Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Anbar
College of Computer Science and Information Technology
Computer Science Department

Deep Learning Model for Fashions and Clothes Automated Classification

A Thesis

Submitted to the Department of Computer Science - College of Computer Science and Information Technology, University of Anbar as a Partial Fulfillment of the Requirements for Master Degree of Science in Computer Science

By

Mustafa Amer Abeud

Supervised By

Assist. Prof. Dr. Wesam Mohammed Jasim

1442A.H. *2021 A.D.*

بسم الله الرحمن الرحيم

وقل رب زدني علما

صدق الله العظيم

سورة طه: الآية 114

اسم الطالب: مصطفى عامر عبيد

كلية علوم الحاسوب وتكنولوجيا المعلومات ـ قسم علوم الحاسبات

عنوان الرسالة: نموذج التعليم العميق لتصنيف الملابس اوتوماتيكيا

كلية علوم الحاسوب
وتكنولوجيا المعلومات

UNIVERSITY OF ANBAR

جامعة الأنبار

عنوان البحث			
Pre-Convoluted Neural Networks For Fashion Classification			
جهة النشر	Bulletin of Electrical Engineering and Informatics	نوع المجلة	Scopus / Q3
حالة النشر	Published	رقم المجلة	2089-3191
رابط البحث	**https://beei.org/index.php/EEI/article/view/2750**		

اسم وتوقيع المشرف　　　　**اسم وتوقيع رئيس القسم**

Supervisor's Certification

I certify that I read this thesis entitled ***"Deep Learning Model for Fashions and Clothes Automated Classification"*** *that was carried out under my supervision at the Department of Computer Science of the University of Anbar, by the student "****Mustafa Amer Abeud****" and that in my opinion it meets the standard of a thesis for the degree of Master of Science in Computer Science.*

Signature :

Name : ***Assist. Prof. Dr. Wesam Mohammed Jasim***

Date : / /2021

Certification of the Examination Committee

*We the examination committee certify that we have read this thesis entitled "**Deep Learning Model for Fashions and Clothes Automated Classification**" and have examined the student " **Mustafa Amer Abeud**", in its contents and what is related to it, and that in our option it is adequate to fulfill the requirements for the degree of **Master of Computer Science**.*

Signature:
Name: ***Prof. Dr. Murtadha Mohammed Hamad*** ***(Chairman)***
Date: / / 2021

Signature:
*Name**: Assist. Prof. Dr. Sawsen Abdlhadi Mahmood*** ***(Member)***
Date: / / 2021

Signature:
Name: ***Lecturer. Dr. Omar Munthir Al Okashi*** ***(Member)***
Date: / / 2021

Signature:
Name: ***Assist. Prof. Dr. Wesam Mohammed Jasim*** ***(Supervisor)***
Date: / / 2021

Approved by the Dean of the College of Computer Science and Information Technology, University of Anbar.

Signature:
Name: ***Prof. Dr. Salah Awad Salman***
Title: ***Dean of the College***
Date: / / 2021

Student name: Mustafa Amer Abeud

Thesis title: Deep Learning Model for Fashions and Clothes Automated Classification

ABSTRACT

In recent years, the computer vision techniques played a major role in most of the multimedia applications and one among the important criteria to classify the application is the image classification. Image classification is one of the maximum introductory glitches in computer vision. It is used extensively in most of the digital multimedia applications in association with practical applications such as video and image indexing. In spite of the major issues in identifying the multimedia images manually has set to large weaknesses for the human beings to classify the high-resolution images and it is an insignificant weakness. The generic algorithm has a lot of weaknesses in determining the accuracy. Therefore, to classify the images, the generic algorithm with an existing strategy to an invariant number of variations has been generated. Recently, a multitude of problems have been applied to deep neural networks to obtain optimal results. Specifically, convolutional deep neural networks illustrated the best results in terms of image recognition, image segmentation, problems with computer vision and issues with the representation of natural languages.

The aim is to train CNN network using Sequential model classify fashion Data set image and Our task main is to create an effective model of deep learning to recognize accessories for clothing and augment Fashion-MNIST dataset that we use.

The Fashion-MNIST clothing classification problem is a modern standard dataset used in computer vision and deep learning. It is relatively easy because due to sharing the exact image size, training and testing data, and format splits structure. The data has been initially pre-processed for resizing and reduce the noise. data is augmented where one image will be in three forms of output; i.e. output image is rotated, shifted and zoom as an output. Finally, the data is sent to the proposed model. The proposed model which consists of three convolutional layers that extract 32, 64, 128 filters of size 3x3 with ReLU as activation function to each layer and SoftMax in last layer , then the output of the activation function of the is fed to a max pooling of 2x2 window, then Dropout To address the problem of overfitting, finally used the 'Adam' optimizer for optimization of the loss function. The trained model with the projected framework had 94% accuracy that achieved and compared with the existing works. The accuracy of the pre-Trained CNN model is used for classifying MNIST-Fashion data revealed that it is the best suited for the selected dataset.

Keywords: convolutional neural networks, Deep Learning, Fashion-MNIST dataset, Image classification, pre-Training.

Acknowledgments

First of all, I would like to express my thanks and gratitude to ALLAH the Almighty, who granted me all graces and providing me with this opportunity.

I would like to express my thanks and gratitude to my supervisor Assist. Prof. Dr. "Wesam Mohammed Jasim" for support and assistance for me in the most difficult circumstances and did not lack any scientific or moral consultation.

I am extremely grateful for the college Dean, Prof. Dr. Salah Awad Salman for all the support.

I would like to express my gratitude and appreciation for the head of the computer science department, Assist. Prof. Dr. Wesam Mohammed Jasim and the postgraduate rapporteur Dr. Mazin Abed Mohammed and Dr. Ruqayah Rabeea Al-dahhan for support and encouragement have been invaluable throughout this study and staff.

Special thanks are due to Dr. Mohammed Ibrahim and my friend Sameer I. Ali, Muhammad Abdul Razzaq, Asif Raad and Salim Hamad.

Finally, I express with my thanks to my family, my father, my mother, my brothers, my sisters and my wife, who made a great effort in order to get out to what I am now. Special thanks to my friends and my colleagues for their support.

Mustafa Amer Abeud

2021

Dedication

Firstly, I will dedicate this thesis to our Almighty Allah, who gives his strength and knowledge for my everyday life.

To The eyes that did not tire of staying up late for my comfort, and the two candles that were burning their bodies to enlighten me in my path in the dark of circumstances, Those who have endured difficulties for my comfort, and who have endured grief for the sake of my joy, are the apple of my eyes ((My dear parents)).

To My brothers, Engineer Muhammad, Professor Dr. Khaled, Professor Ibrahim and Engineer Tariq who provided me with all academic, family, moral and material needs.

To my sisters who are my role models.
To my wife, who endured the difficulties for my comfort and the provision of the appropriate atmosphere.

To my sons Bilal and rafif and rowad and sons my brothers and my brother's wives.

Mustafa Amer Abeud

2021

Contents

List of Tables

List of Figures

List of Pseudo Code

Abbreviations

Ann	Artificial Neural Network
Art	Adaptive Resonance Theory
B-Cnn	Branch Convolutional Neural Network
Bn	Batch Normalization
Bt-Strategy	Branch Training Strategy
Cnns	Convolutional Neural Networks
Convnets	Convolutional Neural Networks
Dl	Deep Learning
Dnn	Deep Neural Network
Eden	Evolutionary Deep Networks
Gpu	Graphical Processing Unit
Hd-Cnn	Hierarchical Deep Convolutional Neural Networks
Hog	Histograms Of Oriented Gradients
Knn	K-Nearest Neighbors
Lbp	Local Binary Pattern
Lira	Limited Receptive Area
Maxent	Maximum Entropy Method
Mpp	Massive Parallel-Processing
R-Cnn	Regions With Cnn
Relu	Rectified Linear Unit
R-Fcn	Region-Based Fully Convolutional Networks
Rgb-D	Red Green Blue Depth
Roi	Region Of Interest
Rpn	Region Proposal Network
Rvnn	Recursive Neural Network
Sgd	Stochastic Gradient Descent
Sift	Scale Invariant Feature Transform
Som	Self-Organizing Maps
Ssd	Single Shot Multibox Detector
Surf	Speeded-Up Robust Features
Svm	Support Vector Machines
Vgg	Visual Geometry Group

CHAPTER ONE

GENERAL INTRODUCTION

CHAPTER 1: General Introduction

1.1 Introduction

Deep learning has recently been commonly used in a wide variety of applications. A Convolutional Neural Network (CNN) is a category of deep neural networks that have the most extensive effects on demonstrable problems. Style organizations have utilized CNN on their web-based business to take care of numerous issues, for example, garments acknowledgment, garments search and suggestion. A center advance for these usages is picture order. Garments characterization is a test task as garments have numerous properties, and the profundity of garments classification is exceptionally convoluted. This muddled profundity makes various classes to have fundamentally the same as highlights, thus the order issue turns out to be extremely hard. One of the most testing multi-classes arrangement issues is the design characterization in which marks that describe the garments type are distribute to the pictures[1]. The trouble of this multi-classes design grouping issue is because of the prosperity of the garment's properties and the high profundity of garments order too. This convoluted understanding makes various names classes to have comparable highlights. There are a few issues to consider in order of design. Firstly, pieces of clothing can be handily misshaped by extending design. Secondly, a few pieces of clothing may be considered as different as per the sentiment, and different articles of clothing may be considered as same. Thirdly, a number of article of clothing things are strong to be recuperated because of their little size. Fourthly, photographs can be considered in various cases, for example, the distinction in the point, request and light foundations. Fifthly, some article of clothing classes have comparative highlights and can be fluffy, for example, pants and leggings. Sixthly, an article of clothing picture is diverse depending on whether it is only a photograph of an article of clothing or a photograph of the model's wearing article of clothing. And therefore, a calculation that could be utilized to obtain multi-classes style grouping execution is of incredible need [2].

Large data is connected with colossal enlightening assortments and the size is over the versatility of standard data base programming gadgets to get, store, manage and survey. Enormous data assessment is the key for specialists, examiners and monetary experts to make better decisions that were at that point not accomplished. The term, Big Data has been initiated to insinuate the tremendous heave of data that can't be overseen by standard data dealing with techniques [3].

Deep learning has been utilized in big data. Big data is a huge volume of both organized and unstructured data that is enormous to the point that it is hard to handle utilizing conventional information base and programming strategies. Big data advances impacts affect logical revelations and worth creation [4].

1.2 Research Motivation

Constructing a neural network model to identify the fashion objects, the study achieved that objective. The model will take an image of size 28×28 pixels grayscale format as input and return the corresponding category. The key motivation of our work is the previous succeeded work in image classification based on deep learning. Various deep learning models were introduced to classify different fashion products from the given images.

1.3 Related Works

Several researchers addressed the cloths image classification problem by focusing on the image dataset augmentation and classification issues and performance aspects of CNN as it significant in different applications. Several types of standard datasets were used in the literature with different techniques. This section contains the previous works which are related to this work. Some of these researches are summarized as follow:

Morelli et al. (2013) [5] proposed a CNN network based framework to characterize the drifting concept. Additionally, there is an optimal correlation among normalization and skip links of the batch and these are recommended to be rejected in order to enhance the learning process. The result indicated with two distinctive sort of CNN model. In addition to that, the first CNN architecture included two important convolutional layers and the second one is called SoftMax layer. It obtained such a satisfactory result with an accuracy of 90.1% while CNN model was of five convolutional layer and one SoftMax layer which give a precision of 89.9%.

Jmour et al. (2014) [6] prepared the CNN on style of ImageNet dataset in picture characterization. CNN is utilized to learn includes and characterize RGB-D pictures task. Different boundary had impacted on precision of preparing results. The authors introduced the impact of small group size on preparing architecture. The architecture that was chosen by the researchers obtained 93.33% accuracy in total on the selected test dataset with a base clump size of 10.

Wei et.al (2015) [7] prepared the CNN on ImageNet dataset of advanced information base on a back-propagation calculation (LeNet-5) convolutional network. It was utilized for distinguishing outrageous evolving designs. The Info layer was worked with 28×28 neurons, in connection with proper size of pictures. The Shrouded layer had 100 neurons in total with using sigmoid initiation work alongside yield layer of 10 neurons, speaking to the classes of penmanship pictures. The Stochastic Gradient Descent (SGD) preparing calculation was utilized for irregular slope plunge preparing to limit the misfortunes. Results acquired were 89.00% accuracy as testing precision in 100 images. The authors utilized two convolution and two completely associated layers with 50 group size to obtain 91.10% testing accuracy the ImageNet dataset.

Yawen et al. (2018) [8] proposed a combination of five distinctive arrangement models, KNN, RFs, DTs, SVM, and GDBTs to develop a special multi-model gathering framework to order the MNIST dataset. To dodge overfitting condition differential quality articulation investigation was utilized to choose instructive pictures. These qualities were given as contribution to five characterization calculations and afterward deep learning model used to group the five diverse yields to get the last expectation output. The assessment measurements were based on the performance evaluation metrics techniques using recall, precision, and accuracy. There are three diverse data sets were analyzed and gathering depends on profound model which capable to produce a more noteworthy precision of 89.20%, 88.78%, and 88.41%.

Zeiler and Fergus (2015) [9] introduced a Large Convolutional Network models CNN. The Architecture of ImageNet model is practically comparative aside from that the authors had diminished first layer channel size to 7×7 rather than 11×11 and utilized step 2 convolutional layer in both first and second layers to hold more data in those layers' highlights. The authors attempted to clarify the explanation of the exceptional exhibition of huge profound CNN. They had utilized a novel perception strategy which was a de-convolutional network with numerous organizations called deconned. It was used to plan initiation at higher layers back to the space of info pixel to perceive which pixels of the information layer were responsible for a given actuation in the element map. They have utilized a lot of impediment tests to check whether the model is sensitive to neighborhood or worldwide data.

Karpathy et al. (2015) [10] tested enormous scope picture arrangement utilizing CNN with MNIST dataset. The authors found that CNN gain the incredible highlights even from feebly named information. The exhibition of the proposed model thought about utilizing picture order of MNIST dataset. Huge execution improvement up to 63.3% was introduced. The finds demonstrated that CNN gave enhance orientation and age grouping outcome even if there should be an occurrence of more modest size of contemporary unconstrained picture sets.

Krizhevsky (2014) [11] introduced a segment of the novel strange features of their association's model. The author gives arranged a tremendous significant convolutional neural association to describe the 1.2 million significant standard pictures in the ImageNet LSVRC - 2010 into the 1000 classes. To make planning faster, a non-drenching neuron and a profitable GPU were used with the convolution movement. A variety of this model was also entered in the ILSVRC-2012 contention and achieved a victorious top-5 test bungle movement of 15.3%. The accuracy is about 26.2% higher than other existing framework.

Younis et al. (2017) [12] recommended a significant neural association by using the upgrade of transfer learning limits. In this aspect, they utilized a tremendous MNIST dataset. Moreover, transfer learning is a methodology used to change the overall pre-arranged architectures which utilize a comparative sort of data sets for desire. In order to find the proper solution, this specific work allied the going with arranged significant learning networks explicitly GoogLe Net, Alex Net, and VGC Net to get the features from the insightful burdens. They surveyed the recently referenced organizations on the plant data sets with various association limits and data development. The researchers eventually join the optimal classifier estimation to enhance the simulation of the system entirely. The results show that the change in GoogLeNet and VGC Net was reestablish the best precision of 78.44% diverged from the mix of Alex Net with VGC Net.

Szegedy et al. (2015) [13] proposed engineering of GoogLeNet which is unique in relation to regular CNN. They have expanded the quantity of units in each layer utilizing equal channels called commencement module of size 1 × 1, 3 × 3 and 5 × 5 in every convolution layer (conv layer). They have additionally expanded the layers to 22. While planning this model, they have considered the computational financial plan fixed. To make the design computationally productive they have utilized commencement module with dimensionality decrease rather than the innocent rendition of beginning module. The quantity of boundaries utilized in GoogLeNet was multiple times lesser than AlexNet yet its exactness was fundamentally better.

Geier Christopher (2019) [14] addressed the image classification issue on the MNIST and Fashion-MNIST datasets. Five CNN architectures were used. The results obtained indicate that 89 percent precision for the MNIST dataset is provided by any CNN model. The third architecture (3 convolutional layers and 2 fully connected layers) provides the Fashion-MNIST dataset with better testing accuracy. In architecture 1, ADAGRAD is the best optimizer algorithm observed, sigmoid is the best activation function, batch size is 64, iteration number is 50 and dropout is 0.1. In architecture 2, ADAM is the best optimizer observed for the algorithm, SoftMax is the best activation function, batch size is 128, iteration number is 50, dropout is 0.25 and kernel size is 2×2. Architecture 1 has only one input layer and two completely linked layers, so training takes less time. With the rise in convolutional layers, the training time increases. Training time is substantially increased with a filter size of 3×3. The images data has been classified by means Random Classifier attains the maximum Accuracy of about 84.4%.

Bhatnagar et al. (2017) [15] presented a feasible classification error, the design had a few parameter numbers and low computational costs. In terms of some dense blocks with skip links, it included densely linked layers to maintain the knowledge flow in a deep CNN model. The Fashion-MNIST is classified by means of higher Deep Learning Architecture by CNN2 with Batch Normalization. The model had

26 times fewer parameter numbers with an 8% better classification error than the first generalised CNN model. It is worth noting that this smallest model had the same accuracy level compared to the 92.54 percent much higher than the smaller model size of the AlexNet. The results showed that there are unacceptable justifications for some levels of complexity in CNN models, especially in early CNN models.

Agarap (2017) [16] presents an important method for an efficient and precise classification and identification of photographs of fashion items. Upon successful implementation of the classification system for fashion articles using CNN feature space and multi-class SVM classifier. It has shown that this system provides relatively good fashion object classification efficiency compared with the available literature works. It attains the classified accuracy of about 90.72%.

Callet et al. (2006) [17] the researchers framed a CNN for item classification of style MNIST dataset. They done the examination with three diverse sorts of pooling strategies to be specific max pooling, mean pooling, and stochastic pooling. They at last infer that the stochastic pooling technique delivers more precision. The outcome acquired by them was 92.48% accuracy.

Xiao et al. (2017) [18] proposed an ensemble Artificial Intelligent (AI) algorithm to group the pictures and it concentrates to make a decent benchmark dataset. It has all the openness of Fashion-MNIST, to be specific its little size, clear encoding and tolerant permit. The different ensemble AI algorithm have been dissected with the Fashion-MNIST dataset. For example, Decision Tree Classifier with precision of 78%, Extra Tree Classifier with exactness of 77%, Gaussian NB with exactness of 51%, Gradient Boosting Classifier with exactness of 88%, K Neighbors Classifier with exactness of 85%, Linear SVC with exactness of 84%, Logistic Regression with precision of 84.2%, and MLP Classifier with exactness of 87%. Fashion-MNIST also represents a more testing arrangement task than the straightforward MNIST digits information, though the last has been prepared to exactness's above 89.7%.

Kussul et al. (2010) [19] proposed a novel neural network classifier of Limited Receptive Area (LIRA) for picture acknowledgment. It contains three neuron layers as sensor, acquainted and yield layers. The proposed classifier tried on MNIST information bases. The classifier demonstrated 88.41% precision. The issue of transcribed digit acknowledgment in optical character acknowledgment was indicated here. The uninhibitedly accessible design MNIST information base of transcribed digits was utilized. While probing design MNIST, falsely mutilated renditions of the first preparing tests were presented in preparing set that elaborate arbitrary mixes of jittering, shifts, scaling, deskewing, deslanting, obscuring and pressure.

Sun et al. (2019)[20] discovered a methodology of chart based semi-administered learning plan with the deep convolutional neural organization for order of design MNIST dataset. This proposed technique utilizes limited quantity of marked information with enormous measure of unlabeled information for the deep learning calculation. It contains three unique stages; information gauging, feature selection and co-preparing information naming. The unlabeled information gave more extra data to characterize and they acquired the most elevated precision of 82.43% utilizing just 100 named information with all leftover unlabeled information.

Manessi et al. (2018) [21] proposed a new methodologies based on LeNet-5, ResNet-56, and AlexNet . The proposed approaches implemented a deep learning architecture LeNet-5, ResNet-56, and AlexNet by utilizing three typical datasets; CIFAR-10, Fashion-MNIST, and ILSVRC-2012. Results show generous upgrades in the general execution, for example, an expansion in the main precision for AlexNet on ILSVRC-2012 of 3.01 rate focuses.

Han et al. (2018) [22] broke down the forecast exactness of three distinctive convolutional neural organizations on most famous prepared and tested datasets in particular CIFAR10 and CIFAR100. They zeroed in the investigation on 10 classes of each dataset as it were. The fundamental intention was to discover the precision of the various organizations on same datasets and assessing the consistency of expectation by every one of these CNNs. They had introduced an exhaustive forecast investigation for looking at the organizations' presentation for various classes of items. Not many items like "seat", "train" and "closet" were entirely perceived by 147 layered organizations though protests like "vehicles" were totally perceived by 177 layered organizations. They was summarized that neural organizations were the best arising procedures for making a machine shrewd for tackling some genuine item order issues.

Deep CNNs have achieved good performance on data based on time series nature or images. Deep CNNs are considered a black box which is difficult to guess and interpret the outcomes of any experimental research. Therefore, it is difficult to verify the training of CNN on noisy image data to increase the misclassification error. Deep CNNs are based on supervised learning mechanism. However, the availability of a large and annotated data is required to obtain such a good learning method. Hyper-parameter selection is considered such and effective and robust to increase performance of CNN. The main drawback of CNN is that, unable to show good performance, when applied to estimate the pose, orientation, and location of an object. Lower layers should handover their knowledge only to the relevant neurons of the next layer.

1.4 Problem Statement

Image classification is a fundamental challenge in computer vision, including image and video indexing. In order to identify a visual entity from an image, humankind is facing a big problem to perform with images accurately Moreover, it is considered a slightly easy for computer algorithm to with achieving an acceptable accuracy and performance. Deep neural networks have recently been used to solve a wide range of problems with excellent results. Convolutional neural networks, in particular, have shown excellent results in image segmentation, mage classification, natural language processing problems, image recognition and machine vision problems. Some Bayesian Belief Networks and Hidden Markov Models based probabilistic models have also been extended to image classification problems with features based on grey level, colour, motion, depth, and texture. In this Thesis, we investigate the concept of classifying Fashion MNIST images and how augmentation Fashion MNIST dataset using Proposed convolutional neural network variants developed by us.

1.5 Aim of The Thesis

The aim of this Thesis is to train CNN network using Sequential model based on fashion Data set image. Furthermore, it is also to create an effective model of deep learning that recognize accessories for clothing and augment Fashion-MNIST dataset that we use. The used Fashion-MNIST images were augmented based on three different augmentation techniques. The augmented Fashion-MNIST images were used in fashion classification by applying the pre-Trained. The accuracy of the pre-Trained model of augmented data for the classification of MNIST-Fashion data has shown that it is best suited to the selected dataset. To evaluate the performance of the given dataset, the training and testing image data were applied. Image classification is considered a major phase for both of these implementations. Clothing classification however is not such any easy task as clothing has a number of properties, categorization depth of clothing is extremely complex. Furthermore, this complex depth makes various groups and similar characteristics, and so the issue related to the classification becomes difficult.

1.6 Contribution of the Thesis

This empirical study proposes a new methodology to train Deep Neural Network using Sequential model based on fashion Data set image. Our methods provides a robust data pipeline for pre-processing fashion Data set; and data modelling using Deep learning. This study is proposed model which consists of three convolutional layers that extract 32, 64, 128 filters of size 3×3 with ReLU as activation function to each layer and SoftMax in last layer. In addition to that, the output of the activation function is fed to a max pooling of 2×2 window. While Dropout to address the main problem related overfitting. Eventually, we used the 'Adam' optimizer for optimization of the loss function. This Thesis also contributed to the up-to-date understanding of how to successfully handle the early stages of the classification process for fundamental advances in computer vision. The holistic review added to the Conventional research the classifying of a group of 10 classes by incorporating an augmented data with the pre-convoluted. The significant features that should be considered in the early stages of the phase of innovation, and the Pre-Trained with the five-network model. To holistically analyze the priorities of these characteristics of augmenting the images that not been analyzed before. The literature gap confirmed outcomes of existing framework that also highlighted the position of certain characteristics like availability of classification (Dibner et al. 2003 [23], e.g. O'Shea et al. 2005 [24], Decter et al. 2007 [25], Powers and McDougall 2005 [26], Link and Siegel 2005 [27], Hall and Bagchi-Sen 2007 [28]). Therefore, the current research also indicated performance measures like classification accuracy has increased with emphasized in existing innovation management and technology transfer literature.

1.7 Thesis Outlines

The thesis contains four chapters in addition to this chapter, and it is organized as follows:

- ***Chapter Two***: Explains all the information about thesis subjects. It starts with the definition and the description of the deep learning approaches and focuses on the deep methods used in this thesis.
- ***Chapter Three***: Describes all the materials used in the proposed system and work steps from the dataset of Fashion-MNIST and classification using proposed neural networks.
- ***Chapter Four***: Presents the results of the practical implementation and the statistical resulted data and shows it in tables and figures then comparing these results with other research work in the same field to evaluate this work.
- ***Chapter Five***: Displays the main conclusions of this research and giving some of the suggestions for future works.

CHAPTER TWO

THEORETICAL BACKGROUND

CHAPTER 2: Theoretical Background

2.1 Introduction

In all different aspects of the economy and being, the systematic study further goes on to elucidate the applications of Big Data. After incorporating it with digital capabilities to do authenticate business creation and its visualization to make it understandable to theoretically qualified business analyzers, the use of Big Data Analytics has been addressed in depth. In addition, the integration of Big Data to enhance population health, to improve telecommunications, banking, the food industry, and to speculate the fraud and evaluate sentiment has been outlined. The basic weaknesses hindering the growth of Big Data Analytics are described in detail in literature. This subject has been divided into two fields; the practical weaknesses and the theoretical weaknesses. The barriers to data protection and democratizing have been established, among many others, such as the inability to find sound data practitioners in the necessary quantities and software capable of processing data at a high speed. Data is produced from numerous sources in the digital world and the rapid transition from digital technology has led to the growth of big data. In several areas, the collection of massive datasets offers evolutionary breakthroughs. In general, it refers to the collection of large and complex datasets that are difficult to manage using standard methods for managing databases or applications for data processing.

