Image Compression using Convolutional Autoencoder

MSc. Research Project
Master of Science in Data Analytics

Shreya Bhattacharya
Student ID: 18185207

School of Computing
National College of Ireland

Supervisor: Mr.Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Shreya Bhattacharya

Student ID: 18185207

Programme: Master of Science in Data Analytics **Year:** 2019-2020

Module: Research Project

Supervisor: Mr. Hicham Rifai

Submission Due Date: 17-08-2010

Project Title: Image Compression using Convolutional Autoencoder

Word Count: 7042 **Page Count** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shreya Bhattacharya ………………………………………………………………………………………………………

Date: 17-08-2020 ………………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Image Compression using Convolutional Autoencoder

Shreya Bhattacharya
18185207

Abstract

The rapid emergence of several online platforms has led to the generation of an enormous amount of data, mostly in the form of images and videos. Multimedia data including graphical, audio and video data in the uncompressed form needs noticeable amount of storage space and bandwidth for its transmission. To combat with this excessive data traffic, the necessity of suitable image compression methods has become a necessity. Image compression is the reduction in the dimension of an image for beneficial storage and transmission. Various methodologies had immerged to solve this problem, but mostly suffered a major drawback, that is the reconstructed image suffers significant data loss. To cope with this challenge this research work advocates the contribution of deep learning, by creating a convolutional autoencoder. A convolutional autoencoder model has been created with 20 different layers and filters to get a better image compression model. This unsupervised machine learning algorithm will do the image compression by applying the backpropagation and reconstruct the input image with minimum loss. To eliminate the minimum data loss, a new instance has been introduced into the architecture for performing denoising. The architecture has proven its success in image compression and denoising, however it has paved a new path in further investigation regarding the improvisation of the model in terms of better compression factor and further reduction of data loss in case of higher dimensional image.

Keywords- Image compression, Autoencoder, denoising, image

1 Introduction

The digitalization has led to the growing importance of various online forums and social networks where images and videos are the primary sources of information. With the growing usage of the web host there is a pressing need for faster image upload with an optimum file size where no significant data is lost. Extensive research work in computer vision collated with the advancement in deep learning methodologies have successfully addressed the concern with image compression techniques. Image compression techniques have evolved over the decades with its increasing importance. Initial image compression algorithm like the JPEG was majorly based on block diagrammed encoder or decoder that required human involvement. However, the application of the traditional algorithms was limited to specific image contents and formats as it was centred around rigid remodelled matrixes like the wavelet transform and discrete cosine transform paired with entropy coder and quantization. Moreover, spanning of high-resolution images and improved formats posed a challenge to traditional image compression techniques. Advent of deep learning techniques have illustrated enhanced image compression with the usage of autoencoder through minimizing the loss function (Cheng et al., 2020).

Recent deep learning algorithms have exploited the autoencoder for image compression. Implementation of LSTM recurrent network utilizing the binarization neural network layer, which substitutes the entropy coder and quantization have successfully compressed thumbnail images. This was further tested on full resolution images resulting in enhanced coding performance. However, the downside of the approach lies in the fact that it failed to utilize the actual entropy for generating the final codes in addition to missed analysis on the energy compaction property. The research work in (Cheng et al., 2018) utilized the symmetric CAE architecture followed by PCA to address the early research gaps in the domain.

To overcome this shortcomings of the traditional methods, a convolutional autoencoder has been developed, which has the potential of successful image compression, such that the reconstructed image produced by the decoder undergoes minimum loss along with the introduction of a new instance to the existing architecture that has the potential of denoising. The architecture has undergone three experiments for the evaluation of its performance for image compression and denoising along with the discussion of its success, shortcomings, and future scope.

The rest of the paper is well organized in a sequential manner. Section 2 illustrates a detailed study of the related work and the literature review, section 3 and 4 describes the research methodology and design specification respectively. Section 5 describes the implementation, section 6 is the evaluation of the results and then the final section consists of the conclusion and future work, followed by the references.

1.1 Motivation

With the rapid digitization, there has developed a necessity for the optimum storage and transfer of information in form of images and videos in almost every domain. With the ever-increasing traffic resulting from the information from the multimedia and digital representation of image, the necessity of image compression has become the priority. Multimedia data including graphical, audio and video data in the uncompressed form needs noticeable amount of storage space and bandwidth for its transmission. Despite having rampant growth in storage of mass density, the speed of the processor and the digital communicational system have developed the need for the higher data storage capability. This has overtaken the capacity of the existing technologies. Also, with the increase of the data intensive web dependent applications, the need for optimized signal compression for storage in the central repository had arisen along with the necessity for signal encoding and compression technology (Sudhakar et al., 2005). Image compression has evolved as a solution to these challenges, thus motivating the commencement of the research work that involves the development of an autoencoder model having the potential of image compression with minimum possible loss.

1.2 Research Objective

Creation of a Convolutional Autoencoder for image compression with minimum possible data loss in the reconstructed image.

With the emerging growth in data generation, the necessity of the image compression has come into place. Lossy image compression is a major challenge in the recent times. Many methodologies have been used to handle this problem but have faced a lot many challenges. This project highlights a deep learning approach of compression of data, that is a convolutional autoencoder has been created for image compression such that the reconstructed image has the minimum data loss.

2 Related Work

Image compression represents compression of image dimension without compromising with the loss of essential data. Over the period of time, with the emergence of the importance in the domain of image compression, several methodologies have immerged. The need for more adaptable image compression techniques had become a necessity in the field of media, medical sciences, and many other fields (Theis et al., 2017). Autoencoders have immerged as an optimum solution to a lot of challenges being faced in compressing an image. In this section, a detailed literature review of the most popular research works in this domain are done, which inspired the development of a novel convolutional autoencoder architecture having the potential of image compression with exceptional accuracy.

Autoencoder is an artificial neural network that works on the unsupervised machine learning algorithm for image compression. The autoencoder model learns how to compress the data efficiently by encoding the actual data and then reconstruct the data back from the compressed data. It has the ability to reduce the dimensions of the input image by ignoring the data noises (Kingma & Welling, 2014) which works with its four primary components which are the encoder, the bottleneck the decoder and the back propagation loss. Different architectures of autoencoders can be developed by varying the hyper-parameters of an autoencoder, which needs to be set before training them.

Several methods like post-processing neural methods have been applied in earlier stages but none of them showed that effectiveness as seen in the case of an autoencoder. Thus, a detailed study and a literature review of the research works on this domain has been done in this section which inspired the development of a novel architecture of a convolutional autoencoder for image compression. The review of the literary works is explained below in different sections.

