

Detection of handwritten text using Convolutional Neural Network

by

Student Name: Rabib Bin Jasim

Student ID: 12221015

Student Name: Rokeya Sultana Mahin

Student ID:15101135

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
April 2019

© 2019. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rabib Bin Jasim

12221015

Rokeya Sultana Mahin

15101135

Approval

The thesis/project titled “Detection of handwritten text using Convolutional Neural Network” submitted by

1. Rabib Bin Jasim (12221015)
2. Rokeya Sultana Mahin (15101135)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on December 24, 2019.

Examining Committee:

Supervisor:
(Member)

Jia Uddin

Associate Professor(On Leave)
Dept. of CSE
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement (Optional)

Abstract

Machine replication of human functions, like reading, is an ancient dream. However, over the last five decades, machine reading has grown from a dream to reality. We have tried to make it more obvious through a hand writing recognition system. This research paper describes a text-line extraction based method. It offers a new solution to traditional handwriting recognition techniques using concepts of Deep learning and computer vision. An image can have hand writing, typed letters, different characters and other images. Our intention is to detect all the characters and display them. Some images can also have unnecessary lines or unclear letters. This system will clear the picture through pre-processing system and will be able to identify the letters or characters. It will help people to identify any unclear messages. It will also avoid unnecessary images and will focus on the text only. Sometimes we want to ignore unnecessary advertisement images from the newspapers. Our system will do a great work for this. It will clear all the images and unnecessary lines etc. and will only display the text what people want to read.

Keywords:

Dedication

I would like to dedicate this thesis to my loving parents . . .

Acknowledgement

We would like to thank Almighty Allah for providing us the opportunity to complete our thesis without any sort of difficulties. A major thanks goes to our dear Advisor Dr. Jia Uddin. For his constant support, feedback and advice during the process of our work we have made it happen. He helped us in every possible way and shaped our thesis in what it is today. We would also like to thank our whole judging panel for their reviews and feedbacks. All the feedback and recommendations they gave us which helped us a lot. And last but not the least, our parents, friends and mentors who helped us throughout the journey. Without their support it may not be possible at all.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	1
1.3 Thesis orientation	2
2 Background Study	3
2.1 Literature review	3
2.2 Image Processing	4
2.3 Purpose of Image processing	4
2.4 Types of image processing	5
2.5 Fundamental steps of digital image processing	5
2.5.1 Image Acquisition	6
2.5.2 Image Enhancement	6
2.5.3 Image Restoration	7
2.5.4 Color Image Processing	7
2.5.5 Wavelets and Multi resolution	7
2.5.6 Compression	7
2.5.7 Morphological processing	7
2.5.8 Segmentation	8
2.5.9 Representation and description	8

2.5.10	Object recognition	8
2.5.11	Knowledge base	8
2.6	Algorithms	9
2.6.1	Support Vector Machine (SVM)	9
2.6.2	K-nearest Neighbor (KNN)	12
2.7	Convolutional Network Architectures	13
2.7.1	LeNet-5 (1998)	13
2.7.2	AlexNet	14
2.7.3	ZFNet	14
2.7.4	GoogLeNet	15
2.7.5	VGGNet	15
2.7.6	ResNet	15
3	Proposed method	17
3.1	Data Training	17
4	Experimental tests and results	18
4.1	Dataset	18
4.2	Training	18
4.2.1	Character Segmentation and Prediction	20
5	Conclusion	22
	Bibliography	23

List of Figures

1.1	Example of a handwritten document	2
2.1	Finding a line to separate	9
2.2	Green line separates these classes	9
2.3	Optimal hyperplanes	10
2.4	Support vectors	10
2.5	2D and 3D feature hyperplane	11
2.6	Finding a line	11
2.7	Line separation	11
2.8	Line changing into circle	12
2.9	KNN example	13
2.10	LeNet5	14
2.11	AlexNet	14
2.12	ZFNet	14
2.13	GoogLeNet	15
2.14	VGGNet	15
2.15	ResNet	16
3.1	Workflow	17
4.1	First and Second Segment Prediction (Bat First)	18
4.2	First and Second Segment Run Prediction (Bat Second)	19

List of Tables

4.1	First and Second Coefficient (Bat First)	19
4.2	First and Second Segment P-value (Bat First)	19
4.3	First and Second Segment Coefficient (Bat Second)	20
4.4	First and Second Segment P-value (Bat Second)	20