2.2 Basic Features of Big Data

Big Data states that the gigantic, larger (volume) datasets, more complicated than before, including structured, semi-structured, and unstructured (variety) data, and closer (velocity) arrivals[3]. This is called the 3V as shown in Figure (2.1).

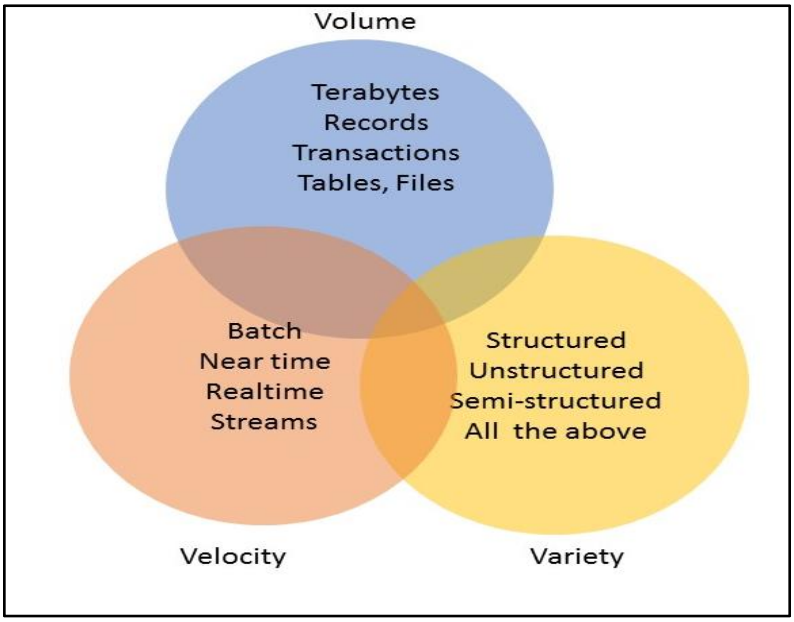


Figure (2.1) 3V Concept.

2.2.1 Volume

This means the amount of the collected data is handled and controlled within the system. The volume growth is explained by the rise in the amount of data generated and processed[29].

2.2.2 Variety

It reflects the increase by an information system of the handled data types. The complexity of data types refers to the amount of multiplication of data and this represents the amount of data between the connections. The potential uses of raw data refer to the spectrum limit of data[29].

2.2.3 Velocity

It reflects the frequency of generation, capture and sharing of the data. The data is streamed and needs to be examined in actual time.

To this classical characterization, two other "V"s is important to be added to the first three Vs. The five Vs are shown in Figure (2.2)[29].

2.2.4 Veracity

It reflects the level of quality, accuracy and uncertainty of data and data sources.

2.2.5 Value

The value and potential derived from data.

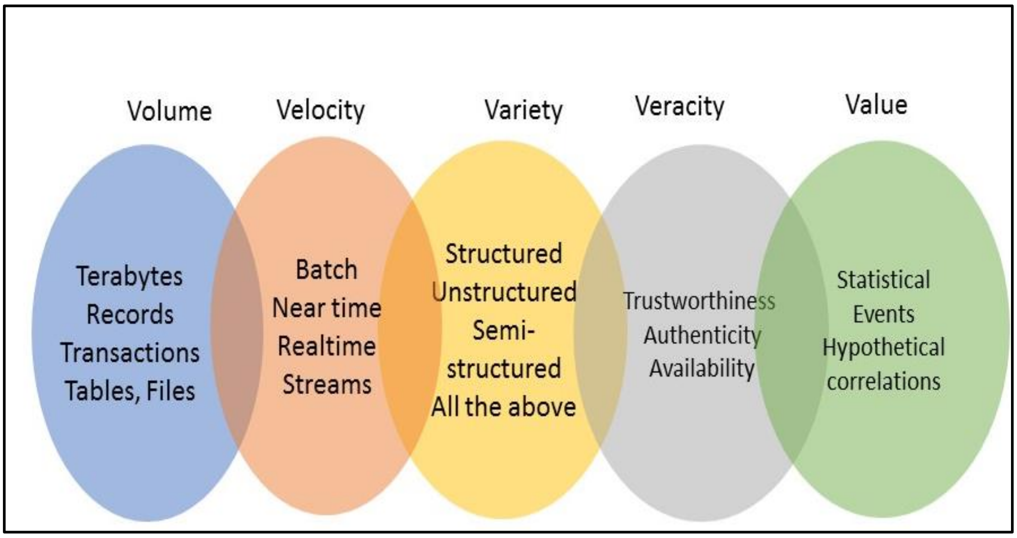


Figure (2. 2) 5V Concept.

2.3 Classification in Big Data

The method of grouping data into categories for its most effective and efficient use is data classification. A well-planned method of classification of data makes it easy to locate and retrieve important data. There are three key aspects of data classification, namely strategies, domains and combinations. The common techniques used for classification are decision trees, instance-based methods, neural networks, probabilistic methods and support vector machine[30]. The various resources from multimedia, time-series, text, network, discrete sequence and various vague data are applied as data in the big data platform. Due to the recent importance of the big data model, it also covers broad data sets and data sources. Classification method variations address ensembles, rare-class learning, distance feature learning, active learning, visual learning, transition learning, and semi-supervised learning, as well as classifier assessment aspects[31].

There are three categories of classification of big data types, namely social networks, conventional business systems and the Internet of Things.

To understand their features, Big Data is categorized into different categories. There are five dimensions to the classification: content format, data staging, data sources, data stores, and data processing as shown in Figure 2.3. Each classification involves new algorithms and techniques in the big data domain to perform classification tasks efficiently.

Big Data typically refers to data that exceeds traditional databases and data analysis techniques' usual storage, processing, and computational power. Therefore, more sophisticated tools and techniques needs to be applied to estimate the large-scale data in Big-data platform [29]. Due to the variety and velocity of manipulated data, structured data analysis is developed. Therefore, the large variety of data suggests that the structures in place must be able to assist in the analysis of data and it is no longer adequate to interpret data and evaluate the final outcomes. The analysis consists of automatically identifying the associations between the data within a spectrum of constantly evolving data in order to assist in its exploitation [31].

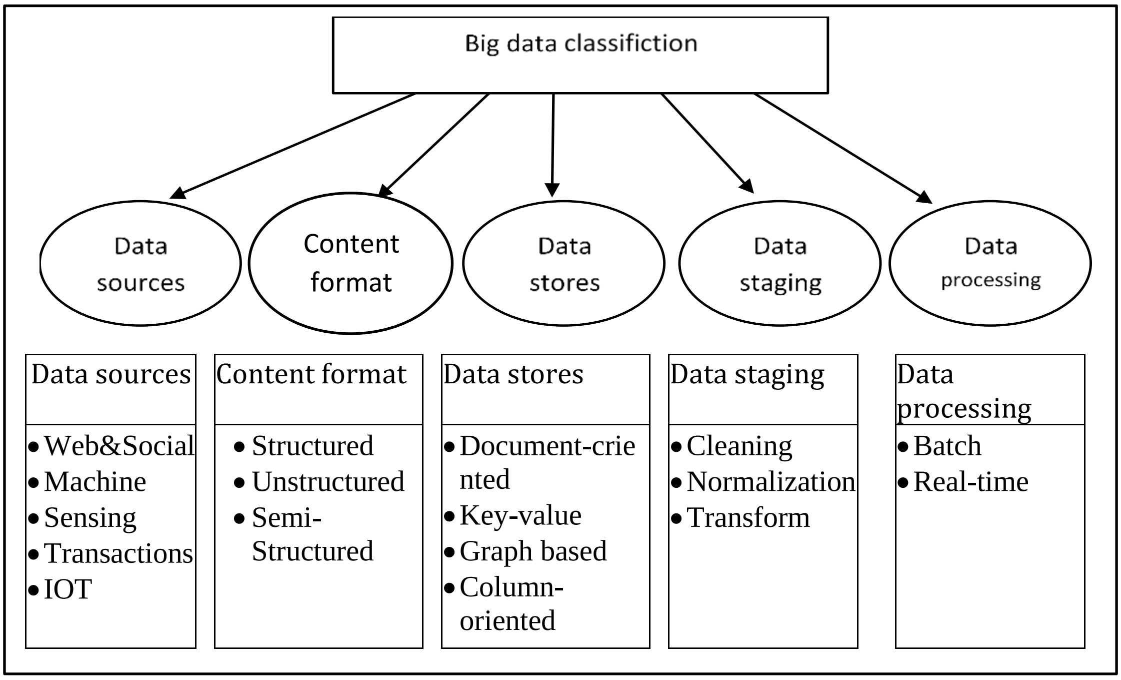


Figure (2.3) Big data classification.

By facilitating optimization process, empowering the exploration of insight and enhancing decision-making, the Big Data revolution promises to change how we live, function and think. The realization of this great potential depends on the ability to derive value through data analytics from such vast data; deep learning is at its heart because of its ability to learn from data and provide insights, decisions, and forecasts powered by data.

In a different era, however, conventional Deep Learning methods have been developed and are thus based on certain assumptions, such as the dataset fitting entirely into memory, which is sadly no longer true in this modern setting. Along with the Big Data functionality, these broken expectations build barriers to conventional techniques.

2.4 Deep Learning

The McKinsey Global Institute has suggested that one of the main drivers of the Big Data revolution would be DL [32]. The explanation for this is its ability to learn from knowledge and provide observations, decisions, and forecasts guided by data [33]. It is based on statistics and can derive patterns from data in the same way as statistical analysis, but it does not require the clear use of statistical evidence.

Through using a hierarchical multi-level learning approach, deep learning models can easily extract meaningful representations from the collected raw data. In which, more abstract and complex representations are learned at a higher level depends on the representations and concepts at the lower level of the learning hierarchy. Although this kind of algorithm can be implemented to learn easily from the collected labeled data set that adequately large quantities, it is mainly effective for learning from large quantities of unsupervised data sets [34]. This type of model can extracting important patterns and representations from Big Data. In this scenario, once the abstractions of hierarchical data are learned with Deep Learning from unsupervised data, more traditional discriminatory models can be trained with the help of comparatively less supervised/labeled data points. The labeled data is normally collected by human/expert feedback. In contrast to relatively shallow learning architectures, Deep Learning algorithms have shown to work better at extracting non-local and global relationships and patterns in the data [35]. Other useful features of Deep Learning's learned abstract representations include; firstly, relatively these models able to work robustly and effectively with the information gained from more abstract data representations and more complex. Secondly, raised automation of the extraction of data representation from unlabeled data sets makes its comprehensive specific application to various types of data sets, such as texture and image. Thirdly, it is possible to acquire relational and semantic information at the higher levels of raw data abstraction and representation.

In addition, Deep Learning handles data representations and abstraction, it is most possible suitable for the study of the collected raw data presented in various formats sources, i.e. Big Data variety. It can reduce the need for human expert feedback to abstract variables from each new form of data observed. In this case, Big Data Analytics offers such a significant opportunity to create new models to solve particular problems relevant to Big Data, while posing numerous obstacles for more traditional approaches to data analysis. Deep Learning principles provide data analytics experts and practitioners with one such solution spot. For example, Deep Learning's extracted representations can be used as a realistic source of information in Big Data Analytics for data retrieval, semantic indexing, decision-making, and for other purposes [36].

2.5 Deep Learning Techniques

Several common networks of deep learning are discussed, including the Recursive Neural Network (RvNN), RNN, CNN, and deep generative models. However, as deep learning has evolved very rapidly, every few months many new networks and new architectures are emerge. Some of these networks are illustrated in the next subsections.

2.5.1 Convolutional Neural Network (CNN):

Recently, the computer vision plays a vital role in platform of the classification of images. The most popularized deep learning model used to organize the classification is the Convolutional Neural Network (CNN). It is most popularly used in Image Classification and Object detection, Whereas the non-linear function and activation function is integrated with multiple layers such as Convolutional layer (Conv), pooling layer, Drop Out Layer and finally with a Fully-connected (FC) layer. With each and every element, the model with a non-linear function and activation function above linear and non-linear function impinges on each element of the input. The pooling function reduces the output size the multiple perceptron examines the image inputs. It is trained to segregate pixel values with learnable weights and bias meaning to multiple sections of images. One of the key benefits of using CNN is that the input images use a local spatial domain. It shares a few sharable parameters and less weights. Due to less computational complexity and considered takes less memory consumption, this method is usually more efficient than other models [37]. The architecture of the CNN is shown in Figure (2.4).

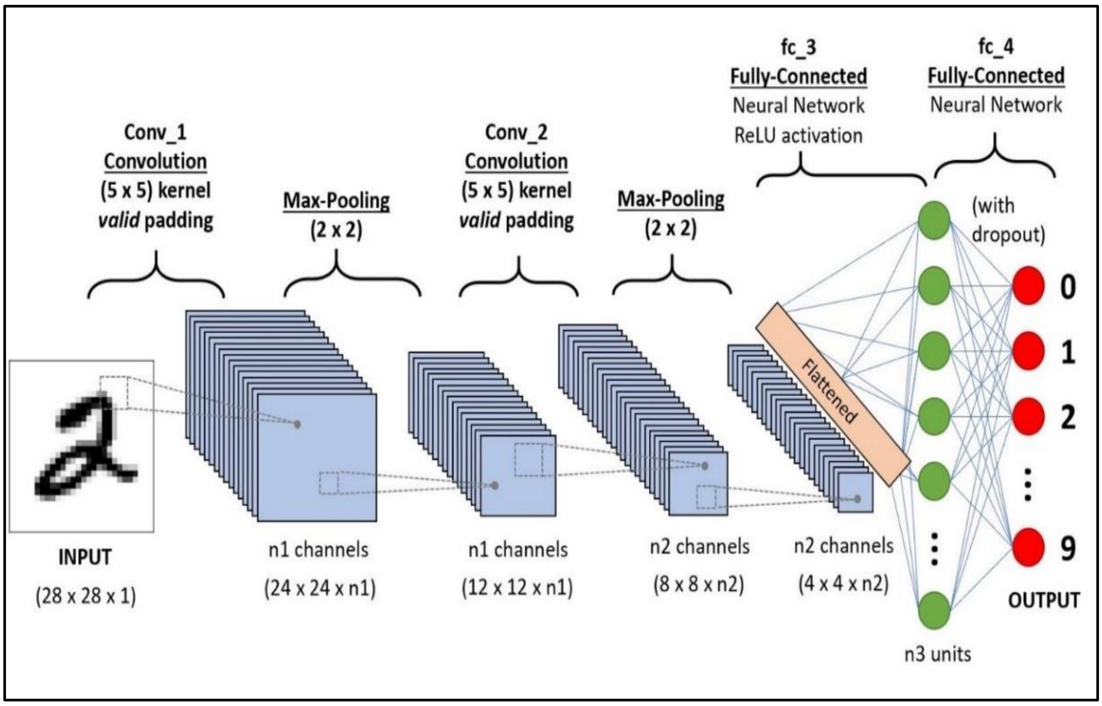


Figure (2.4) Architecture of Convolutional Neural Network[38].

However, neuro-biologically is encouraged the CNN. CNN is self-possessed of three unique layers. Initially, the primary levels furnish with a specific number (ten to thousands) of filters or kernels of typically very negligible dimensions, commonly 3×3, 4×4 or 5×5 transparency. It is used to create the feature map over the input image. The filter values element wise dot product and the image section slides over have been introduced as the slide by the image kernel. This kind of memory-efficient operation, as the image is operated on by the same kernel. A graphical processing unit (GPU) can calculate results extremely quickly.

The Convolutional layer is the principal layer of CNN. A mathematical procedure behind the Convolution is to convolute two groups of data. In the architecture, the Convolution is furnished to primary raw data to determine the feature map by applying convolution filter. The main benefit of Convolutional layers is to furnish the feature maps, the major concern to encode all the images in terms of 1s and 0s. Whereas the feature of the image is constructed from the feature detector and it is a matrix and assign values (i.e. pointy ears, slit eyes ...). The matrix overlays the image segment and performs bit-wise multiplication at that position for all values. The outcome of the bit-wise multiplications are summarized and positioned at the appropriate position on the map of the function. Then, it travels to additional part of the image and concludes the process until the whole image has passed through it. As explained in Equation (2.1) and (2.2).

$$\mathbf{S}(\mathbf{i},\mathbf{j}) = (\mathbf{I} * \mathbf{K})(\mathbf{i} * \mathbf{j}) = \sum_{m}\sum_{n} \boldsymbol{I}(\boldsymbol{m},\boldsymbol{n})\boldsymbol{K}(\boldsymbol{i}-\boldsymbol{m},\boldsymbol{j}-\boldsymbol{n}) \qquad \textbf{(2.1)}$$

Where K is Kernel (convolution function), l is the input (may be 2D Array) [15]

For 3×3 kernel, the equation becomes.

$$\mathbf{S}(\mathbf{i},\mathbf{j}) = (\mathbf{I} * \mathbf{K})(\mathbf{i} * \mathbf{j}) = \sum_{m=1}^{3}\sum_{n=1}^{3} \boldsymbol{I}(\boldsymbol{i}-\boldsymbol{m},\boldsymbol{j}-\boldsymbol{n})\boldsymbol{K}(\boldsymbol{m},\boldsymbol{n}) \qquad \textbf{(2.2)}$$

Therefore, to convolute the raw image from the input, it is held on the left side of the architecture to be transformed. The CNN filter which is also called the kernel, and these relationships are interchangeably used. The 3×3 convolution is done by sliding the filter shape with the input and then executing the convolution technique. It is the element-wise multiplication of the matrix and the summation of the results. This function is used to extract the feature and the feature extraction gap. The convolution takes place at the left area region and the receptive field is determined.

.

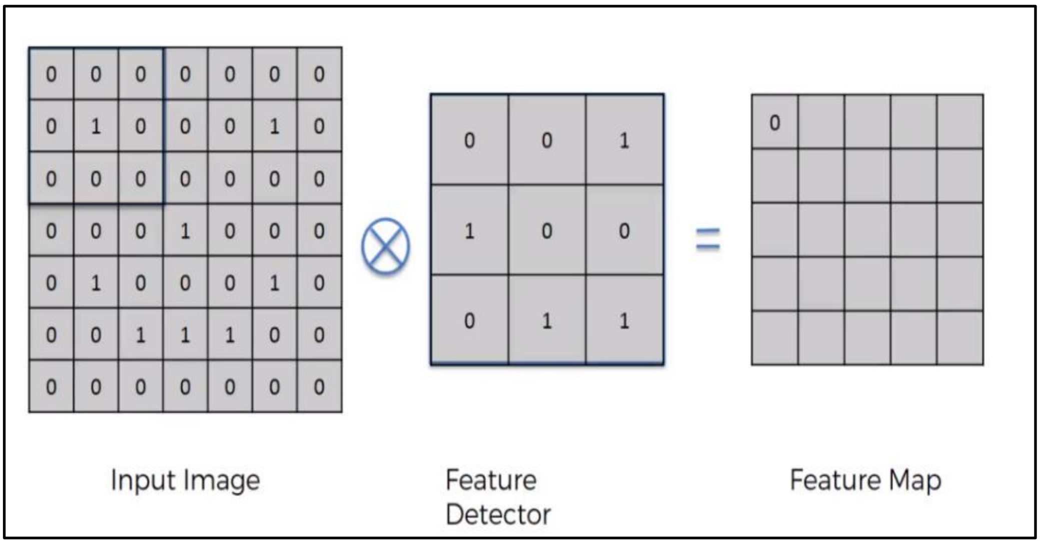


Figure (2.5) Extraction of feature map from the input image.

An example of a 2D convolution operation using a 3×3 filter is shown in Figure (2.5). It extracts the features from the input image by convoluting the input image with the 3×3 filter. Even those convolutions are accomplished in 3D in nature. In order to achieve high accuracy and performance, an image is produced as 3D matrix with the three important dimensions of depth, height, and width where the color channels (RGB) correspond to depth.

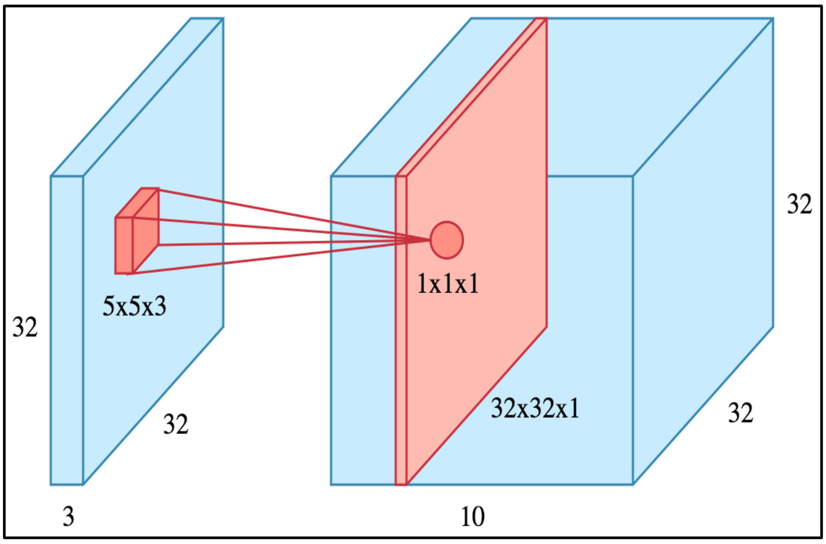


Figure (2.6) Visualization of convolution with 5×5 filter[39].

A raw image with an image size of 32×32×3 and a filter size of 5×5×3 is shown in the Figure (2.6). The depth of the convolution filter corresponds to the depth of the image and it is 3. When the filter is in a specific location, it conceals a small volume of the input and performs the convolution operation described in Figure (2.6). The only difference is that the number of the matrix multiplies in 3D instead of in 2D and the whole procedure results in scalar, sliding the filter over and over the input matrix. As in Figure (2.6), at each location the slide of the filter over the input perform the convolution, aggregating the result into a feature map. The size of this function map is 32×32×1 and it is clearly visualized in the Figure (2.6). It was visualized and it is still 32. The width and height of the feature map are unchanged. As in the function map, due to the padding and scaling obtained at each position, the sliding process is calculated at four positions and it is actually done over the entire input [40].

Figure (2.7) shows how two function maps are weighted along the depth axis. The convolution method is achieved separately for each filter and the result feature maps are disjointed.

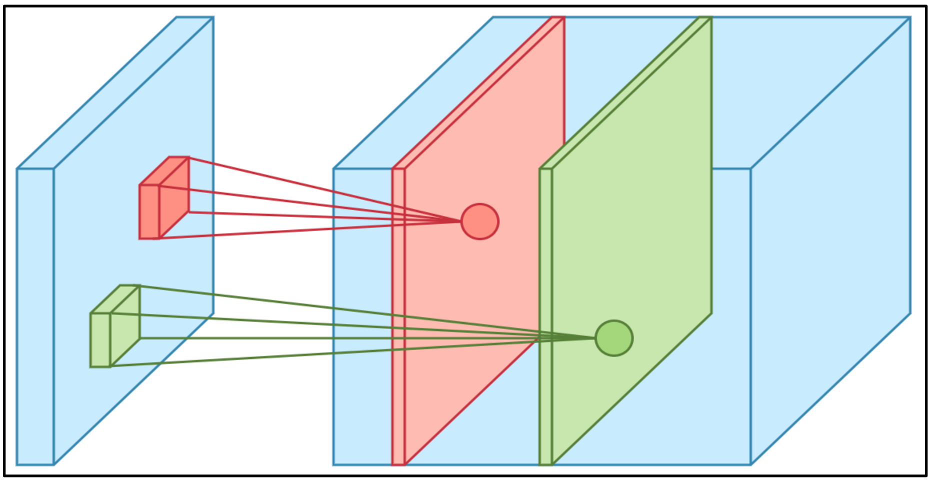


Figure (2.7) Function maps weighted along the depth axis[39].

2.5.1.1 Non-linearity

The fact that the neural network needs to include non-linearity is more powerful. By multiplying the weighted sum of its inputs, the ANN and the auto encoder are moved to the activation function and CNN is so different at that time. It transfers the product of the convolution operation again using the ReLU activation function. Therefore the values are not just the numbers in the resulting feature extraction, but the ReLU function applied to them. It has omitted this in the above statistics for convenience. It determined that some sort of convolution is necessary for a ReLU operation, and without it the network cannot reach its true potential [41]. The operation is illustrated [15]

$$\boldsymbol{A(i,j)} = \begin{cases} \mathbf{0}, & \boldsymbol{if\ S(i,j) < 0} \\ \boldsymbol{S(i,j)} & \boldsymbol{if S\ (i,j) \geq 0} \end{cases} \qquad \mathbf{(2.3)}$$

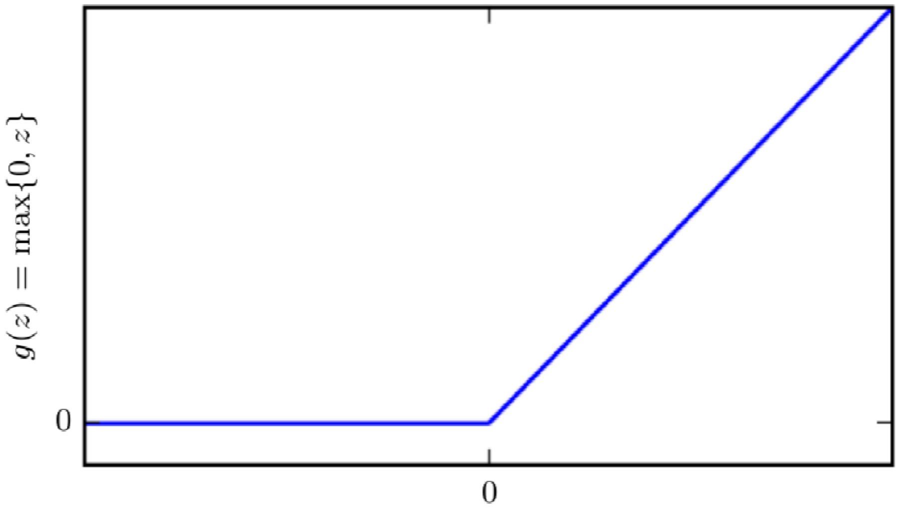


Figure (2.8) The Rectified linear unit outputs zero across half its domain.