2.1 Deep convolutional auto-encoder with pooling and unpooling layers

According to the research paper that was published in 2017 in the domain of deep learning, Turchenko presented how in the Caffe deep learning framework, different deep convolutional autoencoder models can be made. Five autoencoder models were created that varied in architecture by altering the layers of encoder and decoder and critical evaluation was done on the evaluation of these five models (Turchenko et al., 2017). The importance of the prevalence of the max-pooling layer in the convolutional neural network model has been proven long time back in various renowned published papers. A paper published by Nagi in the year 2011, projected the production of a real time hand movement dependent HRI interface for locomotive and adaptable robots (Nagi et al., 2011).s

2.2 Generative and Coupled Generative Adversarial Network

The work of Goodfellow in the year 2013 developed a new framework which was designed for reckoning generative models through an adversarial method, where consecutive training of two models are done that consisted of a generative model and a discriminative model((Makhzani et al., 2015). Wan illustrated in the year 2017, the success of unsupervised learning with the generative adversarial network (GANs). Two networks were structured and the advantages of these frameworks were posed vividly in the paper (Mao et al., 2017). The work of Sun highlighted the data augmentation which is sophisticated and domain-specific and explained the working of generative adversarial networks in two parts, generator and discriminator for preserving the actual data (O'Gara & McGuinness, 2019) . According to the research work of Liu, the Coupled generative adversarial networks are used for multi-domain images dataset. It helps the model or network to learn the joint distribution of the multi-domain images (Liu & Tuzel, 2016).

2.3 Deep convolutional autoencoder-based lossy image compression

Image compression has been the most widely discussed topic of this decade, which deals with the reduction of sizes of images, that are sent across web and captured on the data storage platforms. In the work by Cavigelli in the year 2016, a 12-layer convolutional network for the purpose of image compression. was proposed that included a stratified skip connection along with varied scale loss function (Cavigelli et al., 2017). Feng Jiang had developed an end-to-end compression framework which had a significant impact as it could gain image compression with additive features that work at low bit rate using two CNNS (Jiang et al., 2018). A recently published paper by Cheng in the year 2018, highlights the production of a lossy image compression architecture of autoencoders by exploiting its benefits for high level of proven efficiency (Cheng et al., 2018). According to Baldi, there is an extension for the autoencoder model which is known as Deep Autoencoder. The layers of the Deep Autoencoders works on the features of their respective layers (Baldi, 2012). Xie explained how the second layer features have corresponded to the appearance patterns of the first for constructing a model to learn the high order features. It is used for topic modelling, image search, and data compression (Xie et al., 2012).

2.4 Fast image scanning with deep max-pooling CNN

The maxpooling is used in the encoder part of the convolutional autoencoder to extract the features from the actual input image, as described by the work of Giusti. It had no parameters to train and had the unique ability of extraction of features from most robust images and becomes very good for saving time and fast operation (Giusti et al., 2013). According to Tolias, the maxpooling has 0 parameters as it has the lowest computation cost. That is why it is a fast image scanning method. The work vividly describes the re-encounter of both the search stage and re-ranking stage, by developing compact feature vectors by encoding various image regions which helped in preserving the valuable information and controlling the computational cost (Tolias et al., 2016) .

2.5 Convolution in convolution for the network in network

The concept of 'Network in network' is clearly explained in the research work of Min Lin, where micro network is instigated inside each and every layer of a convolutional neural network for complicated computation (Lin et al., 2014). The recent work of Pang showed how the network in network is an expansion of the deep convolutional neural network, that consists of alternate convolutional and pooling layers, and hence the network got its name as the 'Convolution in Convolution' network (Pang et al., 2018). Theis, introduced a new approach towards the lossy image compression optimization, which had reduced the computational cost by proposing a sub-pixel architecture (Theis et al., 2017).

2.6 Adversarial, Sparse and Variational Auto encoder

The research work of Makhzani, highlights the development of a variational autoencoder, for generating a variational inference by comparing the aggregated posterior of the hidden code vector of the autoencoder to the arbitrary prior (Makhzani et al., 2015). The combination of intelligent fault diagnosis technology and deep learning algorithms has become very prevalent in the recent time, which had a huge challenge in production of enormous amount of data. A very recent study by Hang Yin, in the year 2020 proposed a solution to this problem by proposing a data generation strategy based on WG-CNN which could extend a small sample dataset to a huge of a very good quality, by using a generator and a discriminator (Yin et al., 2020). A suitable alternative to the bottleneck autoencoder is sparse autoencoder. In the research work by Alireza and Brenden, they highlighted how sparsity helped in improvising performance in classifying subjects, that includes a good blend of activation function, sampling steps and several types of penalties (Makhzani & Frey, 2014). Further improvisation like removal of noise from an image can be suitably done by stacked sparse denoising autoencoders, as explained by the research work of Forest Agostinelli, that uses sparsity in the hidden layer assuring that the generated output should not be the copy of input (Agostinelli et al., 2013). As per Doersch, there is one fundamentally unique property of the variational autoencoder which makes it special from the vanilla autoencoders, which is useful for generative models (Doersch, 2016). Ballé explained that the variational autoencoders latent spaces allows the interpolation and the random sampling in an easier way (Ballé et al., 2017).

2.7 Least squares generative adversarial networks

LSGAN is the extension of the generative adversarial network which is used to resolve the problem of loss saturation and vanishing gradient as explained in the research work of Mao (Mao et al., 2017). The work of Johnston showed a method for 'Lossy Image Compression' that how it was dependent on recurrent and CNN for performing better than the BPG (Johnston et al., 2018).

The detailed study of the various literary works of researchers has led to a vivid understanding of the gradual evolution of various proposals of methodologies in the domain of image compression. It highlighted the advantages of different methods and their major drawbacks. Inspired by their work an architecture of a convolutional autoencoder has been developed that has the ability of compressing image with high accuracy.

3 Research Methodology

Data mining is a process of extracting the hidden and valuable information from the data, for which proper storage and analysis is an absolute necessity. For maintaining the quality of a data mining project, a proper data mining methodology is needed that ensures the quality standard of the project. Two of the renowned analytic tool providers, SPSS and Teradata with three user corporation of adopter assembled a special interest group and with the due course of time and advancement, CRISP-DM (Cross Industry Standard Process for Data Mining) was introduced. CRSIP-DM is universal, standardized and is easily accessible for the regularization of industries and for supporting proper deployment of data mining projects.

Realising its advantages, the business data mining tools slowly started to advocate the CRISP-DM methodology (Wirth, 2000).

CRISP-DM methodology is widely popular methodologies because of its numerous advantages. This being a very easily available technique, empower the people belonging to the business domain to actively participate in the process. The model is iterative, where the decision model is kept in the middle. This iterative process helps the IT professionals and business domain people to continuously incorporate changes and make the model and process relevant to the ever-changing business scenarios. The decision model ensures the maintenance of business goal and proper evaluation procedures leading to proper deployment and running of the project (Taylor, 2018).

The CRISP-DM has six stages, they are business understanding, data understanding, data preparation, modelling, evaluation and deployment. The flow between the business and data understanding are bi-directional as they are done based on the situation. The flow between the data understanding and the preparation stage is unidirectional which is followed by the modelling, if needed it may have to return to the data preparation phase. After the modelling stage is done then comes the evaluation, now this stage may lead to deployment if successful or can return to the business understanding phase, depending on the situation. The entire method follows a cyclic pattern starting from business understanding till deployment.

Below diagram shows the flow of the CRISP-DM methodology.