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ϵ Epsilon

v Upsilon

IPL Indian Premier League

LBW Leg before Wicket

MR Runs scored by Home team

MRN Home Team Run Rate

ODI One day International

OR Runs scored by the opponent team

ORN Opponent Team Run Rate

T20 Twenty Twenty

Chapter 1

Introduction

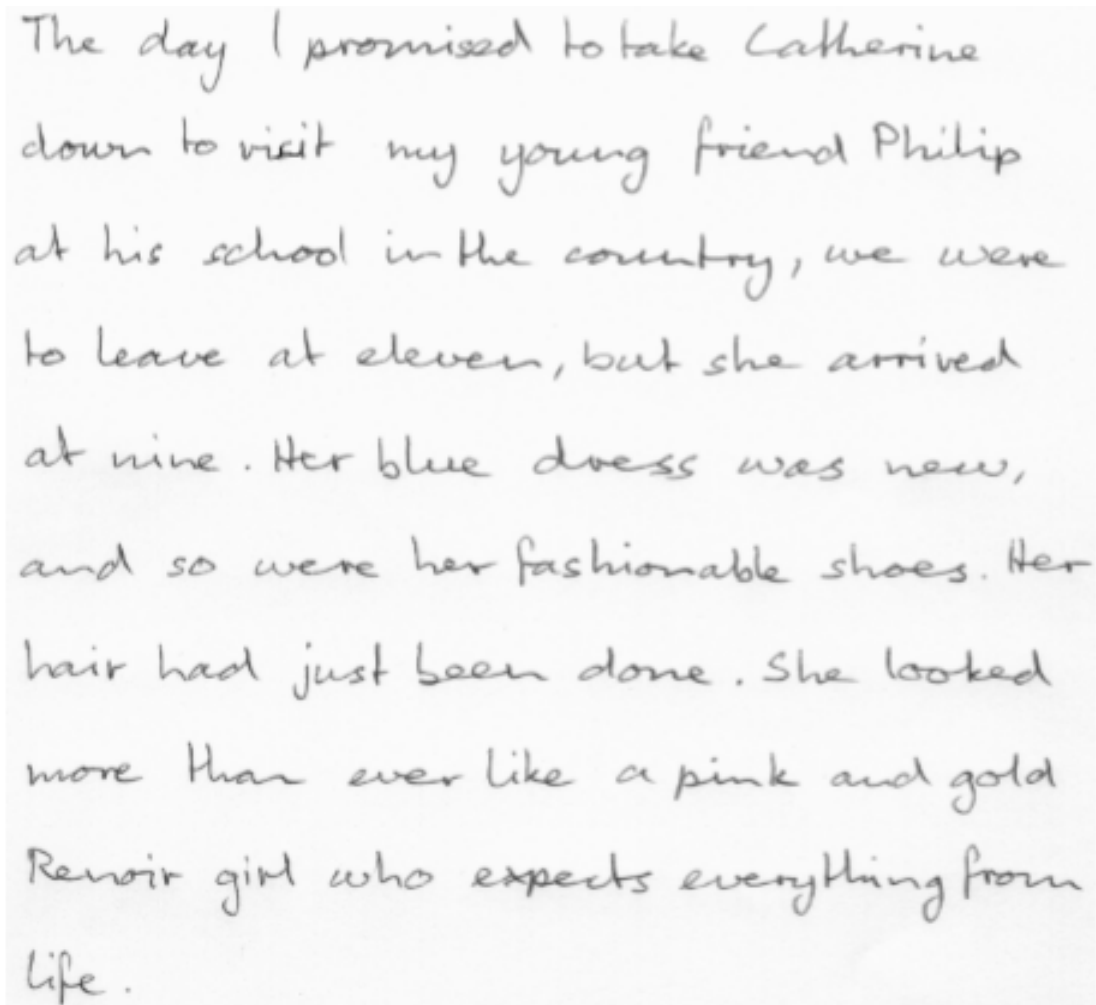
1.1 Motivation

In the modern world, text detection has become a major part of many important applications and industries to shape up the user experience and saving up enough time for more work. Text detection is an old topic on which several research has been done. In order to achieve more efficiency and accuracy we can work on the remaining scopes for improvement. Text detection is basically a method of identifying characters based on their shape, color, size, characteristics and much more. The primary approach is to construct a computer vision that emulates the detection process of normal vision. The color based approach is the mostly used text detection method used because of its efficiency. However, the pace of the detection needs improvement.