2.5.1.2 Stride and Padding

Each step of the convolution filter is moved by the figure clearly specifying the filter and the value with stride1 by default and it is clearly shown in the Figure (2.9).

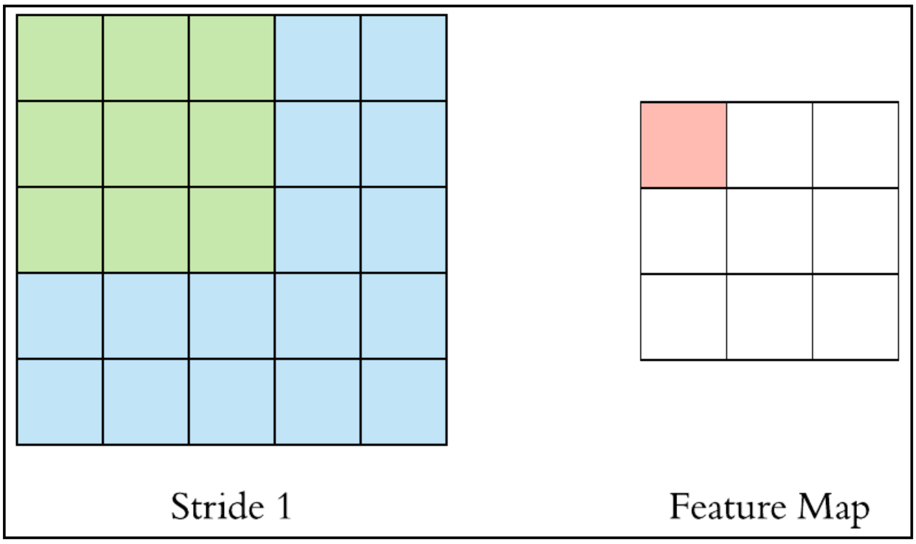


Figure (2.9) Stride1 with Feature Map[39].

This achieves greater progress if less variation between the receptive fields is to be decided. This also reduces the resulting diagram of the function, since it skips over potential locations. Figure (2.10) shows the two stages and it can be noticed that the map of functions has been reduced by applying the Stride 2 [42].

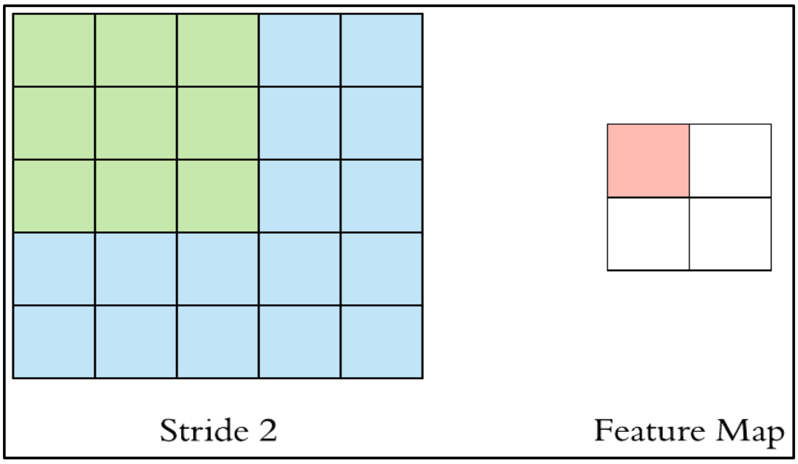


Figure (2.10) Stride 2 with Feature Map[39].

It is highly predictable that the dimensions of the extraction map of the function are smaller than the raw input. This has been calculated by the convolution filter. The padding is surrounded by zeros and it shows the representation of Stride1 with padding in Figure (2.11) to achieve the same dimensionality.

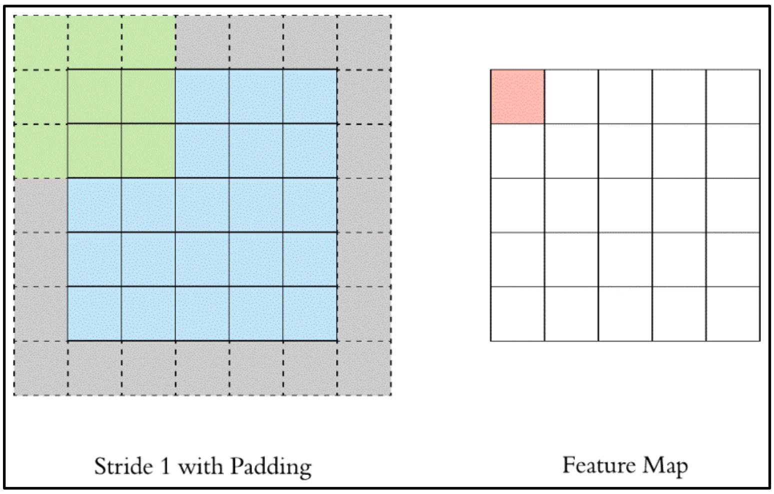


Figure (2.11) Stride 1 with Padding.

The output dimensions corresponding to a convolution operation with a pre-defined zero padding (p), filter size (f), dilation factor (d), stride (s), an input with width (w), and height (h) is given as[43]:

$$h' = \frac{h - f - (d-1)(f-1) + s + 2p}{s} \quad (2.4)$$

$$w' = \frac{w - f - (d-1)(f-1) + s + 2p}{s} \quad (2.5)$$

Padding is however a grey area for the results. It either pad with the zeros, or it pad with the edge values. The dimensionality of the function map now matches that of the data. In CNN, padding is generally used to maintain the size of the feature maps. The resulting convolution would decrease the rows on each non-furnishable sheet. 3D convolution figures then specify the padding, which is why the height and width of the function map were the same as the input (both 32×32), and only the depth was changed [31].

2.5.1.3 Pooling Layer

Usually, after the convolution method the pooling is applied to decrease the dimensionality. It is very significant to be applied in the proposed model. This enables to reduce the number of parameters, which helps to reduce the time of training and prevent overfitting. Every feature map is individually sampled by pooling layers, keeping depth unchanged, reducing height and width, and keeping depth intact.

Max pooling is the most common method of pooling in the pooling window that takes only the max value it is clearly represented in the Figure (2.13). Unlike the method of convolution, pooling has no parameters. It slides a window over its data and simply brings the max value into the window. The window size and the process are similar to the convolution. Using 2×2 windows and stride 2 Max pooling results are illustrated in Figure (2.12). The window size is calculated in separate colours from the window size and the stride is 2. The window size does not overlap.

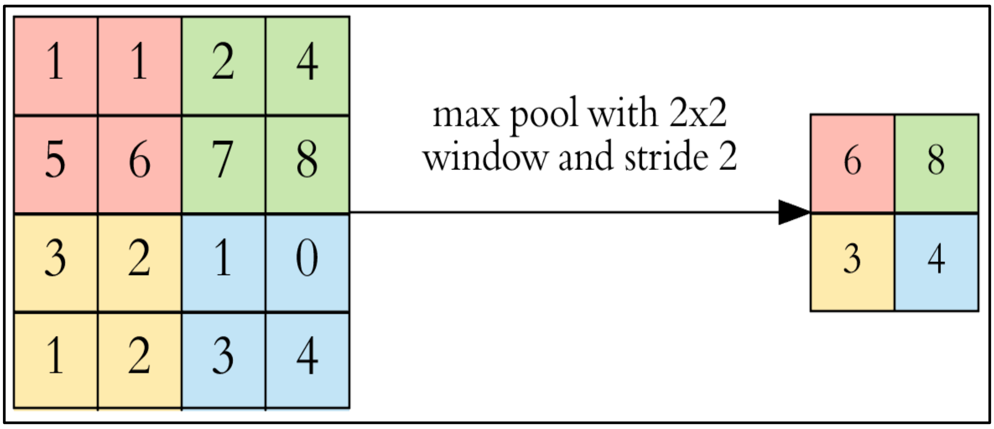


Figure (2.12) Max pool with 2×2 Window and Stride 2[39].

The layout of this window and phase are half the size of the function map and this is the main use case of pooling, down sampling and the extraction of the feature while retentive important information. If the dimensionality of the input to the pooling layer is 32×32×10 as shown in the Figure (2.13), the output is a map extracted from the 16×16×10 function using the same pooling parameters as mentioned in Figure (2.13). The partition does not alter the depth of the pooling self-sufficiently on each depth slice of the input; both the width and height of the function extraction are split.

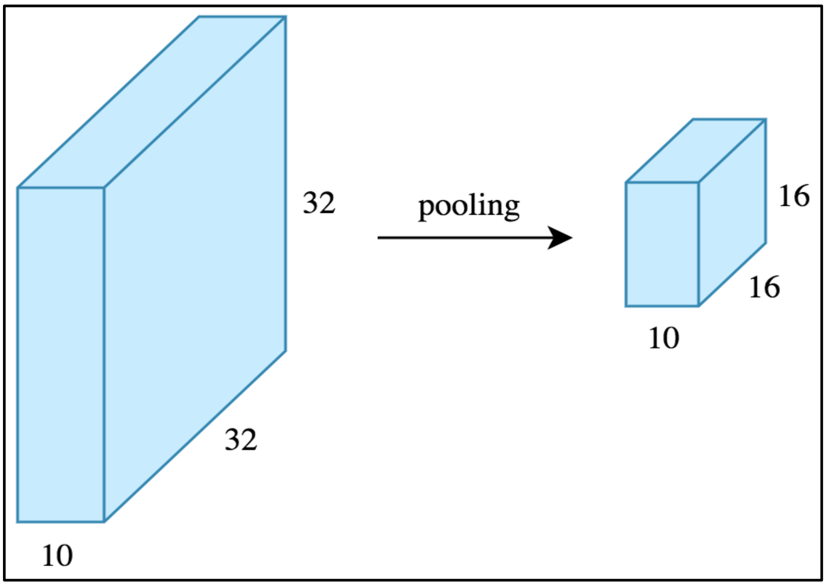


Figure (2.13) Pooling Layer 32×32×10.

Halving the height and the width decreases the number of weights to quarter of the input. Even that millions of weights are typically managed in CNN architectures, the reduction is a pretty big deal. In CNN architectures with 2×2 slots, stride 2 and no padding, the pooling is usually determined. Converting with 3×3 windows is complete. While converting, step 1 and padding are done with 3×3 windows. It has four major hyper parameters and it has been explained below [44].

Filter size: Apply 3×3 in general, 5×5 or 7×7 filters are depending on the application. There are also 1×1 filters, which may seem odd at first glance, but they have motivational applications. These filters are commonly used in 3D and it is also have deep dimensions. The depth of the filter at a given layer is equal to the depth of its input image dimension.

- **Filter count:** This is the highest customizable parameter; it is a power of 2 anywhere in the image and it is between 32 and 1024. In an additional context, by adding supplementary filter results, this leads to an increased number of risk overfitting parameters. Usually, it starts with a small number of filters on the initial layers and progressively increase the count as it move deeper into the network.

- **Stride:** It is usually assign the mandatory value to 1.

- **Padding:** This typically defines the padding.

2.5.1.4 Fully Connected

After the sequence of convolution and pooling layers, the CNN architecture is covered by two or three fully connected layers. Convolution and pooling layers are carried out at 3D volumes to conduct the 1D vector of the fully connected layer. The flattened output of the pooling layer is the input to the fully connected layer. The fully connected layer flattens the 3D vector to the 1D vector.

2.5.1.5 Dropout

It has been generally abbreviated as "fall". It is normally used in the feedback to eliminate the overfitting. It is technique used to facilitate the generalization of methods in deep learning. It sets the weights connected to a certain percentage of network nodes to 0. In the two dropout layers, CNN sets the percentage to 0.5 as in Figure (2.14)[45]. In which, the densely connected neural net explained in the left of the figure. An example of a smaller network produced by applying dropout is explained in the right of the figure. Crossed units have been deactivated by dropout [46].

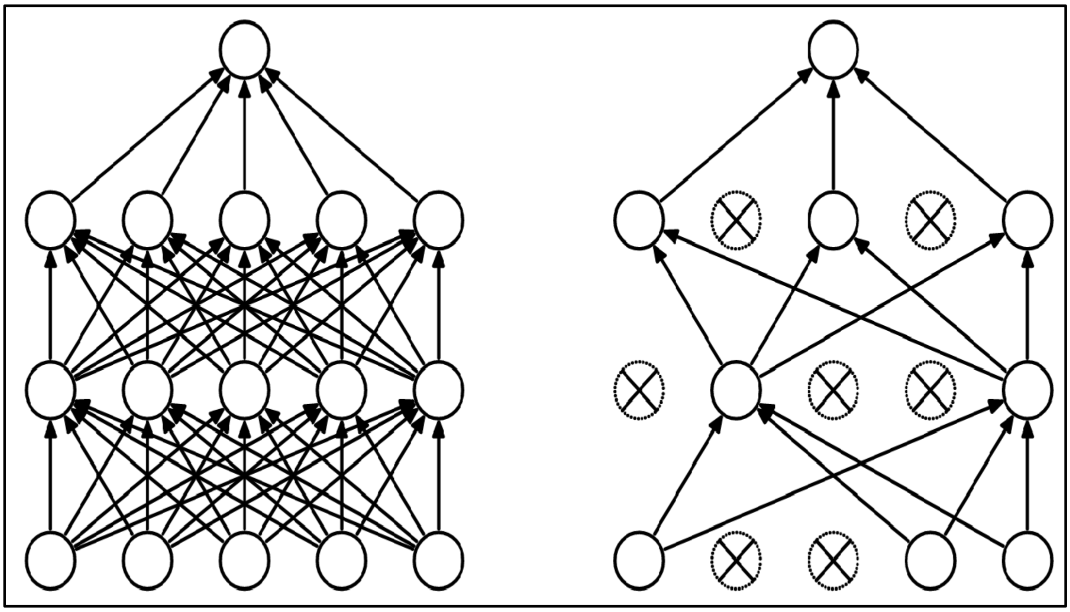


Figure (2.14) Dropout[45].

2.5.1.6 SoftMax

The SoftMax function commonly abbreviated as softargmax or normalized exponential function. It is a multi-dimensional description of the logistic function. It is often used to identify multiple classes and it is also used as a neural network's last activation function to normalize a network output to a distribution of probability over expected output classes. The SoftMax function takes a vector z of K real numbers as input, and normalizes it into a distribution of probability consisting of K probabilities proportional to the input numbers' exponentials. There is certain vector components may be negative or greater than one before applying SoftMax, and may not amount of 1. After applying SoftMax, each component is in the interval (0,1) and the components add up to 1, then they can be interpreted as probabilities. In addition, the larger components of input would lead to larger probabilities. The standard unit function of the SoftMax $\sigma : R^K \rightarrow R^K$ s defined by equation (2.6)[15].

$$\boldsymbol{\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \; for \; i = 1, \ldots, K \; and \; z = (z_1, \ldots, z_K) \in R^K} \qquad \textbf{(2.6)}$$

2.5.2 LeNet-5 architecture

Figure (2.15) shows the LeNet-5 architecture of 32 x 32 x 1 grayscale input image and the goal is to identify handwritten digital patterns. It uses a filter of 5 x 5 and has a phase of 1. By applying the receptive field measurement equation (2.7) and (2.8), the result of the output volume is 28 x 28. The derivation is obtained as [38].

- *W* x *H → 32* x *32 (Width* x *Height)*
- *F(w* x *h) → 5* x *5 (Filter)*
- *S → 1 (Stride)*
- *P → 0 (Pooling)*

$$\left(\frac{w-Fw+2P}{Sw}\right)+1 => \left(\frac{32-5+0}{1}\right)+1 => 27+1=28 \quad (2.7)$$

$$\left(\frac{H-Fh+2P}{Sh}\right)+1 => \left(\frac{32-5+0}{1}\right)+1 => 27+1=28 \quad (2.8)$$

Output volume = 28×28

The next layer is a pooling layer. To measure the pooling layer, the following derivation was applied.

- IM → 28 (Input Matrix → Convolution output volume., *Refer above derivation output*)
- *P → 0 (Pooling)*
- *S → 2 (Stride)*

$$\frac{IM+2P-2}{S}+1 => \frac{28+2*0-2}{2}+1 => \frac{28-2}{2}+1 => 14 \quad (2.9)$$

Output Matrix = 14×14

Finally, it goes to the 120-node fully connected (FC Layer) layer and is preceded by another 84-node FC Layer. It uses nonlinearity functions from Sigmoid or Tanh functions. An output variable with 10 possible digit values ranging from 0 to 9.

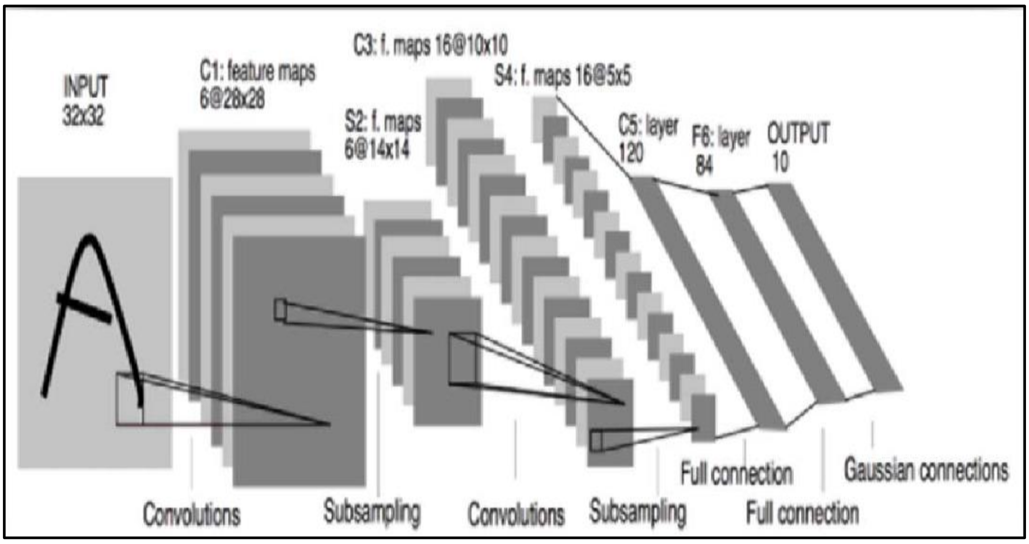


Figure (2.15) LeNet-5 Architecture [47].

The LeNet-5 CNN layers are

First Layer (C1): A 32 × 32 gray scale image is the input for this layer. This image passes through the first convolutional layer with six size 5×5 function maps or filter and a phase of 1. The dimensions of the image will be updated from 32×32×1 to 28×28×6[15].

Second Layer (S2): It is an average pooling layer with a scale of 14 ×14 for six feature maps. In the identical feature map in C1, each unit in each feature map is connected to a 2×2 block. It is also has twelve parameters that can be trained and 5880 links. The image dimensions resulting from this will be reduced to 14×14.

Third Layer (C3): It is convolutional layers with 16 5×5 function maps and a phase of 1. As comparable to positions in a subset of S2's feature maps, each unit in each feature map is connected to different 5×5 blocks. Lastly, the last one loads input from all maps of S2 functions.

Fourth Layer (S4): It is also an average pooling layer with a filter size of 2×2and a stage of 2 with 16 size 5×5 function maps. In the identical feature map in C3, each unit in each feature map is connected to a 2×2 block, and to the same path as C1 and S2. This layer is the same as the second layer (S2), if it does not have 32 trainable parameters and the performance of 2000 links 77s is reduced to 5×5×16.

Fifth Layer (C5): It is completely linked convolutional layers with 120 maps of functions. On all 16 of S4's feature maps, each unit is connected to a 5×5 block. Instead of a fully linked layer, C5 is known as a convolutional layer because if LeNet-5 inputs are rendered larger with everything else covered fixed, the dimension of the function map will be larger than 1 × 1. Each of the 120 units in C5 is connected to all the 400 nodes in the fourth S4 layer (5×5×16).

Sixth Layer (F6): A 84-unit fully connected layer. It is entirely associated with C5. It contains 10164 parameters that can be trained.

Output Layer: A SoftMax layer completely connected with 10 potential rates similar to the digits from 0 to 9 [15].

2.5.3 VGG- 16 CNN

The framework of the VGG-16 CNN is given in Figure (2.16). It represents the pictorial view as a chart. It doesn't use many hyper parameters. It is a simpler architecture model. It often uses 3 × 3 filters with step of 1 in the convolution layer and uses SAME padding with step of 2 in the pooling layer 2× 2 [48].

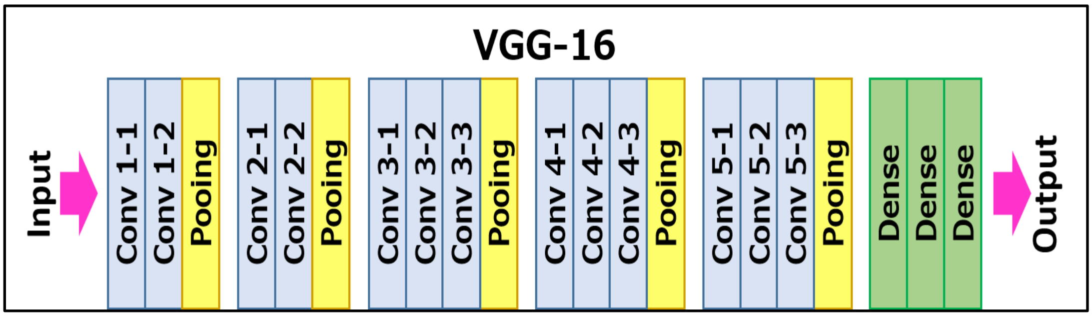


Figure (2.16) VGG-16 Architecture[49].

2.5.4 GoogLeNet

The ILSVRC 2014 winner and the architecture of GoogLeNet are also known as the Inception Module. It goes deeper with different receptive field sizes in parallel paths and has achieved a top-5 error rate of 6.67%.

This design consists of 22 deep layers. It reduces the number of parameters to four million from 60 million (GoogLeNet). it is clearly represented in the Figure(2.17) [13].

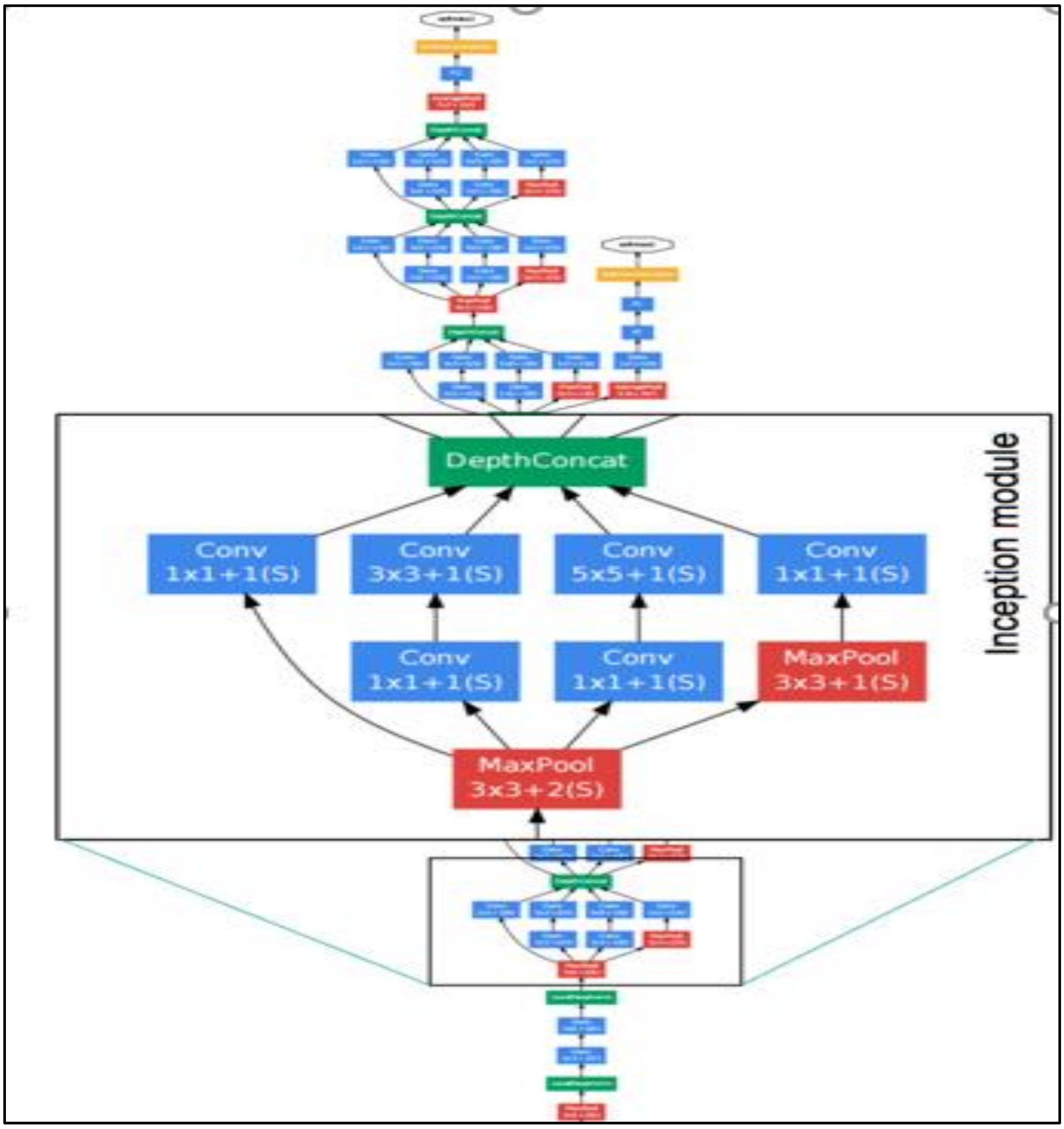


Figure (2.17) GoogLeNet Inception Module[49].