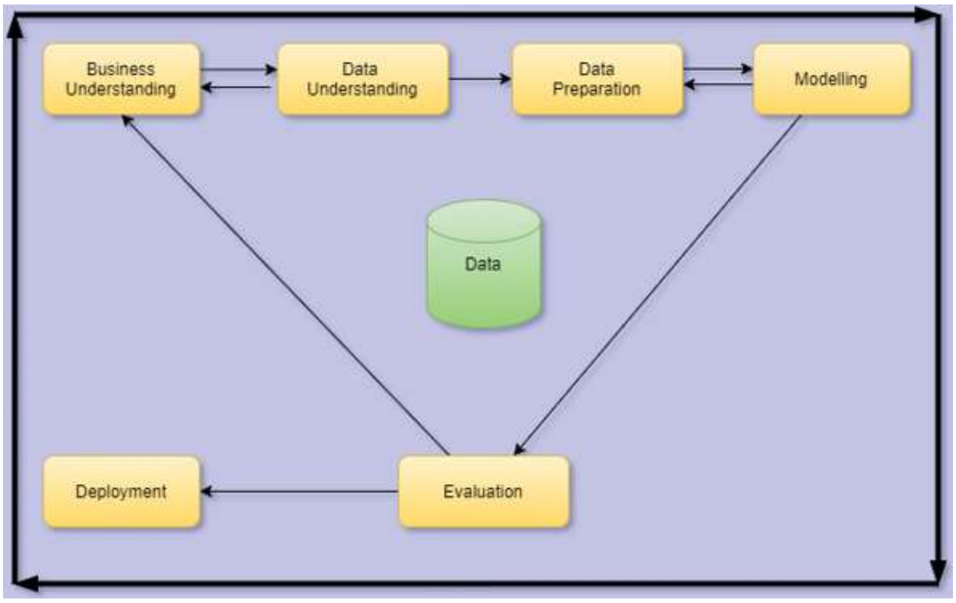

Fig 1: CRISP-DM Methodology Diagram

3.1 Business Understanding:

For efficiently developing the framework for the problem, the first and primary stage is to develop the proper understanding of the business related to it. It involves assessment of the current condition, understanding and defining the data mining goals and finally develop an appropriate plan. Business understanding can further be drilled down to two more stages. The first is understanding the business goal, where the problem needs to be clearly accessed as only a clear assessment of the problem can ultimately lead to an appropriate planning. The second is to determine the objective of the data analysis after understanding the existing scenario, thus leading to the setup of the strategy to go ahead (Wirth, 2000).

The concept of image compression is to reduce the size of an image that is measured in bytes without compromising with the image quality. If the file size gets reduced to an optimum level, a greater number of images can be stored in the disc space. The image processing time also gets reduced to a significant level while transferring through the internet or while downloading from the web sites, as it needs less bandwidth for transmission. With the growth of the data intensive web dependent applications that are intensively data driven, the need for the signal compression became a necessity along with maintenance of the adequate fidelity. (Sudhakar et al., 2005).The image compression plays a very significant role in several domains like in the field of multimedia, medical sciences, entertainment industry, IT and many more, where despite the presence of many methods and tools that could compress the image, the major challenge of image compression without compromising the quality of it still persisted. This led to the development of an autoencoder which uses an encoder to create a compact representation of the input image and a decoder to recreate the input image with minimum loss with the help of several layers.

3.2 Data Understanding:

This phase involves collection of data. Many procedures are consecutively executed to understand this collected data. Rigorous examination of the various datasets collected from different data sources are done to ensure whether the collected data is adequate to solve the business problem. Exploratory data analysis (EDA) is done to analyse the hidden information in the collected data by dividing the data into various subsets for examining the data feature and trend along with the identification of the presence of any outliers or anomaly in the data (Wirth, 2000).To be precise, this stage can be divided further into various sub stages:

3.2.1 Data Collection:

Here the data relevant to the problem statement is collected in various forms from different data sources. **MNIST digits dataset** consisting of digits that are handwritten originally sourced from a much large dataset named NIST is used. **Fashion-MNIST dataset** belonging to Zalando's article which was created to replace the MNIST digit dataset has also been used. Both have been made a part of the Keras inbuilt dataset, which are used directly using Keras.

3.2.2 Data Description:

Once the data set is being described, the contents of the dataset is vividly described in this stage, and exploration of insights for good understanding and the business understanding.

MNIST-digits. For this research, the Modified National Institute of Standards and Technology database (MNIST) dataset consisting 70,000 handwritten digit images of 28x28 pixels which are black and white are used. Along with this, Fashion Modified National Institute of Standards and Technology database (MNIST) has also been which consists of 70,000 various clothing and fashion images from 10 different classes. These are also of 28x28 pixels and are grayscale.

3.2.3 Exploratory Data Analysis:

This involves basic visualizations for studying the data. Various basic features of data including the outliers, the abnormalities, the trend are identified and studied in this phase, through various representations and visualizations. Exploratory data analysis has been done with the random samples of images from both the dataset, which shows that the images are in good quality and condition.

3.2.4 Quality assurance of the data:

Once it has been identified that the collected data is suitable for solving the business problem, the quality of the collected data is being assessed (Wirth, 2000). All the images of both the datasets are of dimensions 28 by 28 pixels, that are of high dimensional data which satisfies the research theory of reduction of the high dimensional data to lower dimensional latent variables. This assures the quality of the dataset for research work.

3.3 Data Preparation:

In this stage, data that are being spread through different files are collated from multiple data sources to form an integrated dataset. In this last stage of CRISP-DM, after the dataset is being studied, the outliers and other anomalies that are observed are handled to make the suitable for the model fit. This is done through various data cleaning steps, transformations, and pre-processing after the abstraction of the data, depending on the characteristics of the model. This step is generally performed multiple times not maintaining a specific order (Wirth, 2000). This phase can also be divided into certain sub steps:

3.3.1 **Data selection:** This step involves the selection of dataset, suitable for the implementation of the project. MNIST digits dataset consisting of 70,000 images handwritten digits in black and white and Fashion MNIST dataset consisting of 70,000 images from 10 different classes, both having dimensions suitable for performing the image compression with the architecture has been selected. Thus, it shows that the image datasets are suitable for the model fit.

3.3.2 **Data Integration**: In this stage coalition of data is being done from multiple data sources to form a single dataset which is to be used for the data analysis. Keras inbuilt datasets are used separately for the project implementation. So, the data integration step is skipped in this case.

3.3.3 **Data Cleaning:** Data cleaning is done by removing the abnormalities and doing suitable changes to make the data ready for use. There is no step involved in the cleaning of the data as these images unlike numeric or string data have no chances of presence of any abnormalities or outliers. So, the step involving the data cleaning is not valid in my case.

3.3.4 **Data Construction and Formatting:** In this stage sometimes from the existing data new characteristics are derived and introduced. This helps in gaining better insights of the data. Formatting of the data are consecutively done for efficient fitting of the model. In our model, we have used the MNIST digits and MNIST fashion dataset, both consists of 70,000 image data. Out of total images, 60,000 images were used as the training dataset and 2000 images as the validation dataset. The image dataset having dimension 28x28x1 is used. The images are in black and white form. The matplotlib library is used to view the raw data.

