

Performance Analysis of Various Image Classifiers and Formulation of Convolution Layer of CNN

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF DEGREE
OF

MASTER OF TECHNOLOGY
IN
SOFTWARE ENGINEERING

Submitted By
ROHIT TYAGI
2K17/SWE/14

Under the supervision of

Mrs. SONIKA DAHIYA
Assistant Professor
Department of Computer Science and Engineering
Delhi Technological University



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)
SHAHABAD, DAULATPUR, BAWANA ROAD, DELHI – 110042

JUNE, 2019

M. Tech (Software Engineering)

ROHIT TYAGI

2019

Department of Computer Science and Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Rohit Tyagi, 2K17/SWE/14, student of Master of Technology (Software Engineering), hereby declare that the Major Project-II Dissertation titled “**Performance Analysis of Various Image Classifiers and Formulation of Convolution Layer of CNN**” which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of requirement for the award of degree of Master Of Technology (Software Engineering) is original and not copied from any source without proper citation. This work has not been previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

ROHIT TYAGI

Date:

2K17/SWE/14

Department of Computer Science and Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled “**Performance Analysis of Various Image Classifiers and Formulation of Convolution Layer of CNN**” which is submitted by Rohit Tyagi, (2K17/SWE/14) to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of requirement for the award of the degree of Master of Technology, is a record of project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Mrs. SONIKA DAHIYA

Date:

(Supervisor)

Assistant Professor

CSE Department

Delhi Technological University

(Formerly Delhi College of Engineering)

Shahbad, Daulatpur, Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without accomplishing the people who made it all possible and whose constant guidance and encouragement secured us the success.

First of all, I would like to thank the Almighty, who has always guided me to follow the right path of the life. My greatest thanks are to my parents who bestowed the ability and strength in me to complete this work.

My thanks is addressed to my mentor **Mrs. Sonika Dahiya**, Department of Computer Science and Engineering who gave me this opportunity to work in a project under her supervision. It was her enigmatic supervision, unwavering support and expert guidance which has allowed me to complete this work in due time. I humbly take this opportunity to express my deepest gratitude to her.

Date:

Rohit Tyagi
M.Tech (SWE)-4thSem
2K17/SWE/14

ABSTRACT

Convolutional Neural Networks (CNNs) are a kind of deep neural networks which are designed from the biologically driven models. Researchers focused on how human perceives an image in their brain. As image is passed through different layers in human brain, in the same way CNNs have many layers. In this work, structure of CNN is described, along with the guidelines on the design of convolution layer and decision making on when to use pre-trained CNN model with transfer learning and when to design own custom architecture CNN model. This will help future researchers in a quick start with CNN modeling. Experimentation is done on two popular image datasets i.e., CIFAR-100 and Stanford Clothing Attribute Dataset, where CIFAR -100 is a clean dataset of 60,000 images belonging to 100 classes and Stanford Clothing Attribute Dataset is highly noisy and imbalanced data as it has uneven distribution of samples for different attributes and many of the samples do not have a clear distinction between the classes resulting in overlapping training data.

To display the results, four CNNs were designed, where two models were pre-trained CNN models and two were customized CNN models. A comparative analysis of their performance on image classification task and treatment of the missing data in dataset is being done. Based on this comparison and related study, the guidelines for designing convolution layer and making choice between using a pre-trained (transfer learning) or customized CNNs are made. Along with this work, the performance of various machine learning classifiers such as Multinomial Logistic Regression, Support Vector Machine, Multi-Layer Perceptron, Random Forests, Naïve Bayes, K-Nearest Neighbors, ADA Boost and Convolutional Neural Network is compared over two popular image data sets CIFAR-10 and MNIST. The analysis is done based on performance variance, feature extraction and feature selection. The results show that CNNs outperformed amongst all the image classifiers by automatically extracting and selecting features and giving better results.

TABLE OF CONTENTS

Content	Page No.
Candidate's Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
List of Abbreviations	ix
CHAPTER-1: INTRODUCTION	1-16
1.1 Overview	1
1.1.1 Machine Learning	1
1.1.2 Computer Vision	4
1.1.3 Image Classification	5
1.1.4 Artificial Neural Network	5
1.1.5 Convolutional Neural Network	6
1.1.6 Feature Extraction and Feature Selection	11
1.2 Motivation	13
1.3 Problem Statement	14
1.4 Organization of the Dissertation	16
CHAPTER-2: LITERATURE REVIEW	17-34
2.1 Background Work	17
2.2 Image Classifiers	22
2.3 Transfer Learning	33
CHAPTER-3: FORMULATION OF DECISION ON THE CHOICE OF CNNs AND FRAMEWORK OF CONVOLUTION LAYER	35-44
3.1 Feature Extraction from Images	35
3.1.1 Pre-processing the Original Images	35
3.1.2 Feature Extraction	37

3.2 Basic Description of CNN Structure	39
3.3 Decision making on choices of, Pre-trained CNN or Customized CNN	42
3.4. Designing guidelines for Convolution Layer	44
CHAPTER-4: IMPLEMENTATION AND RESULTS	45-54
4.1 Datasets Used	45
4.2 Technologies Used	46
4.3 Implementation	48
4.3.1 Code	48
4.4 Results and analysis	51
CHAPTER-5: CONCLUSION AND FUTURE WORK	55
CHAPTER-6: REFERENCES	56-58

LIST OF FIGURES

S.NO	FIGURE NAME	PAGE NO.
1	Figure 1.1: Cards with some printed Images on them	1
2	Figure 1.2: Schematic of Supervised Learning	2
3	Figure 1.3: Schematic of Unsupervised Learning	3
4	Figure 1.4: Reinforcement Learning Process	4
5	Figure 1.5: Different Types of Filters/Kernels	8
6	Figure 1.6: Convolution layer Operation	10
7	Figure 1.7: Operations of Activation layer, Pooling layer and Flattening	10
8	Figure 2.1: Classification Accuracies, CIFAR-10 image dataset	20
9	Figure 2.2: Support Vector Machine	23
10	Figure 2.3: Schematic diagram of Biological Neuron	26
11	Figure 2.4: Mathematical model of an ANN's neuron/node	27
12	Figure 2.5: Schematic diagram of an ANN structure	28
13	Figure 2.6: Representation of a k-Nearest Neighbor graph	33
14	Figure 3.1: Grayscale Image from FER-2013 dataset	36
15	Figure 3.2: Gray-scale image conversion using channel drop	37
16	Figure 3.3: Cycle for Image Classification	38
17	Figure 3.4: Basic structure of CNN	42
18	Figure 4.1: Performance of Custom and Pre-Trained CNNs on CIFAR-100 and Stanford Clothing Attribute dataset	54

LIST OF TABLES

S.NO	TABLE NAME	PAGE NO.
1	Table 4.1 Comparative analysis of accuracies of Image Classifiers on CIFAR-10 dataset	51
2	Table 4.2 Classifiers' Accuracies on MNIST dataset	51
3	Table 4.3 Classifiers' Accuracies on CIFAR-100 and Clothing Attribute Dataset	54

LIST OF ABBREVIATIONS

ML	:	Machine Learning
AI	:	Artificial Intelligence
CV	:	Computer Vision
RGB Image	:	Red, Green and Blue Image
HSV Image	:	Hue Saturation Value Image
ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
ReLU	:	Rectified Linear Unit
MLR	:	Multinomial Logistic Regression
SVM	:	Support Vector Machine
RF	:	Random Forests
MLP	:	Multi-Layer Perceptron
K-NN	:	K-Nearest Neighbor
ADA	:	Ada Boost
NB	:	Naive Bayes
CIFAR-10	:	Canadian Institute For Advanced Research
MNIST	:	Modified National Institute of Standards and Technology
IDE	:	Integrated Development Environment
RAM	:	Random Access Memory

CHAPTER-1

INTRODUCTION

Machine Learning (ML) is the new emerging concept from artificial intelligence with the ideology to make the machines imitate the human brain. This idea help machines to assimilate and learn things naturally and rapidly, for this the machines are programmed and able to resolve problems automatically. It has applications in, finding, separating and condensing pertinent information, making expectations dependent on the investigation information, ascertaining probabilities for explicit outcomes, adjusting to specific advancements independently, improving procedures dependent on perceived examples.

1.1 Overview

1.1.1 Machine learning

Machine Learning is the quintessential skill of this digital age. ML is basically an application of artificial intelligence (AI). It empowers the computer systems or the machines to “learn” data driven choices, instead of being explicitly programmed for carrying out a specific task. The Machine Learning algorithms or programs are structured such that they learn and improve after some time when they are given a new data.

There are three types of machine learning techniques as follows:

1.1.1.1 Supervised Learning

Supervised learning is one of the widely accepted types of machine learning techniques. It is easy to grasp and simple to code.

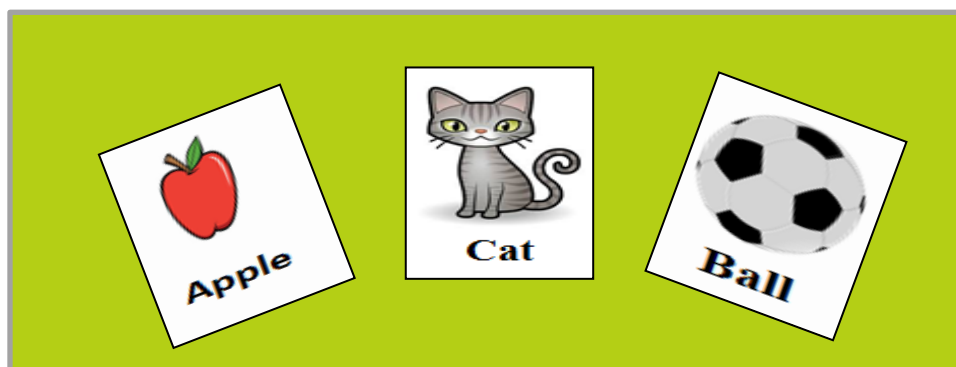


Figure 1.1: Cards with some printed Images on them

It is like introducing a child with the usage of the cards with some images printed on them as shown in figure 1.1. We can provide these example-label pairs one by one as input to a learning algorithm, which would enable the algorithm to predict the label for every example-label pair. Feedback such as right or wrong prediction will also be provided to the algorithm. After several feedbacks algorithm will learn to identify the mapping between example-label pairs. After getting fully-trained, the supervised learning algorithm will be able to predict label for a completely new example which was not observed before.

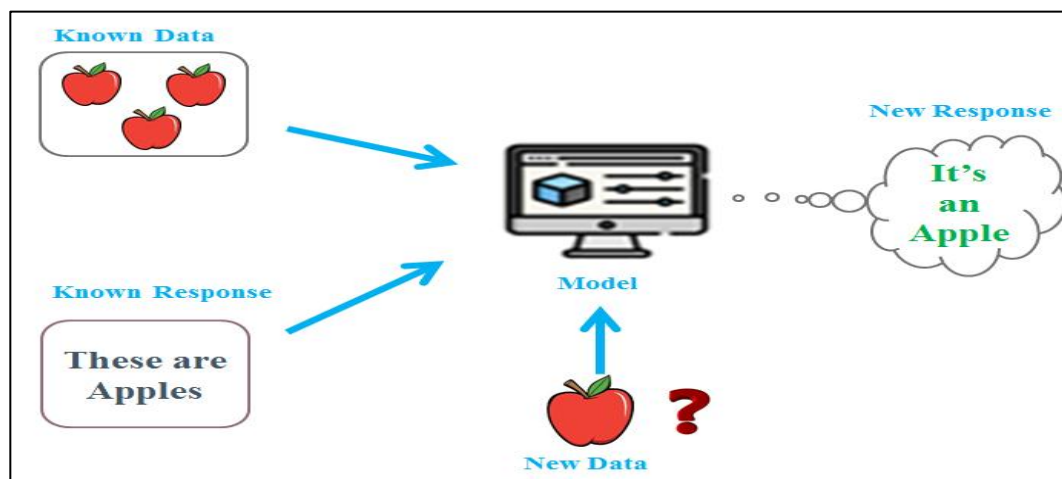


Figure 1.2: Schematic of Supervised Learning

Following are the common applications of supervised learning:

- **Advertisement Popularity:** Supervised learning is utilized to select the advertisements which will prove to be better attention grabber.
- **Spam Classification:** Systems learn how to automatically filter out malicious emails. Mostly these algorithms behave in a manner that a user will feed new labels into the system and preference of the user can be learnt from it.
- **Face Recognition:** Reading facial images and then identifying them or mapping them with the database, is also an application of supervised learning.

1.1.1.2 Unsupervised Learning

It is quite opposite to the aforementioned supervised learning. Here no labels are provided, a lot of data is provided as input to these algorithms and using the given tools features of

the data will be understood by the system. Furthermore it learns to organize, cluster and/or group the given data, so that a human can make a meaningful conclusion from the organized data.

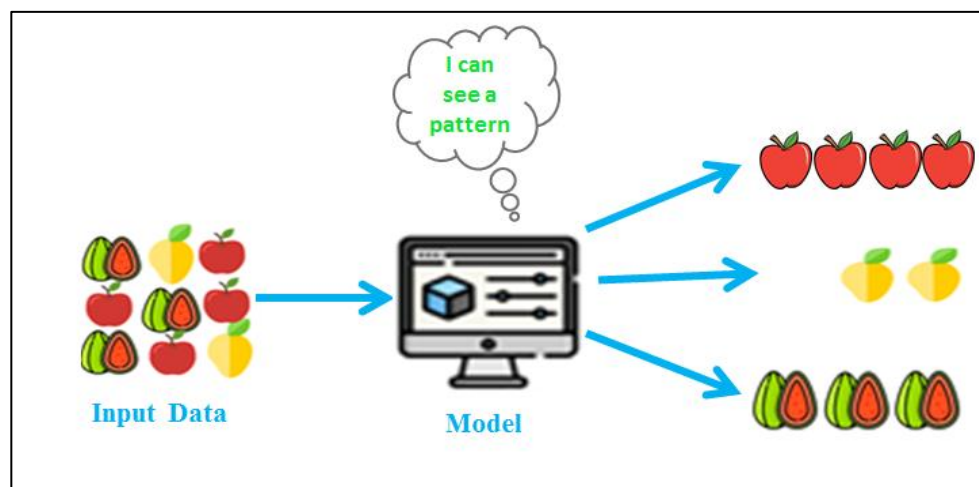


Figure 1.3: Schematic of Unsupervised Learning

Few applications where unsupervised learning is used:

- **Buying Habits:** Buying habits of the customers are stored in databases and unsupervised learning groups the buyers into similar segments of purchase. This allows companies target these groups effectively.
- **Recommender Systems:** Using the watch history of many users. The users who have watched similar videos as you and watched other videos which you are yet to see, the recommender system utilizes this relationship in the data and provides suggestions.