1.2 Objectives

In order to extract information from images, scholars from all over the world applied many methods. Segmentation is one of the most important steps for information extraction from images. At present, optimal scale calculation method mainly uses calculation models, expert experience, text functions, and much more.

This paper proposes a new considerable option for text detection in Bangladesh through OpenCV library. OpenCV (Open Source Computer Vision) is one of the most widely used machine learning software library in order to solve computer vision problems. OpenCVPython is a fast Python API for OpenCV.

A photograph of a piece of white paper with handwritten text in black ink. The text is written in a cursive, slightly slanted script. The paper is slightly wrinkled and has a soft shadow on the right side, suggesting it's resting on a surface. The text describes a day where the writer promised to take Catherine to visit Philip at his school, but she arrived early and looked very fashionable.

The day I promised to take Catherine down to visit my young friend Philip at his school in the country, we were to leave at eleven, but she arrived at nine. Her blue dress was new, and so were her fashionable shoes. Her hair had just been done. She looked more than ever like a pink and gold Renoir girl who expects everything from life.

Figure 1.1: Example of a handwritten document

1.3 Thesis orientation

The rest of the thesis is organized as follows :

1. Chapter 2 discusses the fundamental of image processing and algorithms.
2. Chapter 3 presents the working procedure of the research.
3. Chapter 4 demonstrates the results found in our research.
4. Chapter 5 concludes the thesis and states the future research direction.

Chapter 2

Background Study

2.1 Literature review

In recent years, text detection using image processing has been a major area of research, with various approaches proposed by researchers. Several studies and papers in this field are worth mentioning:

1. **IEEE Paper by Supriya Das, Purnendu Banerjee, Bhagesh Seraogi, Himadri Majumder, Srinivas Mukkamala, Rahul Roy, Bidyut Baran Chaudhuri:**
 - Conducted text detection at the word level of AEC documents.
 - Utilized a standard classifier based on Support Vector Machines (SVM) for text classification.
 - Achieved an overall accuracy of 98.45
2. **Paper by A. Zahour, B. Taconet, P. Mercy, S. Ramdane:**
 - Focused on Arabic text line extraction from printed binary documents.
 - Used horizontal projection analysis and connected component regrouping for segmentation.
 - Worked with a database containing handwritten Arabic texts from various writers.
3. **Kevin Hughes:**
 - Although not a formal research paper, Kevin Hughes contributed to the field by providing blogs, project tutorials, and instructions related to using OpenCV, covering various aspects of computer vision and image processing, including text detection.
4. **IEEE Paper by Muhammad Labiyb Afakh, Anhar Risnumawan, Martianda Erste Anggraeni, Mohamad Nasyir Tamara, Endah Suryawati Ningrum:**
 - Focused on distinguishing between handwritten and machine-printed text in Bangla document images.

- Presented a classification system for connected components in document images.
- Achieved an overall accuracy of 94.49

Text line extraction is a crucial operation in the development of an optical handwriting reading system. This system is designed to clear unnecessary images, such as advertisements, and extract text, even from images with unclear or old content, providing a valuable solution for users.

2.2 Image Processing

Image processing is a crucial tool used to dissect an image into its constituent parts and perform various operations on those sub-parts to extract valuable features. Essentially, it digitally transforms an image to unveil key characteristics, allowing us to understand patterns and recognize important elements. This field plays a significant role in computer science, where it is employed for tasks such as person recognition, object characterization, and much more.

Image processing typically involves three fundamental steps:

1. **Image Acquisition:** The initial step involves the scanning of the image. This can be achieved through digital photography or by using an optical scanner.
2. **Image Manipulation:** The acquired image is then processed and manipulated to meet specific requirements. Various techniques are applied to enhance or modify the image.
3. **Feature Extraction and Analysis:** Subsequently, the data from the image is analyzed to extract essential features that may not be visible to the naked eye. This step is vital for identifying patterns or significant elements within the image.

Based on the processing steps, the output is generated, fulfilling the desired requirements, which could be patterns or an entirely new image **reference3**.