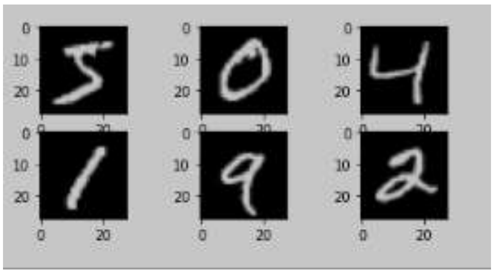

Fig 3: Visualization of the sample from MNIST-digits

The image is then represented in the form of NumPy array for further processing. The NumPy array values vary from 0 to 255, as the intensity of the pixel varies from 0 to 255. The normlaization process has been performed in the NumPy array, so that the data can be get to a certain range.

3.4 Modelling:

This phase mostly deals with the choosing of models and implementing those algorithms. A single business problem can be dealt with multiple algorithms when applied to a single dataset. Every modelling technique has their own specifications and features, based on which they are applied for solving a business problem. After the model implementation is being done, the model undergoes a testing procedure. This phase can be subdivided into multiple stages, of which the first stage is model selection, the second stage is to develop the test cases for model validation, in the third stage further models are developed according to the need of the business problem and the final stage is the examination of the models to validate the business objectives (Wirth, 2000).

To deal with the problem of image compression, a lot of methodologies have been adopted by identifying their benefits. Lossy image compression is a challenging task in deep learning in recent times. Several methods like post-processing neural methods have been applied in earlier stages but none of them showed that effectiveness as has been seen in the case of an autoencoder. The autoencoder has an encoder which creates a compact representation of the image and a decoder that recreates the input image with least possible data loss. convolutional autoencoder is responsible for keeping the spatial information of the image or data that is being passed as an input in the model using the convolutional layer. The convolutional layer used to retain the data spatial relationships. The encoder and the decoder possess the same number of layers and have hyper parameters like strides, Max pooling layer, up-sampling layer etc. It can be used for many functions like image reconstruction, latent space clustering and image colourization (Theis et al., 2017). The project here dwells on the creation of an individual convolutional autoencoder so that it could be optimized at any time rather than many other models. A convolutional autoencoder consisting of 20 different layers with different filters of different sizes, having different nodes in each layer has been constructed, with tanh activation function in the output layer. The constructed convoluted autoencoder has shown very high accuracy with minimum loss such that the loss generated is minimum.

3.5 Evaluation:

This stage mostly deals whether the implemented model could successfully justify or solve the business problem and confirmed whether the business objectives are met or not. This step highlights the future scope of the model and the research work, also it determines whether the research work can be deployed in real time or there is a need of re-evaluation by iterating the modelling task. Both modelling and evaluation work together as an iterative process for getting result to the result of the satisfaction. (Wirth, 2000)
The evaluation of the autoencoder model is determined by the value of the reconstruction loss, which determines how much the reconstructed image is close to the original image. The lesser the loss value the better the performance of the model. The performance has been judged graphically by illustrating the loss function. Three experiments were performed with the model to assess the performance of the model more intricately with respect to the generation of the reconstructed image with minimum loss and denoising performance of the model under different conditions. The evaluation has been done visually and graphically and the mean squared loss function has been identified and adam optimizer has been chosen for the model optimization.

3.6 Deployment:

After the building of the model is done, then comes the final stage where the model is fed on the live data, throughout the different sections of an organization involving the customers who are mostly involved in the implementation. Thus, the results of the model need to well be represented in front of the customers, for their proper usage. The modelling tools are maintained and monitored at a regular interval, so that any kind of bias that occurs are avoided while business decisions are being taken (Wirth, 2000)The limited time frame of the project did not permit to perform the deployment phase of the CRISP-DM methodology. Moreover, this project is more inclined towards the domain of research rather than industrial implementation. However, with the increasing demand and importance of the research work in this domain, the future objective of this project would be to deploy in the production environment.

CRISP-DM is considered as a universal process which is very useful for developing proper plan, documenting them and to maintain proper communication. This method can be considered very flexible for different problems of the business. This methodology is of great use when a large number of people are involved, and also the process being iterative, it can be modified recurrently with the change or modification of the business data. However, the CRISP-DM method has certain distinctive disadvantages like the CRISP-DM methods cannot be assessed optimally in terms of cost and time. Seldom the improvement for the small-scale business are distinctively low in case of CRISP-DM. In future if the performance of the CRISP-DM methods can be well evaluated or quantified then the judgement becomes accurate and the progress of the project can be easily controlled (Wirth, 2000).

4 Design Specification

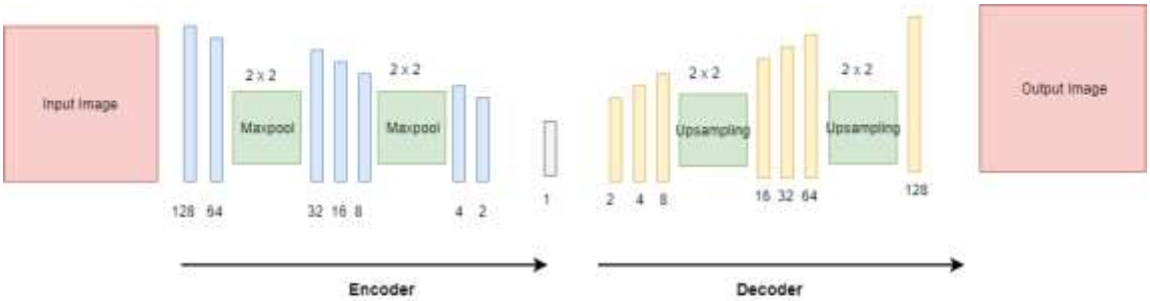

Fig 4: Architecture of the Convolutional Autoencoder

A convolutional autoencoder has been created which consists of 20 layers with different filter sizes. Convolutional autoencoder maintains the data spatial relationship, where the spatial information of the image data is fed as an input in the model which is extracted in the convolution layer. This convolutional layer is responsible for the image compression. It has the potential of learning the non-linear transformation through the multilayers and the linear activation function. Different nodes are used in each layer. In the encoder, the number of nodes decreases from 128 to 1 and in decoder increases from 1 to 128. Both the encoder and the decoder consist of 10 layers each. Convolutional layers are included responsible for successful compression of an image with high level of accuracy.

The autoencoder layer starts with an input layer, whose main function is to take the input image into the further layers of the autoencoder. In the convolution layers, the 3x3 filter having the same padding and ReLu activation function has been used. Activation functions are used in neural network for determining the output of the network, which are basically mathematical equations. This function is connected to each neuron which is responsible for determining whether the activation function would be triggered or not and is responsible for normalizing the result from each neuron. ReLu is an activation function stands for Rectified Linear Activation function which possesses incredible computational efficiency, that leads to convergence of a network. It is a nonlinear function, that is responsible for back propagation.

