Jaypee Institute of Information Technology, Noida



MINOR PROJECT   
(Synopsis)   
  
Image De-Noising Model using CNN-Auto-Encoder

Team Members:   
Dharmesh Malav - 19803005  
Nikhil Paleti - 19803024  
Divyanshu Tiwari – 19803025

OBJECTIVE

The objective of the proposed project is to use CNN-Auto-Encoder Models to De-Noise Images, and compare their performance with the commonly-used BM3D De-Noising Algorithm.

The main aim for the Auto-Encoder is to find the best transforming algorithm to de-noise a given “noisy” image, with the minimum loss of information from the original image.

The objective of the resulting model is to improve the results and provide better measuring scores than the conventional, Golden-Standard for De-Noising Images, i.e BM3D.

INTRODUCTION

Image de-noising can be described as the problem of mapping from a noisy image to a noise-free image. The best currently available de-noising methods approximate this mapping with cleverly engineered algorithms. In this work we attempt to learn this mapping directly with a plain multi-layer perceptron (MLP) applied to image patches. While this has been done before, we will show that by training on large image databases we are able to compete with the current state-of-the-art image de-noising methods. Furthermore, our approach is easily adapted to less extensively studied types of noise (by merely exchanging the training data), for which we achieve excellent results as well.

1. WHAT IS BM3D?

Block-matching and 3D filtering algorithm (BM3D) was predominantly used for image de-noising. This algorithm has a high capacity to achieve better noise removal results as compared with other existing algorithms at the time.   
Nevertheless, there is still much room for improvement in this algorithm to achieve more attractive results. To address the shortcomings of BM3D filtering, our paper algorithm makes the following contributions:

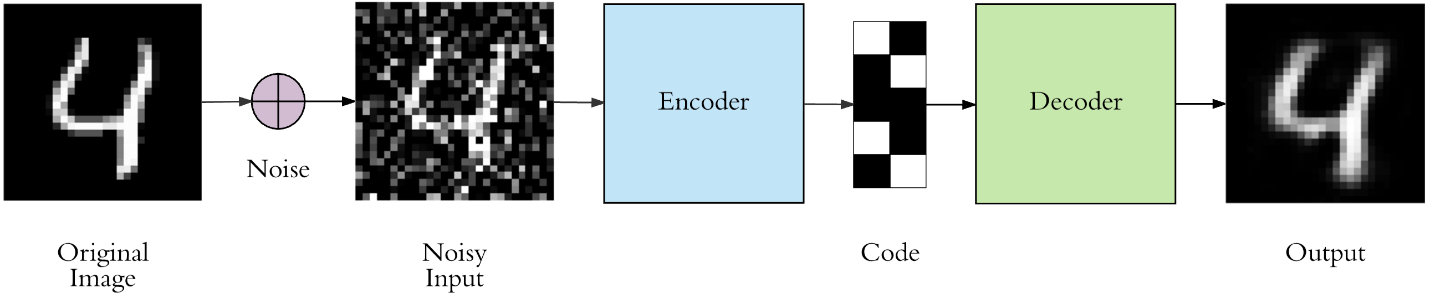
Firstly, the traditional hard-thresholding of the BM3D method is substituted by an adaptive filtering technique. This technique has a high capacity to acclimate and change according to the noise intensity. More accurately, in the proposed algorithm, soft-thresholding is applied to the high-noise areas, whereas the total variation filter is applied to the light-noise areas.

Secondly, since too small a threshold leaves the most amount of the noise without removing, in contrast, a too large threshold fails to maintain the significant information of the image such as edges. BM3D is moreover a static algorithm, irrespective of the noise-factor or the condition in which the image is taken.

With the rise of ML and Computer vision, methods and models like Auto-Encoders that can work on the very basis of retrieval of features for the noisy images can be used to improve the noise removal models.

2. AUTOENCODER

An auto-encoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The aim of an auto-encoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise”. It is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible.



An Auto-Encoder consists of 4 main parts:

1. Encoder: In which the model learns how to reduce the input dimensions and compress the input data into an encoded representation.

2. Bottleneck: which is the layer that contains the compressed representation of the input data. This is the lowest possible dimension of the input data.

3. Decoder: In which the model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.

4. Reconstruction Loss: This is the method that measures how well the decoder is performing and how close the output is to the original input.