1.1.1.3 Reinforcement Learning

Reinforcement learning is looked upon by many as learning from mistakes. When this learning algorithm is placed in a new environment, initially it will make many errors. When we provide feedback signals to the system for its output, we can reinforce our algorithm as required. Eventually, our learning algorithm learns to make less error as before.

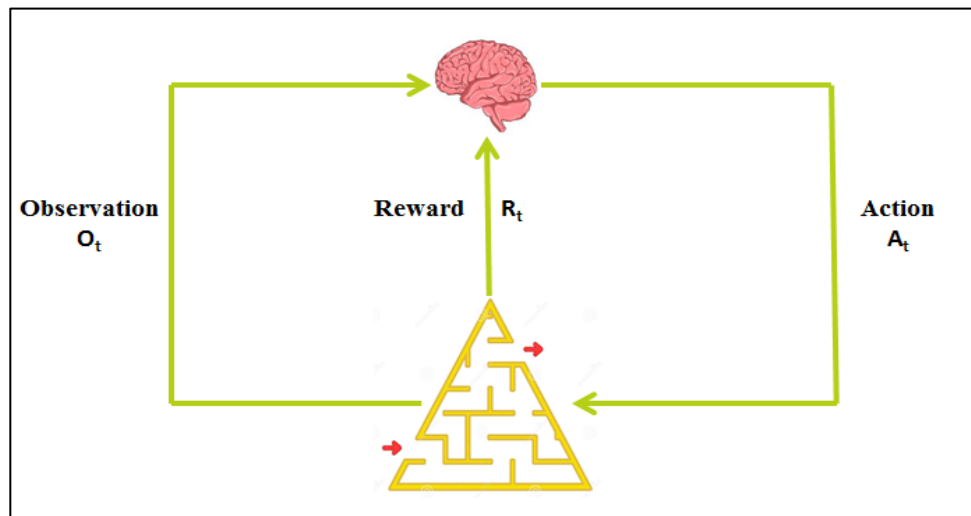


Figure 1.4: Reinforcement Learning Process

Application:

- **Video Games:** Learning to play videos games is one of the areas where reinforcement learning is exploited.

1.1.2 Computer Vision

Computer Vision (CV) aims to automate the tasks which humans do naturally by using their visual capabilities. Computer machines are designed to observe a very high-level meaning from digital videos or images. Computer vision typically follows the steps, which are, analyzing, processing, understanding and acquiring the images. Eventually they retrieve high-dimensional data from the real scenarios to create numerical information out of it. By understanding in computer vision it means that, visual data image is used to produce descriptions of the real world scenarios which along with the other views provide effective and relevant action. The understanding of image provides with, useful symbolic information from data images, using models. Computer vision involves artificial systems which retrieves information from the data images. The data images may have multiple forms, like views from multiple cameras, video data sequence, data images from a medical scanner. Computer vision aims to utilize theories for the production of computer vision systems.

1.1.3 Image Classification

In this world of digitization, images play a very important role in various areas of life including scientific computing and visual persuasion. Technically images can be Binary images, Gray scale images, RGB images, Hue Saturation Value (HSV) or Hue Saturation Lightness images etc. Image classification alludes to the assignment of extricating data classes from a multiband raster image. The subsequent raster from image classification can be utilized to make topical maps. Contingent upon the connection between the expert and the personal computers during classification, there are two sorts of classification: supervised and unsupervised. In the case of supervised classification, earlier information is necessary before experimenting and it must be collected by the analyst. The main benefit of supervised classification is that a supervisor can identify mistakes and correct them. And in the case of unsupervised classification, no earlier knowledge is required. It does not need any kind of human intrusion. This algorithm assists in recognizing clusters in data. The benefits of the unsupervised technique are that it is faster, escaped from human faults and there is no need of full prior information.

1.1.4 Artificial Neural Network

Artificial Neural Network abbreviates as ANN that is based upon the biological neurology system to compute the outcome. ANN is basically inspired by the structure and functionalities of biological neural networks. Generally, by biological neural network we mean the structure and working of human brains. ANN tries to imitate the functioning of human brain by following its principles. As human cerebrum is made up of billions of nerve cells called neurons. Similarly, an artificial neural network consists of many artificial neurons called nodes which behave as the same way a biological neuron does. Biological neurons are consists of three parts:

- Dendrites - they accepts the input from the previous layer.
- Axon - neurons are connected to each other through axon.
- Synapses- they transfer the output to the next layer neurons.

The input is taken by dendrites and sent to the nucleus, then nucleus decides whether to generate an output signal or not. If an output is fired by the nucleus, it goes to synapses

through the axon to be passed onto the next layer neuron. In the similar way, an ANN also consists of neurons called nodes, they are connected to each other by links and each link is associated with some weight. A node receives input from many nodes and based upon the activation function used in a node, action is taken or rejected. If the sum of input values reaches up to a threshold value, then the output is generated and passed onto the next layer input. Otherwise the input is rejected and no action is taken.

1.1.5 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a kind of deep neural networks which were designed from the biologically driven models. Researchers focused on how human perceives an image in their brain. As image is passed through different layers in human brain, in the same way CNNs have many layers. Hence they have been proven very efficient for all the image processing, pattern recognition kind of applications. CNN emerge from deep learning that inspired from biological neuron that form a network to connect. This is based on multilayer perceptrons. Convolutional Neural Network is basically an extension of Artificial Neural Network which is popularly used for many applications like medical image analysis, recommender systems, natural language processing, image classification, image and video recognition etc. Input images, used in CNN are of 3 dimensions i.e. height , width and number of channels, where height and width specify image resolution and number of channels specify whether the image is RGB image having 3 channels (i.e. red, green, blue color) or a Gray Scale image having 1channel(i.e. gray color). There are many layers in CNN, which includes: Convolution layer, Activation layer, Pooling layer, Batch norm Layer, Dropout Layer, Fully connected Layer.

1.1.5.1 Convolution Layer:

Convolution is the first layer used to extract the features from an input picture/image. Convolution conserves the correlation between pixels by learning image features using small squares of input data. It is a mathematical process that takes two inputs such as image matrix and a filter/kernel. A filter is used to run over the image in fixed gap intervals called strides. Selecting the size of stride is crucial to achieve desired results. During running the

filter over the image, dot product of filter with part of image on which filter lies is calculated. Then sum of all values of product matrix is copied to the corresponding position in convolved feature map matrix. Thus we get a reduced dimension feature map of image. Filters may be of many kinds, where each filter is used to extract different kind of feature from the image. For e.g. one filter may be responsible for extracting one kind of feature from the image based on shapes and edges and another filter may be used to extract feature based on color intensities. Stride defines the number of pixels by which we have to move our filter over the image so that we can focus on a new set of pixel while doing convolution. Stride's value ranges from 1 to 3 depending upon the amount of loss which we can be accommodated during convolution. The amount of loss in image increases with the increasing value of stride. Padding is a process of adding zeroes around the border of original image symmetrically. This helps us obtaining the feature map output to be of size as per our requirement. Commonly it is used to preserve dimension of image after convolution. Filters are also called kernels. These may be of many types. Some of the filters are explained under and also their filter matrices are shown in Figure 1.5.

- a.** Sobel filter (horizontal) - This filter is utilized to identify horizontal borders in the image.
- b.** Sobel filter (vertical) - This filter is used to detect vertical edges in the image.
- c.** Laplace filter (both horizontal and vertical) - This filter is used to detect horizontal edges as well as vertical edges in the image.
- d.** Blurring filter – This filter is used to blur out an image.
- e.** Sharpening filter – This filter is used to sharpen an image.

Each filter increases the depth of the output generated after convolution. So if we are using 3 filters then the depth of the output will be 3. There are 3 parameters on which the output of convolution layer depends namely stride, depth and padding. We must finely tune these parameters to obtain the desired output.

An e.g. of dimension reduction during convolution is: a 9×9 RGB image with 3, 3×3 kernels at stride 1(per channel), will produce an RGB feature of $7 \times 7 \times 3$. Here 3rd dimension represents the depth of the convolved image.

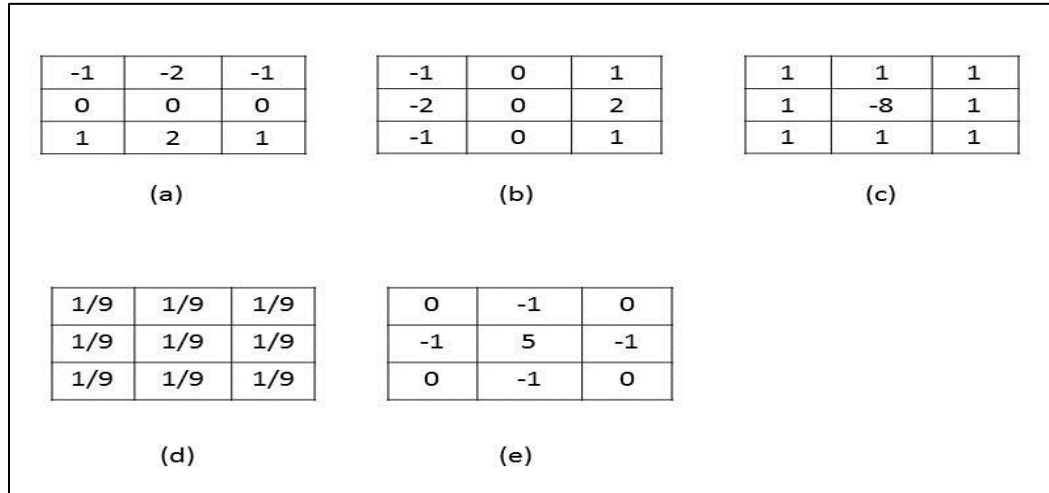


Figure 1.5: Different Types of Filters/Kernels

1.1.5.2 Activation Layer:

An activation layer in a Convolutional Neural system comprises of an activation function that takes the convolved feature map produced by the convolutional layer and makes the activation map as its yield. This layer mostly uses ReLU as an activation function. ReLU is a function which is used to set all negative values to zero and keeps positive value as it is. The ReLU Activation Function definition is: $R(z) = \max(0, z)$.

1.1.5.3 Pooling Layer:

The pooling layer would decrease the number of parameters when the images are extremely large. Spatial Pooling also called downsampling which diminishes the dimensionality of each map but preserves the important information. Spatial Pooling can be of different types such as sum pooling, average pooling, max pooling etc. Pooling layer operates a small kernel on the image at fixed stride. It is used to pick the pixel with highest intensity and discard other pixels. The resultant matrix will be a reduced dimensional matrix of feature map. This helps in reducing the unnecessary sparse cells of image which are of no use in classification.

1.1.5.4 Dropout Layer:

Dropout Layer is usually applied after the layer containing neurons in the fully connected

network. Dropout layer is a regularization layer. It helps to create robustness in the layer by dropping a fraction of units randomly from the previous layer usually kept around 20-50% of the original input.

1.1.5.5 Fully Connected Layer:

Fully connected layers take the high-level filtered images and translate them into the desired classes. Usually images which are fed into the neural network are reduced in dimensions so as to reduce the processing time and avoid the problem of under fitting. For e.g., if we take an image of size $224*224*3$ which when converted in to 1 dimension will make an input vector of 150528. Even this input vector is still too large to be fed as input to any neural network. The structure from top to down usually forms a pyramid structure, the number of parameters in these layers keep on converging till they finally reach the number of desired classes. Increasing the number of hidden units in the layer can increase the learning ability of the network, but there is saturation of the increase in accuracy of the network.

So, how CNN actually works is basically a step by step process of corresponding layer's operations such as convolution layer operation, activation layer (ReLU) operation, pooling layer operation, flattening and finally, flattened vector data is fed to fully connected layer as an input, where one vector value is given to one input node of the network.

Taking an example:

First of all, CNN uses a 'filter matrix' over array of image and perform 'convolution' to get 'convolved feature map' and this operation is called Convolution operation.

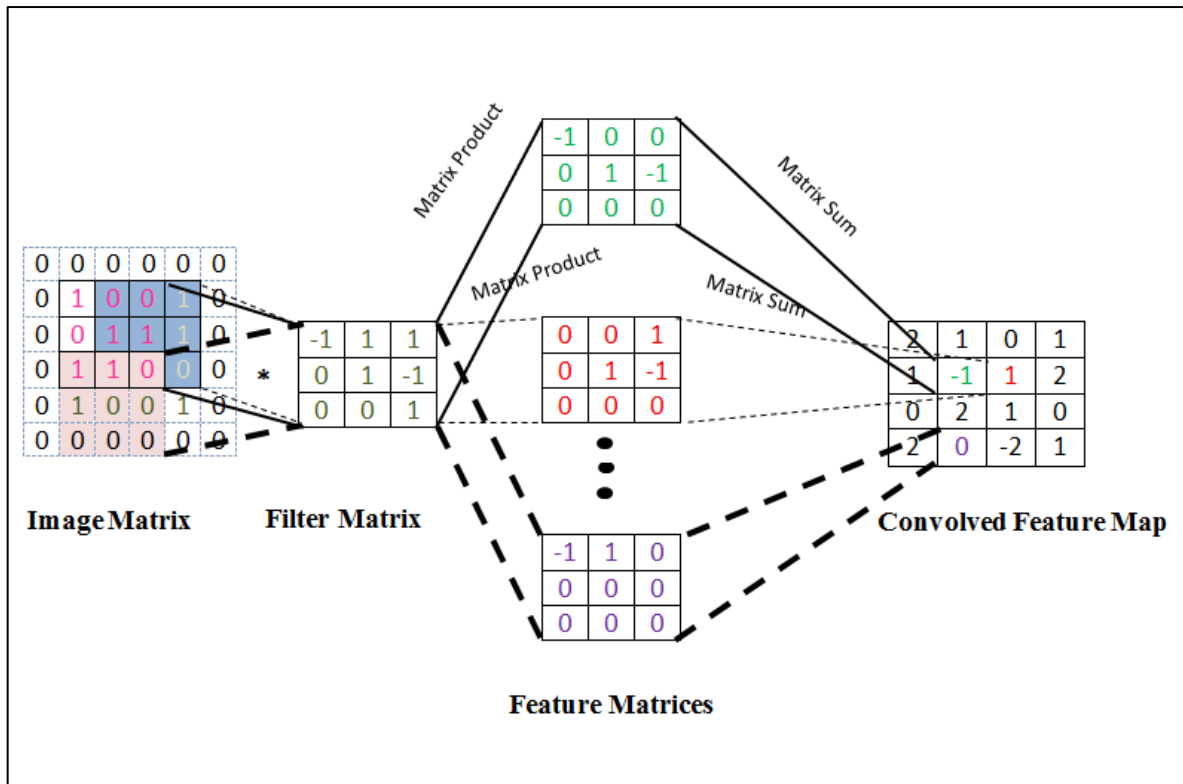


Figure 1.6: Convolution layer Operation

After this operation, Activation layer (ReLU as activation function) operation, Max pooling and Flattening occurs.