Following the input layer are two convolutional layers whose main function is to compress the image. Following the two convolutional layers there is a maxpool layer. Max pool layers are also used in the encoder part to select the more robust features from the input image or dataset. In the maxpooling layer, we have used the 2x2 size of the filter. After the Maxpool layer there are three convolutional layers, which are again responsible for performing the convolution function. After the maxpool layer there are again three convolutional layers that is followed by another maxpool layer. Following the maxpool layer there are two convolutional layer and one code layer. This code layer is basically the bottleneck consisting of the lowest possible compressed data with the minimum data which is 7x7x1. After the code layer there are three convolutional layers, responsible for slowly restoring the compressed image from the code layer. Upsampling layer is also included in this model in the decoder part to increase the dimension of the image from the encoded representation having filter size 2x2. The upsampling layer is further followed by three convolutional layer and then again by one three convolutional layers, and then one upsampling layer. After the upsampling layer there is one final convolutional layer, followed by the final output layer that shows the output function. For the output layer, we have used the tanh activation function, this function is also a differentiable function which can be considered as the better version of the logistic

sigmoid function whose basic advantage is that the inputs that are negative will get the value that is extremely negative and the inputs that is zero would get a value near to the zero, and is mostly used for classification between multiple classes.
The 28x28 size of image is passed as input in the autoencoder, which then passes through every layer, thus retaining the original image dimension. The model consists of 198290 parameters with the 10 layered encoders possessing 98601 parameters and the 10 layered decoders having 99689 parameters. The autoencoder is created by considering the limited computational cost for good efficiency.

5 Implementation

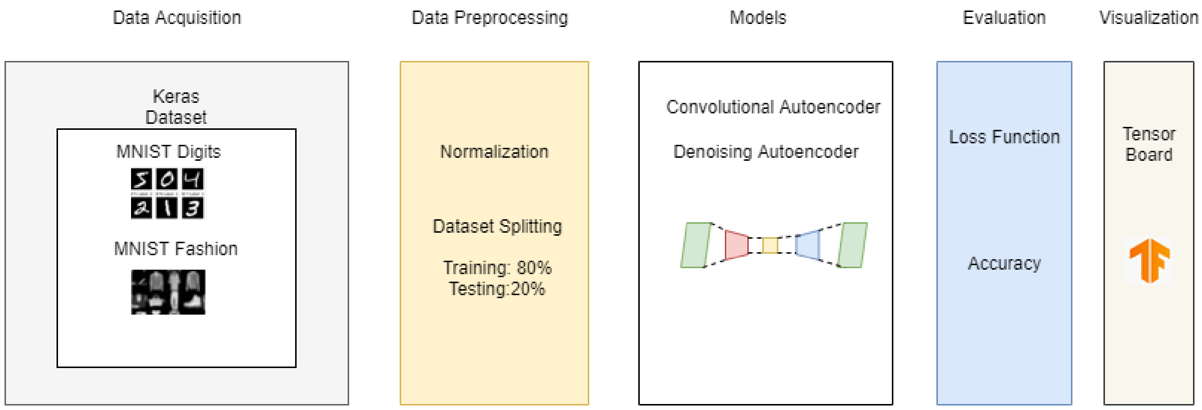

Fig 5: Project Implementation Diagram

Python programming language has been used for the application of this project that involves Python libraries like TensorFlow, NumPy, sklearn and Keras. This project has been written in the Jupyter notebook format by using Google Colab, as it provides access to the high-end GPUs, thus helps in reducing the computational cost and time. The data mining methodology CRISP-DM has been followed for the implementation of the project in an organized manner.
The above flow diagram shows a vivid description of the processes or stages followed for the implementation of the project.

At the first stage, a detailed study and analysis of the domain of the image compression is being done for understanding the business requirement of the project. With the rampant growth and advancement of the digital world there has immerged a necessity of image compression for optimum storage and transmission of data. This business understanding has motivated this research work which aims at the development of a convolutional autoencoder that possesses the ability to compress an image with minimum data loss and ability to denoise the minimal loss by introducing a denoising instance.
In the data acquisition stage, the MNIST digits and Fashion MNIST dataset has been used for the project implementation, where both datasets are present as an inbuilt Keras dataset. The MNIST digits dataset contains handwritten digit images having 70,000 images in black and white colour and are of 28x28 pixels dimensions. The Fashion MNIST dataset consists of images of various clothing and fashion items from 10 different classes in greyscale and in 28x28 pixels dimensions. Visualizations are done with random samples of images, which assures the quality of the images along with the assurance of the good condition of the images. As a part of pre-processing these images are normalized by dividing by 255 as the

pixel values vary from 0 to 255. Each of the two datasets are divided into 80 percent and 20 percent for training and testing purpose respectively.
In the modelling phase, the convolutional autoencoder consisting of 20 layers has been created with various filters and nodes in each layer. The images when passed through the encoder gets compressed at the bottleneck code layer. Then the encoded image gets reconstructed at the decoder end, with the minimum possible loss. This loss is the mean squared error loss function, that determines how accurately the reconstructed image resembles the input image. The entire process is done with the help of various layers, filters, and activation function. A new instance was introduced to this architecture for denoising the minimal loss produced for the reconstructed image.
Three experiments are then performed to assess the performance of the constructed model in the Evaluation phase. The evaluation metrics like loss and accuracy of the reconstructed image and the denoising factor is assessed visually and graph with the help of visualizations using TensorBoard. Also, the loss and accuracy of the reconstructed images and the denoising factor is judged visually by studying the output image.

6 Evaluation

This section describes the performance of the model vividly with the help of the validation dataset used for validating the graph, and by using various graphical visualizations for evaluating the model. Three experiments are performed with convolution autoencoder architecture that has been developed, where the both the datasets are divided into 80 percent training and 20 percent for the testing purpose. In the first experiment, the architecture has been trained with the MNIST-digit dataset and the various performance parameters are documented. In the second experiment the same model has been run against Fashion-MNIST dataset and the evaluating parameters of the model are re-evaluated, thus confirming the stability and consistency of the performance of the model as the dataset used is the fashion data which is similar to the original MNIST digits dataset as it also contains 28 by 28 pixels and also grey scale. In the third experiment the de-noising ability of the autoencoder has been studied, where it has been evaluated whether the minimum loss of the model can also be eliminated by the novel architecture of the model created.

6.1 Experiment 1

In this section, the autoencoder model has been trained against the MNIST-digit dataset and the performance of the model has been documented.
The model has been trained for 20 epochs, where the reconstruction loss of the model is found to be 0.0022, thus confirming the high accuracy and performance of the model. The Reconstruction Loss function that has been used for evaluating the model is Mean Squared Error Loss function, which implies how well the output image represents the input image. Mean squared error is the most commonly used loss function that is calculated by the summation of the square of the differences of the actual and estimated value. The function of the convolution autoencoder is to compress the image not compromising much with the data quality of the image. The difference between the output and the input image is determined by the value of the loss function, the smaller the value the better the performance. The decoder could finally be able to reconstruct most of the input data, with minimum possible loss.
The reconstruction loss is found to be decreasing and gets stable at the 20th epoch. The comparison of the input and the output image can evidently demonstrate the performance of the

model with respect to the minimum loss of the data. The test data and the decoded images are shown below which shows how accurately the model has the potential of producing the output images with minimum loss.