The training then involves using back propagation in order to minimize the network’s reconstruction loss. The main essence of an Auto-Encoder is in the encoding of features it extracts from the data also called “The Code”. To find the best feature encoding, we train the model on different kernel sizes, different noise sigma, and different image sizes.

3. LATEST ADVANCEMENTS

The most recent advancements in the field of Image-De-Noising are the CNN-Auto-Encoders called CNN-DANs.

They use dense layers, high complexity loss functions like Adamax, are stacked with overfitting handling regularizers like R2, LeakyRelu and many more other features that allow these models to work the magic of removing the noise from images efficiently and preserving the features keeping the information stored in these images intact.

Experimental Design

The process of the project can be broken down into the following steps:

1. Image preprocessing

● Labelling the images,

● Crop the images to required input-resolution

● Adding Gaussian noise for training purposes.

2. The noisy training datasets are then used in the Auto-Encoder model to train the model on de-noising them.  
3. A BM3D function is also developed for the same purpose.  
4. Finally their results are compared.

Datasets and their Analysis

1. Handwritten Digits images Dataset  
2. LFW Dataset of Faces

The purpose of using these two is that the digits dataset is fairly simple and has only a single color channel, for the initial training purpose it's the best candidate. The LFW dataset can then be used with other noises and used for hyper tuning of parameters for the final model, as it's more complicated.

Handwritten Digits Dataset:

The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset. The original creators of the database keep a list of some of the methods tested on it. In their original paper, they use a support-vector machine to get an error rate of 0.8%. An extended dataset similar to MNIST called EMNIST has been published in 2017, which contains 240,000 training images, and 40,000 testing images of handwritten digits and characters. It is a classical dataset used for image classification, the dataset contains images of size (28, 28, 1) of handwritten digits. It is simple and due to having one-channeled dimensions it is good for building an initial Auto-Encoder.

LFW (Labelled Faces in Wild) Dataset:

Labeled Faces in the Wild, a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector. More details can be found in the technical report below.

1. It’s a dataset of labelled faces of personalities with images in a zip file, a text labels file and an attribute file with features like color, shape of eye, ethnicity etc.  
2. Including 13k images of facial photos of famous personalities with 73 attributes.

Research Variables

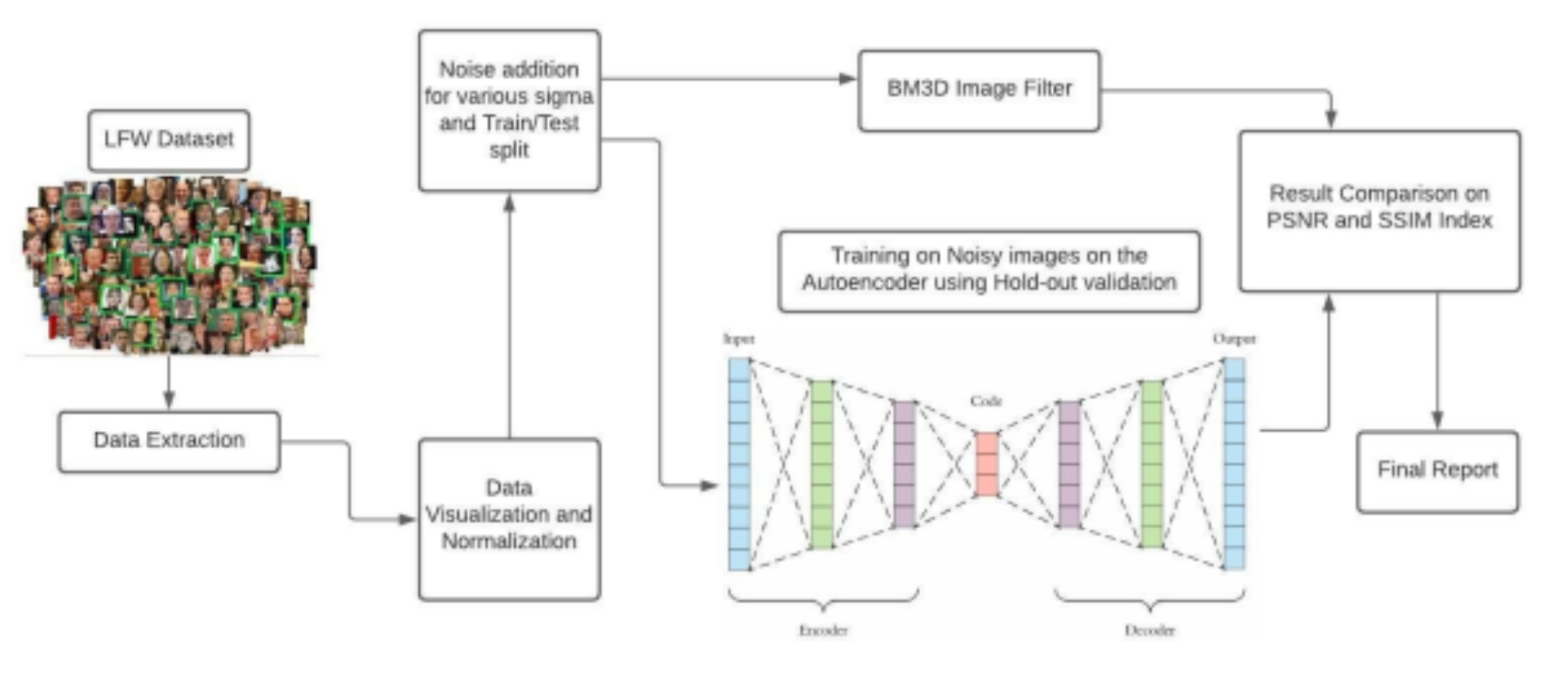
Auto-Encoders are unsupervised learning techniques in which we leverage neural networks for the task of representation learning.