The Inception Module

The inception modules is alternate view of the architecture of Google Net as shown in the Table (2.1). The inception modules can also decrease the size of the data by providing pooling while performing the inception computation. This is basically identical to performing a convolution with strides in parallel with a simple pooling layer, which at a first glance is basically the parallel combination of 1×1, 3×3, and 5×5 convolutional filters. But the great insight of the inception module was the use of 1×1 convolutional blocks Network In Network (NiN) to reduce the number of features before the expensive parallel blocks [50].

Table (2.1) GoogLeNet Architecture[13].

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

2.5.5 ResNet

Figure (2.18) illustrate the architecture of the ResNet. It was also named by Kaiming as the Residual Neural Network (ResNet) winner of ILSRVC 2015. A term called "skip links" was implemented by this architecture. Usually, the input matrix is determined with the ReLU activation function in to two linear transformations. It directly copies the input matrix to the second transformation output in the residual network and sums the output to the final ReLU function [51].

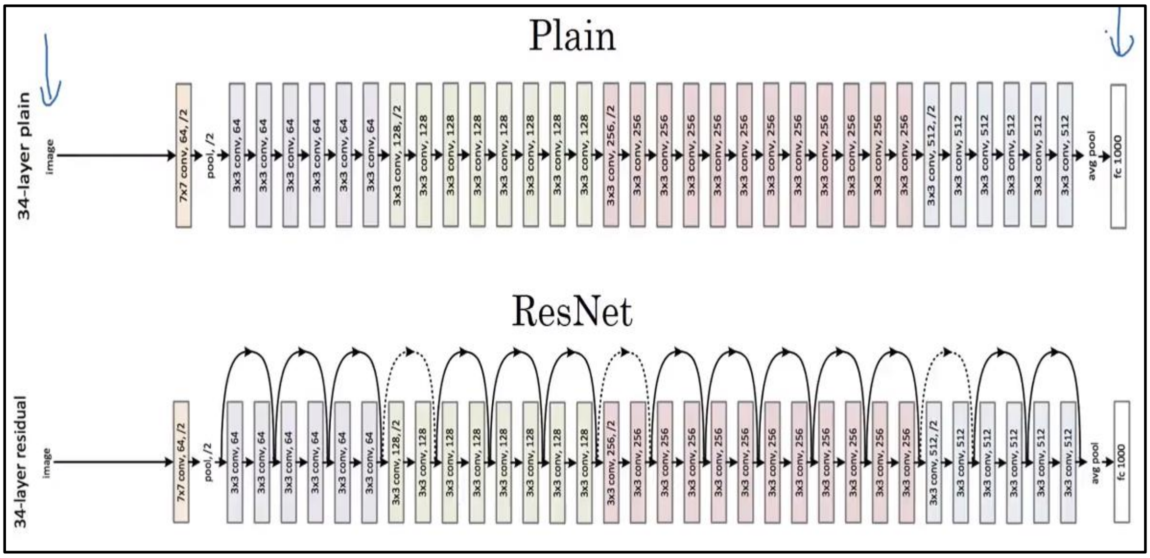


Figure (2.18) ResNet Architecture comparison with plain[49].

2.5.6 AlexNet

In this framework as shown in Figure (2.19), it is initially starts with 227 × 227 × 3 images size, and 96 of 11 × 11 filters with strides of four to be added to the next convolution layer. The final layer volume is reduced by 55 × 55 in its dimensions. The next layer is a pooling layer along with stride 2. It applies a maximum pool of 3 × 3 filters. It goes on and finally hits the 9216 parameter FC layer along with the two 4096 node FC layers. At the end, with classified 1000 output classes, it uses the SoftMax function and it has been incorporated with the 60 million parameters. So, it has 60 million parameters [52].

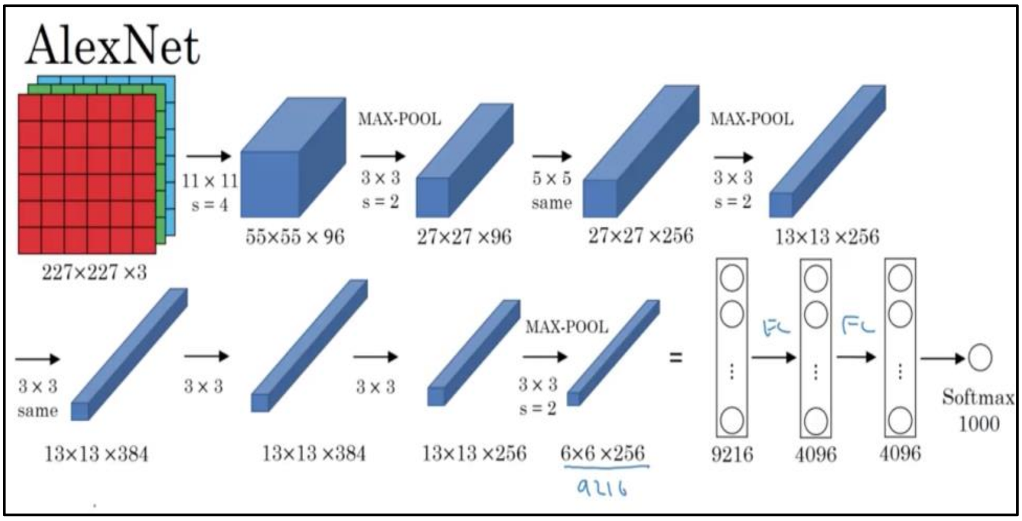


Figure (2.19) AlexNet Architecture[49].

The AlexNet Architecture's highlights; it practices Sigmoid or Tanh features instead of the ReLU activation function. It accelerates more than five times faster with the same accuracy. Instead of regularization, it utilizes dropout to deal with the overfitting. Yet with a dropout ratio of 0.55, the training time has doubled with seven secret layers, 650 K units and 60 M parameters. The architecture is the bigger model with a various parameters [53].

2.6 Deep Learning Applications

It is much necessary to implement the outcomes by applying Deep Learning concept to analyze the Big Data. Deep learning focuses mainly on the nature of the V’s concept, which is of two ways to be analyzed. The most important features are volume and variety. DL is the best solution in extracting the valuable data and it is get extracted based on the previous historic data. It obtained from both vast volumes and from the various sources.

Microsoft speech recognition (MAVIS) is the main application for audio and video speech recognition applied by implementing DL. It is allowing audio and video files to be searched as an example of the application of Deep Learning in Big Data through human voices and speeches. Whereas content-based image retrieval in the Big Data world is a recent trend application and it is introduced using DL. It is commonly used for image search services by Google. Researcher faces some problems when they implement DL. In this work, we will try to overcome some of these problems [54].

2.6.1 Deep Learning for Massive Amount of Data

Initially, it is sufficient to practice all the Big Data input or not. The first one is whether or not one can use any of the feedback from Big Data. In general, researcher uses DL algorithms for training purposes in a percentage of existing Big Data to extract abstract representations. In this scenario, the data volume is required for training in the given domain applications. In the next paragraphs some of the problems faced the researchers in this field.

- **The domain**: in applications where it is important to come in to aspect that the training data is dissimilar from the circulation of test data the domain adaptation is the most important issue. From another point of view, one may point out how DL's generalization ability can be increased. Generalizing learned patterns where there is a shift between the input domain and the target domain.
- **The specification of the parameters:** enabling data representations based on the specification of the parameters to provide useful future semantic meanings is another issue. In simple words, it should not be permitted to assign useful significance to each extracted data representation. To obtain better data representations, one has to have certain parameters.
- **Define the loss:** The most important issue in the DL techniques is to define the loss and it is much important to mention the target to get extract. It is simply a tedious process to extract the data from the huge volume of data. as shown below[15]:

  $$H_{P(y)} = -\sum_i y_p * \log p_i \qquad (2.10)$$

  where pi is the predicted probability distribution of class i and yi is the true probability distribution of class i. As we have ten classes in our dataset the summation is computed over 10 terms. After every minibatch iteration of the training process we compute the total cross entropy error and optimize the parameters of the network so that the loss is gradually minimized. We have used the 'Adam' optimizer [55] for optimization of the loss function.
- **The empirical outcomes:** the other issue is that most of researcher does not have clearly understood the empirical outcomes. In other words, it cannot

evaluate the process easily because of its complexity. In a Big Data world, this problem gets worse.

- **Combining:** Due to its ability to learn abstract representations, Deep Learning seems ideal for combining heterogeneous data with multiple modalities.
- **Labeled data:** The last but not least significant problem is that they need labelled data. They will have bad output if they cannot have labeled data. The optimal solution should utilize reinforcement learning. In which, the system collects knowledge on its own and it only need for them to reward the system [55].

2.6.2 High Variety Data in Deep Learning

Nowadays, information arrives from a number of sources in all sorts of formats are potentially with distinct distributions, for example, a large collection of audio streams, images, graphics and animations, unstructured text, and videos. It is included that related to multimedia data. In this respect, as some of them are offered as open resources for the big data analysis, these are open questions that need to be answered.

- Conflicting details can be obtained from specifically sources. The, how can researchers effectively and efficiently overcome the struggles and rage the information from various sources?
- Is the efficiency of the system paybacks from dramatically expanded modalities?
- At what stage the deep learning architecture is suitable for heterogeneous data feature fusion?

2.6.3 High Velocity in Deep Learning

Deep learning are considered the optimal solution for learning from such high-speed data. In recent years, only modest progress has been achieved in online deep learning. In this sector, there are various weaknesses as data is always non-stationary and the distribution of data varies over time. The big question is how to benefit from Big Data related to the learning architecture [56].

2.7 Data Augmentations

CNN is applied to visual object recognition, data augmentation. This is often utilized to generate additional data without introducing extra labeling costs. Data Augmentations is considered one of the most effective tools in association with image manipulations geometric transformations depicts various enlargements depending on mathematical changes and numerous other picture handling capacities. The class of enlargements examined beneath could be portrayed by their simplicity of execution. Understanding these changes will give a valuable base to additional examination concerning data augmentation methods. The diverse mathematical expansions with regards to their 'security' of use will depicted. The security of a data augmentation strategy alludes to its probability of protecting the name post-change. For instance, Rotation and flips are commonly protected on ImageNet difficulties, for example, T-shirt/top versus Shirt.

A non-mark protecting change might reinforce the model's capacity to yield a reaction showing that it isn't sure about its expectation. Notwithstanding, accomplishing this would require refined marks [11] post-growth. On the off chance that the name of the picture after a non-name saving change is something like [0.5 0.5], the model could learn more vigorous certainty forecasts. Building refined names for each non-safe data augmentation is a computationally costly cycle. Because of the test of building refined names for post-increased information is essential to consider the 'security' of an enlargement. This is to some degree area subordinate, giving a test to creating generalizable increase arrangements. There is no picture preparing capacity that can't bring about a mark changing change at some contortion greatness. This shows the information explicit plan of expansions and the test of creating generalizable increase strategies. The significant thought concerning the mathematical expansions recorded are:

2.7.1 Flipping

Level center flipping is considerably more average than flipping the vertical turn. This expansion is easy to execute and has exhibited important on datasets, for instance, CIFAR-10 and ImageNet. On datasets including text affirmation, for instance, MNIST or SVHN, this isn't a name protecting change[57].

Concealing space digital picture data is commonly encoded as a tensor of the estimation (Height × width × channels). Performing extensions in the concealing channels space is another strategy that is amazingly feasible to realize. Incredibly direct concealing growths consolidate binding a lone concealing channel, for instance, R, G, B. An image can be quickly changed over into its depiction in one concealing channel by detaching that lattice and adding two zero structures from the other concealing channels. Additionally, the RGB characteristics can be conveniently controlled with direct structure errands to augmentation or diminishing the splendor of the image. Further created concealing amplifications come from deriving a concealing histogram portraying the image. Changing the power regards in these histograms achieves lighting adjustments, for instance, what is used in photo modifying applications [47].

2.7.2 Cropping

Trimming pictures can be utilized as a viable handling venture for picture information with blended tallness and width measurements by editing a focal fix of each picture. Moreover, arbitrary trimming can likewise be utilized to give an impact fundamentally the same as interpretations. The contrast between irregular trimming and interpretations is that the editing will decrease the size of the information, for example, (256,256) → (224, 224), though interpretations safeguard the spatial components of the picture. Contingent upon the decrease limit picked for editing, this probably won't be a mark safeguarding change.

2.7.3 Rotation

Pivot expansions are finished by turning the picture right or left on a hub somewhere in the range of 1° to 359°. The wellbeing of turn expansions is intensely controlled by the revolution degree boundary. Slight turns, for example, somewhere in the range of 1 to 20 or −1 to −20 could be valuable on digit acknowledgment errands. Interpretation shifting pictures left, right, up, or down can be a helpful change to keep away from positional inclination in the information. For instance, if all the pictures in a dataset are focused, which is regular in face acknowledgment datasets, this would also require the model to be tried on entirely focused pictures. As the first picture is interpreted toward a path, the leftover space can be loaded up with either a steady worth, for example, 0 s or 255 s, or it tends to be loaded up with irregular or Gaussian commotion. This cushioning jams the spatial components of the picture post-enlargement.

2.7.4 Noise Injection

Clamor infusion comprises of infusing a framework of irregular qualities typically drawn from a Gaussian dispersion. Commotion infusion is tried on nine datasets from the UCI vault [58]. Adding commotion to pictures can enable CNNs to learn more vigorous highlights. Mathematical changes are awesome answers for positional inclinations presented in the preparation information. There are numerous possible wellsprings of predisposition that could isolate the dispersion of the preparation information from the testing information. On the off chance that positional predispositions are available, for example, in a facial acknowledgment dataset where each face is completely focused in the casing, mathematical changes are an incredible arrangement.

A helpful answer for exorbitantly awesome or dull pictures is to hover through the photos and reduction or augmentation the pixel regards by a steady worth. Another quick concealing space control is to unite out individual RGB concealing cross sections. Another change includes restricting pixel regards to a particular min or max regard. The regular depiction of concealing in automated pictures fits various strategies of broadening. Concealing space changes can moreover be gotten from picture modifying applications. An image's pixel regards in each RGB concealing channel is gathered to outline a concealing histogram. This histogram can be controlled to apply channels that change the concealing space ascribes of an image. Like mathematical changes, a weakness of shading space changes is expanded memory, change expenses, and preparing time. Moreover, shading changes may dispose of significant shading data and accordingly are not generally a mark safeguarding change. For instance, while diminishing the pixel estimations of a picture to recreate a hazier climate, it might get difficult to see the items in the picture. Another circuitous illustration of non-name saving shading changes is in Image Sentiment Analysis [59].

2.8 History of The Fashion-MNIST dataset

The Fashion-MNIST dataset is suggested as a more demanding replacement dataset for the MNIST dataset. Fashion-MNIST is a dataset of the article images of Zalando which consists of a training set of 60,000 examples and a test set of 10,000 examples. The sample is a grayscale 28×28 pixels image associated with 10 classes. Fashion-MNIST is served as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of the training and the testing splits[15].

However, the MNIST dataset is popularly used by the deep learning researchers because of the constructed values are scaling and it used to give the possible outcomes for all the corner test cases. It determined the output very quickly for all complicated input values. The Zalando's website is a platform for collecting the fashion and it has got the collection of the Fashion-MNIST. It has a series of images taken by professional photographers screening various characteristics of the product such as foreground and background, photos, model appearances and suit looks. Therefore, raw image is a grey contextual (hexadecimal color: # fdfdfd). It is deposited a JPEG format of 762 × 1000. The original image is re-sampled with numerous resolutions for the well-organized serving of different frontend elements such as medium, thumbnail, large, small and tiny. Fashion-MNIST is created using the front-look thumbnail unique images of about 70,000 items. These images are collected from the various classes of genders, men, women, children and neutrals. Objects are not applied to the dataset in white colour images because they have low contrast in the background. Then the thumbnails (51 x 73) are fed to the pipeline shown in Figure (2.20)[18]. As follows

- The primary step is to convert the image to a PNG image.
- Cutting edges near the colour of the corner pixels. In 5% of the "closeness" supreme conceivable intensity in RGB space is defined by the distance.
- Resize the image's longest edge to 28 by presampling the pixels, i.e. skipping several rows and columns over.
- Sharpening pixels using a Gaussian operator of the radius and standard deviation of 1.0, with increasing effect near outlines.

- Extending the shortest edge to 28, and positioning the image in the canvas middle.
- Negating the intensities of the image
- The picture is translated to 8-bit grayscale pixels[18].

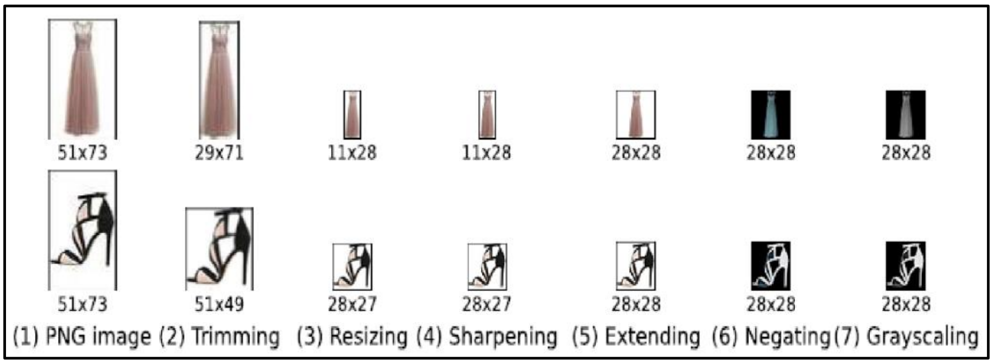
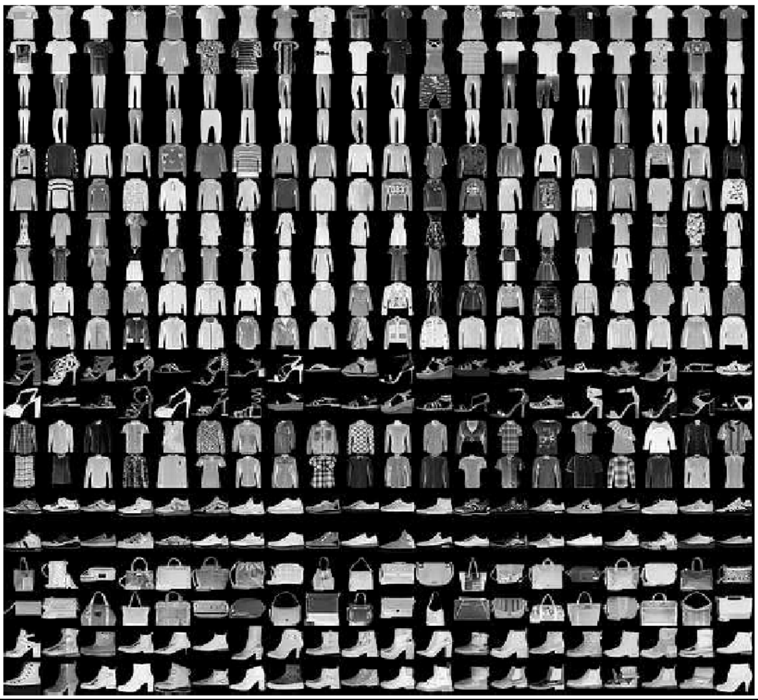


Figure (2.20) Fashion-MNIST dataset conversion diagram.

Figure 3.1 lists two examples from the categories of dress and sandals. The class labels are named after a silhouette code of the product is taken. The in-house design experts mark the silhouette code manually and it reviewed by a separate Zalando team. Each product contains one single silhouette code. Table (2.2)[18] shows a list of all Fashion-MNIST class labels with examples.

Table (2.2) Class labels in Fashion-MNIST with examples.

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

Lastly, the dataset is broken down into a trained set and Validation set. The training set collects one randomly chosen 6,000 examples from each class. In the same file format as the MNIST data collection, images and labels are stored, which is designed to store multidimensional matrices and vectors. The outcome is referenced as illustrated in Table (2.3). The trained data are sorted by their labels and stored for further processing. It results in a reduced label files subsequently firmness associating to the MNIST. Examples with a certain class mark are often easier to retrieve. Therefore, the task of shuffling data is left to the creator of the algorithm.

Table (2.3) Files contained in the Fashion-MNIST dataset[18].

Name	**Description**	**# Examples**	**Size**
train-images-idx3- ubyte.gz	Training set images	60, 000	25 MBytes
train-labels-idx1-ubyte.gz	Training set labels	60, 000	140 Bytes
t10k-images-idx3-ubyte.gz	Test set images	10, 000	4.2 MBytes
t10k-labels-idx1-ubyte.gz	Test set labels	10, 000	92 Bytes

CHAPTER THREE

MATERIALS AND WORK METHODS

CHAPTER 3: Materials and Work Methods

3.1 Introduction

The aim of this Thesis is to train CNN network using Sequential model based on fashion Data set image and create an effective model of deep learning that recognize accessories for clothing and augment Fashion-MNIST dataset that we use. For a Fashion-MNIST dataset the accuracy of existing works achieved is 91%. The used Fashion-MNIST images were augmented based on three different augmentation techniques. It is done by rotate the image to a particular angle and shift the image. There will be one image in three output types, such as rotation of the image, shift, Flip and zoom image as an output for all the 60,000 training images from the Fashion-MNIST dataset and 10,000 testing images from Kaggle dataset. Initially the data has been preprocessed for resizing and remove the noise. Then this image has been given as the input for the augmentation. This augmented data contributes as input images to the pre- training model.

3.2 Dataset Proposing

The Fashion-MNIST images have matrices of 28 x 28 that has 28 pixels both in height and width. Entirely the matrix has 784 pixels. Every matrix with pixels has been allotted with the value along with their reliant of pixel lightness or the darkness. A data set integer has limits 0-255 that is drawn. This integer limit has assigned as minimum value with example would be lower when compared with the maximum rate from the dataset. Almost 785 row data pixels will be for both training and testing dataset. At first column, both data sets will have class labels which have been represented as sneaker, dress, coat or others. In general, clothing category is in the first column. The remaining columns possess the value of pixels of the image on basis of maximum or minimum features with 0-255 integer value . The classified training and testing datasets are inside these pixels values (0-255). The image pixel has been predicted as $i \times 28 + j$ for pixel X. Among the numbers of 0 and 27 have an integer i and j. The row i is positioned as pixel along with the matrix 28 x 28 standards of column j. An individual image is in each row, class label is in each column and each pixel value has been predictive in lower pixel range. Class labels of the fashion images are predicted with the category in the dataset. The Fashion-

MNIST dataset are augmented before the dataset images are pre- Trained with the category and it is predicted with the class labels. Those Fashion-MNIST dataset images are alone utilized and hence pre- Trained CNN trained data has been tested and trained to classify the input images. The following steps have been given for the data preparation.

At the beginning, we applied Python programming language on the Google Collab platform. Python is an open-source, dynamically-typed, high-level, portable, easy to learn, and code programming language. Google Collab provides 12 hours of free access to K-80 GPU power. It run on Google Drive. The best part of that is, there is no need to install any ML or DL library. Without installing we can use very easily. Google Collab is a free cloud service and now it supports free GPU! You can: improve your Python programming language coding skills. develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV

- Initially the required libraries have been traded in namely tensor flow, keras, numpy, matplotlib and pylot.
- The next step is stacked up by the Fashion-MNIST dataset. This has been extracted from Kaggle. The dataset is downloaded from the Kaggle server. Dataset includes photographs of clothing pieces and accessories in fashion.
- A target dictation connecting fashion images or the inputs with the target variables or the 10 classes is now coded. This corresponding to the existing data definition, and a typical assignment would be as target classes as 0: ‘T-shirt/top’ 1: ‘Trouser’, 2: ‘Pullover’, 3: ‘Dress’, 4: ‘Coat’, 5: ‘Sandal’, 6: ‘Shirt’, 7: ‘Sneaker’, 8: ‘Bag’ and 9: ‘Ankle boot’.
- The data set is divided into two datasets training and validation processes.
- The data set was finally normalized. The normalization is necessary to ensure where all the information is on the same scale and that this typically enhances efficiency. For the Fashion-MNIST dataset, the normalization for the training dataset and the validation dataset would be handled as mapping [0,255] to the [0,1] that will increase the training speed.

3.3 The Proposed Architecture

The dataset taken from Fashion-MNIST data is used as input for both the training and the testing phases. Figure 3.1 is the proposed architecture for the Pre-Trained CNN networks. First the input data has been preprocessed for resizing the image and to filter out the noise. This filtered data is augmented where the image has been rotated, shifted and Flip to obtain three various sets of input. The images are sent to augmented data generator where the training image is three times the original one. Then, it is pre-Trained using the proposed CNN. Its output is the trained data, whereas the tested output is compared with the trained data and then the performance measure process takes place After compute the total error. The output is the classified output of ten-classes.