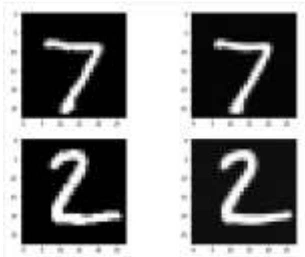

Fig 6: Sample showing the validated image on the left and the reconstructed image in the right

By comparing the sample from the validation dataset in the left and the output image in the right, it is visually evident that the image construction of the decoder happened with minimum loss, as the output image almost replicated the input image.

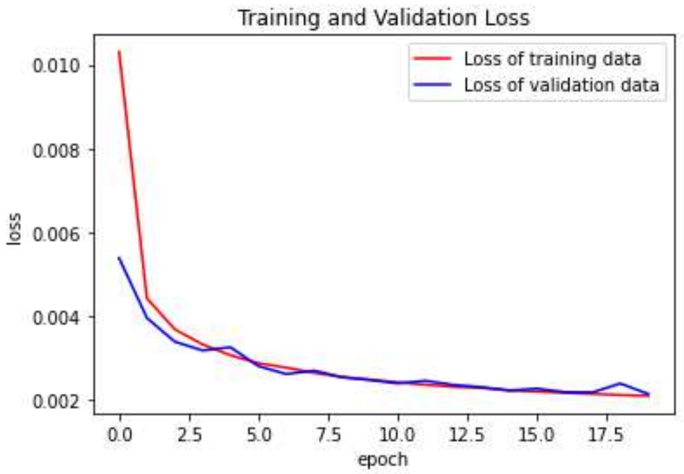

Fig:7 Graph showing the loss function of the training and the validating data.

The above graph graphically visualizes the reconstruction loss that is the mean squared error loss for the validation and training dataset. The x-axis represents the epochs that are run for training the model, and the y-axis shows the loss value for the validation and training data. Both the training and the validation data is assessed against the same number of epochs run. The line in blue represents the loss of the validation data and the line in red represents the loss of the training data.

It can be observed initially that the loss of the training data is very high as at the initial stage the model has just started to learn. While starting to train, it can be observed that the training loss has undergone a steep fall, and the rate of decrease of the loss is slowly getting reduced from the third epoch. The reconstruction loss for the training data is continuously decreasing and also gets stable at the minimum loss reaching the value of 0.0022 reconstruction loss value. The loss of the validation data also decreases and gets stable. This represents that the model is learning at every epoch. After training the model for few more epochs, that is from the epoch value 3 to the 12.5 value, there is a decreasing trend in the value of the loss and then slowly the loss value gets stabilized. The validation loss curve has almost been in line with the training loss curve until approximately the 16.5th epoch where the two lines start to diverge however at the 20th epoch, the two curves finally converges again. However, this divergence is not significant enough. Adam optimiser is a stochastic gradient descent method that is based on the principle of adaptive learning rate optimization, where each parameter learns optimally. This optimizer contributes highly to the performance of the model in cases where the dataset size is huge. It also does not require much memory loss. Hence, keeping the advantages in mind, Adam Optimizer has been used in this project for optimizing the model performance

6.2 Experiment 2

The result of the first experiment shows that the autoencoder could successfully perform image compression and the decoder can accurately reconstruct the original image with minimal loss. Hence, to re-evaluate the performance of the autoencoder for image compression, another similar dataset has been selected, which is the Fashion MNIST dataset consisting of 70,000 images of fashion images of clothing and other accessories belonging to 10 different classes. The first experiment has proven the performance of the autoencoder model but the dataset used was handwritten digits, so though the performance of the model was evitable by studying the loss function but to study the performance visually, the fashion dataset can provide better aid to the visual understanding. This fashion dataset being similar to the original dataset as it consists of images in grey scale and 28 by 28 pixels has been used to re-evaluate the performance of the model.

The model was trained with the fashion MNIST dataset, and after 20th epoch the loss value is found to be 0.0064, which is significantly very less. We can fairly say that the model performed very well in reconstructing the original image, by studying the input and the output visually.

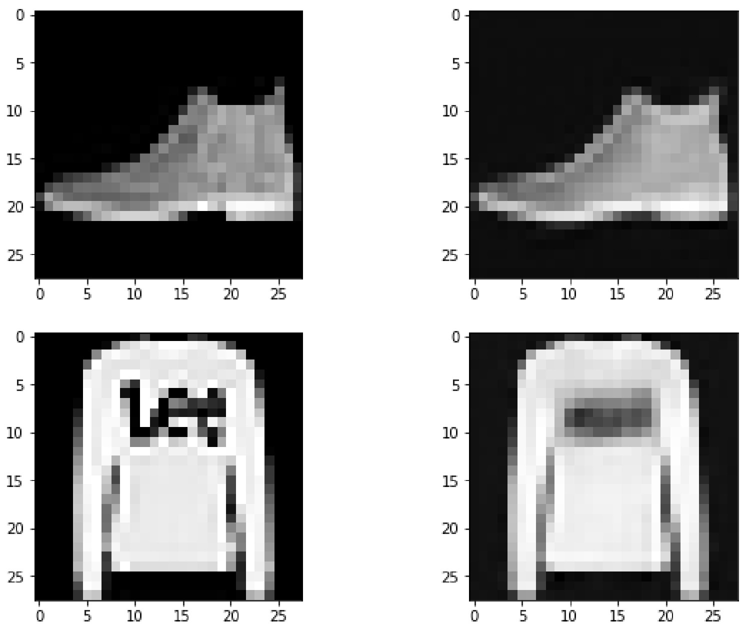

Fig:8 Sample showing the validated image at the left and the reconstructed image in the right

By comparing the reconstructed images of the autoencoder, it can be clearly stated that the novel architecture has also perform well with the second similar dataset with very minimal loss. Thus, this re-confirms the performance of the model.

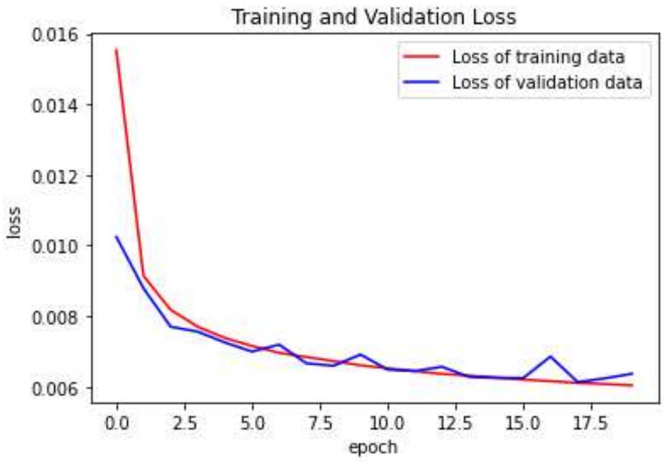

Fig 9: The graph showing the loss function of the training and the validating dataset

The graph of the loss function for the training and the validating dataset has been plotted to study the performance of the model using loss function. The x-axis represents the epochs run and the y-axis represents the loss value of the model for both the training and the validating dataset. Both the training and validation dataset are assessed against same number of epochs run. The blue line represents the loss of the validation dataset and red line shows the loss of the training dataset.