The independent variable for this model is the input image that has been infused with added Gaussian noise, and the dependent variable is the resulting de-noised image.

We have used 2 image sizes, 32x32 and 64x64 for our final Model. Another important variable is the noise factor, we will use 2 factors - 0.1 and 0.2. The final variable for the model is the kernel size, we have used and experimented on 2 variants, (3, 3) kernel size and (5, 5) kernel size, what we mean by it is the size of the kernel used in CNN layers of our model, the bigger the size the more image is covered for its convolution operations.

The other variable is the Latent variable, the central representation of the data that includes its most important features just like PCA, its dependent on the input as well.

Project Procedure Diagram



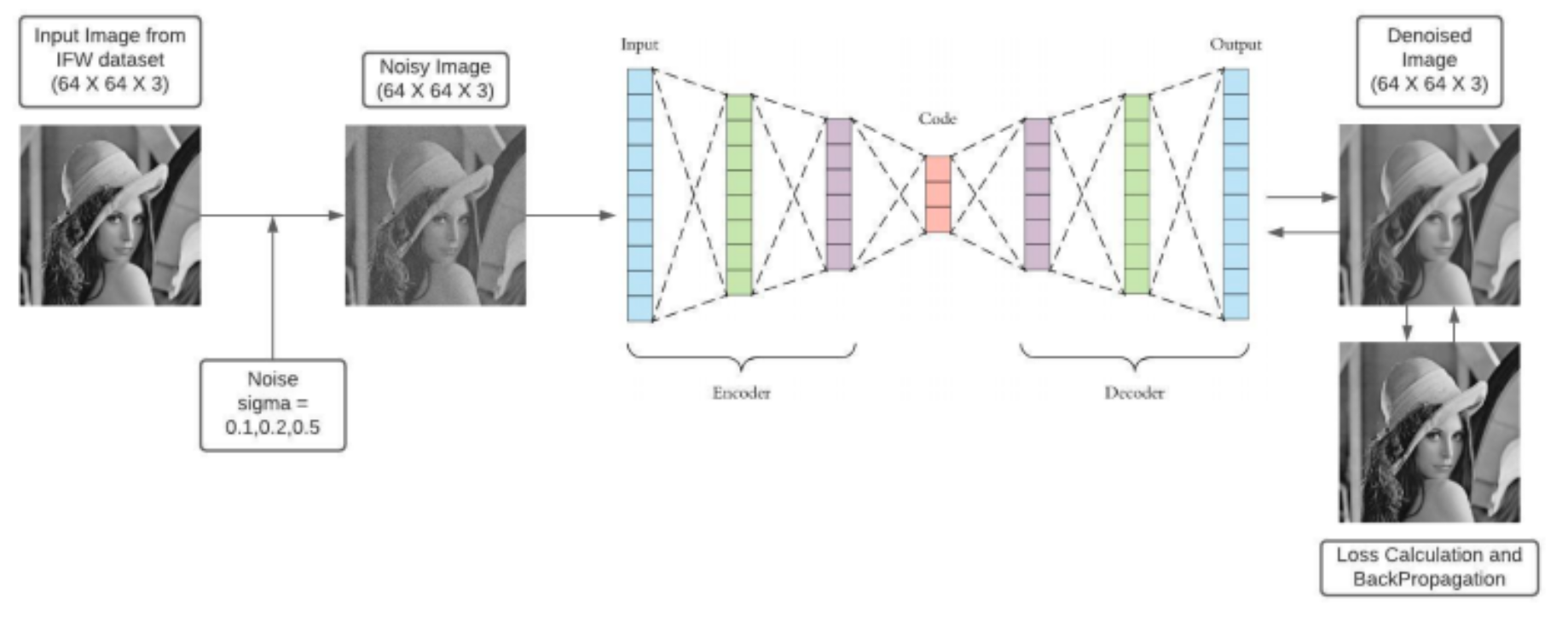
As the Project flow suggests, the first process executed for each of our models and experiments is the data preparation, adding noise to it creating ground truth images to be compared later to the outputs.