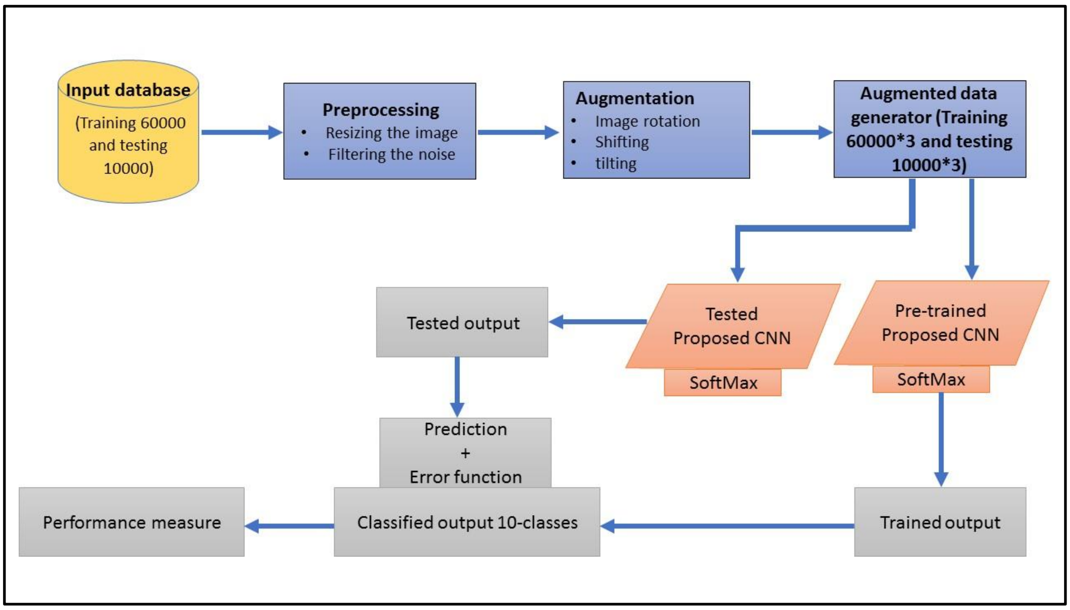


Figure (3.1) Framework of the proposed deep learning model.

3.3.1 Data Preprocessing

The initial step is to preprocess the data that involves loading of data. The loading of data is related to the function used for proportional evaluation of CNNs. These processes are initializing files, size of the image, classes and images per file along with path for data has been created. Initializing files, count for the file is assigned as three. The resize of the data have been augmented and the augmented image has been assigned for preprocessing. For the data loading, both images and labels have been loaded for training the dataset. The pseudo code of the data preprocessing phase is shown in Pseudo code (3.1).

Pseudo code (3.1) Data Preprocessing.

Step1: Load the Fashion-MNIST dataset.
Step2: Resizing the input augmented image as the input to the file is the input data.
Step 3: Converting the input augmented data in a n-dimensional array using numpy.
Step 4: The input data is resized to the limit of 28×28,1.
Step 5: Return the data for the further process.
Step 6: Preprocessing the listed data values.
Step7: The array_data is augmented and it is used for further processing.

3.3.2 Data Normalization

Normalization of data is a function of preprocessing method. The CNN operates with data normalization of batch is carried out by the following steps, the batch normalization is utilized in enhancing the training process of dataset. In addition, it will also enhance the conditioning during the training process. Generally, the enhancement in conditioning of the problem leads to the gradients from backpropagation has become easier. The obtained function of convergence average count of iterations has been minimized. Complexity of the process has been slowed along with the limitations like phase intensity due to the huge dataset that has to be minimized. Batch normalization through itself is not intensive computationally and the phase also has been decreased per iteration. The memory of the bandwidth for batch normalization has been limited in these characteristics. Two passes are computed with the input data. The first pass is basically compute the statistics of the batch being read. The second pass is the normalization of the output of the layer. In this research, it has been identified that the use of BN is to reduce the training time to around one by fourth of the total training time of the pre-Trained. Therefore; it has been used in the proposed pre-Trained network model for
classification purpose. The convolutions are going to be more optimized and the use of BN helps reduce the run time even more. Normalization is necessary to ensure all the information is on the same scale, and this typically enhances efficiency. For the Fashion-MNIST data set, the normalization for training dataset and the validation dataset would be handled as in Pseudo code (3.2).

3.3.3 Data augmentation

Data augmentation is a process of adding various changes to the raw input images, resulting in several copies of the same image that have been transformed. However, depending on the applied augmentation techniques such as moving, spinning and flipping, each copy is distinct from the other in some ways. Applying these tiny quantities of variations to the original image does not alter its target class, but only provides a new perspective in real life to capture the object. We use it very frequently to create models of deep learning. The ImageDataGenerator class ensures that at each epoch the model gets new variations of the images. But only the converted images are returned and not added to the original corpus of images. If that was, in fact, the case, then sometimes the model would see the original images that would certainly over-fit our model. Another benefit of the ImageDataGenerator is that lower memory consumption is necessary. This is because we load all the photos at once without using this class. When we use it, we load the images into batches that save a lot of memory. Data preparation is needed when working with Deep Neural Network Models. Tasks for object recognition are more complex and increasingly involve data augmentation. To achieve the best efficiency, the Deep Networks need very large amounts of sample data. Image augmentation is one method to get more data for training. Data augmentation techniques are used to enhance the performance of deep neural networks which used to create an image classifier using very little data. The selected data called (Keras) data augmentation processes used in this thesis is illustrated in Figure (3.2). The augmentation techniques with ImageDataGenerator class used in this thesis are as follows.

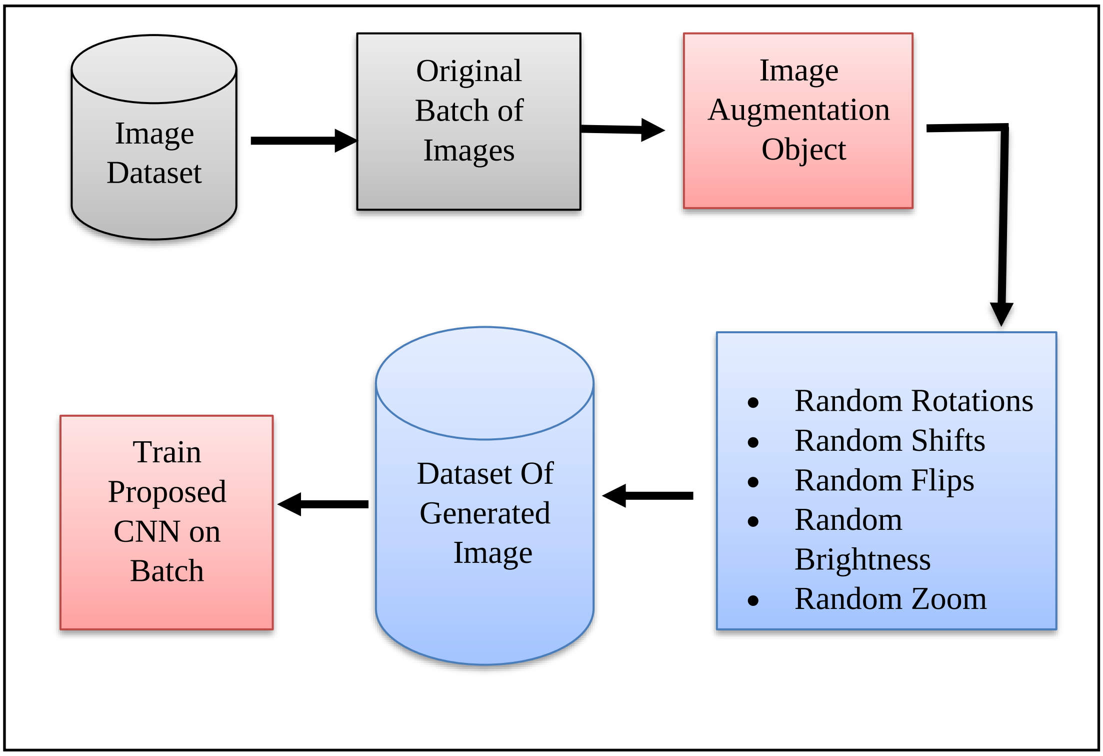


Figure (3.2) Work of Keras Data Augmentation.

3.3.3.1 Random Rotations

One of the commonly used techniques for augmentation is image rotation. It enables the model to become invariant to the object's orientation. The ImageDataGenerator class allows one to rotate images randomly to any degree between 0 and 360 by supplying the rotation range statement with an integer value. Some pixels can travel outside the image when the image is rotated and leave an empty region which needs to be filled in. It can be filled in by various means, such as a constant value or the nearest pixel value. This is defined in the fill mode argument and "nearest" is the default value, which simply replaces the empty region with the nearest pixel value. In this thesis, the Random Rotations augmented images of fill_mode = 'nearest' are used in Fashion-MNIST input and it is shown in Figure (3.3).

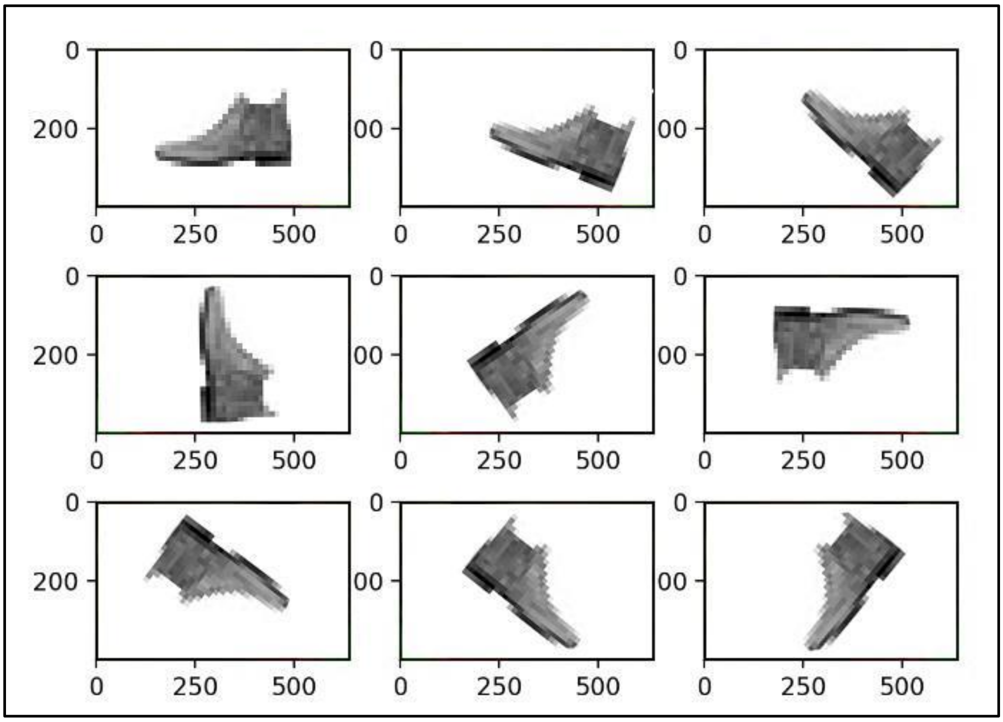


Figure (3.3) Random Rotations augmented Images.

3.3.3.2 Random Shifts

It can happen that the object may not always be in the picture's middle. The pixels of the image can be moved either horizontally or vertically to solve this problem. It is achieved by adding all the pixels to a certain constant value. The ImageDataGenerator class has the height-shift-range argument for vertical image shift and the width-shift-range argument for horizontal image shift. This will mean the percentage of the width or height of the picture to be shifted if the value is a float number. Otherwise, whether it is an integer value, these various pixel values simply change the width or height. In this thesis, the Random Rotations augmented images of Random Shifts = 10 is used in Fashion-MNIST input and it is shown in Figure (3.4).

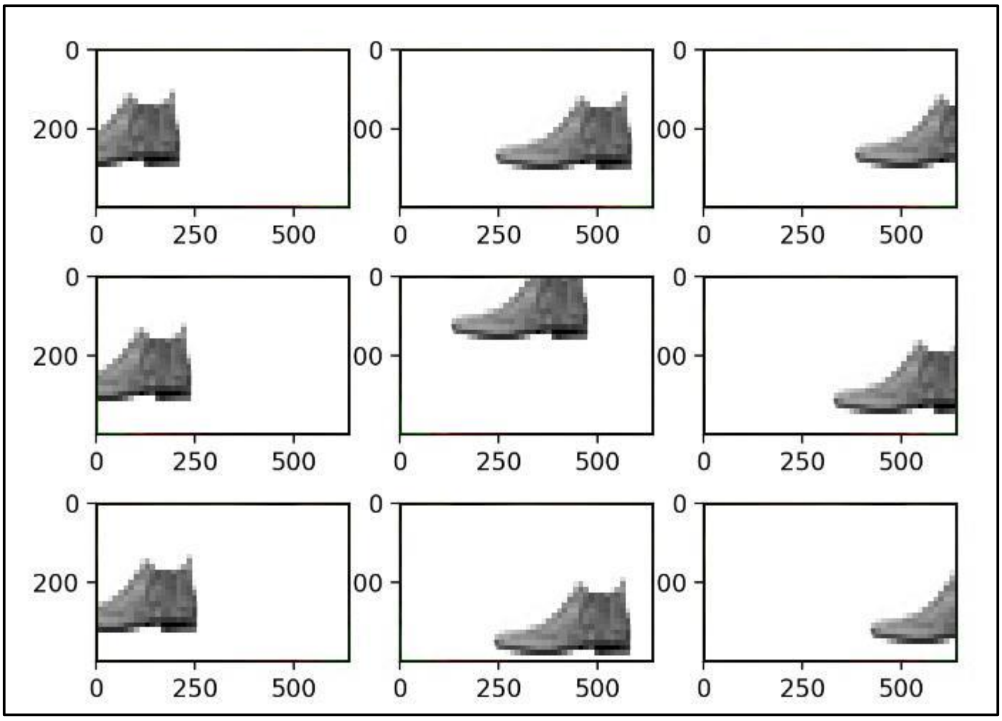


Figure (3.4) Random Shifts augmented Images.

3.3.3.3 Random Flips

Also, flipping images is a brilliant augmentation method and doing it for a lot of different objects makes sense. In order to find the proper position for that, flipping connected to the horizontal or vertical axis, the ImageDataGenerator class has vertical flip and horizontal flip parameters. This form, however, should be chosen according to the object of the image. This method should, however, be chosen depending to the selected object of the image. In this case, vertical flipping of a vehicle, for example, not sensible in comparison with to symmetrical artefacts such as football. In this thesis, the Random Flips Augmented images of horizontal_flip = True is applied in Fashion-MNIST input and it is shown in Figure (3.5).

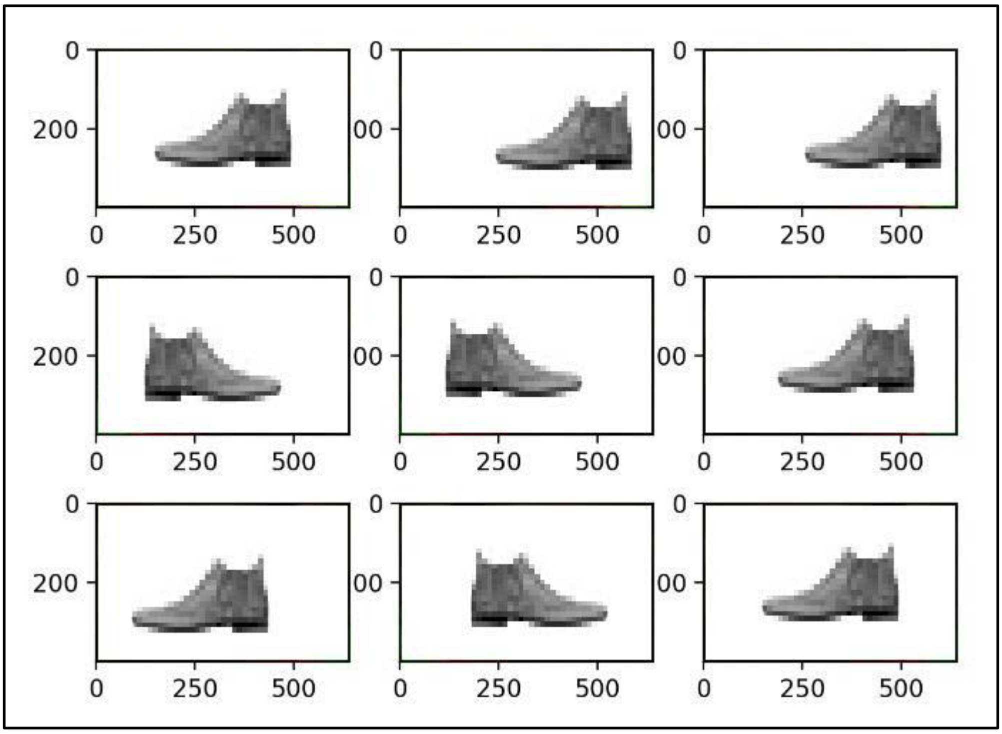


Figure (3.5) Random Flips augmented Images.

3.3.3.4 Random Brightness

The image brightness is randomly altered in this process. It is also a very useful augmentation tool since our object is not under optimal lighting conditions much of the time. So, under various lighting conditions, these types is considered imperative to train the appropriate model on pictures. Brightness can be managed through the brightness range argument in the ImageDataGenerator class. In order to obtain such an optimal brightness, this type is considered a list of two float values as shown in Figure. (3.6). Values smaller than 1.0 make the image darker, while values above 1.0 brighten the image.

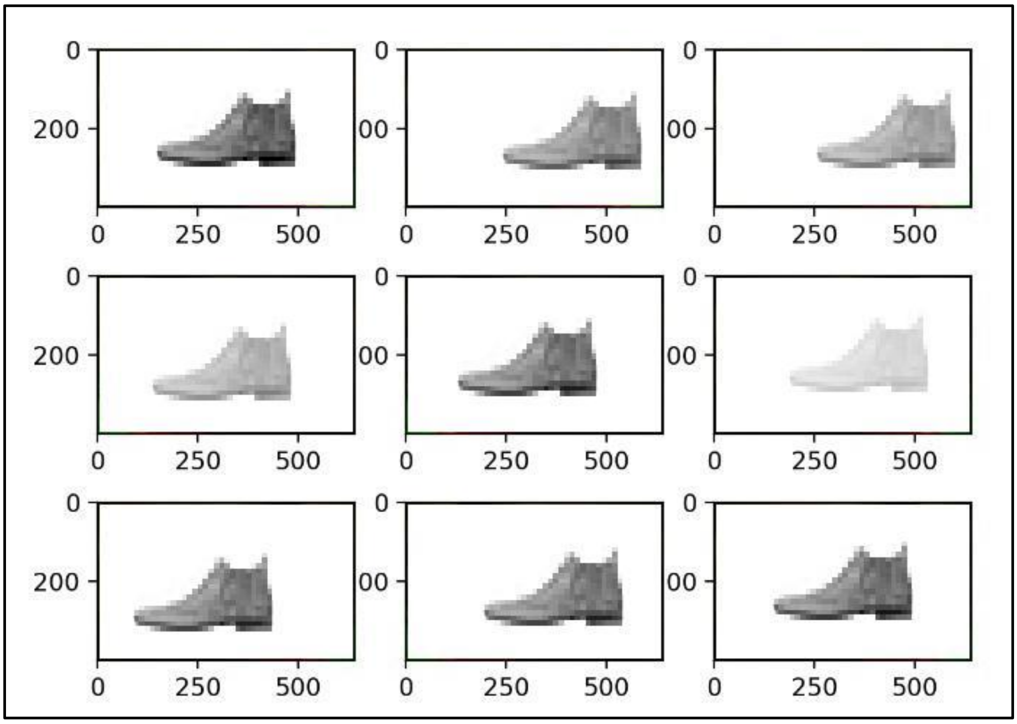


Figure (3.6) Random Brightness augmented Images.

3.3.3.5 Random Zoom

Zooming is either randomly zooming in or zooming out of the picture. In the zoom range argument, the ImageDataGenerator class takes a float value for zooming, as seen in the Figure (3.7). A list of two values defining the lower and the upper limit could be given. In addition, if a float value is defined, the zoom is carried out in the [1-zoom range, 1+zoom range] range. The image contains values smaller than 1 zoom and the image contains values greater than 1 zoom.

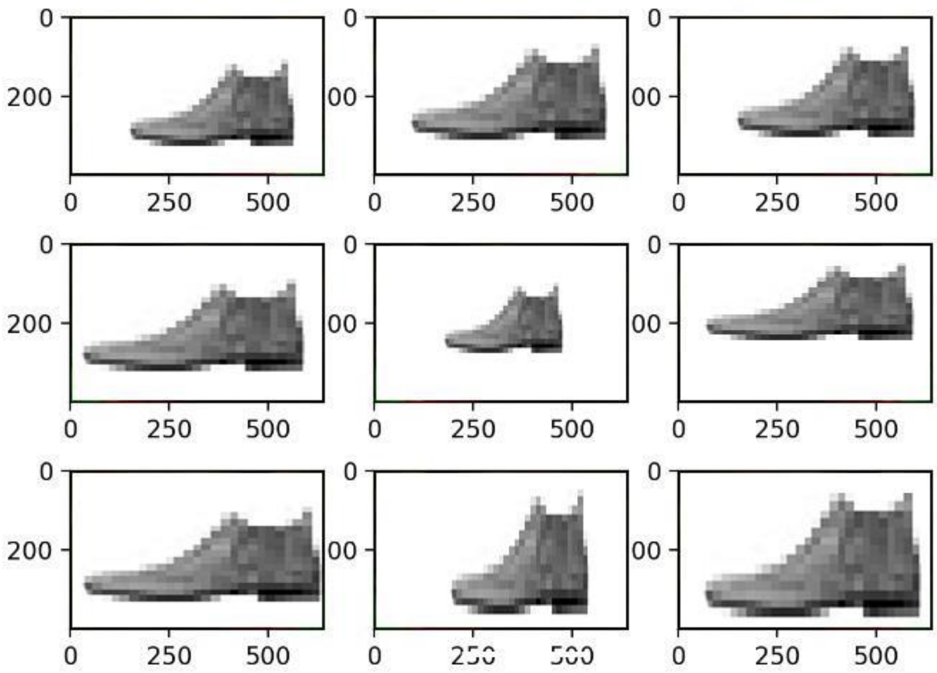


Figure (3.7) Random Zoom augmented Images.

The pseudo code (3.2) for the data augmentation and this augmented process of artificially generating new images for training is given below. This is the basic procedure applied for data augmentation based on applying transformations such as random rotations, shifts, shear, flips and etc.

Pseudo code (3.2) Data Augmentation.

Step 1: The input image is passed to the function (tf.keras.preprocessing.image.ImageDataGenerator( )).
Step 2: The function is passed with the values as rotation_range=10, horizontal_flip=True, fill_mode='nearest'.

3.4 Data Training

Initially in training process the data to be trained has been loaded. The set of images has been started with the images No, size of the image, No of the channels and set of labels has been created along with the images counts, integer type. The operation of load data for training is done before CNN pre-trained. The initial stage for training is the generating of Epochs and the labels for the inventive operation. The first step is to set the number of epochs to 150 that took 25 - 30 minutes to implement and test the function of labels top to labels Ankle boot.

The training data sequence operation is operated by training labels; Fashion-MNIST data has been arranged in step 2 to process the implementation as shown in Pseudo code (3.3). The function generation are the class counts in the first stage that is equivalent with 4 along with the class counts in the second stage is equivalent with 3 for top, then the class counts in the second stage is equivalent with 2 for shoes. Finally, the channel counts are equivalent with MNIST data, and the class count is equivalent with 7. The pseudo code of the data training and the procedure of it is implemented in the projected framework is given below.

Pseudo code (3.3) Data Training.

Step1: Initialize the packages TensorFlow, NumPy, Matplotlib.
Step2: Load the Fashion-MNIST dataset.
Preprocessing:
Step3: Split the dataset into training and testing data.
Step 4: Initialize the input shape and it considered as the first layer of the network model.
Step 5: The augmented data is done in the training and testing data.

3.5 Proposed CNN with Pre-Trained

Neural network is focused in accounting the data processing through the shape of the input. It looks similar to the 2D image and the input images are pr-processed by augmenting the resized image. This approach has taken the advantage of image recognition along with pattern detection and image classification. In general, the image is measured by pixels of the matrix based on the shape of the image. The operation for distant training in CNN is utilized in detecting the image and the image classification has been done based on the pre-annotated classes. CNN is Supervised Learning. Supervised Learning In supervised learning algorithm the classic machine learning setting is use. Where the training data is a combination of tuples (xi,yi), where xi represent the input and yi the corresponding target vector. The case in which the target value is discrete, such as the images recognition problem, where the images are mapped to a finite number of discrete categories, is called classification problem.

In the extraction of image characteristics, the convolution is the first layer. The connection among the pixels uses small squares in input data for classifying the image characteristics based on the convolution process. The mathematical function uses multiple inputs; two input namely image matrix and filter or kernel. **As explained in Equation (2.1) and (2.2) in Chapter Two.**

The non-linear activation function is used in the next phase. The result is at x = max(0,x). **As explained in Equation (2.3) in Chapter Two.**

The data can be operated through the width and the Height of the image that accompany through sampling in down direction. Data volume has been minimized as their significance. When the specific features has been indicated such as boundaries, it could be identified through the convolution function extracted from the previous convolution and it does not require the additional process to load the data for processing. The elaborated compression has been neglected from the information. In addition, the convolutional series of nonlinear and layers of pooling have been fully connected and it is important to be fully connected. From the above the output of convolution networks has been obtained. The network outputs with the final stage has been related with the layer of fully connected for dimensional vector N, and N represents the class counts that selects the necessary class.

In convolution, the computation part has been given with the same signals and using this process the image has been identified and classified. Generally, for CNN, it has multiple convolution layers; this means many altered convolutions have been generated. For calculation, the matrix weight has been used and the tensor form is of 5 x 5 x n; where n indicates the convolution numbers of CNN. The proposed CNN is a pre-convoluted, and therefore the function of distance is trained for evaluating identities among the fashion images. Semantic noise problem has been managed with optimized using the CNN. In this research, we have evaluated the fashion images calculation using CNN in classification that obtains optimal accuracy. According to this differentiation of category, there is option to train from scratch or to fine tune. Then by classifying based of product repeatedly is reliant on Pre-Trained again. However, fine tuning is possible in acquiring a High-performance value. The existing CNN classification supports the notion even that it has the enormous fashion images with higher standards. The Pre-Trained and fine tuning has been done in preparation of distance function that has been assisted with the performance.