It can be observed that the initial loss of the training dataset is very high as the model has just started to learn. It can be observed that after running approximately 2 epochs there is a steep exponential decrease in the loss. The decrease then continues however with less rate of decrement from 2 to 10 epochs. After the 10th epochs the loss continues to fall at a steady rate to reach the lowest value at the 20th epoch that is 0.0064. The validation loss curve seems to be in line with the training data as well, which also falls at around 2 epochs. It then further shows a decreasing trend, with slight oscillations throughout. At around 15th epoch a certain spike of the validation loss has been noticed, which again seems to converge at around 20th epoch. However, this spike is not significant enough. Hence, by studying the graph it has been proven graphically the optimum good performance of the model has been re-confirmed. Here also the Adam optimizer has improved the performance of the model.

6.3 Experiment 3

Image compression has been performed by the autoencoder architecture, which has proven its efficiency on both the datasets MNIST digits and the fashion MNIST dataset. However, there remains some minimal loss in the reconstructed image. To eliminate that minimal loss, a denoising step has been introduced.

For denoising the loss of the reconstructed images, a new step has been introduced and the model is trained on a newly generated dataset consisting of the reconstructed images and the original images of the MNIST fashion dataset. A new instance of the same model has been used for de-noising and trained such that the input image is the previously generated reconstructed images and the target of the model is the original images of the MNIST fashion dataset that was initially fed.

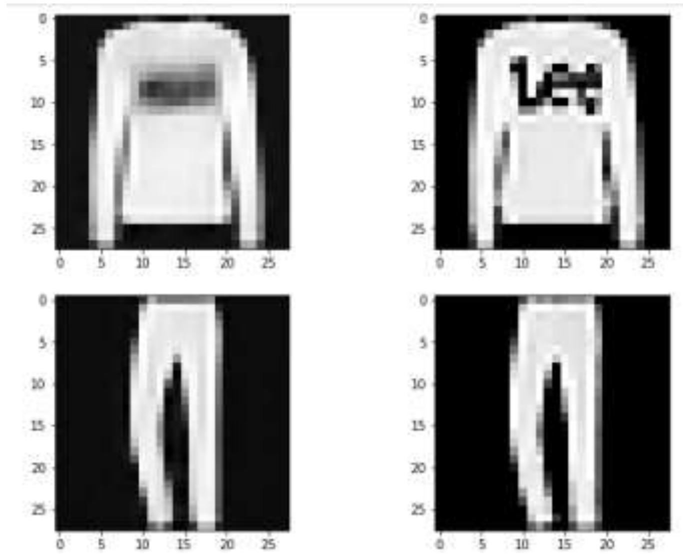

Fig 10: Sample showing the input and the target for denoising

The above snapshot is a sample of the newly constructed dataset for the denoising training, where the images in the left side shows the input which is the previously reconstructed images having noise and the images to the right shows the target image, which is the original image of the MNIST fashion dataset initially fed to the model. The model was trained until 100th epochs, where the loss value was found to be 0.1242, and the output images was found to be fairly close to the input image, which was the previously generated reconstructed image with noise.

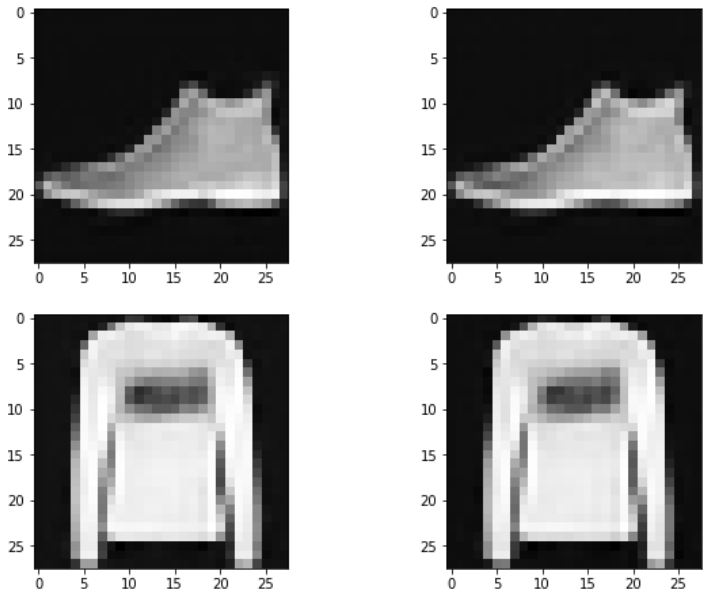

Fig 11: Sample showing the validation and the output

The new instance of the trained model has the accuracy of 0.124 at the 100the epoch. The above images on the left shows the validation dataset and the images to the right shows the output after the introduction of the denoising instance. Visually it can be clearly stated that the output image can accurately replicate the input image, thus it has the potential of successfully eliminating the noise.

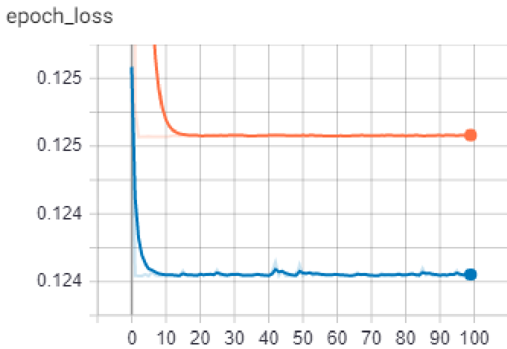

Fig 12: Validation and the training loss

The graph has been plotted using TensorBoard where the x axis shows the number of epochs run and the y-axis shows the validation and the training loss generated. The line in the red shows the train loss and the line in blue shows the validation loss. For the training loss, the initial loss value was quite high as the model was starting to learn which experiences a very steep fall at around 8^{th} epoch, after approximately 12th epochs the loss got stabilised to the value of 0.125. The validation loss showed similar trend, where the loss was also initially high, which experienced a sharp exponential fall at the third epoch, after which it gets stabilised to an accuracy value of 0.124. Thus, it can be said that the denoising loss got stabilized at the value of 0.124, thus proving the performance of the new instance, that is the denoising performance.