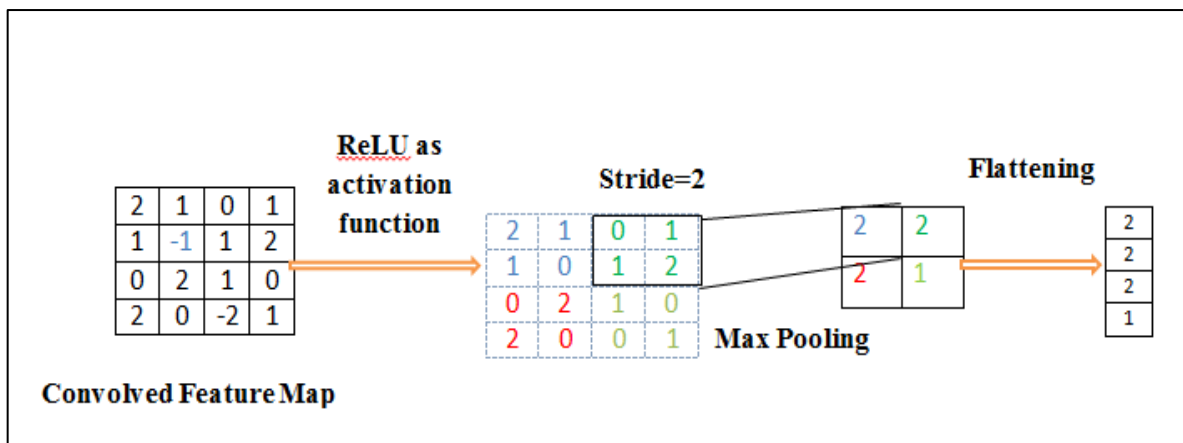


Figure 1.7: Operations of Activation layer, Pooling layer and Flattening

And in the last, this flattened vector data is fed to fully connected layer/neural network as an input, where one vector value is given to one input node of the network.

1.1.6 Feature Extraction and Feature Selection

Feature extraction is the process of extracting the numeral features from the image datasets that form the basis for training and prediction of the model. This forms the major basis during the machine learning process to identify how a machine learns to classify and the inputs or the raw materials needed for learning the specifics of the desired task, features or attributes forms the basis of what we actually feed in the learning algorithm. Features may vary from simple binary inputs to highly complex matrix data. The data can be sequential, temporal, sparse, dense, 2D, 3D or high dimensional. As well as data can be represented in tables, graphs, documents, images etc. The collection of data objects, data records, vector data, data tests, data cases or data entity is called dataset. There are different types of data in the datasets:-

- **Record-Based data:**

In Record based data, the data is stored in the form of a table. In the table each column is called a field and each row is called a record, which is fixed.

- **Graph-Based Data:**

A graph defines a type of structure that compromises of two things, i.e., nodes and edges. Nodes represents (stores) the actual data and edges represents the relationship between different nodes.

- **Ordered Data:**

In an ordered data, the attributes of data have relationship that involves order in time or in space.

- **Image Data:**

Data that can be represented in pictorial or image form is said to be an image data. It provides the information and behavior of data in form of remotely stores sensory images. For e.g. satellite images. Images are usually of 3 dimensions [height, width and depth]. The height and width are the dimensions of the pixel's resolution grid while the depth defines the RGB color value of a particular pixel on the grid. The values of RGB (red, green and blue) can range from anything between 0-255, where higher values signify greater density.

In this world of digitization, images play a very important role in various areas of life including scientific computing and visual persuasion. There exists different diversification of images in nature, specified as follows:-

- **Binary Images:** In this, each pixel is represented in a one bit binary encoded form based on their intensity. So the image is expressed as either black or white. The white color encodes for foreground image whereas black color encodes for background image.
- **Gray Scale Images:** These images constitute the different obscuration of grey color. This range is decoded using 8 bits per pixel. These combinations give a range from the darkest color as grey and fluorescent as white. Grayscale plays a very crucial role as without grayscale decoding, the whole image turns into a black picture.
- **RGB images:** A digital image comprises of various colors along with grey and white. So there is a need of another representation for such images. Here is where RGB scale comes into picture. In this, every pixel is expressed using 24 bits, involving 8 bits for every red, green and blue pixel individually. Using the grayscale decoding, RGB can be represented using the intermediate 254 combinations, excluding the all zeros' and all ones'. All zeros' encode for black color and all ones' encode for white color. That is why these images are also known as 8 bit grayscale images.
- **Hue saturation value (HSV):** In this representation, images are represented along with their angular information in the 3-dimensional space. Hue is a proportion of the wavelength found in the dominant color gotten by the sight while the Saturation is the size of the measure of white light blended in hue.

So, technically images can be Binary images, Grayscale images, RGB images, Hue Saturation Value or Hue Saturation Lightness images etc. Pixels of an image are exemplified using matrices. Each value of the pixel is represented in the matrices combines to form dense and sparse matrices. Each data record can be represented via a huge number

of features. But all features are not necessarily significant for analysis or classification. Thus “feature extraction” and “feature selection” are significant research areas.

Feature extraction involves gathering set of information from the given data and transforming it into smaller number of attributes that carry the maximum information about the original data.

Feature selection can be defined as a problem of choosing the minimal set of features that are able to address the problem in a more effective, compact and computationally efficient manner. Basically, there are many processes involved in features selection that gives the final features set that we use for analysis:

- **Creating new features from existing ones**

New features can be derived from existing records, for e.g., the feature to decide, student is in state or out of state can be added on the basis of student state of residence.

- **Removing features**

Separating correlated data, such as the price of an item and the tax paid on that item are correlated, as if the price is more tax is also. Removing the features contains large number of missing values. Also remove data with irrelevant information have no useful task in analysis.

- **Combining features**

Features can be combined if the new combined feature is more relevant, for e.g., calculating speed from the feature of distance and time.

- **Breaking up features**

Separating one feature into multiple to get more meaningful features, for e.g., feature address is partitioning into house number, city, street, state, zip.

1.2 Motivation

With the growing world of digitization, videos and images have come out as an important aspect of the advancing industry. This brings to its connection with the enriching field of

AI, by using classification techniques of ML with the images. The major motivation of this report is to analyze how image classification can help in improving the existing problems. This can be done using a sample of techniques such as Logistic Regression, SVM, Random Forests, Naïve Bayes, etc. Logistic regression is a linear classifier model but SVM can be used as a multi-classifier as well.

So, to analyze the advantages of these techniques and the drawbacks associated, this report gives a comparative analysis of the existing classifiers. Then the solution to the existing problems in them is given using deep learning leading techniques. As deep learning has been state-of-the-art in most of the image classification tasks. Images can be represented in various forms. Some of them include RGB and Grayscale representations. These are used to find how they affect the classification task as well. Deep Learning is very useful because it has automatic feature selection and extraction which helps to develop good models giving an edge to the developers who do it from the scratch.

This report also discusses the structure of CNN. CNN consists of many layers which include: Convolution layer, Activation layer, Pooling layer, Batch norm Layer, Dropout Layer, Fully connected Layer. Designing CNN is a big task in itself. There is no fixed formula for it yet. So a research is being conducted to narrow it down to a few possibilities to show the classification results over different datasets. Designing a CNN itself for the image classification is a challenging task because these CNNs only take out the patterns that exist in the data itself. So, how CNN varies on the data, that is the motivation for this work.

1.3 Problem Statement

Machine learning is a very growing field these days and in this report the problems specific to the task of image classification have been seen. As there are different types of images such as Binary images, Grayscale images, RGB images, Hue Saturation Value or Hue Saturation Lightness images etc., so this report aims at finding how these images affect the image classification task for a variety of famous image classifiers that are existing such as Multinomial Logistic Regression, SVM, Random forests, Multi-Layer Perceptron,

K-Nearest Neighbor, Ada Boost, Naive Bayes, etc. and also deep learning CNNs are the state of art these days. During this it was found that there is a problem actually with feature selection part of the data. Feature extraction is not automatic as well. So, the problem is in the designing as to what kind of classifiers are present, and found that CNNs actually perform well.

Then a CNN model was designed which concludes that there is a problem in the structure of the CNNs. It is very difficult to decide that how many layers a CNN should have and what should be the architecture of that. And it is very difficult for someone new to that to do this. So, this work aims at narrowing down the possibilities and classifying the images based on the data itself and the kind of patterns that exists in the images. A pattern is formulated especially for the convolution layer that what kind of structure can be implemented for better results and what should be its width and depth depending on the layers itself. And that's how the motivation for this work, to study the image classification for various kind of images and various classifiers and then focusing on the deep learning CNNs, so as to understand what should be the design structure of these non-trivial solutions.

So, the task is to identify what are the potential gaps are in the field of machine learning, deep learning and particularly for the tasks specific of Image Classification. Essentially Image data is quite varying from numbers itself. So, first the gaps are identified that how the numerical machine learning techniques normally work on numbers, does not apply to Image data. Next, the mathematical representation of Image data is formulated, where there are ranging arrays moreover as raw pixels cannot be feed into these algorithms because the inputs are quite high. For example, in Iris dataset, there is an image but to do feature extraction the features like the length of petal, length of sepal, the width of petal and width of sepal, have to be chosen manually. And later work on these new features rather than the pixels itself carried away. Following this, it was observed that how classifiers have performed and how the task of feature extraction and feature selection actually affect the problem of Image Classification. Subsequently deep learning and especially the field of

Convolutional Neural Networks have been implemented in order to get good results particularly for the Image Classification task.

It was found that CNNs performed well for the task of Image Classification, feature extraction, and feature selection moreover helps get better results in terms of the algorithms, as CNNs are very deeply in nature having much depth of hierarchical features. The trade-off in this is in designing the CNN so a set of guidelines is recommended concerning someone new toward the field of CNN. The field of transfer learning and pre-trained networks are also inspected to find how they are useful for the tasks such as Computer vision, Image processing, etc.

1.4 Organization of the Dissertation

As there are different types of images such as Binary images, Grayscale images, RGB images, Hue Saturation Value or Hue Saturation Lightness images etc., so this report aims at finding how these images affect the image classification task for a variety of famous image classifiers. So, this report includes a comparative analysis of existing classifiers and to analyze how image classification can help in improving the existing problems. This report also discusses the structure of CNN. As there is no fixed formula for designing the CNNs yet, so a set of guidelines are given for CNN layers and recommendation for the choices of different CNNs.

The current chapter describes the overview behind carrying out this study and also the motivation and problem statement for doing this study.

Chapter 2 describes the literature review behind this study.

Chapter 3 describes the formulation of decision on the choice of CNNs and framework of Convolution Layer.

Chapter 4 provides a description of the implementation and the results that were obtained in this study.

Chapter 5 contains a description of the conclusion and the future scope of this study.

CHAPTER-2

LITERATURE REVIEW

This module discusses about the work being conducted by various researchers in the field of image classification. In the area of machine learning for image classification tasks, researchers have done various researches on ‘Image Classification’ by using different techniques. It also discusses about the existing work of some researches that will help us to know about the Image Classification in depth.

2.1 Background Work

Machine learning is the quintessential skill of this digital age. During the machine learning process of identifying how the machine learns to classify and the inputs or the raw materials needed for learning the specifics of the desired task, features or attributes forms the basis of what is actually passed as input in the learning algorithm. Features may vary from simple binary inputs to highly complex matrix data. In this world of digitization, images play a very important role in various areas of life including scientific computing and visual persuasion. Technically images can be Binary images, Gray scale images, RGB images, Hue Saturation Value or Hue Saturation Lightness images etc. Pixels of an image are exemplified using matrices. Each value of the pixel is represented in the matrices combines to form dense and sparse matrices. Each data record can be represented via a huge number of features. But all features are not necessarily significant for analysis or classification. Thus “feature extraction” and “feature selection” are significant research areas.

“Feature extraction” involves gathering set of information from the given data and transforming it into smaller number of attributes that carry the maximum information about the original data. “Feature selection” can be defined as a problem of choosing the minimal set of features that are able to address the problem in a more effective, compact and computationally efficient manner. Feature selection involves creating new features from existing ones, removing redundant and insignificant features, combining a number of features to a minimal count, as well as splitting a feature to a number of features.

For example: **Iris Data Set [4]**, which consists of, 150 instances of 3 classes namely, Iris Virginica, Iris Setosa and Iris Versicolor. The Iris Dataset features 4 attributes, sepal length, petal length, sepal width and petal width in centimeters, only these 4 attributes are sufficient for the classification of flowers in this dataset. As there can be many features for any one particular flower image like number of leaves, plant cell, length of the stem, plant structure, chloroplast, photosynthesis process, sepal length, petal length, sepal width and petal width etc . The transformation of the image data consisting of pixel data to numerical attributes is “feature extraction”. This allows different classifiers to operate on image data without having any information about the original image (pixel form image). The selection of only these 4 features (sepal length, petal length, sepal width and petal width) for classification or any further processing is “feature selection”.

Jović, A. et al. [6] have majorly classified the feature selection methods into three categories, **wrapper methods, hybrid methods and filter methods**. Filter Methods use ranking, co-relation, mutual information or other criteria to rank the features in order, and then one can choose, the k-best features out of the n features ($0 < k \leq n$). These methods though, do not depend upon the learning algorithms used, as they are just features independent selection, but this may have negative impact on the learning algorithm, as one algorithm may perform good on some selected features independently, but there cannot be a sure proof that it will work well. Wrapper methods consider feature subsets by the status of the performance on a modeling algorithm. As getting every subset of features is a NP hard problem, one can use optimization algorithms to find subsets and approach accordingly. For N features, there can be 2^N subsets, now if N is small like 10, 20, it may sound feasible to go through the whole subset of features and make the selection for the best feature subset, however when $N > 100$, things start to get tricky, as the number of subsets to be tested are huge. Hence one can apply algorithms to use optimal subsets making on a partial selection of features and have approaches such as incremental addition of a feature to a subset. Hybrid methods are a combination of filter methods and wrapper methods. Filter method is used to diminish the dimension space whereas wrapper method is used to find the best candidate sets. With the help of hybrid method one can get the better

accuracy i.e., feature of wrapper methods and also get the better efficiency i.e., feature of filter methods. It takes on the specific discussion of Image classification, and the challenges to represent an image for the classifier as well.

Lu, D. et al. [10] have presented tables for ‘Major Advanced Classification Methods’ and ‘Taxonomy of image classification methods’ which give insight into the classifiers’ performance based on the desired task and type of features. Logistic Regression, Support Vector Machines (SVMs), Neural Networks, plethora of classifiers can be used, but each have their advantage based on the distribution of data and type of features. As type of features forms the basis for further type of analysis, feature extraction and feature selection form a strong foundation for image classification. This may be done externally by applying the techniques separately for extraction and selection or automatically using a classifier such as Convolutional Neural Network (CNN), which internally performs this hierarchy, meaning, we don’t have complete control over the feature extraction over for each layer, specifically telling the network to extract only a particular feature of choice and then performing different feature selections on it to get the best set of features but rather the weights are updated and features are learnt hierarchically.