The proposed CNN model has a number Convolutional Network models Which we have tried that process on datasets of two convolutions discussed above. Initially, the network model has a single 28 filter layers. Its kernel size is 3×3. Then ReLU layer activated and process the implementation. Next for maxpooling the 2 × 2 kernel size is used. Their dropout will be 20% that is utilized in standardizing the overfit. At the end, the computation of loss function is done through implemented SoftMax layer.

The network model 2 consists of 32 filters with the convolutional layers correspondingly. In this convolution layers, the kernel size is 3×3. The activation of ReLU has been utilized similar to the initial network model for every layer. In which, the used kernel size for maxpooling is 2×2 in every layer. Its dropout is 20% which is below the first layer that has been implemented and 25% of dropout has been utilized in layer 2 in the prevention of over fit. The loss function has been evaluated by the SoftMax which is deployed finally.

Network model 3 consists of 64 filters with the convolution layers correspondingly. Layer 1 uses 3×3 kernel size. Both layer 2 and layer 3 use kernel size of 3×3. The three layers were implemented in ReLU activation. The layer 1 Kernel size 3 × 3 max pooling is functioned in layer 1, 2 and 3 and it is implemented as 2 × 2. After implementation of layer 1 and 2 there is 25% of dropout. After layer 3 the dropout activated up to 20% to reduce the overfitting. Finally, the loss function calculation SoftMax is fully connected like the previous network models.

The network model 4 is on basis of pre-Trained CNN and it consists of 128 filters. In general, CNN are unstable sometimes, eventually during gradient propagation by the extended window that might generate gradient desertion and over burst. Hence the design developed uses Pre-Trained network model. A sequential model is used by the network model 4 which remains stack at linear layer. The model has been tested on 512 batch size.

The network model 5 consists of stack of linear layers. It is on basis of pre-Trained CNN and it consists of 1024 filter. Layer 5 is the layer of Pre-Trained that has implemented with the ReLU of the previous network models that have Kernel_regularizer of 12 units. The dropout layer is implemented in preventing the overfitting of the model after Pre-Trained layer 2. At the end, the last layer has SoftMax activation with fully connected layer. The projected pseudo code for the pre_convolution is given below and it has built in with five network model and the inputs to the network are the augmented images.

After experimentation on the previous five networks and the implementation was achieved the best accuracy to obtain by the following model.
We decided to implement the CNN architecture since it is more powerful than regular DNN. Our CNN has 3 convolutional layers and 3 Max pooling layers followed by two fully connected layers and finally the output layer.

Cross validation was used with 20% of the datasets is randomly selected as a validation set. we stacked three convolutional layers that extract 32 filters of size 3x3 with ReLU as activation function, then the output of the activation function of the is fed to a max pooling of 2x2 window in layer firstly. The same architecture is adopted in the following layers with a few changes in the number of filters, size of filters, max pooling layers. The second convolutional layer 64 filters of size 3x3 with ReLU as activation function, then the output of the activation function of the is fed to a max pooling of 2x2 window. The third convolutional layer 128 filters of size 3x3 with ReLU as activation function, then the output of the activation function of the is fed to a max pooling of 2x2 window the exact architecture is shown in Figure (3.8). After that, we added two fully connected layers that receive flatten output of the previous layer with 1024 neurons and 512 neurons respectively. Which also follows activation function (ReLU) to fourth layer, (softmax) to fifth layer.

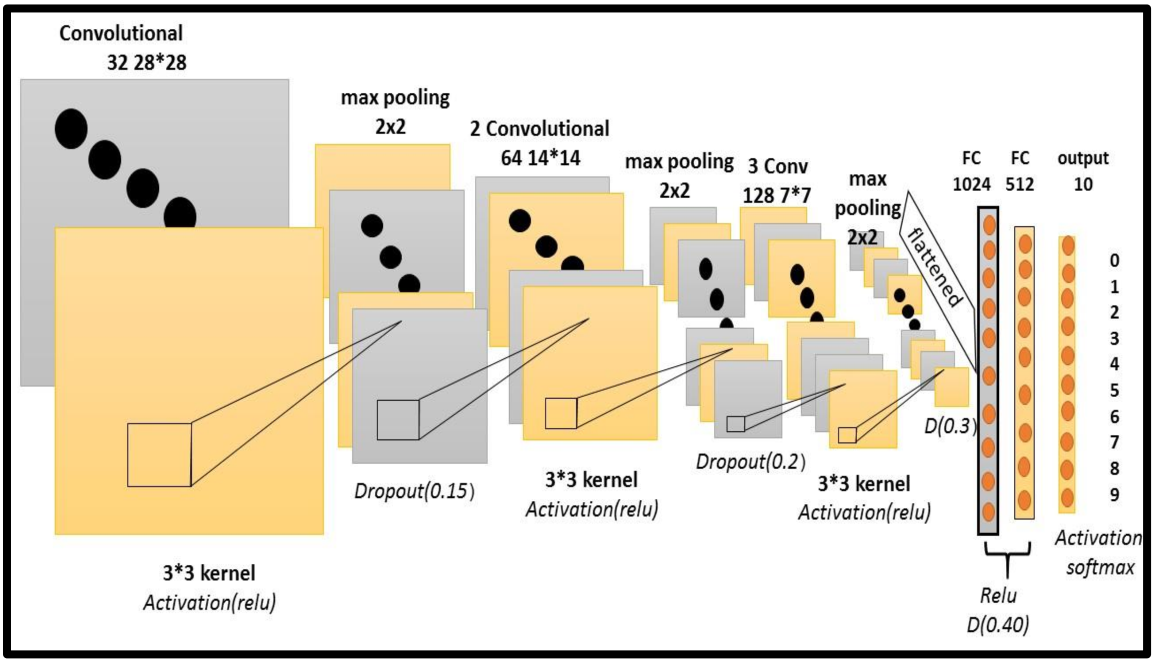


Figure (3.8) Internal Architecture of The proposed Pre-Trained CNN.

To address the problem of overfitting, we use the Dropout, that was explained in chapter two in Section 2.5.1.5, with 0.15 probability in first layer, 0.2 probability in the second layer, 0.30 probability in the third layer, and 0.4 probability in the fourth and fifth layer fully connected. To reduce overfitting, we also used Early Stopping method. With this method, the model stops learning new weights if the loss function is not reduced for a certain number of epochs. This improves the learner's performance on data outside of the training set. Finally, a 10 neurons layer with a Sigmoid activation function is added to produce the final output of the conventional neural network.

After every minibatch iteration of the training process we compute the total cross entropy error and optimize the parameters of the network so that the loss is gradually minimized. We have used the 'Adam' optimizer for optimization of the loss function

Pseudo code (3.4) The proposed CNN with pre-Training.

INPUT: Fashion-MNIST dataset
OUTPUT: classify the fashion images and they are classified as Class= {class0, class 1, class2, class 3, class4, class 5, class6, class 7, class8, class 9}
Begin
Define some values as
IM_WIDTH, IM_HEIGHT, CHANNELS =28, 28, 1
BATCH_SIZE = 512
NB_EPOCHS = 150
NB_AUGMENTATION = 3
1. Initial
Step1: Load the input data for training.
Step2: Preprocessing the input data.
2. Image Augmentation:
Step1: Generate new Augmented images.
Step2: NB_AUGMENTATION = 3
Step3: Random Rotations, Shifts, Flips, Brightness, Zoom.

Step3: Resize the Augmented images as to IM_WIDTH, IM_HEIGHT, CHANNELS = 28, 28, 1.
3.Sequential Model
INPUT: Data Augmented Image.
STEP1: Process: CONV (3×3), ReLU MaxPool (2×2), ReLU Dropout, then Conv (3×3) ReLU CONV (3×3), ReLU Repeat: Process it for 3 times
STEP 2: Flatten the Convoluted data.
STEP3: FC (1×1), ReLU Dropout, then FC (1×1) ReLU
STEP4: CONV (1×1), SoftMax.
OUTPUT: Predict and Classify the Fashion Image.
loss function (optimizers. Adam(lr=0.0001))

In this thesis, the proposed CNN with pre-Training is implemented according to the following architecture:

(1) INPUT: 28 × 28 × 1.
(2) CONV5: 3 × 3 size, 32 filters, same stride.
(3) Activation ReLU(x).
(4) MaxPooling (POOL: 2 × 2 size).
(5) DROPOUT: p = 0.15.
(6) CONV5: 3 × 3 size, 64 filters, same stride.
(7) Activation ReLU(x).
(8) MaxPooling (POOL: 2 × 2 size).
(9) DROPOUT: p = 0.2.
(10) CONV5: 3 × 3 size, 128 filters, same stride.
(11) Activation ReLU(x).
(12) MaxPooling (POOL: 2 × 2 size).
(13) DROPOUT: p = 0.3.
(14) Flatten(x).
(15) DROPOUT: p = 0.4.
(16) FC: 1024 Hidden Neurons.
(17) DROPOUT: p = 0.4.
(18) FC: 512 Hidden Neurons.
(19) DROPOUT: p = 0.2.
(20) Activation SoftMax(x).
(21) FC: 10 Output Classes.
(22) loss function (optimizers. Adam(lr=0.0001))

Table (3.1) describes the detailed report of the convolutional network model and it is clearly shows that the convoluted pixel values are in each and every stage.

Table (3.1) Summary of Convolutional Network Models Analyzed.

Layer (type)	Output Shape	Param #
input_28 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d_81 (Conv2D)	(None, 28, 28, 32)	320
activation_81 (Activation)	(None, 28, 28, 32)	0
max_pooling2d_81 (MaxPooling	(None, 14, 14, 32)	0
dropout_162 (Dropout)	(None, 14, 14, 32)	0
conv2d_82 (Conv2D)	(None, 14, 14, 64)	18496
activation_82 (Activation)	(None, 14, 14, 64)	0
max_pooling2d_82 (MaxPooling	(None, 7, 7, 64)	0
dropout_163 (Dropout)	(None, 7, 7, 64)	0
conv2d_83 (Conv2D)	(None, 7, 7, 128)	73856
activation_83 (Activation)	(None, 7, 7, 128)	0
max_pooling2d_83 (MaxPooling	(None, 3, 3, 128)	0
dropout_164 (Dropout)	(None, 3, 3, 128)	0
flatten_27 (Flatten)	(None, 1152)	0
dropout_165 (Dropout)	(None, 1152)	0
dense_81 (Dense)	(None, 1024)	1180672
dropout_166 (Dropout)	(None, 1024)	0
dense_82 (Dense)	(None, 512)	524800

CHAPTER FOUR

RESULTS AND DISCUSSION

CHAPTER 4: Results and Discussion

4.1 Introduction

The performance analysis of the proposed technique is illustrated in the next paragraph. The parameters to be considered for evaluation are the accuracy, the precision, the recall, and then the F1-score. Confusion matrix is used to calculate the various performance metrics. The datasets used to evaluate the Applied methodology is the Fashion-MNIST dataset. The performance of the model is analyzed by choosing the test data arbitrarily from the dataset as outcome data.

Labels

Each training and test example is assigned to one of the following labels:

> 0 T-shirt/top, 1 Trouser, 2 Pullover, 3 Dress, 4 Coat, 5 Sandal, 6 Shirt, 7 Sneaker, 8 Bag and 9 Ankle boot.

Accuracy

It shows correctly the classified instances percentage in course of classification. It is evaluated as:[15]

$$Accuracy\ rate = \frac{True\ Positive + True\ Negative}{Total\ Instances} * 100 \tag{4.1}$$

Precision

This measure gives the proportion of data sets in association with clothes. The predicted positives (Fashion predicted is belonged to TP) and the Fashion actually having an important factor which is considered the main role of TP. This is used to measure the performance and accuracy of the classifier as shown below:[15]

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{4.2}$$

Recall

Recall is the ratio of real positives which are correct the predicted positive and it is defined as:[15]

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad \textbf{(4.3)}$$

where

True Positive is an outcome where the model correctly predicts the positive class.
True Negative is an outcome where the model correctly predicts the negative class.
False Positive is an outcome where the model incorrectly predicts the positive class.
False Negative is an outcome where the model incorrectly predicts the negative class.

F1-Score

F1 Score is determined from the precision and recall of the test values. The F1 score is an indicator of the test's accuracy to assess the binary classification. Where the precision is the number of True Positive Outcomes, divided by the number of all the positive outcomes, the recall is the number of True Positive Outcomes, divided by the number of all the positive samples that should have been detected. It is calculated as:[15]

$$F1 - \boldsymbol{Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \textbf{(4.4)}$$

4.2 Confusion Matrix

The table frequently applied to determine a classification model's output on a collection of test data for which the true values are known as the Confusion matrix. Therefore, the estimated Performance of our classification model on the data using confusion matrix is shown in Figure (4.1).

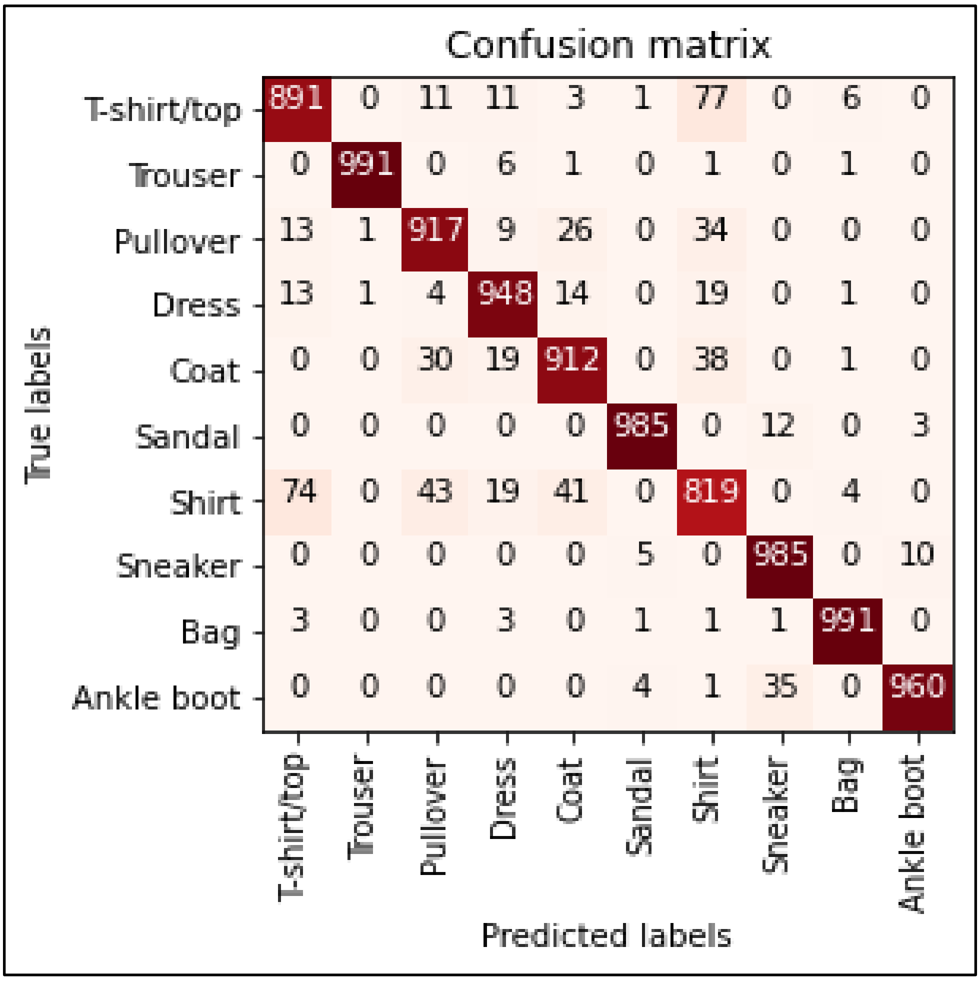


Figure (4.1) Confusion matrix for the Pre-Trained technique.

You can see that in Table (4.1), with all the scoring metrics; precision, recall and f1 score are somewhat high for the Trouser, Dress, Sandals, Sneaker, Bag and Ankle Boots classes. While that of the T-shirt/Top, Pullover, Coat, and Shirt are somewhat low compared to the rest of the classes. Among the other six classes, images of these four classes are most commonly misclassified. One potential explanation may be that 28×28 grayscale images are not really adequate for the algorithm to distinguish with very good precision between them. This is because in such small images, these groups look mostly comparable. Distinguishing features such as colour, texture are only available in RGB images, and it is likely that the algorithm will be able to exploit the knowledge and determine correctly only in coloured images between these groups.

Table (4.1) Precision, Recall and F1 Score for all classes.

Class	Class Label	Precision	Recall	F1 Score
T-Shirt/Top	0	90	89	89
Trouser	1	100	99	99
Pullover	2	91	92	91
Dress	3	93	95	94
Coat	4	91	91	91
Sandals	5	99	98	99
Shirt	6	83	82	82
Sneaker	7	95	98	97
Bag	8	99	99	99
Ankle Boots	9	99	96	97
Overall		94	94	94

4.3 Training and Validation Accuracy and Loss

Figure (4.2) clearly depicts the training and validation accuracy for the proposed Conv-technique. The accuracy increases with the Epoch exponentially and attain the maximum accuracy at 150 Epochs then it attains the saturation after that. The training and evaluation accuracy is predicted at the same number of epochs. The 'Adam' optimizer is used for loss function optimization

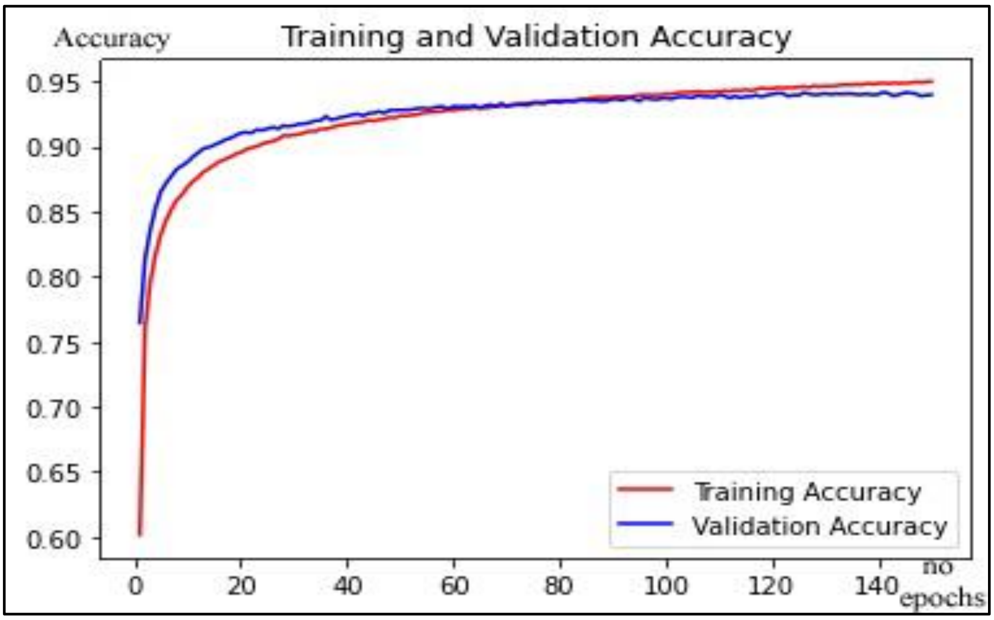


Figure (4.2) Training and Validation Accuracy for the Possible Epoch steps.

Figure (4.3) depicts the training and validation loss for the proposed Conv-technique. The accuracy is decays with the Epoch exponentially and it attains the minimum loss at 150 Epochs and it reaches the saturation after the 125 Epochs. The training and evaluation Loss is predicted at the same number of epochs.

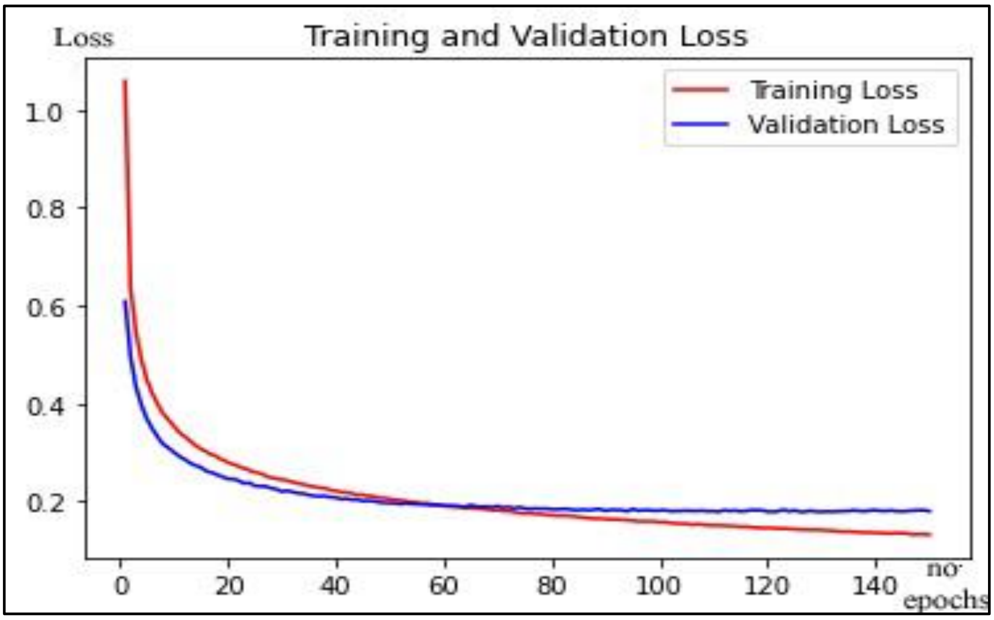


Figure (4.3) Training and Validation Loss for the Possible Epoch steps.

4.4 Correctly Predicted Classes

Figure 4.4 shows the outcomes pictorially that the projected model predicts the predicted classes and the correct classes.

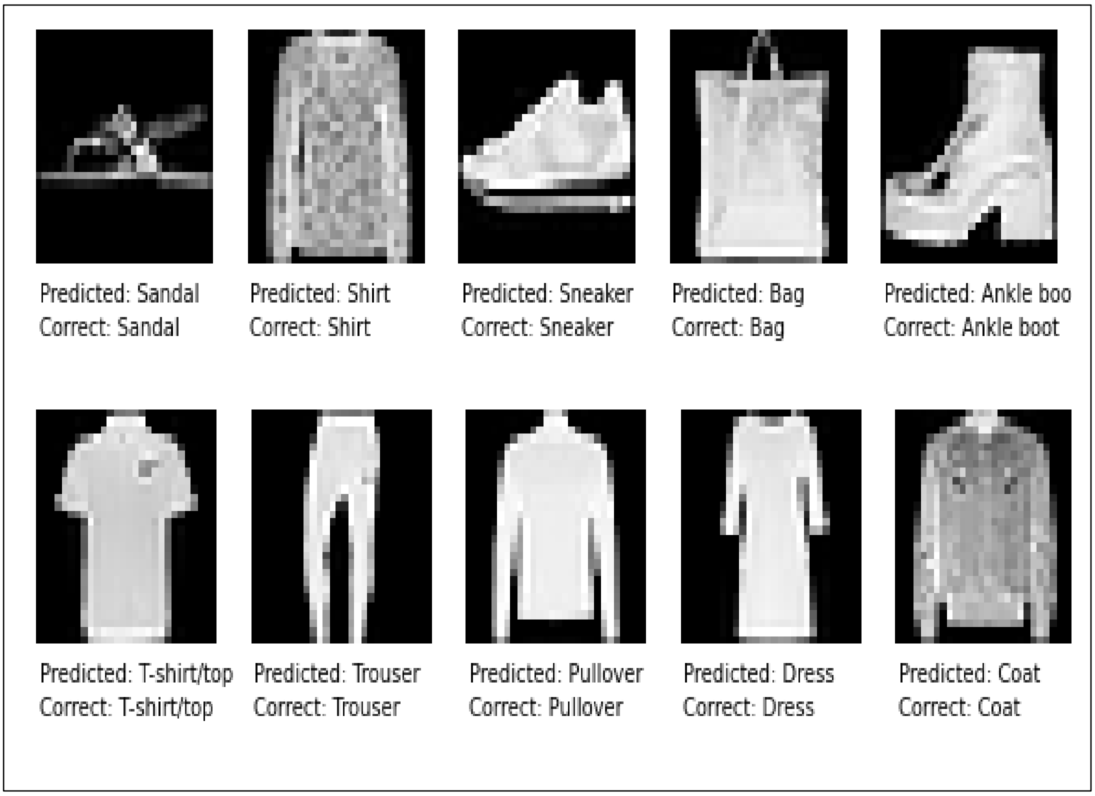


Figure (4.4) The Predicted and Correct Images of the Fashion-MNIST dataset

Our model predicted 94% of the test images correctly, which indicates that the model did pretty good job in generalizing the data. that took 25 - 30 minutes to implement of 150 epochs.