6.4 Discussion

A new architecture of autoencoder has been constructed which is responsible for successfully compressing an image from 28x28x1 dimension to lower dimension 7x7x1 producing a compression factor of 0.0625, such that the reconstruction loss is as low as possible. The new 20 layered architecture can successfully compress the image with minimum possible reconstruction loss. Three experiments were performed to evaluate the performance of the model. In the first experiment, the MNIST digit dataset was used to train the model. The model has produced a very good result at 20th epoch itself, where the reconstructed image produced by the decoder can fairly replicate the input image with a very minimal loss value of 0.0022. Also, the rate of learning of the model is also found to be fairly high as the loss has been continuously decreasing with the number of epochs run. The model could be well evaluated graphically as the reconstruction loss with the digits dataset is clearly understood. However, since the dataset consisted of only the handwritten digits, loss in the reconstructed images was not visible enough by comparing the validation and the output image data visually. So, to get a better understanding of the performance of the model, another similar dataset named fashion MNIST dataset is used. The output of the model is also very good, at the 20th epoch the model produced a reconstruction loss of 0.0064, which is evitable both graphically as well as visually. This helped in understanding that there is still some minimal loss in the reconstructed image. In the third experiment, a new instance has been introduced for the elimination of this noise. It is evaluated how well the model can remove even the minimal loss. The model performed well at denoising as at the 100th epoch the loss value was found to be 0.124. Though by evaluating the loss graph it can be derived that the denoising performance is very good, but still the drawback remains that the model could not produce absolute loss free reconstructed images. The model performance could have been better assessed by running the model against some higher dimensional dataset like CelebK dataset consisting of coloured images of celebrities, thus opening the scope for future work regarding further tuning of the model.

7 Conclusion and Future Work

Image compression has become the most significant problem in almost every domain like the medical sciences, entertainment, and IT field as with the rapid digitization, the rate of growth of data generation has become very high. This has led to the necessity of the development of compression of images for optimum storage of these large number of generated images and their transmission. Various methods have been used for image compression, but those methods had their own glitches. Hence the research objective of this project was to handle image compression with the deep learning approach. A convolutional autoencoder has been constructed that has successfully compressed the image of higher dimension 28x28x1 to 7x7x1 where the compression factor is 0.0625 and the decoder could successfully reconstruct the compressed image with minimum possible reconstruction loss. Introduction of a denoising instance to the architecture has been done that could successfully eliminate those minimal loss as well, thus highlighted the denoising power of the model.
Though the performance of the model was very good but there persists a future scope of optimizing the model further, so that it can compress much higher dimensional data with almost no loss and also the compression factor can further be optimized from 0.0625. A comparative study of the performance of this model can be done with multiple other methods like PCA, which are also renowned for image compression so that the model can be further optimized by identifying those optimizing factors, by studying those competitive existing models.

8 Acknowledgement

I would like to acknowledge my supervisor, Mr. Hicham Rifai for the guidance, support, and encouragement that he has provided throughout the course module.

References

Agostinelli, F., Anderson, M. R., & Lee, H. (2013). Adaptive Multi-Column Deep Neural Networks with Application to Robust Image Denoising BT - Advances in Neural Information Processing Systems. *Advances in Neural Information Processing Systems*, 1–9. http://papers.nips.cc/paper/5030-adaptive-multi-column-deep-neural-networks-with-application-to-robust-image-denoising

Baldi, P. (2012). Autoencoders, Unsupervised Learning, and Deep Architectures. *ICML Unsupervised and Transfer Learning*, 37–50. https://doi.org/10.1561/2200000006

Ballé, J., Laparra, V., & Simoncelli, E. P. (2017). End-to-end optimized image compression. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

Cavigelli, L., Hager, P., & Benini, L. (2017). CAS-CNN: A deep convolutional neural network for image compression artifact suppression. *Proceedings of the International Joint Conference on Neural Networks*, *2017-May*(November), 752–759. https://doi.org/10.1109/IJCNN.2017.7965927

Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2018). Deep Convolutional AutoEncoder-based Lossy Image Compression. *2018 Picture Coding Symposium, PCS 2018 - Proceedings*, 253–257. https://doi.org/10.1109/PCS.2018.8456308

Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2020). Energy Compaction-Based Image Compression Using Convolutional AutoEncoder. *IEEE Transactions on Multimedia*, *22*(4), 860–873. https://doi.org/10.1109/TMM.2019.2938345

Doersch, C. (2016). *Tutorial on Variational Autoencoders*. 1–23. http://arxiv.org/abs/1606.05908

Giusti, A., Cireşan, D. C., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2013). Fast image scanning with deep max-pooling convolutional neural networks. *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, 4034–4038. https://doi.org/10.1109/ICIP.2013.6738831

Jiang, F., Tao, W., Liu, S., Ren, J., Guo, X., & Zhao, D. (2018). An End-to-End Compression Framework Based on Convolutional Neural Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(10), 3007–3018. https://doi.org/10.1109/TCSVT.2017.2734838

Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T., Jin Hwang, S., Shor, J., & Toderici, G. (2018). Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4385–4393. https://doi.org/10.1109/CVPR.2018.00461

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, *Ml*, 1–14.

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 1–10.

Liu, M. Y., & Tuzel, O. (2016). Coupled generative adversarial networks. *Advances in Neural Information Processing Systems*, *Nips*, 469–477.

Makhzani, A., & Frey, B. (2014). k-Sparse autoencoders. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). *Adversarial Autoencoders*. http://arxiv.org/abs/1511.05644

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). Least Squares Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 2813–2821. https://doi.org/10.1109/ICCV.2017.304

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., & Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. *2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011*, *November*, 342–347. https://doi.org/10.1109/ICSIPA.2011.6144164

O'Gara, S., & McGuinness, K. (2019). Comparing Data Augmentation Strategies for Deep Image Classification. *Irish Machine Vision and Image Processing Conference (IMVIP)*. https://doi.org/10.21427/148b-ar75

Pang, Y., Sun, M., Jiang, X., & Li, X. (2018). Convolution in convolution for network in network. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(5), 1587–1597. https://doi.org/10.1109/TNNLS.2017.2676130

Sudhakar, R., Karthiga, M. R., & Jayaraman, S. (2005). Image Compression using Coding of Wavelet Coefficients – A Survey. *Icgst-Gvip*, *5*(June), 25–38.

Taylor, B. J. (2018). *CRISP-DM & Decision Management*. 1–9.

Theis, L., Shi, W., Cunningham, A., & Huszár, F. (2017). Lossy image compression with compressive autoencoders. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 1–19.

Tolias, G., Sicre, R., & Jégou, H. (2016). Particular object retrieval with integral max-pooling of CNN activations. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 1–12.

Turchenko, V., Chalmers, E., & Luczak, A. (2017). *A Deep ConvolutionTurchenko, V., Chalmers, E., & Luczak, A. (2017). A Deep Convolutional Auto-Encoder with Pooling - Unpooling Layers in Caffe. 1–21. Retrieved from http://arxiv.org/abs/1701.04949al Auto-Encoder with Pooling - Unpooling Layers in Caffe*. 1–21.

http://arxiv.org/abs/1701.04949

Wirth, R. (2000). CRISP-DM : Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining, 24959,* 29–39. https://doi.org/10.1.1.198.5133

Xie, J., Xu, L., & Chen, E. (2012). Image denoising and inpainting with deep neural networks. *Advances in Neural Information Processing Systems, 1*, 341–349.

Yin, H., Li, Z., Zuo, J., Liu, H., Yang, K., & Li, F. (2020). Wasserstein Generative Adversarial Network and Convolutional Neural Network (WG-CNN) for Bearing Fault Diagnosis. *Mathematical Problems in Engineering*, *2020.* https://doi.org/10.1155/2020/2604191