Popescu, M. C. et al. [11] have designed 48 features for the dataset of public pollen image dataset. Specific to their task, they took features such as height, width, the dimensions of ellipse enclosing the object. This is totally different for some other form of data. For CIFAR-10 dataset, usually the automated form of CNNs or Deep CNNs is preferred. With huge number of images, it is difficult to calculate features of different objects such as height, width, color information, the contours and creating this set of numerical features and have human verification for it to pass over to the network. They proposed a model in which they have classified the varieties of plants by using the pollen grains images. However this paper also focused on feature selection and feature extraction methods. To get the best accuracy of this model, they took the twelve dataset and twelve machine learning based classifiers. For an example, it was found from a random experiment that MLP classifier performs well on pollen dataset whereas PART classifier did not perform well on the same dataset.

Figure 2.1 provides a table of the leading classifiers particularly for the CIFAR-10 images dataset [15]. As visible from the figure, the leading classifiers are mostly CNNs, RNNs and other automated methods. Other normal classifiers may work well for these datasets, but the accuracies are mostly dominated by variants of the CNN. Here, an important transformation to understand is the pixel matrix transformation to numerical attributes before the classification algorithm actually operates on it. So, care has to be taken for datasets involving sparse values in such cases, as a transformation without check will lead to mostly zero values in the features and the classifier will have the tendency for over-fitting.



















Result	Method
96.53%	Fractional Max-Pooling 
95.59%	Striving for Simplicity: The All Convolutional Net 
94.16%	All you need is a good init 
94%	Lessons learned from manually classifying CIFAR-10 
93.95%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree 
93.72%	Spatially-sparse convolutional neural networks 
93.63%	Scalable Bayesian Optimization Using Deep Neural Networks 
93.57%	Deep Residual Learning for Image Recognition 
93.45%	Fast and Accurate Deep Network Learning by Exponential Linear Units 
93.34%	Universum Prescription: Regularization using Unlabeled Data 
93.25%	Batch-normalized Maxout Network in Network 
93.13%	Competitive Multi-scale Convolution 
92.91%	Recurrent Convolutional Neural Network for Object Recognition 
92.49%	Learning Activation Functions to Improve Deep Neural Networks 
92.45%	cifar.torch 
92.40%	Training Very Deep Networks 
92.23%	Stacked What-Where Auto-encoders 
91.88%	Multi-Loss Regularized Deep Neural Network 

Figure 2.1: Classification Accuracies, CIFAR-10 image dataset [15]

Krizhevsky, A. et al.[8] proposed a neural network architecture based on convolutional neural networks for image classification. They actually proposed a large, deep convolutional neural network to analyze the 1.2 million images from the ImageNet dataset into 1000 distinct classes. They achieved a top-1 error rate of 37.5% and top-5 error rate of 17.0% on the testing data which was considerably stabler than the earlier state-of-the-art neural networks. They designed the network architecture which consists of five convolutional layers, followed by max-pooling layers and three fully-connected layers including in the last a 1000-way softmax. The proposed neural network also consists of 650,000 neurons and 60 million parameters. To start with the faster training, they used non-saturating neurons and the convolution operation's implementation with a very efficient GPU. They applied a most recent regularization method called "dropout" to reduce the over-fitting in the fully-connected layers. In this study they have designed a architecture that reduced the lower level kernel in convolutional neural network by differentiating the color information from the real picture. Convolutional Neural Networks (CNNs) have gained immense popularity since AlexNet won the ImageNet Challenge in 2012.

Boureau, Y. L. et al.[16] have provided a detailed theoretical analysis of max pooling and average pooling, and give extensive empirical comparisons for object recognition tasks. They have also shown that the reasons underlying the performance of various pooling methods are obscured by several confounding factors, such as the link between the sample cardinality in a spatial pool and the resolution at which low-level features have been extracted.

In numerous fields such as Computer vision, Image processing, etc. CNNs are becoming state-of-the-art achieving near human or better performance. They sound fascinating but designing a CNN is a herculean task in itself. Till now there is no fixed formula for the design of CNN. Many researchers have come up with the general suggestions but they don't always hold and even with small changes at critical places in a CNN, huge improvements have been seen. From the review of the above research papers, one can say that these conditions (such as pooling design studied by [16], smaller convolutional filters for better accuracy in case of Imagenet dataset [17], and study of Very Deep CNNs for

Large-Scale Image Recognition [18]) do not hold always, as the dependency of the classification task on a dataset is as important as its dependency on an algorithm. CNN exploits the spatial hierarchical features of data, extracts features and helps classify them into different classes. This has led to development to a stream of data augmentation and pre-processing to increase the data, as more data allows for chance of better training and avoiding over-fitting. This helps build models that are more robust to new samples as we try to make it more generalized to noise at training phase.

2.2 Image Classifiers

This section describes the preliminary information of some of the machine learning algorithms that is requires to under the work done in detail. These include:

2.2.1. Support Vector Machine (SVM) [3]

SVM is basically a supervised machine learning technique. SVM is a classification algorithm and when used for classification tries to find a hyper-plane differentiating two (or more) classes or it draws a hyper-plane which discriminates a set of data points in two different classes. For example, in a 2D space SVM draws a line that separates the data points of different classes from each other. It is a linear classifier but can be used as a non-linear classifier using kernel implementation by mapping it to a higher dimensional space. There may be many hyper-planes separating the two classes but an optimal hyper-plane is defined as the linear decision function with max margin between the class vectors.

During training phase of SVM, at first any random hyper plane is drawn but while learning process, according to the error, the plane adjusts its position to reach at optimal distance from both the classes.

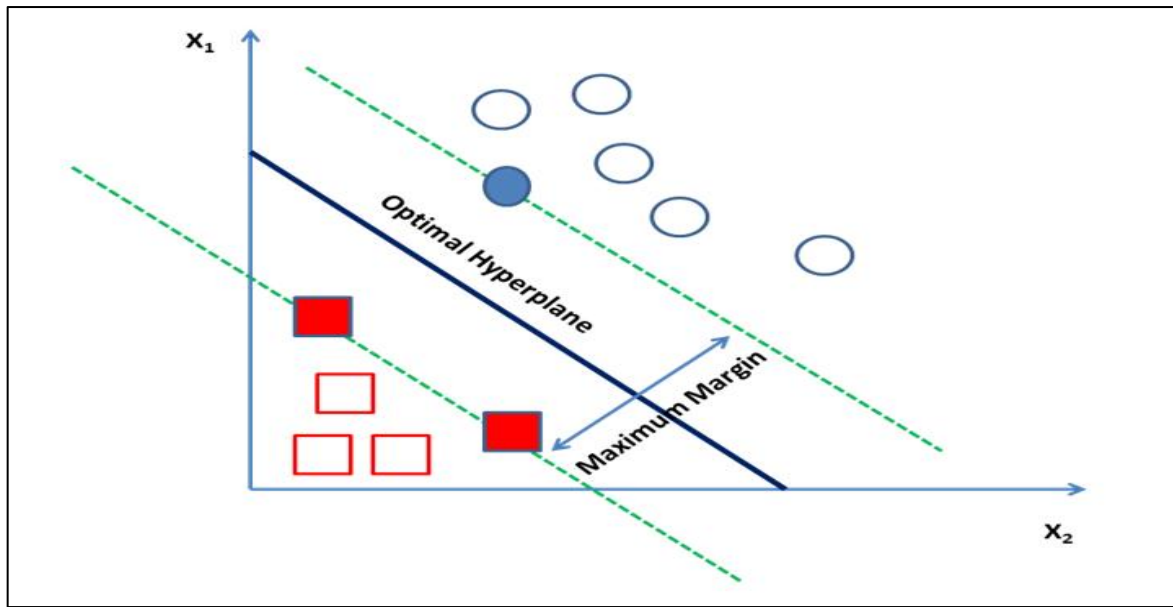


Figure 2.2: Support Vector Machine

Advantages of SVM-

- It works very well when we have no idea on the input data.
- Even with the unstructured and semi-structured data SVM's efficiency doesn't reduce.
- The real strength of SVM is its kernel function. Any complex problem can be solved with an appropriate kernel function.
- While comparison with ANN, SVM always gives better results with high accuracy.

Disadvantages of SVM-

- It is tricky to choose an appropriate kernel function which usually consumes a lot of time.
- Training time is very long for large databases.
- Fine tuning of hyper parameters like Cost and gamma is not an easy task.
- In multi dimensions model, it is not possible to visualize the model, that's why it is not possible to do small changes also in kernel function.

2.2.2. Multinomial Logistic Regression (MLR) [9]

MLR classifier predicts probabilities of binary classes known as logistic regression and if

for more than two classes known as multinomial regression, in terms of the dependent output variables. Means, this classifier predicts the category of any member of a dependent variable according to the different values of independent variables. It is an extension to binary logistic regression with more than two categories to classify. As binary logistic regression uses probability to predict the belonging, multinomial logistic regression uses maximum likelihood estimation. MLR considers sample size and outliers to predict the category more accurately. In this method multivariate diagnostics are used to accurately assess inclusion and exclusion of outliers or special cases. No ranking order in dependent variables removes the possibility of using estimators such as least square for predicting the results. In such cases different types of classifiers such as Multinomial Logistic Regressions are used. In comparison to other estimators it is simple to understand and faster in terms of underlying calculations. In this method linearity, homoscedasticity or normality is not assumed by itself that's why it is considered to be an attractive analysis technique. Logistic function predicts the probability for a particular outcome by formation of a linear combination of independent variables/features. Hence, it is a linear classifier. It follows a Bernoulli distribution for dependent variables in case of two classes.

$$\text{Logit} = \ln \left(\frac{P}{1-P} \right), \quad \text{here 'P' is the probability of success.}$$

Advantages of MLR -

- Normal distribution of independent variables is not necessary, so this makes it robust.
- It is not necessary for independent variables to be in intervals or unbounded.
- Non-linear effects can be handled effectively.

Disadvantage of MLR-

- MLR doesn't assume linear relationship between independent and dependent variables.

2.2.3. Naive Bayes (NB) [12]

Naive Bayes uses an approximate Bayesian distribution over the dataset and predicts the most probable class based on the features. Although it assumes, independence of features,

which is not always the case, but still in many cases, NB has given competitive results because optimality in terms of classifier error is not always dependent on the quality of approximate distribution.

Here, the Dataset is separated into two sections: list of features and response set. Based on the feature set the prediction is made where nearness of one specific feature does not influence the other. The feature set representation is given by $X=(x_1, x_2, \dots, x_n)$, where the x_i represents the i^{th} feature and the outcome set yes/no is given by variable y . The formula of Naive Bayes classifier is given below:

$$P(y|x_1, \dots, x_n) \propto P \prod_{i=1}^n P(x_i|y)$$

Advantages of NB-

- Better classification in lesser training set.
- Implementation is less complex.
- It scales straightly with the quantity of indicators and information focuses, so it is exceedingly adaptable.
- It manages both persistent and discrete information.

Disadvantages of NB-

- Outcomes are based on prediction model so accuracy may get loss.
- Naive Bayes is that on the off chance that you have no events of a class mark and a specific characteristic worth together at that point the recurrence based likelihood gauge will be 0.

2.2.4. Multi-Layer Perceptron (MLP) [13]

Multilayer perceptron is generated using more than one perceptrons. This is basically a kind of feed-forwarded artificial neural network. It is inspired by the structure and functionalities of biological neural networks. Generally, by biological neural network we mean the structure and working of human brains. ANN tries to imitate the functioning of human brain by following its principles. As human cerebrum is made up of billions of nerve cells called neurons. Similarly, an artificial neural network consists of many artificial

neurons called nodes which behave as the same way a biological neuron does. Biological neurons are consists of three parts:

- Dendrites - they accepts the input from the previous layer.
- Axon - neurons are connected to each other through axon.
- Synapses- they transfer the output to the next layer neurons.

The input is taken by dendrites and sent to the nucleus, then nucleus decides whether to generate an output signal or not. If an output is fired by the nucleus, it goes to synapses through the axon to be passed onto the next layer neuron. The information received from dendrites is passed onto the cell nucleus, and then it decides whether to transfer stimuli to the next layer or to reject the signal based on some threshold value.

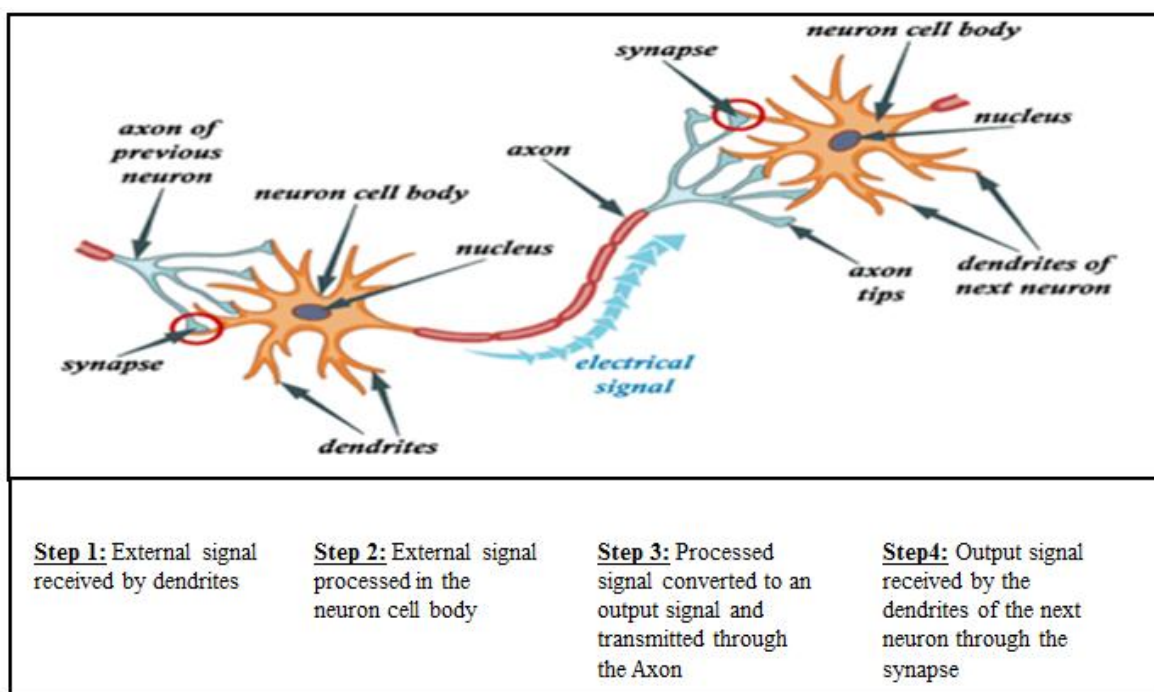


Figure 2.3: Schematic diagram of Biological Neuron

In the similar way, an ANN also consists of neurons called nodes, they are connected to each other by links and each link is associated with some weight. A node receives input from many nodes and based upon the activation function used in a node, action is taken or rejected.

