking batch size of 128 and 100 epochs.This step may take several minutes to execute depending upon your GPU power.
- Our next step is plotting the difference between loss in training and testing data as well as the difference between the accuracy of training and testing data with the help of matplotlib python library.