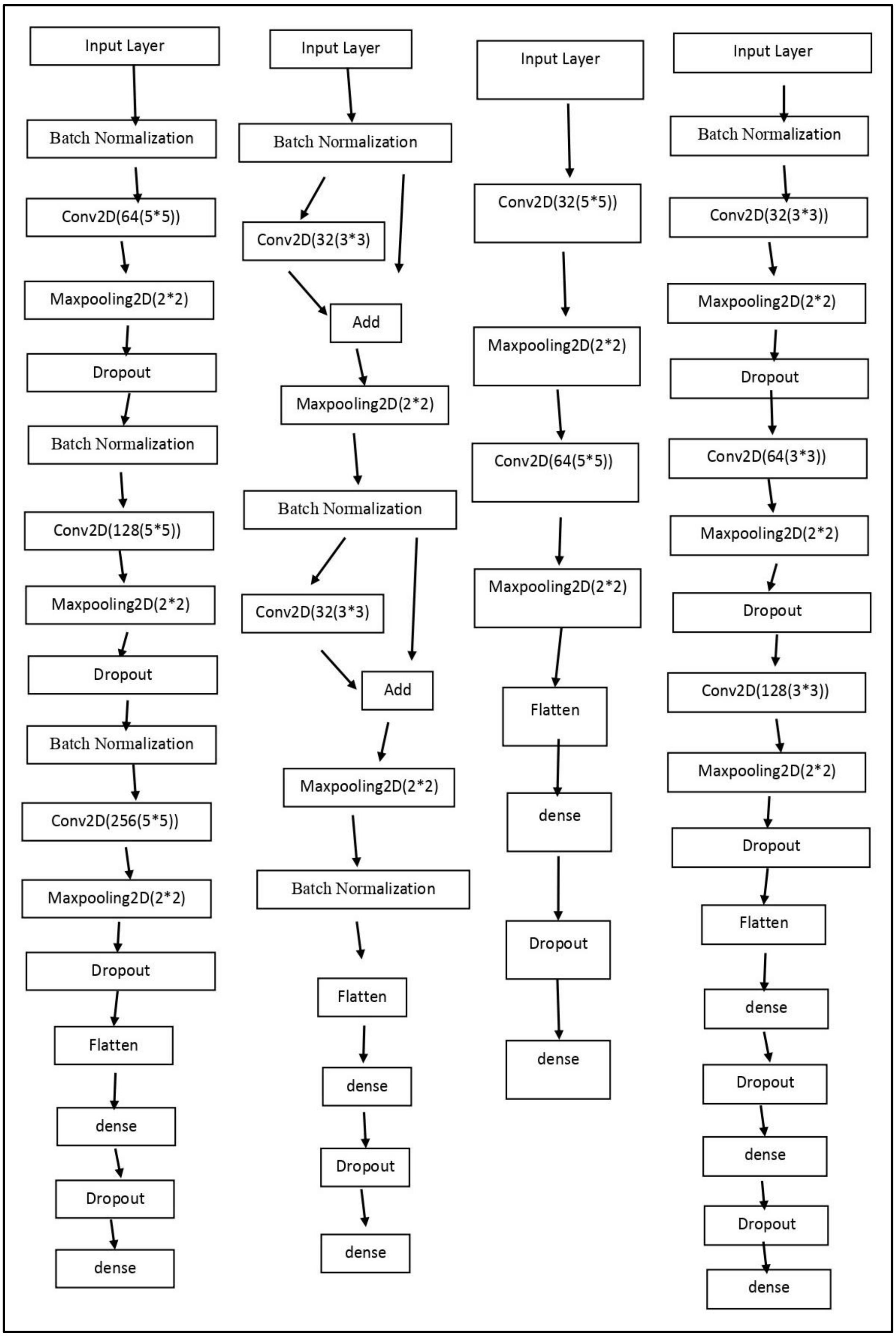
4.5 Comparing the Results

In this section, the proposed model architecture and the obtained results from applying it is compared with the closest models to the proposed model and the best results published in the literature. These models are CNN+Randomforest [14], CNN2 + BatchNorm + Skip model [15], and CNN + SVM [16]. Figure 4.5 illustrates the structure of the proposed model compared with that of the other three models. These three models are illustrated in the next paragraphs.

CNN+Randomforest: [14]To address the image classification issue on the MNIST and Fashion-MNIST datasets, five CNN architectures were used. The results obtained indicate that 89 percent precision for the MNIST dataset is provided by any CNN model. The third architecture (3 convolutional layers and 2 fully connected layers) provides the Fashion-MNIST dataset with better testing accuracy. In architecture 1, adagrad is the best optimizer algorithm observed, sigmoid is the best activation function, batch size is 64, iteration number is 50 and dropout is 0.1. In architecture 2, Adam is the best optimizer observed for the algorithm, SoftMax is the best activation function, batch size is 128, iteration number is 50, dropout is 0.25 and kernel size is 2×2. Architecture 1 has only one input layer and two completely linked layers, so training takes less time. With the rise in convolutional layers, the training time increases. Training time is substantially increased with a filter size of 3×3. The images data has been classified by means Random Classifier attains the maximum Accuracy of about 84.4%.

CNN2 + BatchNorm + Skip model:[15] With a feasible classification error, the design had a few parameter numbers and low computational costs. In terms of some dense blocks with skip links, it included densely linked layers to maintain the knowledge flow in a deep CNN model. The Fashion-MNIST is classified by means of higher Deep Learning Architecture by CNN2 with Batch Normalization, the model had 26 times fewer parameter numbers with an 8% better classification error than the first generalized CNN model. It is worth noting that this smallest model had the same accuracy level compared to the 92.54 percent much higher than the smaller model size of the AlexNet. The results showed that there are unacceptable justifications for some levels of complexity in CNN models, especially in early CNN models.

CNN + SVM:[16] It presents an important method for the efficient and precise classification and identification of photographs of fashion items. Upon successful implementation of the classification system for fashion articles using CNN feature space and multi-class SVM classifier. It has shown that this system provides relatively good fashion object classification efficiency compared with the available literature works. It has a Classified Accuracy of about 90.72%.



CNN+Randomforest | CNN2+ BatchNorm + Skip | CNN +SVM | Proposed Pre-Trained

Figure (4.5) Model Architectures.

Table (4.2) shows some of the observation of Fashion-MNIST Dataset, the outcome of the classifier has been estimated from the Fashion-MNIST datasets instances. Then, classifying the instances with the same observation is performed. Next, the performance measures of the various techniques of CNN+Randomforest, CNN2 + BatchNorm + Skip model, and Conv+SVM are calculated. Finally, these results are compared with that of the Pre-Trained technique. Table (4.2) shows the comparison of the performance in terms of precision, recall and F1-score. It has been analyzed from the actual and predicted values from the objective of three classes class 0, class 1 and class 2 in confusion matrix and it is represented in percentage.

Table (4.2) Performance for class 0, 1 and 2.

CLASS 0-T-shirt/TOP					
S.NO	Techniques	CNN+Randomforest [14](%)	CNN2 + Batch Norm + Skip model [15](%)	CNN +SVM[16] (%)	Proposed Pre-Trained (%)
1	Precision	87	88	83	90
2	Recall	86	86	82	89
3	F1-score	85	87	81	89
CLASS 1-Trouser					
1	Precision	85	99	91	100
2	Recall	89	98	89	99
3	F1-score	88	99	87	99
CLASS 2-Pullover					
1	Precision	87	90	85	91
2	Recall	86	85	87	92
3	F1-score	85	88	81	91

Figure (4.6) is the graphical representation of the Table (4.2) contents. It shows the precision comparison for Fashion-MNIST dataset among the existing and the proposed techniques. As shown in this figure, the Pre-Trained achieves the maximum precision percentage in the three classes 0, 1 and 2 compared with the other techniques. Whereas, the Conv+SVM approach obtained the worst performance by furnishing the precision value of about 83% for class 0, 91% for class 1 and 85% for class2. Simultaneously, the CNN2 + BatchNorm + Skip model acquires more precision compared to the previous one of about 88% for class 0, 99% for class 1 and 90% for class 2. The CNN+Randomforest precision value is gradually decreases to about 87% for class 0, 85% for class 1 and 87% for class 2. Finally, the proposed Pre-Trained technique operates more efficiently compared with the other models performance by acquiring the maximum precision value of 90% for class 0, 100% for class 1 and 91% for class 2.

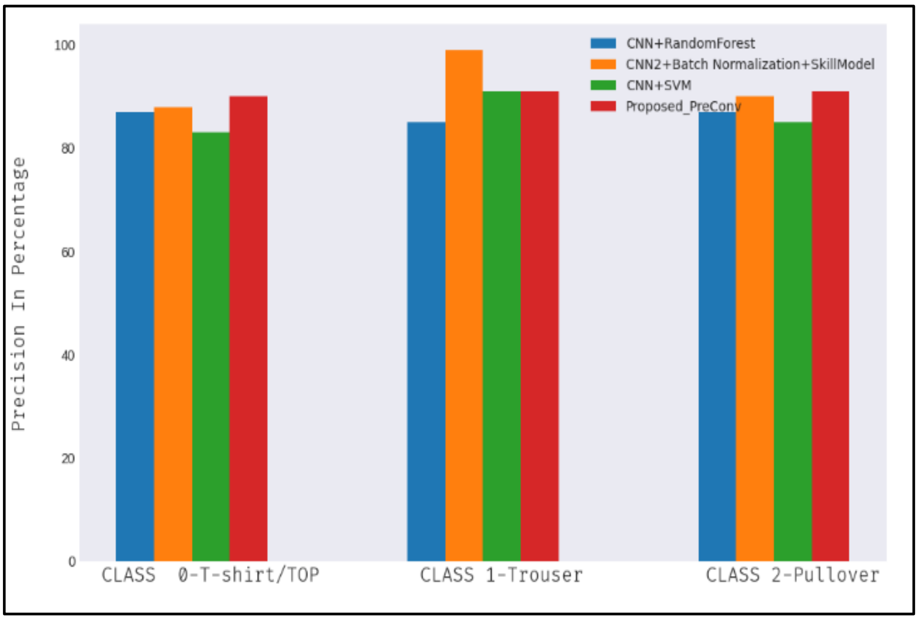


Figure (4.6) The precision for class 0, 1 and 2.

The graphical representation for Table (4.2) in Figure (4.7) highlights the contrast in terms of recall with different approaches. It is a recall contrast between proposed and existing techniques based on the Fashion-MNIST dataset. The proposed Pre-Trained achieves the highest recall percentage in the three classes 0, 1 and 2 compared with that of the other techniques. In comparison, the Conv+SVM technique resulted in the worst results by offering the minimum recall values of approximately 82% for class 0,89% for class 1 and 87% for class 2. Moreover, higher recall values are obtained from the CNN+Randomforest model compared to the previous one. it is of about 86% for class 0, 89% for class 1 and 86% for class 2. While, the parameters of the CNN2 + BatchNorm + Skip model is steadily increase the recall values to be about 86% for class 0, 98% for class 1, and 85% for class 2 compared to the other techniques. Finally, compared to other techniques, the proposed Pre-Trained system performs more effectively by obtaining the maximum recall values of 89% for class 0, 99% for class 1 and 92% for class 2.

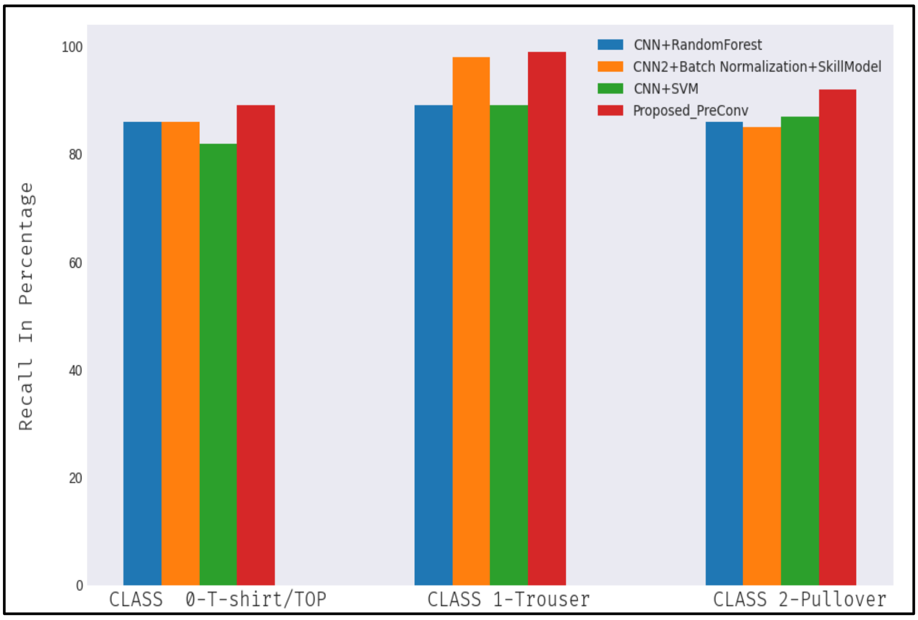


Figure (4.7) The Recall for class 0, 1 and 2.

The graphical representation for Table (4.2) in Figure (4.8) is the F1-Score values. It is an F1-Score comparison between the existing and the proposed techniques for the Fashion-MNIST dataset. The proposed Pre-Trained achieves the highest F1-Score percentage in the three classes 0, 1 and 2 compared with that of the existing techniques. In which, the Conv+SVM approach resulted in the worse performance by providing about 81% for class 0, 87% for class 1 and 81% for class 2. It also clear that the CNN2 + BatchNorm + Skip model obtains higher F1-Score percentage than the previous one; about 87% for class 0, 99% for class 1 and 88% for class 2. The CNN+Randomforest F1-Score values are gradually increased to about 85% for class 0, 88% for class 1 and 85% for class 2 compared to the other existing methods. Finally, compared to the existing methods, the proposed Pre-

Trained technique works more effectively by acquiring the maximum F1-Score values of 89% for class 0, 99% for class 1 and 91% for class 2.

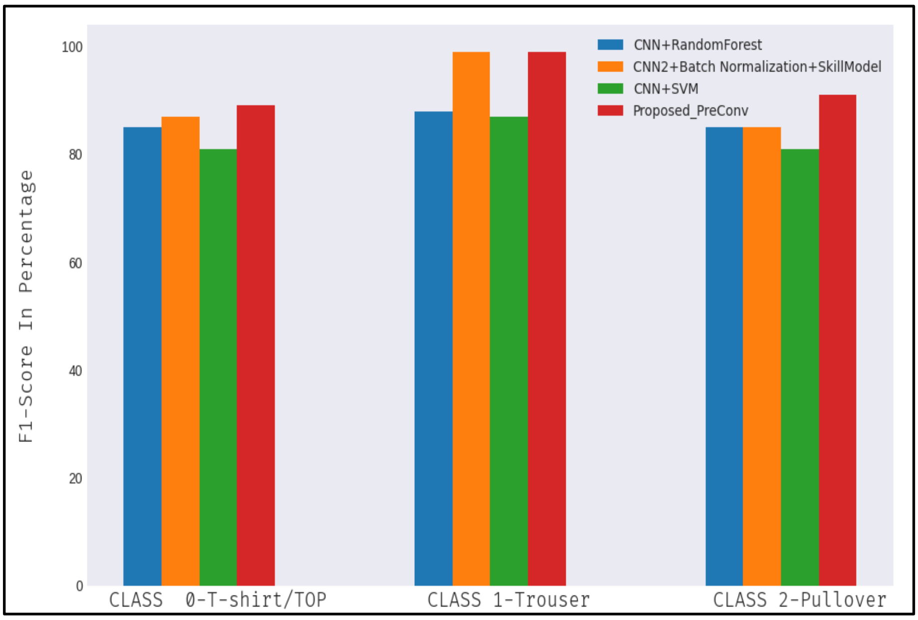


Figure (4.8) F1-Score for class 0, 1 and 2.

The effect of the classifier which was determined from the instances of the Fashion-MNIST datasets in Table (4.3). The instances are classified with the same observation, and then the efficiency measurements of CNN+Randomforest techniques, CNN2 + BatchNorm + Skip model, and Conv+SVM are compared with that of the proposed Pre-Conv technique. In terms of the precision, recall and F1-score, the output relation is shown in Table (4.3). The objective in the confusion matrix of three classes; class 3, class 4 and class 5, were evaluated from the actual and expected value and are expressed in percentage.

Table (4.3) performance for class 3, 4 and 5.

S.NO	Techniques	CNN+Random forest [14] (%)	CNN2 + Batch Norm + Skip model [15](%)	CNN +SVM [16] (%)	Proposed Pre-Trained (%)
CLASS 3-Dress					
1	Precision	87	92	87	93
2	Recall	83	93	89	95
3	F1-score	85	92	97	94
CLASS 4-Coat					
1	Precision	89	87	85	91
2	Recall	87	88	83	91
3	F1-score	85	88	81	91
CLASS 5-Sandal					
1	Precision	85	99	91	99
2	Recall	84	99	87	98
3	F1-score	85	99	88	99

Figure (4.9) is the graphical representation of Table (4.3). It shows the comparative analysis of the precision of the Pre-Trained for Fashion-MNIST dataset compared with the existing, Conv+SVM, CNN2 + BatchNorm + Skip model, and CNN+Randomforest techniques. As shown in Figure (4.9) the precision percentage of the Pre-Trained is clearly increased percentage in the three classes 3, 4 and 5 compared with other techniques. In which, the Conv+SVM approach resulted in worst values by providing about 87% in class 3, 85% in class 4, and 91% in class 5. In addition, the CNN2 + BatchNorm + Skip model acquires higher values compared with the previous one which are 92% in class 3, 87% in class 4, and 99%

in class 5. Moreover, the CNN+Randomforest model obtained 87% in class 3, 89% in class 4 and 85% in class 5. According to these results, the proposed Pre-Trained technique performs more effectively than the other models, by acquiring the full precision values of 93% for class 3, 91% for class 4 and 99% for class 5.

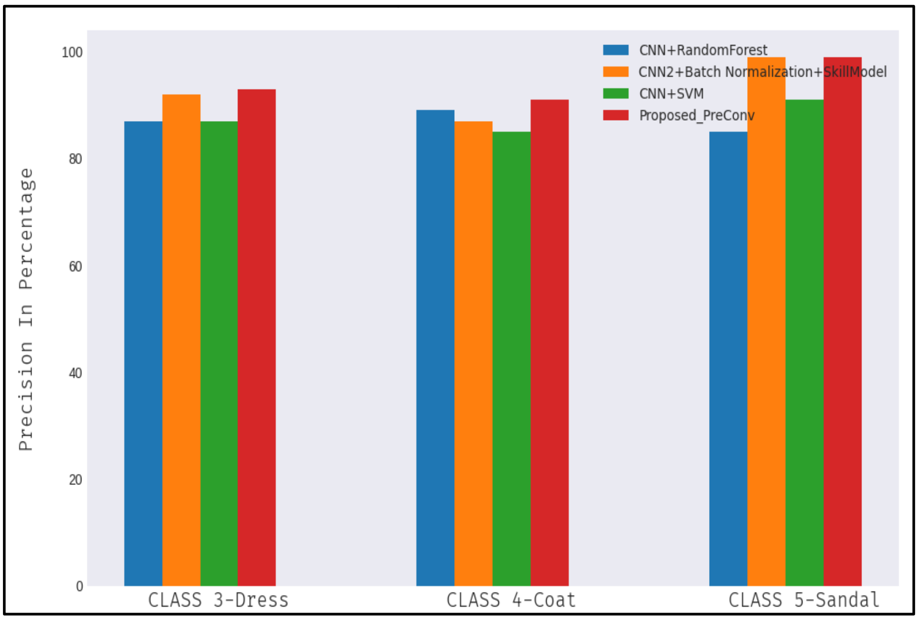


Figure (4.9) The precision for class 3, 4 and 5.

Figure (4.10) is the graphical representation of Table (4.3). It shows the comparative analysis of the recall of the Pre-Trained technique compared with those of the three existing techniques. In which, the recall values of the Pre-Trained in the three classes 3, 4 and 5 are higher than the other techniques. In addition, the Conv+SVM technique resulted in the worse results by offering a minimum recall values of roughly 89% for class 3, 83% for class 4 and 87% for class 5 compared with the other techniques. Moreover, the CNN+Randomforest model obtains more recall values compared to the preceding one of around 83% for class 3, 87% for class 4 and 84% for class 5. The CNN2 + BatchNorm + Skip recall values are steadily raises to be about 93% for class 3, 88% for class 4, and 99% for class 5 compared with the other techniques. Finally, by acquiring the full recall values of 95% for class 3, 91% for class 4 and 98% for class 5, of the Proposed Pre-Trained technique and it performs better than the other techniques.

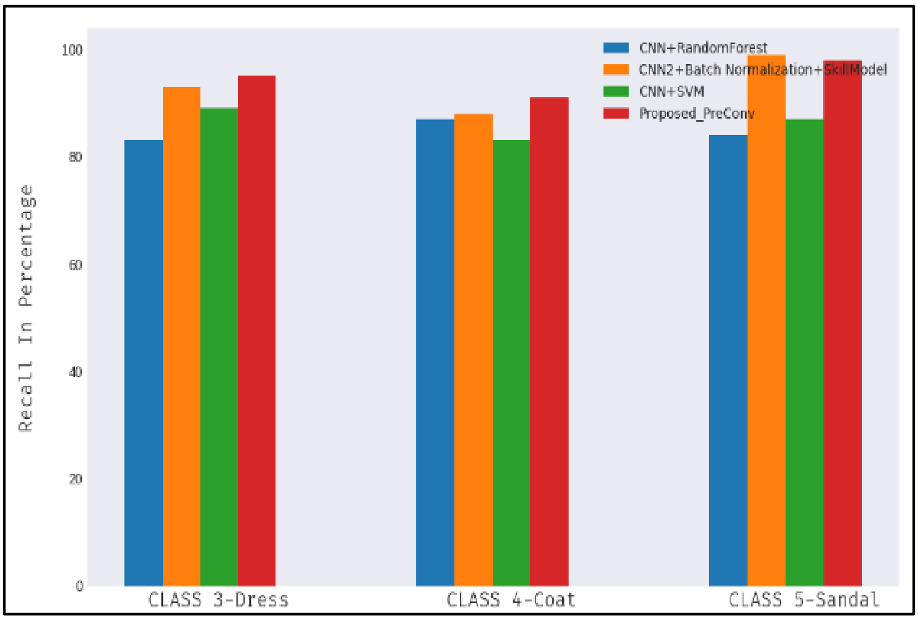


Figure (4.10) The Recall for class 3, 4 and 5.

Figure (4.11) illustrate the graphical representation of Table (4.3). It shows the comparative analysis of the F1-score of the Pre-Trained technique compared with those of the three existing techniques. In which, the F1-score values of the Pre-Trained in the three classes 3, 4 and 5 are higher than the other techniques. In addition, the Conv+SVM technique resulted in the worse results by offering a minimum F1-score values of roughly 97% for class 3, 81% for class 4 and 88% for class 5 compared with the other techniques. Moreover, the CNN+Randomforest model obtains more F1-score values compared to the preceding one of around 85% for class 3, 85% for class 4 and 85% for class 5. The CNN2 + BatchNorm + Skip F1-score values are steadily raises to be about 92% for class 3, 88% for class 4, and 99% for class 5 compared with the other techniques. Finally, by acquiring the full F1-score values of 94% for class 3, 91% for class 4 and 99% for class 5, of the Proposed Pre-Trained technique, it performs better than the other techniques.

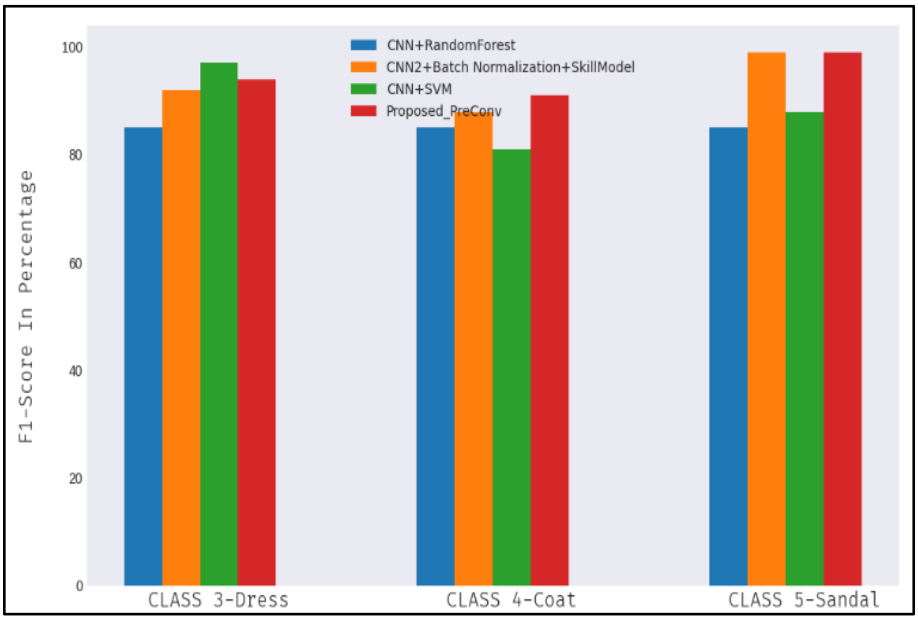


Figure (4.11) F1-Score for class 3, 4 and 5.

The effect of the classifier which was determined from the instances of the Fashion-MNIST datasets is illustrated in Table (4.4). The instances are classified with the same observation and then the efficiency measurements of CNN+Randomforest, CNN2 + BatchNorm + Skip, and Conv+SVM are compared with that of the proposed Pre-Conv technique. In terms of the precision, recall and F1-score, the output relation is shown in Table (4.4). The objective in the confusion matrix of four classes; class 6, class 7, class 8 and class 9, were evaluated from the actual and expected value and are expressed in percentage.

Table (4.4) Performance for class 6, 7, 8 and 9.

CLASS 6-shirt					
S.NO	**Techniques**	**CNN+Random forest** [14] **(%)**	**CNN2 + Batch Norm + Skip model** [15]**(%)**	**CNN +SVM**[16] **(%)**	**Proposed Pre-Trained (%)**
1	**Precision**	81	75	77	83
2	**Recall**	80	80	75	82
3	**F1-score**	78	77	73	82
CLASS 7-Sneaker					
1	**Precision**	83	96	87	95
2	**Recall**	85	97	91	98
3	**F1-score**	84	97	90	97
CLASS 8-Bag					
1	**Precision**	87	99	93	99
2	**Recall**	83	99	89	99
3	**F1-score**	81	99	87	99
CLASS 9-Ankle Boot					
1	**Precision**	87	92	93	99
2	**Recall**	83	97	89	96
3	**F1-score**	85	97	87	97

Figure (4.12) is the graphical representation of Table (4.4). It shows the comparative analysis of the precision of the Pre-Trained for Fashion-MNIST dataset compared with the existing, Conv+SVM, CNN2 + BatchNorm + Skip model, and CNN+Randomforest techniques. As shown in Figure (4.12) the precision percentage of the Pre-Trained is clearly increased percentage in the four classes 6, 7, 8 and 9 compared with other techniques. In which, the Conv+SVM approach resulted in worst values by providing about 77% in class 6, 87% in class 7, 93% in class 8 and 93% in class 9. In addition, the CNN2 + BatchNorm + Skip model acquires higher values compared with the previous one which are 75% in class 6, 96% in class 7, 99% in class 8 and 92% in class 9. Moreover, the CNN+Randomforest model obtained 81% in class 6, 83% in class 7, 87% in class 8 and 87% in class 9. According to these results, the proposed Pre-Trained technique performs more effectively than the other models, by acquiring the full precision values of 83% for Class 6, 95% for Class 7,99% for Class 8 and 99% for Class 9.