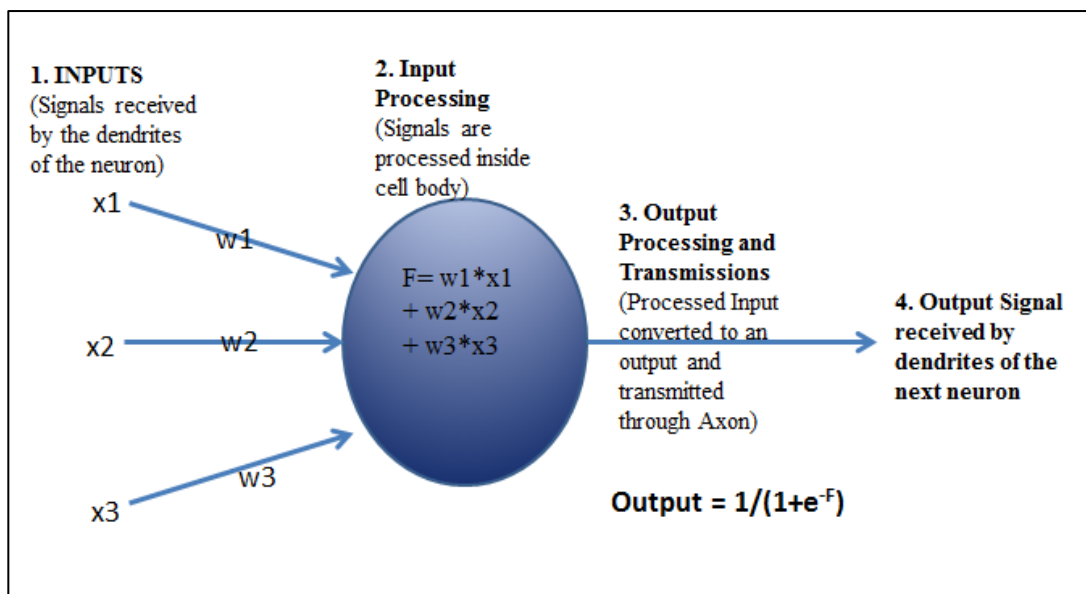


Figure 2.4: Mathematical model of an ANN's neuron/node

So, MLP is basically a feed-forward neural network using input, output and hidden layers with primarily linear relationship between the layers and the activation functions. A basic neuron consists of the structure,

$$y = w \cdot x + b \quad \dots \dots \dots \text{eq.(1)}$$

Here y = output of the layer, w = weights of the hidden layer, x = input to the hidden layer, b = bias term to be added for the neurons. MLP uses a back propagation neural network for classification to calculate the loss and optimize the values of w and b to minimize the corresponding error and in turn approximates the Bayes optimal discriminant function.

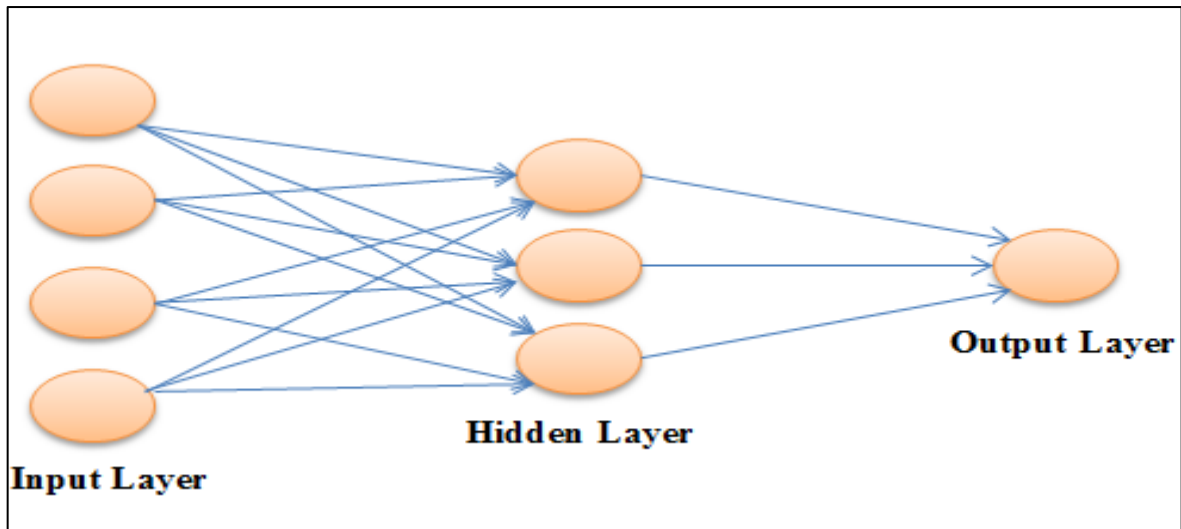


Figure 2.5: Schematic diagram of an ANN structure

Advantages of ANN-

- ANNs can learn by themselves and can create non-linear complex relationships.
- Non-linear relationships are very useful to construct, as all the real life relationships between inputs and outputs are generally non-linear.
- ANNs can be smart as humans, as after learning from a set of big training data. They can create unseen relationships between inputs and outputs. Thus they can be very useful in prediction of output on never seen input data.
- ANNs never put restrictions on input data variables. Also it is studied that ANN produces better results on heterogeneous data which has very high volatility and non-constant variance.

Disadvantages of ANN-

- **Long training times-** As ANN is used for predictions of unseen data, so this is the reason it requires a long training time to train the model. This is especially true if we are using CPU for the training instead of a GPU.
- **Need lots of data-** Architectures with many layers need lots of data to completely build the network model. As there are lots of weights and connections, therefore it requires to adjust weights according to the input data given.
- **Architecture must be fast tuned-** Only having a good data set doesn't assure you

the best result. It also requires the weights of the network to be adjusted optimally so that the predictions made later can be more accurate.

Applications of ANN-

- **Image processing and character recognition-** As images are usually consists of non-linear unstructured patterns, so ANN can work effectively on identifying the never seen objects in images based on some previous learnings. Also recognizing characters is a task where unseen images are need to be processed, so based on some mathematical calculations and prediction ANN finds the class to which an object has most compatibility.
- **Forecasting-** This is also one of the major application of ANN, as forecasting in any fields of life is mostly unpredictable so ANN tries to find the unpredictable behavior and can forecast the outcomes before the happening of the event. Which sometimes prove to be very useful- eg: weather forecasting, stock market forecasting, score predictions, etc.

2.2.5. Random Forests (RF) [2]

RF classifier uses an ensemble of trees to randomly generate the trees using the training input vector to predict the output vector, similar in analogy to generate a random set of weights, independent of the past weight sequences. Then the best one is voted in and the process is repeated for a fix number of times and the best tree is selected as the corresponding classifier. RF Classifier is based on algorithms that classify objects by using two or more classifying algorithms that may be alike or different such as SVM, Decision Tree or Naive Bayes. This algorithm is also called Ensemble algorithm. The role of the RF Classifier is to collect data or subset indiscriminately in-order to create a set of decision trees. In this algorithm, there are two parts, first uses decision algorithms to classify list or list the possibilities available. For this process it uses divide and conquer approach. The collection of the decision tree forms a forest among many trees. The algorithm uses information gain, gain ratio and these are the attribute selection processes to give a single decision tree. Now the next part is to choose the most popular, this process is done by using voting system, where each generated tree are candidate and the vote system decide the

result. The tree with highest vote is recommended, this recommender system is simple and powerful, also not complex as other non-linear classification algorithms.

Advantages of RF-

- As to remove the biases, it takes average of all the predictions, so the algorithm does not have over-fitting problem.
- By using median values or computing proximity-weighted average for replacing missing values, it manages the discontinuity by handling missing values.
- It is highly flexible and can deal problems of classification as well as regression.
- It is more precise, robust and accurate algorithm that combines many other algorithms.

Disadvantages of RF-

- Difficult to elucidate when comparing with simple decision tree method.
- It is very difficult and time consuming to construct the random forest classifier.
- The over-fitting can occur in random forest classifiers very easily.

2.2.6. ADA Boost (ADA) [5]

ADA Boost is used to improve the performance of learning algorithms. It runs small learners over various distributions of the training set and then combines them into a single composite classifier, where associated weights are taken with these classifiers and are updated to improve the training accuracy on different samples. This can sometimes bear huge individual errors in some learners but overall their composite classifier can still give good results, hence it is robust to unstable behavior of the learners. AdaBoost classifier is basically the conjugation of the numerous classifying set of rules and offer blended end result. AdaBoost stands for “Adaptive boosting” is based on meta-heuristic procedure of system getting to know in which the role of booster is to combining diverse weak classifier so one can give a vigorous classifier. The AdaBoost classifier is efficaciously running for type of binary.

The equation for the category is given below:

$$F(x) = \text{sign} \left(\sum_{m=1}^M \theta_m f_m(x) \right)$$

where, f_m stands for weak classifier and θ_m is the weight of classifier.

This process requires multiple classifiers from various selected classification algorithms and at every iteration requires to assign weight accurately for final election. And it requires also choosing training set based on accuracy achieved by it.

Advantages of ADA Boost-

- AdaBoost is a successful classifier for discrete data, and have wide performance in the field of computer vision, image processing, voice reorganization, medical science, etc.
- In AdaBoost the algorithm required external work and parameters and enhance simple implementation.
- AdaBoost is relatively powerful for feature selection.
- AdaBoost does not have the drawback of over-fitting problem for estimation.

Disadvantages of ADA Boost-

- AdaBoost requires more execution time along with higher computation.
- AdaBoost's set of rules are touchy to noisy documents and outliers.

2.2.7. K-Nearest Neighbors (KNN) [1]

K-NN is a form of non-parametric based classifier, which depends on data neighbors rather than intensive training of the parameters of the network. It is said to be one of the most basic yet effective machine learning technique which requires less computations and training time. It classifies the input data based on the class of the nearest item in the dataset using Nearest-Neighbor based distance estimation. Due to this, sometimes over fitting can be avoided and classification can be done faster than parametric learning based classifiers.

It is a supervised learning technique in which new data points are classified by looking at their k-number of neighbors. Subsequently majority class is expanded by some margin results in increased decision boundary. This procedure proceeds till every one of the data points are being classified. The new data point is being associated to the class which has majority among the entire k neighbors. Based on the input samples it creates classes on the input data by calculating the distances of data points among each other. The distance formula used by K-NN can be of many types, such as Euclidean distance, Manhattan distance, etc.

So, when a new input item comes, KNN checks the area under which the input item falls and based on that KNN predicts the class to which input item belongs. The accuracy of the model is tested by using labeled data on which training is not done. The accuracy depends on the number of points KNN is able to predict correctly to the actual classes to which they belong.

Advantages of K-NN -

- Learning process cost is almost zero- As KNN doesn't require learning time, it just draws class boundaries based on input values.
- Can be used for classification and regression - It can have applications in both the areas, classification as well as in regression.
- It evolves constantly- With the increase in the input training data, it can evolve iteratively.

Disadvantages of K-NN -

- **Slow algorithm-** KNN might be easy to implement, but as the input data increases the speed of KNN decreases quickly.
- **Problem with dimensions-** When the number of variables increases KNN struggles to predict the correct output.
- **Optimal value of K-** One of the major issue with KNN is to assume an optimal value for number of nearest neighbors to be considered when predicting the output.

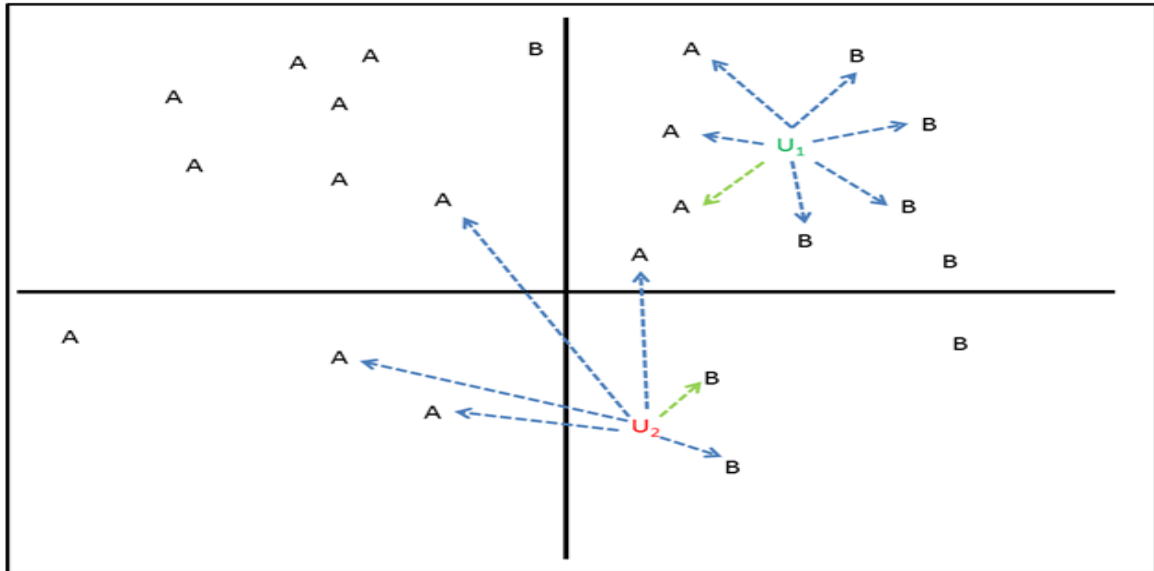


Figure 2.6: Representation of a k -Nearest Neighbor graph

2.2.8. Convolutional Neural Network (CNN) [8]

CNN is basically an extension of Artificial Neural Network which is popularly used for many applications like medical image analysis, recommender systems, natural language processing, image classification, image and video recognition etc. CNNs have huge learning capacity which makes it great for tasks such as image classification and object recognition. It uses variation of breadth and depth of features to extract features and learn from the data. The learning capacity can be varied by changing the size and number of different layers. Although convolution itself is a linear operation, non-linearity can be added using activation layers. There are many layers in CNN, which includes: Convolution layer, Activation layer, Pooling layer, Batch norm Layer, Dropout Layer, Fully connected Layer. But it is computationally expensive to train and is good at capturing spatial features compared to temporal features.