Then the data is divided and used in training and testing models and the results are then compared to that of BM3D.

For the complete test dataset, the images are first run through our model and then they are run through the BM3D function that generates the output image as well.

Both these images are then passed through the PSNR and SSIM functions are used to generate scores for each of these images, and finally a mean score for that particular dataset is generated and recorded.

Model Diagram



The Model we use is a CNN-powered Auto-Encoder, it uses 3 CNN layers in encoder for 64x64 images and 4 for 32x32 images, same for the decoder, which uses stacked Transpose layers.

The architecture is such that the middle ‘encoding’ has the best chance to capture meaningful features from the images, and use them to recreate the images without noise, for 64x64 its size is 128 as the features are more abundant, for 32x32 it is 32.

The loss functions used and optimizers deployed are discussed later.

Data Analysis

1. The images are obtained from the Kaggle in the form of zip, txt and csv formats.   
2. The images are extracted from the zip files and their corresponding labels as well.   
3. We don’t need the attributes csv file for this project.  
4. The data is firstly visualized for outliers and the standard deviation values are fixed to crop the dataset for outliers.

Research Methodology

Data Preprocessing Techniques

1. The first preprocessing technique for any image dataset is its visualization, we have used tensorflow for plotting the RGB scatter plot to detect outliers.

2. All the images below or above the threshold deviation values are removed.

3. The images sizes are fixed and stored in variables.

4. The data is normalized.

5. The Training data is formed by adding noise at various sigma values to original images.

6. The images are then converted from BGR to RGB.

7. Normalization of images is done from

8. Now, for the training data, we introduce Gaussian Noise at different rates \_\_\_\_VALUES\_\_\_\_

9. The final dataset is then split into train and test for model training.

Before Pre-processing After Pre-processing

Image Dataset Visualization

Here we have used Tensorboard web service to upload all our dataset and visualize it and plot it to learn some insights into the dataset, like outliers and what the distribution is.

The different color channels RGB were distributed along this way to detect outliers, we decided to go for pixel values less than 10 and more than 220 for RGB values in outlier detection.

Validation Techniques

1. For validation the validation\_split attribute is set to 0.2 hold-out validation for the initial model.  
2. Then we have moved on to using 5-fold cross validation but it's found to have no effect on the accuracy score.  
3. The final model validation is focused more on batch modulation than on splitting of training data.  
4. We have used ‘mean-square-error’ and ‘categorical\_crossentropy’ for validation, in the final model, both giving similar results, but since the dependent variable is not categorical, ‘cross entropy’ can’t be used.

Performance Measures

1. For the training scores, ‘accuracy’ and ‘loss’ are used to monitor the history of the model training.

2. Optimization is done through ‘adamax’ and activation function used is softmax.

3. The final results are not based on accuracy or other metrics, they are based on the levels on PSNR and SSIM in the final output images, discussed later.

4. A comparative report is done on the basis of these 2 measures between BM3D and our CNN model.

A small brief on PSNR and SSIM

1. Peak signal-to-noise ratio (PSNR) is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation.

2. The Structural Similarity Index (SSIM) index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference.

3. Basically PSNR will guide it for better noise removal, and SSIM for feature preservation.