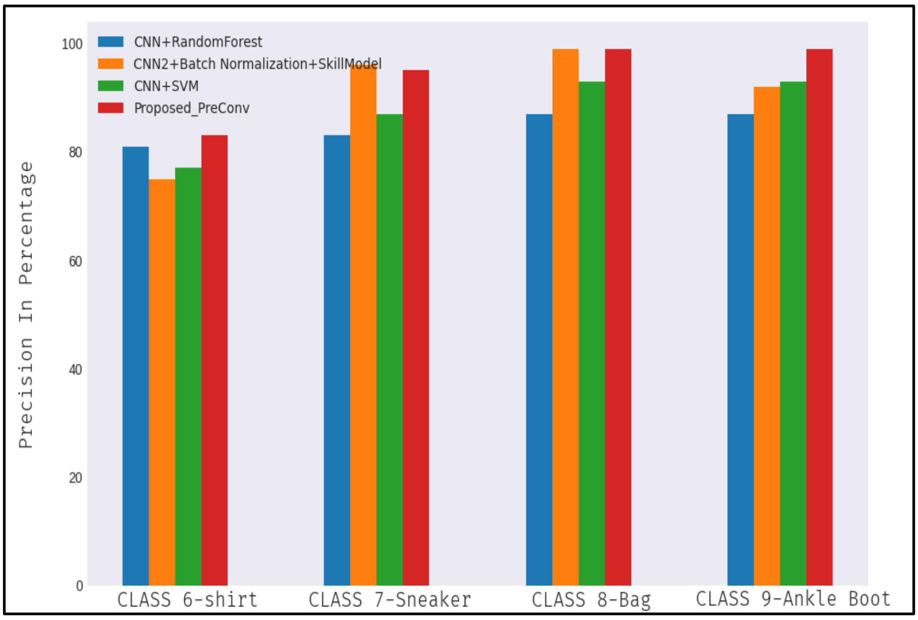


Figure (4.12) The precision for class 6, 7, 8 and 9.

Figure (4.13) is the graphical representation of Table (4.4). It shows the comparative analysis of the recall of the Pre-Trained technique compared with those of the three existing techniques. In which, the recall values of the Pre-Trained in the four classes 6, 7,8 and 9 are higher than the other techniques. In addition, the Conv+SVM technique resulted in the worse results by offering a minimum recall values of roughly 75% for class 6, 91% for class 7, 89% for class 8 and 89% for class 9 compared with the other techniques. Moreover, the CNN+Randomforest model obtains more recall values compared to the preceding one of around 80% for class 6, 85% for class 7, 83% for class 8 and 83% for class 9. The CNN2 +

BatchNorm + Skip recall values are steadily raises to be 80% for Class 6, 97% for Class 7, 99% for Class 8 and 97% for Class 9 compared with the other techniques. Finally, by acquiring the full recall values of 82% for Class 6, 98% for Class 7,99% for Class 8 and 96% for Class 9. of the Proposed Pre-Trained technique and it performs better than the other techniques.

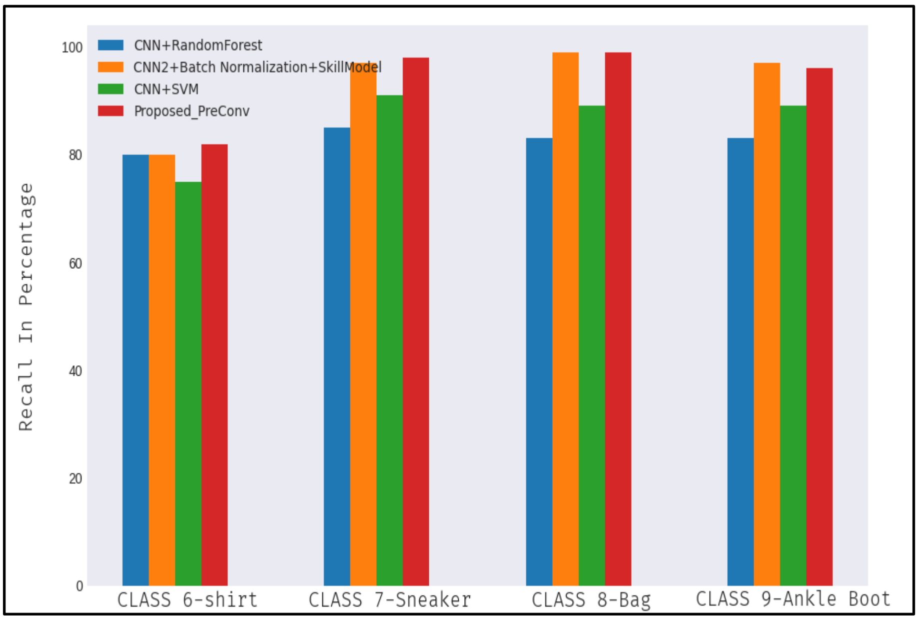


Figure (4.13) The Recall for class 6, 7, 8 and 9.

Figure (4.14) is the graphical representation of Table (4.4). It shows the comparative analysis of the F1-score of the Pre-Trained technique compared with those of the three existing techniques. In which, the F1-score values of the Pre-Trained in the fou rclasses 6, 7,8 and 9 are higher than the other techniques. In addition, the Conv+SVM technique resulted in the worse results by offering a minimum F1-score values of roughly 73% for Class 6, 90% for Class 7, 87% for Class 8 and 87% for Class 9 compared with the other techniques. Moreover, the CNN+Randomforest model obtains more F1-score values compared to the preceding one of around 78% for Class 6, 84% for Class 7, 81% for Class 8 and 85% for Class 9. The CNN2 + BatchNorm + Skip F1-score values are steadily raises to be about 77% for Class 6, 97% for Class 7, 99% for Class 8 and 97% for Class 9 compared with the other techniques. Finally, by acquiring the full F1-score values of 82% for Class 6, 97% for Class 7,99% for Class 8 and 97% for Class 9. of the Proposed Pre-Trained technique and it performs better than the other techniques.

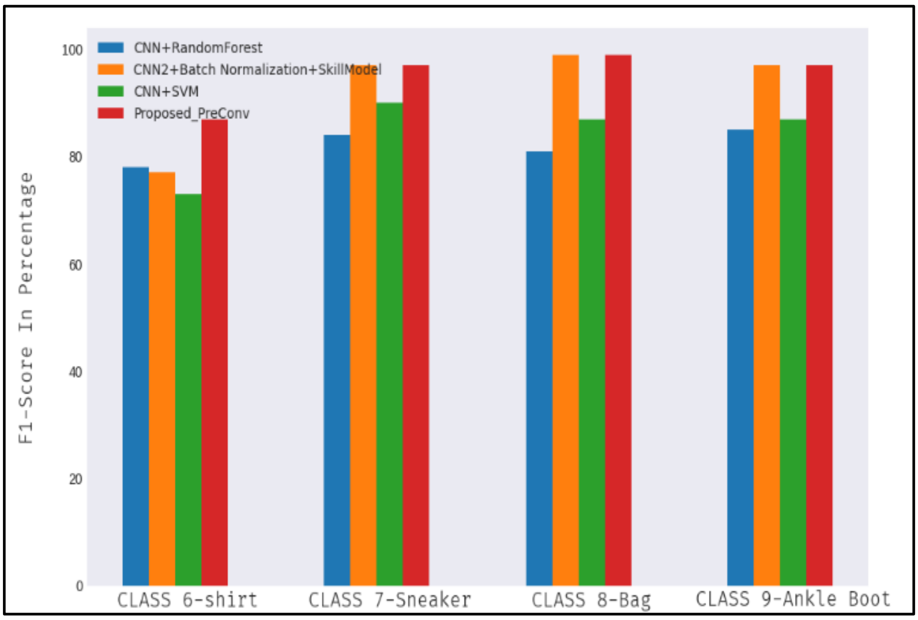


Figure (4.14) F1-Score for class 6, 7, 8 and 9.

Figure (4.15) is the graphical representation of Table (4.5). It shows the comparative analysis of the accuracy of the Pre-Trained technique compared with those of the existing techniques. In which, the accuracy values of the Pre-Trained in all the ten classes 0,1,2,3,4,5,6,7,8 and 9 are higher than the other techniques. This illustrates the efficiency contrast of accuracy. The objective of the ten classes Class 0, Class 1, Class 2, Class 3, Class 4, Class 5, Class 6, Class 7, Class 8 and Class 9 in the confusion matrix was evaluated from the real and projected value and it is expressed in percentage. As shown in Figure (4.15), the Pre-Trained achieves accuracy with maximum percentage in the ten classes 0 to 9 than existing techniques. While, the CNN2+BN+Skill approach resulted in accuracy value of about 92.54% for Class 0 to Class 9. Simultaneously, the CNN+RandomnForest model acquires less accuracy score compared to the previous one of about 84.4% for Class 0 to Class 9. Whereas, CNN+SVM parameters are gradually increase the accuracy value of about 90.72% for Class 0 to Class 9 compared with the other techniques. Finally, the proposed Pre-Trained technique operates more efficiently compared with the other models by acquiring the maximum accuracy value of 94.2% for all classes.

Table (4.5) Comparison of Overall Accuracy.

Accuracy				
Techniques	**CNN+Random forest** [14] **(%)**	**CNN2 + Batch Norm + Skip model** [15]**(%)**	**CNN +SVM** [16] **(%)**	**Proposed Pre-Trained (%)**
Accuracy	84.4%	92.54%	90.72%	94.2%

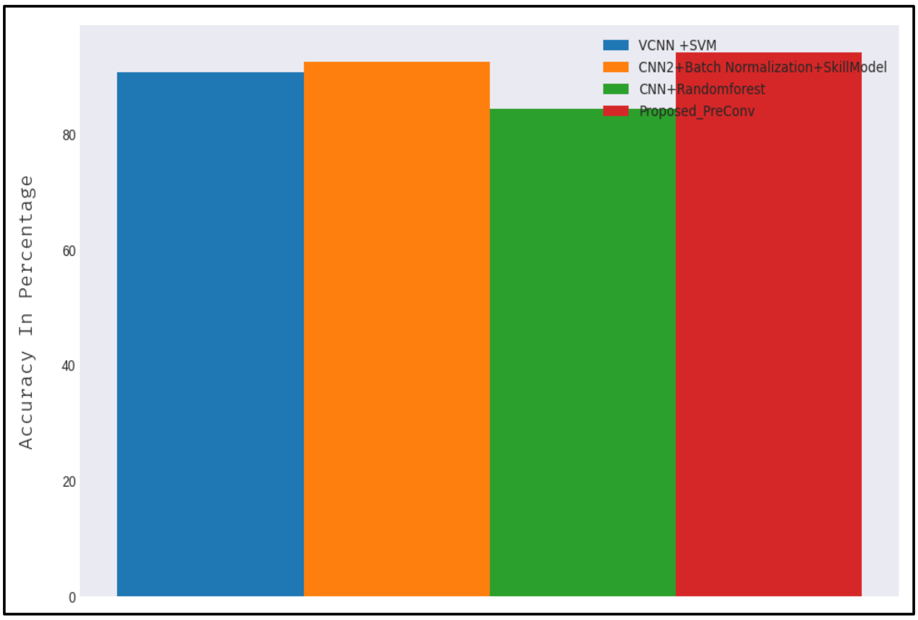


Figure (4.15) Comparison of overall accuracy.

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

CHAPTER 5: Conclusions and Future Works

5.1 Conclusions

- In this thesis we have explored the classification of a dataset with a Proposed CNN with Pre-Trained technique, which achieve accuracy of 94% and it is compared with the existing techniques.
- The images have been taken from the Fashion-MNIST dataset that contains grayscale images of fashion stuffs of 10 categories. These images are analyzed from each category entities.
- Fashion-MNIST dataset is augmented where one image will be in three forms of output; i.e. output image is rotated, shifted and zoom as an output. Finally, the data is sent to the proposed model.
- In this thesis, it is proposed that the augmented data with the pre-Trained architecture train parameters achieved the highest accuracy compared with the other traditional data.
- This augumented data will contribute as input images to the Pre-Trained network training model (basic CNN). The output image leads to improve the obtained accuracy compared with existing works.
- The Thesis contributed Pre-Trained apparel class structure to the apparel images classification process. The proposed technique is the Pre-Trained Networks. This pre-Trained framework is constructed and it comprises of three convolutional layers that extract 32, 64, 128 filters of size 3x3 with ReLU as activation function to each layer and SoftMax in last layer , then the output of the activation function of the is fed to a max pooling of 2x2 window, then Dropout To address the problem of overfitting, finally used the ’Adam’ optimizer for optimization of the loss function.
- The proposed approach was explained and used to classify apparel from given images using the CNNs that use Pre-Trained with batch normalization and dropout that have proved remarkably well performed in the field of image classification. CNN image recognition is widely used in the development of deep learning technology in fashion areas such as human identification, apparel detection, clothing recovery and automatic clothing labelling. Pre-Trained was not generally included in the phase of image classification.

- The obtained results proves that the exploitation of Pre-Trained networks achieves improved than the base model with lower loss and greater accuracy. This leads us to say that some of the challenges in classifying clothing images can be overcomed by representing preconfigured structure on CNN.
- The proposed approach contributes is not limited to the model alone, but instead of an embedded model of expertise that is trained data sets to produce a multi-class classification label levels that convey pre-convolved information. In this thesis, the pre-Trained structure was described in heuristic system, which is the only limit in this work. The drawback of this research is that we are only used the heuristic technique, but we can extend for ablation analysis. Another subject of research might be what technique to utilize to define the pre-Trained function.

5.2 Future Works

There are several aspects were explained in this thesis which worth further enhancements in the future. Our suggestions to improve this work are:

- Experiments with increasing the depth of the auto encoder and the use of pre-trained model that can improve the performance of all the models.
- Other feature extracted method is commonly used in the deep learning is the transfer learning or CNN model.
- Further research is to use the Fashion-MNIST public dataset. However, the quality of the dataset is very different from the real online clothing photos. The public dataset has already been extensively analyzed, which means that no new inferences can be drawn from this dataset.
- The estimated outcomes of the proposed techniques can be enhanced by implementing some rule between CNNs or CNN with other methodologies for extracting features, as multiple CNNs and CNN's ensemble with other handcrafted methodologies. The classification accuracy can be enhanced by incorporating with the VGG-16 with Pre-Trained Networks and VGG-19 with Pre-Trained Networks models together by applying some rule for better classification accuracy.

- Since CNN was extensively used for non-handcrafted extraction of features, we only consider CNN for our classifier and introduce pre-Trained approach on CNN only. Then further expand in the future study is to implement the CNN's convolutional ensemble model with other methods of extracting features.
- Finally, it could be further incorporate data-driven method into other experiments for future analysis. The proposed model had been determined in reverse way of classification with the higher level of classes first and then the finest level and then determine the classification of accuracy. Therefore; the futuristic study could be the determination of the classification in a heuristic way or data-driven techniques to define the convolutional structure of datas.

References

References

[1] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to sequence learning with neural networks," *arXiv Prepr. arXiv1409.3215*, 2014.

[2] C. Clark and A. Storkey, "Training deep convolutional neural networks to play go," in *International conference on machine learning*, 2015, pp. 1766–1774.

[3] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: challenges, issues, and opportunities," in *International conference on database systems for advanced applications*, 2013, pp. 1–15.

[4] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mob. networks Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[5] L. Morelli, P. M. Granitto, and G. L. Grinblat, "Deep Architectures on Drifting Concepts: A Simple Approach," 2013.

[6] N. Jmour, S. Zayen, and A. Abdelkrim, "Convolutional neural networks for image classification," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 2018, pp. 397–402.

[7] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, 2015.

[8] G. N. Shao, H. Kim, and S. M. Imran, "https://www. sciencedirect. com/science/article/abs/pii/S092633731500346X," 2016.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818–833.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[12] K. S. Younis and A. A. Alkhateeb, "A new implementation of deep neural networks for optical character recognition and face recognition," *Proc. new trends Inf. Technol. Jordan*, pp. 157–162, 2017.

[13] C. Szegedy *et al.*, "Going deeper with convolutions," in *Pr*

oceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[14] C. Geier, “Training on test data: Removing near duplicates in Fashion-MNIST,” *arXiv Prepr. arXiv1906.08255*, 2019.

[15] S. Bhatnagar, D. Ghosal, and M. H. Kolekar, “Classification of fashion article images using convolutional neural networks,” in *2017 Fourth International Conference on Image Information Processing (ICIIP)*, 2017, pp. 1–6.

[16] A. F. Agarap, “An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification,” *arXiv Prepr. arXiv1712.03541*, 2017.

[17] P. Le Callet, C. Viard-Gaudin, and D. Barba, “A convolutional neural network approach for objective video quality assessment,” *IEEE Trans. neural networks*, vol. 17, no. 5, pp. 1316–1327, 2006.

[18] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv Prepr. arXiv1708.07747*, 2017.

[19] E. Kussul, T. Baidyk, and D. C. Wunsch, *Neural networks and micromechanics*. Springer, 2010.

[20] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving deep convolutional neural networks for image classification,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, 2019.

[21] F. Manessi, A. Rozza, S. Bianco, P. Napoletano, and R. Schettini, “Automated pruning for deep neural network compression,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 657–664.

[22] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, “Advanced deep-learning techniques for salient and category-specific object detection: a survey,” *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, 2018.

[23] J. J. Dibner, “Review of the metabolism of 2-hydroxy-4-(methylthio) butanoic acid,” *Worlds. Poult. Sci. J.*, vol. 59, no. 1, pp. 99–110, 2003.

[24] J. M. Raser and E. K. O’shea, “Noise in gene expression: origins, consequences, and control,” *Science (80-.).*, vol. 309, no. 5743, pp. 2010–2013, 2005.

[25] M. Decter, D. Bennett, and M. Leseure, “University to business technology transfer—UK and USA comparisons,” *Technovation*, vol. 27, no. 3, pp. 145–155, 2007.

[26] J. B. Powers and P. P. McDougall, “University start-up formation and technology licensing with firms that go public: a resource-based view of academic entrepreneurship,” *J. Bus. Ventur.*, vol. 20, no. 3, pp. 291–311, 2005.

[27] A. N. Link and D. S. Siegel, “University-based technology initiatives: Quantitative and qualitative evidence,” *Res. Policy*, vol. 34, no. 3, pp. 253–257, 2005.

[28] L. A. Hall and S. Bagchi-Sen, “An analysis of firm-level innovation strategies in the US biotechnology industry,” *Technovation*, vol. 27, no. 1–2, pp. 4–14, 2007.

[29] Y. Wang, B. Li, R. Luo, Y. Chen, N. Xu, and H. Yang, “Energy efficient neural networks for big data analytics,” in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2014, pp. 1–2.

[30] A. Fernández, S. del Río, N. V Chawla, and F. Herrera, “An insight into imbalanced big data classification: outcomes and challenges,” *Complex Intell. Syst.*, vol. 3, no. 2, pp. 105–120, 2017.

[31] X.-W. Chen and X. Lin, “Big data deep learning: challenges and perspectives,” *IEEE access*, vol. 2, pp. 514–525, 2014.

[32] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, “Representational learning with ELMs for big data,” 2013.

[33] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[34] R. Vargas, A. Mosavi, and R. Ruiz, “Deep learning: a review,” *Adv. Intell. Syst. Comput.*, 2017.

[35] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Comput. Intell. Neurosci.*, vol. 2018, 2018.

[36] S. Khan and T. Yairi, “A review on the application of deep learning in system health management,” *Mech. Syst. Signal Process.*, vol. 107, pp. 241–265, 2018.

[37] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[38] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3d data: A survey," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–38, 2017.

[39] "Convolutional Neural Networks | Machine Learning Tutorial." https://sci2lab.github.io/ml_tutorial/cnn/ (accessed Apr. 05, 2021).

[40] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, 2017.

[41] Y. Bengio, A. C. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," *CoRR, abs/1206.5538*, vol. 1, p. 2012, 2012.

[42] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. big data*, vol. 2, no. 1, pp. 1–21, 2015.

[43] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, 2018.

[44] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[46] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[48] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE conference on computer vision and pattern recognition*, 2012, pp. 3642–3649.

[49] "CNN Architectures — LeNet, AlexNet, VGG, GoogLeNet and ResNet | by Prabhu | Medium." https://medium.com/@RaghavPrabhu/cnn-architectures-

lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848 (accessed Apr. 05, 2021).

[50] E. Dufourq and B. A. Bassett, "Eden: Evolutionary deep networks for efficient machine learning," in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 2017, pp. 110–115.

[51] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1, no. 2. MIT press Cambridge, 2016.

[52] K. V Greeshma and K. Sreekumar, "Fashion-MNIST classification based on HOG feature descriptor using SVM," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, pp. 960–962, 2019.

[53] A. Iliukovich-Strakovskaia, A. Dral, and E. Dral, "Using pre-trained models for fine-grained image classification in fashion field," in *Proceedings of the First International Workshop on Fashion and KDD, KDD*, 2016, pp. 31–40.

[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.

[56] A. Kori, G. Krishnamurthi, and B. Srinivasan, "Enhanced image classification with data augmentation using position coordinates," *arXiv Prepr. arXiv1802.02183*, 2018.

[57] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *2017 Chinese automation congress (CAC)*, 2017, pp. 4165–4170.

[58] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *Ieee Access*, vol. 5, pp. 5858–5869, 2017.

[59] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," 2010.

الخلاصة

. في السنوات الأخيرة، لعبت تقنيات الرؤية الحاسوبية دورًا رئيسيًا في معظم تطبيقات الوسائط المتعددة، وكان تصنيف الصور أحد المعايير المهمة لتصنيف التطبيق. تصنيف الصور هو أحد أعظم الثغرات التمهيدية في رؤية الكمبيوتر. يتم استخدامه على نطاق واسع في معظم تطبيقات الوسائط المتعددة الرقمية بالاقتران مع التطبيقات العملية مثل فهرسة الفيديو والصور. على الرغم من المشكلات الرئيسية في تحديد صور الوسائط المتعددة يدويًا فقد حددت نقاط ضعف كبيرة للإنسان لتصنيف الصور عالية الدقة وهي نقاط ضعف تافهة. تحتوي الخوارزمية العامة على الكثير من نقاط الضعف في تحديد الدقة. لذلك، لتصنيف الصور، تم إنشاء خوارزمية عامة مع إستراتيجية موجودة لعدد ثابت من الاختلافات. في الآونة الأخيرة، تم تطبيق العديد من المشكلات على الشبكات العصبية العميقة للحصول على أفضل النتائج. على وجه التحديد، أوضحت الشبكات العصبية العميقة التلافيفيه أفضل النتائج من حيث التعرف على الصور وتجزئة الصور ومشاكل رؤية الكمبيوتر ومشكلات تمثيل اللغات الطبيعية.

الهدف هو تدريب شبكة CNN باستخدام نموذج متسلسل لتصنيف صورة مجموعة بيانات الموضة ومهمتنا الرئيسية هي إنشاء نموذج فعال للتعلم العميق للتعرف على إكسسوارات الملابس وزيادة مجموعة بيانات Fashion-MNIST التي نستخدمها.

مشكلة تصنيف ملابس Fashion-MNIST هي مجموعة بيانات قياسية حديثة تستخدم في رؤية الكمبيوتر والتعلم العميق. إنه سهل نسبيًا لأنه بسبب مشاركة حجم الصورة الدقيق، وبيانات التدريب والاختبار، وهيكل تقسيم التنسيق. تمت معالجة البيانات مسبقًا لتغيير الحجم وتقليل الضوضاء. يتم زيادة البيانات حيث تكون صورة واحدة في ثلاثة أشكال من المخرجات؛ على سبيل المثال، يتم تدوير الصورة الناتجة وتحويلها والتكبير كإخراج. أخيرًا، يتم إرسال البيانات إلى النموذج المقترح. النموذج المقترح الذي يتكون من ثلاث طبقات تلافيفيه تستخرج 32، 64، 128 مرشحًا بحجم x33 مع ReLU كوظيفة تنشيط لكل طبقة وSoftMax في الطبقة الأخيرة، ثم يتم تغذية ناتج وظيفة التنشيط الخاصة بـ أقصى تجمع لـ نافذة 22×، ثم Dropout لمعالجة مشكلة overfitting، أخيرًا استخدم مُحسِّن "Adam" لتحسين وظيفة الخسارة. النموذج المدرب مع الإطار المسقط لديه دقة 94٪ تم تحقيقه ومقارنته بالأعمال القائمة. تم استخدام دقة نموذج CNN الذي تم تدريبه مسبقًا لتصنيف بيانات أزياء MNIST التي كشفت أنها الأنسب لمجموعة البيانات المحددة.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة الأنبار
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسبات

نموذج التعليم العميق لتصنيف الملابس اوتوماتيكيا

رسالة مقدمة الى

قسم علوم الحاسبات – كلية علوم الحاسوب وتكنولوجيا المعلومات جامعة الانبار،

وهي جزء من متطلبات نيل درجة ماجستير علوم في علوم الحاسبات

قدمت من قبل

مصطفى عامر عبيد

بإشراف

أ.م.د وسام محمد جاسم

1442 هـ		2021 م