2.3 Transfer Learning

One of the powers of Machine Learning is learning something from one place and makes use of its prior knowledge in performing the similar tasks to get a better initial learning point, hence helping the loss converge faster. Optimization of very deep convolutional networks and millions of images might take up months, and retraining a new network on

such data is not quite feasible in most cases. Alternatively, if the design begins with previously learned networks, use the optimization that they have already done and form layers of our own over it? This question gave birth to the field of Transfer Learning or Knowledge Transfer, where a previously optimized network can be used as a base model for optimization for same or different datasets with room for customization.

It comes with its pros and cons. Although it helps the new classifiers get a better initial learning point, but these pre-trained networks pack a lot of baggage with them for example the VGG16 model trained on Imagenet is quite heavy to be run on normal laptops/desktops and as they are fine-tuned on millions of images, the new data should have a good chunk of samples to make a dent into the previous weights and help the network perform classification on the new data. It is although possible to make custom networks perform better than pre-trained networks with small samples as well which although for lighter weight of the models and faster inference speed but requires prior knowledge of the functions of each CNN layer and might require a lot of hit and trial. One such example is seen for the Stanford Clothing Attribute Classification Dataset [19].

CHAPTER-3

Formulation of Decision on the choice of CNNs and Framework of Convolution Layer

This module describes about feature extraction and feature selection for image specific tasks, and how the ocean of classifiers learn from the image directly or indirectly (after feature extraction or selection) to segregate them into desired classes.

3.1. Feature Extraction from Images

This section describes the image preprocessing and the effect of different processes on the original image along with its advantages and disadvantages. Later, some feature extraction techniques and their applicabilities being limited to the target task are described. This helps gain an overview about the effect of preprocessing and the need of noise addition in some of the datasets to make the features more robust for the models to learn. This leads the models to generalize in a better manner.

3.1.1 Pre-processing the Original Images

Image dataset is mapped to numeral pixel value matrix data for the RGB (Red, Green and Blue) values. The processing stage may involve changing it from the RGB to Grayscale and then performing the feature extraction and selection. The conversion to Grayscale would mean reduction of the data by ~ 67% as we would drop 2 channels which could technically help to speed up the training time which is a crucial factor sometimes in Deep CNNs which may take up to months to train. Zheng et al. implements a compact CNN [14], achieving almost the similar accuracy as RGB images on CIFAR-10 [7] dataset. This actually performs well as although the RGB matrix does have different values for color information but actually the spatial features are not lost in conversion to a gray-scale matrix. Further improvement may be there using Background/Foreground enhancement [11], that would enable the network to identify boundaries more easily rather than separating the background and foreground features itself. Although it seems promising, but in real life problems, the gray-scale classification actually is unable to differentiate two

similar objects of different colors. Gray-scale is good for tasks where only the shape of the object can enable the classifier to perform well. But such classifiers may perform poorly to differentiate out objects such as red and blue t- shirts from each other. Hence we conclude that when choosing the gray-scale preprocessing, the disadvantages of losing the color classification should be taken into account. To compensate for this lack of color differentiation, Zheng et. al, uses a histogram of bins, to store pixel counts for different ranges of bins, somewhat preserving the colored information [14]. This may have more overhead than the original gray-scale network itself, but it does preserve the color information in some magnitude. In case of Foreground enhancement, contours may be drawn over the original images for the object detection or temporal difference methods could be used, to make the actual object of interest easier to identify based on the newly created pixel differences. As seen in Figure 3.1, if the background is converted to black pixels, and the foreground is kept to high valued pixels, then it is easier to differentiate between the two.



Figure 3.1: Grayscale Image from FER-2013 dataset [20]

As seen in Figure 3.2, the task of using the single channel technique for gray-scale conversion by dropping the rest of the channels to preserve the spatial information is being done [26].

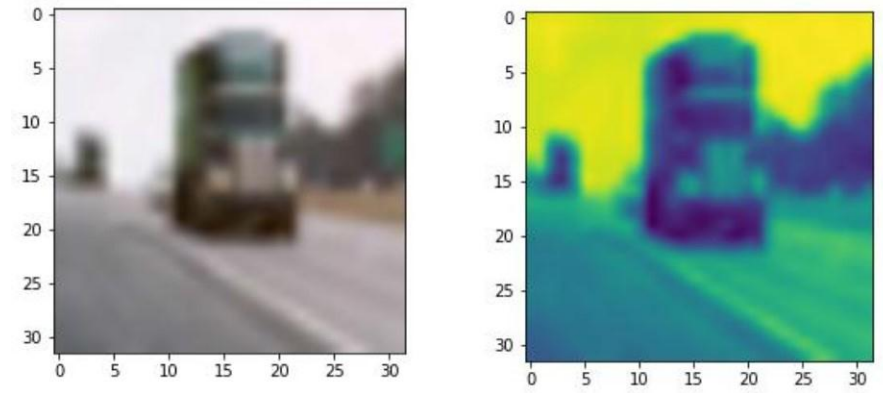


Figure 3.2: Gray-scale image conversion using channel drop [26]

If the dataset is skewed or biased towards a particular class, most classifiers will just overfit and give bad results on testing sets usually. Data Augmentation compensates for this to an extent, as it helps increase the size of the dataset and introduce more variation in the data itself as well, which will in turn help the feature extraction algorithms to get a largest set of values. Most common technique for augmentation using noise addition is using the Gaussian noise addition to the images. Other augmentation techniques involve flipping, translation, scaling and others.

3.1.2 Feature Extraction

It is important to understand the cycle of image classification given in Figure 3.3 to understand the role played by feature extraction [26]. This is one of the most crucial step of the whole learning process. It is a forward pipeline from the original data. This is explained previously in the iris dataset example where the four image attributes are passed based on the dataset of iris. The data is already provided in the dataset, but getting this data from any new image is a lot of work and usually requires human intervention for creation. The object detection or classification datasets mostly involve human classifiers to label the initial data, usually more than 1 human input is taken for it. This form of learning is called the supervised learning. Other technique for forming the features involve wavelet feature extraction, this is particularly useful for facial recognition tasks, as it forms the levels of features based on the energy wavelets as they would have reflected on the face. Popescu et. al designed 48 features for the dataset of public pollen image dataset [11]. Specific to their

task, they took features such as height, width, the dimensions of ellipse enclosing the object.

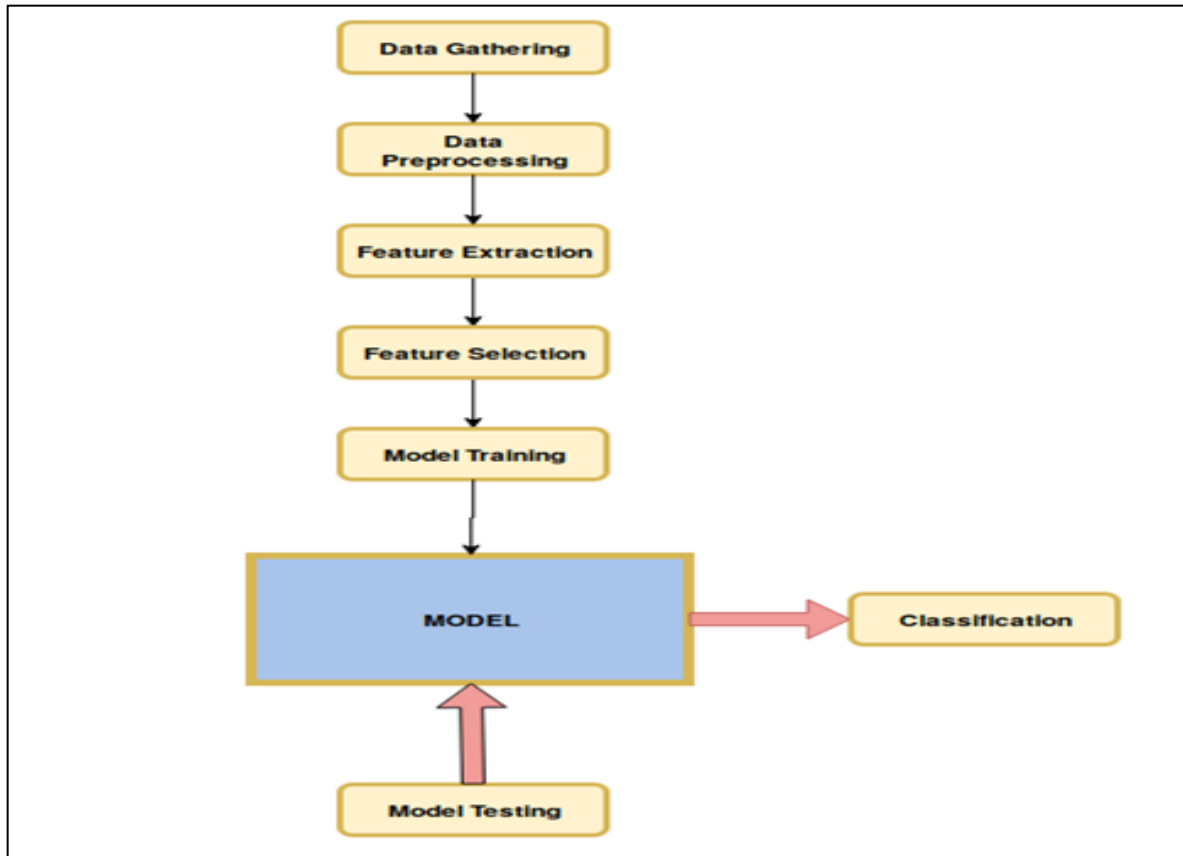


Figure 3.3: Cycle for Image Classification

This is totally different for some other form of data. For CIFAR-10 dataset, usually the automated form of CNNs or Deep CNNs is preferred. With huge number of images, it is difficult to calculate features of different objects such as height, width, color information, the contours and creating this set of numerical features and have human verification for it to pass over to the network. Principal Component Analysis is one of the most popular forms of feature extraction [11]. It is a dimensionality reduction technique, where higher number/dimensions of features can be transformed into smaller dimensions; this is different from selecting a small number of features out of all features but rather involves a transformation of the old features into a smaller set of features. This works on separating out the orthogonal components from the features into linearly uncorrelated features. For

most tasks involving a new dataset, the feature extraction has to be created in a custom manner if not using the automated method. Hence it is important to understand the dataset as it drives the process of feature extraction and it may actually involve more work designing a good feature extractor, rather than the classifier itself.

The aforementioned classifiers in related work section are being implemented and out of them CNN performed much better. That's why further work focuses on CNN models.

3.2. Basic Description of CNN Structure

This section discusses the design structure of CNN based on data and the optimization techniques that can help most of the CNNs with low dependency on the structure. Further the techniques are analyzed in a reverse manner similar to the flow of a back-propagation to demonstrate the learning cycle of a CNN.

3.2.1. Output of CNN

The output of a CNN in tasks such as classification is usually the probability of different outputs. For example, if the final output of a CNN has 4 units (usually denoted as classes), they are normalized into probability of occurrence of these classes and the unit with maximum probability is labeled as the output class. This may actually vary for task specific cases such as facial recognition, where the output is an encoded form of the input rather than the representation of output for particular class. Back-propagation helps minimizing the error by reducing the error/cost function which helps in shaping the values of the weights so that the accuracy is increased. There are different types of optimizers to help train the network by reducing the loss such as Adam optimizer, Momentum optimizer, and others, these are generally used in classification tasks. For special requirements of loss functions such as in case of facial recognition requiring embedding rather than classifying to a particular class, Triplet loss is used.

3.2.2. Fully Connected Layer

The structure from top to down usually forms a pyramid structure, the number of parameters in these layers keep on converging till they finally reach the number of desired

classes. Increasing the number of hidden units in the layer can increase the learning ability of the network, but there is saturation of the increase in accuracy of the network. There is no formulation of the units you choose; it is a hit and trial usually. There are two factors taken into account in terms of fully connected layers, the number of units in each layer and the depth of the mesh of the fully connected layers. Increasing number of units help increase accuracy initially, but reaches a saturation soon and then the accuracy starts decreasing. Depth is very useful to make hierarchy of features but too many layers may cause increase in computational cost and hence decrease in the speed of the network. Most of the networks in research usually perform well with number of units in multiple of 64. 2-3 layer networks are good if there are enough patterns being passed to the network after flattening the outputs of the convolutional layers. Generally, a convolution layer is followed by activation layer and later by dropout layer to help generalize the network. This is one of the problems with designing CNNs from scratch, that you mostly have to settle for acceptable accuracy and trying to achieve always the best solution. As most CNNs already take a long time to train, getting an acceptable range of accuracy is preferred if done within a feasible time frame.

Here transfer learning or using pre-trained proven architectures give a great starting point provided there is enough data to tune the weights to be able to classify a custom dataset. One important point to note while training the CNN is preparing for generalization. CNNs work very well on inference data if its pattern is also similar to the training data using which the CNN was trained with. But giving vastly varying data for classification can lead to abrupt results. Hence, data augmentation for robustness to noise is an important aspect. This will be very crucial to understand when we see Transfer Learning in the following sections.

3.2.3. Dropout Layer

Dropout Layer is usually applied after the layer containing neurons in the fully connected network. Dropout layer is a regularization layer. It helps to create robustness in the layer by dropping a fraction of units randomly from the previous layer usually kept around 20-50% of the original input. This helps create noisy input for the next layer and makes it more adaptable for such noisy samples.

3.2.4. Pooling layer

These are usually used in two settings, max pooling and average pooling .But recently some advanced styles of pooling such as mixed pooling and gated pooling are also used in some networks.

Max Pooling helps to reduce the dimensionality of the previous layer such as scaling down the width and height of previous layer by half by keeping only the maximum values in the nearby range, but this may cause some information loss. The concept behind these is that adjacent or nearby pixels can be approximated by the maximum information carrying pixel.

3.2.5. Activation layer

An activation layer in a convolutional neural system comprises of an activation function that takes the convolved feature map produced by the convolutional layer and makes the activation map as its yield. Activation functions are those functions which maps a specific output to a specific set of inputs. So they are used for containing the output in between 0 to 1 or -1 to 1. They are also used to impart a non-linearity in the machine learning models and are one of the important factors which affect the results and accuracy of the machine learning models. There are some important activation functions used in machine learning such as identity function, sigmoid function, binary step function, Tanh function, ReLU function, Leaky ReLU function. This layer mostly uses ReLU as an activation function. ReLU is a function which is used to set all negative values to zero and keeps positive value as it is. The ReLU Activation Function definition is: $R(z) = \max(0, z)$.

3.2.6. Convolutional Layer

Convolutional Layer introduces the concept of shared weights. The shared weights/filters in these layers usually comprises of three factors, kernel size (square matrix) (width of the kernel), stride of the convolution and number of filters. Although the parameters in these layers are less than the ones in the bottom layers, these present a computational bottleneck for the networks. Even small networks can scale up to millions of parameters, for networks such as AlexNet which has 60 million parameters. These require the most data to train and

hence enough data has to be provided for them to be optimized. Usually, the standard filter sizes used are: 7×7 , 5×5 , 3×3 and most recently 1×1 . The recent inception modules have shown that computation of smaller kernel sizes have faster computation and perform at par with large kernel sizes usually.

The other aspect comes is the depth of the filters or the number of filters. The computation scales highly as the depth increases. [21] shows the factorization of bigger convolutions into smaller convolutions and also gives the example of comparison of 5×5 layer into two 3×3 layers and the performance difference. The mathematical details of the increase in speed are presented in [21].

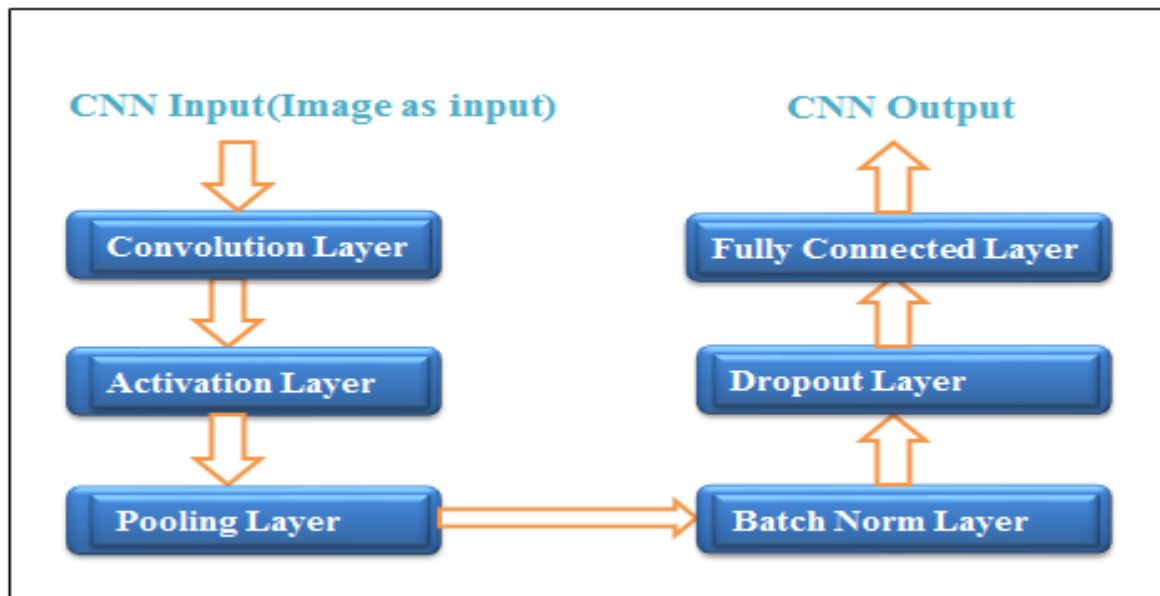


Figure 3.4: Basic structure of CNN

3.3. Decision making on choices of, pre-trained CNN or Customized CNN

The prominent finding of our work is to formulate soft guidelines for the structure of Convolution layer of CNN and to decide when a pre-trained CNN may work better or a custom design may give better results. The structure of CNN does not have a closed best solution, hence, one cannot say for a dataset that this is the best ever CNN which cannot be improved further. Hence we don't target for the best structure but rather providing a

feasible solution with the guidelines. We divide the choice of a CNN based on the availability and pattern of data, as data is very important part of a model because if there are not good patterns to be found, it is very difficult that the CNNs will give good and distinct results.

There are four categories of data which include:

1. Data with high number of images (10K+ overall, approximately 1k per class) and good differentiable pattern between different classes, e.g. one class is a flower and other class is an automobile, they will have quite different spatial patterns. In such a case, it is quite useful to develop a Pre-trained CNN. Although there might be a custom CNN which might give better results, but transfer learning will help you get a substantial solution very quickly as well. The most common choice of Pre-trained CNN is VGG16 or ResNet50. VGG has quite a heavy model size and does require a lot of epochs to train, as you are fine tuning the weights again for your classes. Custom CNN in this case if developed should have 4-6 convolutional layers and approximately 14 overall layers, similar to what the structure of these Pre-trained CNNs to give a good starting point in terms of structure, other hit and trial methods lead to a better structure as well for a particular dataset, because patterns in the data is the key to structuring the CNN, and there is no particular formula to see this degree of pattern.
2. Data with high number of images but somewhat overlapping patterns, such as two classes of very similar shaped flowers, and little difference in the color. This is a tricky one, and a custom CNN design is better in this case, because Pre-trained CNN usually are not good in picking such low differences and get biased towards a particular category. In our observation, in this case, the balance of data for each class also plays a big role, if number of samples for a class are quite high compared to other class with overlapping pattern, the Pre-trained CNNs will give biased results for the class with more samples. Hence, although on checking accuracy sometimes, we might be misled with Pre-trained giving a higher accuracy in some cases, but many of them just give biased class results and not generalized and distinct results. A Custom CNN design with large number of epochs will be very

useful in such cases.

3. Data with low number of images and distinct pattern, both the Pre-trained and custom CNN can give good results here. 2-4 convolutional layer architectures can give satisfactory results in these cases. Transfer learning seems to have a little edge in the training phase though, because they have weights already distributed to different classes compared to initial random weights of a custom design. Hence, although custom CNNs can give good results here, but if one wants to avoid the hit and trial phase, a Pre-trained CNN can give quite significant results still.
4. Data with low number of images and overlapping patterns, a custom CNN design is a better approach in this case, as discussed earlier for the case of high number of images because Pre-trained CNNs mostly give biased results for these cases.

3.4. Designing guidelines for Convolution Layer

In this section convolutional layer is discussed in detail as it is a prominent part of the whole CNN architecture. Convolutional layer comprises of independent filters which are convoluted (dot product in the case of CNN) over the input image or the output of previous layer and helps to extract spatial hierarchical features using shared weights. The filters usually consist of 4 dimensions namely, filter height, filter width, input filter dimension (the one coming from previous layer) and output filter dimension. The choice of filter height and width effect the feature identified and is usually chosen to be an odd dimension between 1-7. The filter height and width is kept the same. As seen in the work of [21], the convolutions of 3x3 and 1x1 are computationally more efficient to have as compared to higher filters such as 5x5 and 7x7. Hence it is recommended to keep the filter sizes between 1 and 3. And size 5 in some cases only, where the patterns seem quite distinct between different classes. For this, filters of size 3x3 are used in the research models. The output dimensions of the filter may vary, usually for Grayscale inputs it is kept to be 8, or for RGB input images many research models use 16 or 32 as the output channels. This is a choice of the user, but it is recommended not to go beyond a multiple of 8 of the previous channel, as the number of weights will increase and there depth of features identified may not be improved comparably in terms of the increased resources required for higher channels.

CHAPTER-4

IMPLEMENTATION AND RESULTS

4.1 Datasets Used

4.1.1 CIFAR-10 dataset [22] is a widely used images dataset. This dataset is a collection of 60,000 RGB images, which belong to 10 classes. The classes of CIFAR-10 images dataset are of dogs, cats, airplanes, deer, automobiles, birds, frogs, horses, ships and trucks. Each class is having 6,000 images. Each image is of 32*32 pixels with 3 channels for each pixel i.e. red, green and blue. These images play an important role to train the models in machine learning and convolution neural network or deep learning and basically are used to perform a classification task for object identification. From 6,000 images of each class, 5000 images are used for training and 1000 images for testing. The classes completely different from each other and not contain any overlapping properties. In convolution neural network the input image is provided and then distortion of image performed by flipping image, cropping, modifying colors and saturations and pass through different CNN layers. Each layer work in order to correctly recognize the class of the image. The accuracy of CNN model depends on the how well the training sets are trained.

4.1.2 MNIST dataset [23] contains a set of penmanship scanned images of numerals, the numerals scale from 0 to 9. Each gray-scale image is of 28*28 pixels. For exploratory data analysis we required visualizing high dimension data. We used a large dataset MNIST where M stands for modified NIST, MNIST is derived from NIST native data set. MNIST is a dataset that contains a set of penmanship scanned images of numerals, the numerals scale from 0 to 9. All the images belong to 10 classes such that the images of digit 0 belongs to class 0, the images of digit 1 belongs to class 1 and so on. The MNIST dataset contains 60,000 images for training and 10,000 images of testing. All the images of numerals in database are standardized in size. There is a data set of two dimensions x_i and y_i , each x_i is an image with 28 pixels vertical and 28 pixels horizontal, where y_i is any digit

from range 0 to 9. For example x_i , is a 28x28 image of a digit 0 then in y_i the image is belong to class 0. The objective is to classify the given object image into one of the given class from 0 to 10. The set of x_i is a column vector by converting an image into data matrix then use row or column flattening to give column vector of 784-dimensional dataset.

4.1.3 CIFAR-100 dataset [24] consists of 60,000 images belonging to 100 classes which can further be divided into 20 super-classes. This allows coarse label of training and identifying the accuracy of network in similar cases for example two objects within a class such as vehicles. Each image is of size 32*32*3 and in each class there are 600 images. Training set is of 50,000 images with 500 images from each class and testing set of 10,000 images with 100 images from each class. Classifying classes with overlapping features is more difficult than the ones with easily differentiating features. Distinguishing a flower from a truck is easier than separating a truck from a car as spatially, car and truck have more features overlapping as compared to a flower.

4.1.4 Stanford Clothing Attribute Dataset [25] consists of 1856 images and 26 ground truth clothing attributes which are collected/extracted using the Amazon Mechanical Turk. These attributes are labelled as Necktie, Collar, Gender, Placket, Skin exposure, Wear scarf, Solid pattern, Floral pattern, Spotted pattern, Graphics pattern, Plaid pattern, Striped pattern, Red color, Yellow color, Green color, Cyan color, Blue color, Purple color, Brown color, White color, Gray color, Black color, Many (>2) colors, Sleeve length, Neckline, Category in the same order. Values of some attribute entries are 'NaN', indicating no acceptable category reached by all the turks mutually. This will help us tackle the problem of missing data. We tackle this two different ways actually, one is taking any of the random classes for these and other is dropping the samples which NaNs in the results category, both produce quite different results and help us prepare better for real life data scenarios as this.

4.2 Technologies Used

4.2.1 Python

Python is a broadly utilized high – level, general – reason, translated dialect. The plan

theory of Python underlines code readability and the linguistic structure has been intended to allow software engineers to express ideas and calculations in less lines of code than conceivable in dialects, for example, C++ or Java. The availability of Python interpreters for many operating systems allows Python code to run on a wide variety of systems.

4.2.2 Anaconda and Jupyter Notebook

Anaconda is basically a bunch of popular python packages and a packet manager similar to 'pip' called 'conda'. Anaconda might be a free and open source dispersion of the Python and R programming dialects for data science and machine learning associated applications. Package forms square measure oversight by the bundle administration framework conda. The Anaconda distribution is employed by over millions of users, and it includes quite 250 in style information science packages appropriate for Windows, Linux, and MacOS. The Jupyter Notebook is AN ASCII text file internet application that permits you to form and share documents that contain live code, equations, visualizations and narrative text. Uses include: knowledge cleansing and transformation, numerical simulation, applied mathematics modeling, knowledge image, machine learning, and far a lot of.

4.2.3 Some of the standard Python Libraries Used

- import os (os: miscellaneous operating system interface) :
This module gives a versatile method for utilizing working framework subordinate usefulness.
- import time (time: Time access and conversions) :
This module gives different time-related capacities. For related usefulness, see additionally the date time and date-book modules.
- System - specific parameters and functions :
This module provides access to some variables used or maintained by the interpreter and the functions that interact strongly with the interpreter.
- Numpy :
Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.
- Matplotlib :

Matplotlib is a plotting library. Matplotlib can be used in the Jupyter Notebook, IPython Shells, Python scripts, web application servers, etc.

4.3 Implementation

The implementation of the work is done over Python language of various image classifiers such as MLR, SVM, MLP, RF, NB, K-NN, ADA, and CNN. The code implementation of Convolutional Neural Network on CIFAR-10 image dataset is given below:

4.3.1 Code

```
import numpy as np
import tensorflow as tf
from sklearn.metrics import confusion_matrix
from time import time
from include.data import get_data_set
from include.newcompact import model2
import matplotlib
import matplotlib.pyplot as plt

train_x, train_y, train_l = get_data_set()
test_x, test_y, test_l = get_data_set("test")
train_orig = train_x
test_orig = test_x

# Reshaping the images
train_x = np.reshape(train_x, (50000, 32, 32, 3))
test_x = np.reshape(test_x, (10000, 32, 32, 3))
train_orig = np.reshape(train_orig, (50000, 32, 32, 3))
test_orig = np.reshape(test_orig, (10000, 32, 32, 3))

# Testing the Histograms
x, y, output, global_step, y_pred_cls, counts_list = model2()

_IMG_SIZE = 24
_NUM_CHANNELS = 3
_BATCH_SIZE = 128
_CLASS_SIZE = 10
_ITERATION = 10000
_SAVE_PATH = "./tensorboard/cifar-10_new_bins/"
```

```

loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=output, labels=y))
optimizer=tf.train.RMSPropOptimizer(learning_rate=1e-3).minimize(loss,
global_step=global_step)

correct_prediction = tf.equal(y_pred_cls, tf.argmax(y, axis=1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
tf.summary.scalar("Accuracy/train", accuracy)

merged = tf.summary.merge_all()
saver = tf.train.Saver()
sess = tf.Session()
train_writer = tf.summary.FileWriter(_SAVE_PATH, sess.graph)
def train(num_iterations):
    """
    Train CNN
    """
    for i in range(num_iterations):
        randidx = np.random.randint(len(train_y), size=_BATCH_SIZE)
        batch_xs = train_x[randidx]
        batch_ys = train_y[randidx]
        start_time = time()
        i_global, _ = sess.run([global_step, optimizer], feed_dict={x: batch_xs, y: batch_ys})
        duration = time() - start_time

        if (i_global % 10 == 0) or (i == num_iterations - 1):
            _loss, batch_acc = sess.run([loss, accuracy], feed_dict={x: batch_xs, y: batch_ys})
            msg = "Global Step: {0:>6}, accuracy: {1:>6.1%}, loss = {2:.2f} ({3:.1f}
examples/sec, {4:.2f} sec/batch)"
            print(msg.format(i_global, batch_acc, _loss, _BATCH_SIZE / duration, duration))

            if (i_global % 100 == 0) or (i == num_iterations - 1):
                saver.save(sess, save_path=_SAVE_PATH, global_step=global_step)
                print("Saved checkpoint.")

    if (i_global % 100 == 0) or (i == num_iterations - 1):
        data_merged, global_1 = sess.run([merged, global_step], feed_dict={x: batch_xs, y:
batch_ys})
        acc = predict_test()

```

```

summary = tf.Summary(value=[
    tf.Summary.Value(tag="Accuracy/test", simple_value=acc),
])
train_writer.add_summary(data_merged, global_1)
train_writer.add_summary(summary, global_1)
def predict_test(show_confusion_matrix=False):
    """
    Make prediction for all images in test_x
    """
    i = 0
    predicted_class = np.zeros(shape=test_x.shape[0], dtype=np.int)
    while( (i < int(test_x.shape[0])) and (i+128<int(test_x.shape[0]))):
        j = min(i + 128, int(test_x.shape[0]))
        batch_xs = test_x[i:j, :, :, :]
        batch_ys = test_y[i:j, :]
        predicted_class[i:j] = sess.run(y_pred_cls, feed_dict={x: batch_xs, y: batch_ys})
        i = j
    correct = (np.argmax(test_y, axis=1) == predicted_class)
    acc = correct.mean()*100
    correct_numbers = correct.sum()
    print("Accuracy on Test-Set: {0:.2f}% ({1} / {2})".format(acc, correct_numbers,
test_x.shape[0]))
    if show_confusion_matrix is True:
        cm = confusion_matrix(y_true=np.argmax(test_y, axis=0), y_pred=predicted_class)
        for i in range(_CLASS_SIZE):
            class_name = "({}) {}".format(i, test_l[i])
            print(cm[i, :], class_name)
        class_numbers = ["({})".format(i) for i in range(_CLASS_SIZE)]
        print("".join(class_numbers))
    return acc
if _ITERATION != 0:
    train(_ITERATION)
sess.close()

```

Similarly, other classifiers are also implemented and their code is attached in the Compact Disk (CD).

4.4 Results and analysis

This report includes implementation of MLR, SVM, MLP, RF, NB, K-NN, ADA, CNN classifiers using Python 3.6, Jupyter Notebook IDE on Windows OS with 8 GB RAM, Intel i7 (4th Gen Processor). The datasets used in this work are CIFAR-10 and MNIST images datasets.

The accuracy results of the classifiers on validation and test data for CIFAR-10 is presented in **Table 4.1** using random shuffle and division to choose the 40,000 training, 10,000 validation and 10,000 testing images.

Table 4.1 Comparative analysis of accuracies of Image Classifiers on CIFAR-10 dataset

S.No.	Classifier	Validation Accuracy	Testing Accuracy
1	Multinomial Logistic Regression	40.64%	40.22%
2	Support Vector Machine	42.55%	41.94%
3	Random Forest Classifier	26.57%	25.74%
4	Multi-Layer Perceptron	43.52%	43.34%
5	K-Nearest Neighbor	31.94%	31.37%
6	Ada Boost	30.67%	30.34%
7	Naive Bayes	29.44%	28.89%
8	Convolutional Neural Network	80.50%	65.54%

Table 4.2 presents the cross-validation accuracies and validation set accuracies for MNIST dataset.

Table 4.2 Classifiers' Accuracies on MNIST dataset

S.No.	Classifier	Cross-Validation Accuracy	Validation Accuracy
1	Multinomial Logistic Regression	[91.20%, 92.22%, 91.41%, 91.71%, 92.64%]	92.60%
2	Support Vector Machine	-----	94.39%
3	Random Forest Classifier	[62.24%, 63.28%, 63.73%, 62.36%, 66.34%]	64.70%
4	Multi-Layer Perceptron	[94.81%, 95.16%, 95.22%, 93.55%, 95.79%]	95.63%
5	K-Nearest Neighbor	[96.80%, 96.88%, 96.89%, 96.59%, 97.08%]	97.00%
6	Ada Boost	[72.49%, 70.14%, 70.52%, 70.77%, 75.58%]	71.26%
7	Naive Bayes	[55.04%, 56.02%, 55.13%, 54.65%, 55.89%]	54.92%
8	Convolutional Neural Network	-----	98.00%

Cross-validation is not done on CNN and SVM due to their high computational cost and the variation of Cross-validation accuracies was observed to be only around 1% in case of other classifiers, hence we skip the cross validation for these two. For Logistic Regression, we used multinomial logistic regression with Limited- memory BFGS (LBFGS) optimizer and L2 penalty. SVM Support Vector Classifier (SVC) tuned with penalty parameter 1.0, rbf kernel and 3 degree polynomial kernel function. Random Forest Classifier is implemented with 100 trees for estimation with a maximum depth of 2 using “gini” criterion. Multi-Layer Perceptron (MLP) classifier is used with a setting of 100 hidden layers, Rectified Linear Unit (ReLU) activation, Adam optimizer and L2 penalty parameter as 1. KNN classifier uses 3 nearest neighbors and uniform weight initialization. Ada Boost Classifier implements 50 estimators with a learning rate of 1.0 using SAMME.R estimator. Multinomial Naive Bayes Classifier is done using additive smoothing parameters with enabled calculation of priors on data. CNN used for MNIST consists of four layers, one convolution layer with kernel size of 3 and 8 output channels, max pooling layer with kernel and stride of 2 each, fully connected layers with 150 and 10 hidden units output each. For CNN used in CIFAR-10 gray-scale network, we center crop the images from (32, 32, 3) to (24, 24, 3) and then use the channel drop technique to keep a single channel to convert the input to (24, 24, 1). The network consists of 4 convolution layers with ReLU activation and 2 of these layers actually work as an inception type model as the filter and stride are set to 1, then 3 normalization layers are used with 2 pool layers along with 2 fully connected layers at the bottom of the model. The results from the table show that CNN outperforms the other classifiers, which is partly due to their high learning capability.

Logistic Regression, Naive Bayes Classifier are linear classifiers and hence have limited capability to capture non-linearity between the input data and output classes. Non- linear classifiers, especially CNN with activation functions can capture a higher degree of relationship. For the sake of simplicity, only a single model of CNN is implemented. RGB CIFAR-10 images are cropped and converted to gray-scale before passing to the CNN. As CNNs capture spatial information, the accuracy drop is not too high in comparison to the

gain in the speed of implementation.

Further the work is done on Windows OS with i7-4th Gen CPU using python frameworks in Jupyter Notebook IDE such as Keras and Tensorflow. Here, custom CNNs trained on Clothing Attribute Dataset did not provide great results on small networks and also the results varied highly based on categories based on the variation of patterns in each particular category for example 'Black Color' category. A VGG16 pre-trained model of Black Color attribute is being trained from the above dataset and compared it to a custom CNN and the result was highly dissatisfying for pre-trained CNNs. There was a lot of missing or unlabeled data in this category, there were choices to either drop such data or to label it randomly to one of the categories and pass to the system for training. The later had an adverse effect on the custom CNN designs, as initial learning is very important for such CNNs to be able to differentiate with pixels into different categories. CNNs take huge advantage of hierarchical spatial features, and black color category corresponds to color attributes and also with corrupt (random labeled data which was previously missing) the features are even harder to classify. We cropped the image around centre so that the background color does not contribute to the color classification of the clothing. Custom design of 4 convolutional layers was able to provide average results for the classification, but many results were biased towards the black color. For pre-trained VGG, the change in output probability was very low for white color, and was mostly biased on black color. This proved our hypothesis of training a custom CNN for less images and difficult to identify features. Good results were expected from Pre-trained CNN from this as well, but due to missing or corrupt data, the features could not change the network weights significantly as there are millions of weights to be optimized in such a big network.

During training, the pre-trained CNN displayed a training accuracy of 82% after 30 epochs, compared to 80% in a 4 layer designed CNN (this could be improved further for different architectures, but this was considered a satisfactory network to verify the hypothesis). But the test results were only biased to black for pre-trained CNN, hence this measure for Pre-trained CNN is considered incorrect whereas the custom CNN did produce different classes for a variety of samples, but it also showed hints of bias due to quantity of data tilted

towards black data. [Clothing reference] provides results of their custom CNN vs Pre-trained model.

For CIFAR – 100, custom CNN that was created for overall 14 layers achieved an accuracy of 49% whereas the VGG 16 transfer learning model for CIFAR 100 achieves accuracy of almost 69% for our model. This proves our hypothesis that for datasets with high number of images and distinct features can be trained easily using a pre trained CNN and would require more hit and trial in case of custom CNNs, hence it is easier to start with pre trained CNNs in such cases. The results are shown in below Figure 4.1 and also the classifiers' accuracies are depicted in Table 4.3.

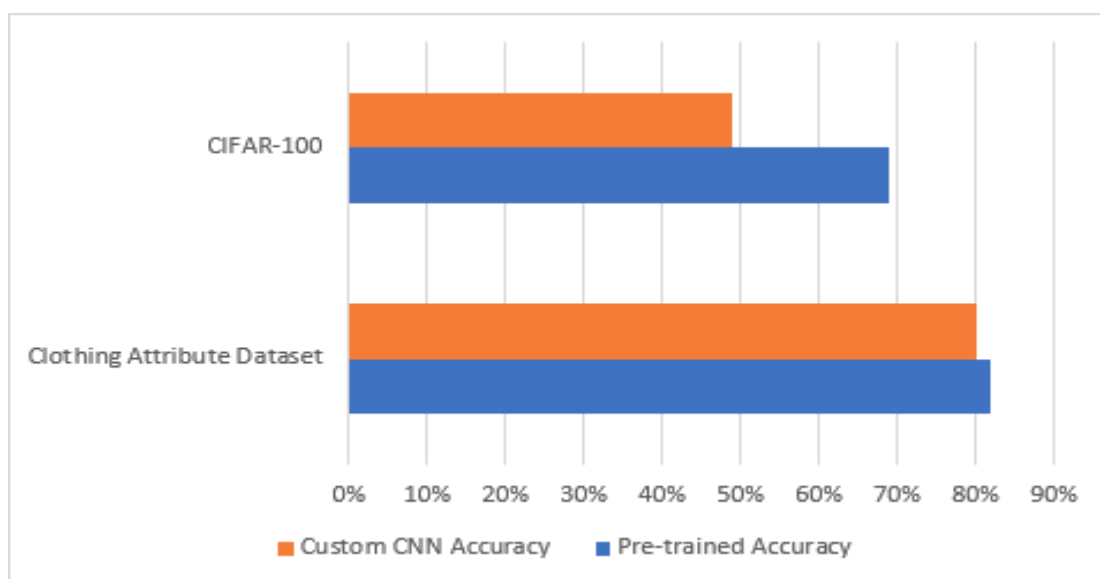


Figure 4.1: Performance of Custom and Pre-Trained CNNs on CIFAR-100 and Stanford Clothing Attribute dataset

Table 4.3 Classifiers' Accuracies on CIFAR-100 and Clothing Attribute Dataset

Dataset	Pre-trained CNN	Custom CNN
CIFAR-100	69%	49%
Clothing Attribute Dataset	82%	80%

CHAPTER-5

CONCLUSION AND FUTURE WORK

Convolutional neural networks have emerged as the most reliable deep learning technique amongst the existing techniques. They outperformed other classifiers such as Logistic Regression, SVM, Random Forest, Multi-Layer Perceptron, KNN, Ada Boost, and Nave Bayes in task of image classification on CIFAR-10 and MNIST images datasets. Their use in image classification can improve the efficiency of the prediction model to great extent. CNNs include weight initialization, bias initialization, learning rate and other layer related parameters to be initialized, setting them appropriately helps in converging to the results faster. This work describes the design of CNNs and develops some soft guidelines based on the data, the number of images and patterns in the images. Experiments have been performed in order to decide when to choose custom CNNs of less or more layers and when to choose transfer learning or pre-trained models for better solutions.

Currently, the major focus is on four sections which include firstly, the Data with a high number of images (10K+ overall, approximately 1k per class) and also good differentiable patterns between different classes. For instance, one class is of a flower and other class is an automobile, they have quite different spatial patterns. Secondly, Data with a high number of images but somewhat overlapping patterns, such as two classes of very similar shaped flowers, and the little difference in their color. Thirdly, Data with a low number of images and distinct patterns too and in the last, Data with a low number of images and overlapping patterns as well.

The future work aims at exploring more mathematical bounds of different kinds of layers and how their combined architecture might affect the patterns in one of these data as well. The work initially includes implementing the project over larger datasets that are greater in number than that of CIFAR-100 dataset and Stanford Clothing attribute dataset. Also the implementation to formulize the current theory as well for different frameworks and analyzing the effect on varying image data such as RGB, Grayscale, and HSV is to be done.

CHAPTER-6

REFERENCES

- [1] Boiman, Oren, Eli Shechtman, and Michal Irani. "In defense of nearest-neighbor based image classification." *Computer Vision and Pattern Recognition*, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008.
- [2] Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32
- [3] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- [4] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository, (<http://archive.ics.uci.edu/ml>), Irvine, CA: University of California, School of Information and Computer Science, Iris Data Set, Fischer 1936
- [5] Freund, Yoav, Robert Schapire, and Naoki Abe. "A short introduction to boosting." *Journal-Japanese Society For Artificial Intelligence* 14.771-780 (1999): 1612.
- [6] Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015 38th International Convention on (pp. 1200-1205). IEEE.
- [7] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
- [8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.
- [9] Kwak, C., & Clayton-Matthews, A. (2002). Multinomial logistic regression. *Nursing research*, 51(6), 404-410.
- [10] Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5), 823-870.
- [11] Popescu, M. C., & Sasu, L. M. (2014, May). Feature extraction, feature selection

- and machine learning for image classification: A case study. In Optimization of Electrical and Electronic Equipment (OPTIM), 2014 International Conference on (pp. 968-973). IEEE.
- [12] Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. New York: IBM, 2001.
- [13] Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., & Suter, B. W.(1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks, 1(4), 296-298.
- [14] Zheng, Z., Li, Z., Nagar, A., & Kang, W. (2015). Compact deep convolutional neural networks for image classification. ICMEW, 1-6.
- [15] ClassificationDataResults,CIFAR-10.
(http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130)
- [16] Boureau, Y. L., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 111-118).
- [17] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.
- [18] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- [19] Patki, R., & Suresha, S. (2016). Apparel classification using CNNs. Unpublishedresults.
- [20] Challenges in Representation Learning: Facial Expression Recognition Challenge (<https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>).
- [21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
- [22] Krizhevsky, A., & Hinton, G. (2010). Convolutional deep belief networks on cifar

10. Unpublished manuscript, 40(7), 1-9.
- [23] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. arXiv preprint arXiv:1702.05373.
- [24] The CIFAR-100 dataset. (<https://www.cs.toronto.edu/~kriz/cifar.html>).
- [25] Chen, H., Gallagher, A., & Girod, B. (2012, October). Describing clothing by semantic attributes. In European conference on computer vision (pp. 609-623). Springer, Berlin, Heidelberg.
- [26] Dahiya,S.,Tyagi,R.,&Gaba,N.(2019). Comparison of ML classifiers for ImageData. Proceedings in "INTERNATIONAL CONFERENCE ON INTELLIGENT MACHINES" (ICIM - March 2019).