2.3 Purpose of Image processing

Image processing serves various critical purposes in different fields. Here are five key reasons why image processing is essential:

1. **Visualization:** Image processing allows for the discovery of elements within an image that are otherwise not visible to the naked eye. It enhances the image to reveal hidden details and patterns.
2. **Image Restoration:** Image processing techniques are used to restore and improve the quality of images. This is especially important for producing clearer and more accurate pictures, particularly when dealing with degraded or noisy images.

3. **Retrieval of Image:** Image processing enables the selective retrieval of essential parts of an image while filtering out irrelevant or unimportant information. This is valuable for efficient data storage and retrieval.
4. **Pattern Recognition:** Image processing methods segment images into distinct parts and analyze each part independently. This process is crucial for identifying patterns or specific features within the image.
5. **Recognition of Image:** After individual image components are analyzed, image processing techniques can be employed to understand the meaning and significance of these components. This is fundamental for tasks such as object recognition and interpretation.

2.4 Types of image processing

Analog and Digital image processing are two types of image processing methods that gives us output in different ways. The analog system is mainly on the hard copies where the papers are scanned or they are being photocopies. It is also used for printing purpose. The result of the image processing is not just what computer processes it for but also how the user interprets it to be. The analyst uses various engineering techniques and their own experience in the field to bring about the best possible outcome.

An image in the beginning has no value as it cannot be interpreted to valuable features. The raw data of the images would only make sense when a certain or various patterns can be found to match our requirements. Here comes the processing by the computers of the image where it translates raw data into something that is useful. In order for that to happen the image has to undergo several phases to convert the raw dataset into a handy result.

2.5 Fundamental steps of digital image processing

Digital image processing involves several fundamental steps to manipulate and analyze images. Here is an overview of these key steps:

1. **Image Acquisition:** The process of capturing or obtaining digital images, often through devices like cameras or scanners.
2. **Image Enhancement:** Techniques used to improve the visual quality of an image, making it more suitable for analysis or display.
3. **Image Restoration:** Methods aimed at restoring images that may have been degraded due to noise, distortion, or other factors, resulting in a clearer representation.
4. **Color Image Processing:** Specific techniques designed to process color images, allowing for the manipulation and analysis of color information.
5. **Wavelets and Multi-resolution Processing:** Approaches that leverage wavelet transforms and multi-resolution analysis to enhance image features and reduce noise.

6. **Compression:** Methods for reducing the size of digital images while preserving important information, often used for efficient storage and transmission.
7. **Morphological Processing:** Morphological operations like dilation, erosion, and opening, used for extracting image components and analyzing shapes.
8. **Segmentation:** The process of dividing an image into meaningful regions or objects to facilitate further analysis.
9. **Representation and Description:** Techniques for representing and describing image features and objects, making them suitable for recognition and analysis.
10. **Object Recognition:** The task of identifying and categorizing objects or patterns within an image based on their features.
11. **Knowledge Base:** Utilizing domain-specific knowledge or databases to aid in image analysis and interpretation.

These fundamental steps form the foundation of digital image processing, enabling various applications in fields such as computer vision, medical imaging, and remote sensing.

2.5.1 Image Acquisition

The first and fundamental stage of digital image processing is the acquisition of the image. The complexity of this task can vary depending on whether the image is already in digital format or if it needs to be converted into a digital form.

In cases where the image is already in digital form, this step is relatively straightforward. However, when dealing with non-digital images, the process involves the conversion of the image into a digital format.

During the acquisition process, it is common to perform preprocessing tasks, which may include image scaling. The image is captured by a sensor or device, and the light from the energy source illuminates the scene elements. Subsequently, this reflected light forms an image on an internal image plane.

To be suitable for digital processing, the output from the image acquisition step is digitized, converting the image into a set of discrete digital values that can be manipulated and analyzed using digital image processing techniques

2.5.2 Image Enhancement

Enhancement of the image is basically turning the image into a clearer version of it. In this case the image is neither compressed nor the resolution is increased. What it does is that it highlights the raw image and brings in a more visible picture. It could also change the hue or saturation to make the image more vibrant. There are tons of apps now that can bring details via these methods **13**.

2.5.3 Image Restoration

Image restoration also deals with improving the quality of the picture but in a different way than the enhancement technique. It fills up the missing data in the image using mathematical models. One of the techniques are probabilistic models of the degradation of image. Several white marks are being removed to reduce the noise and recover the resolution loss using the algorithm ‘deconvolution’ **akhand2016cnn**.

2.5.4 Color Image Processing

This determines the color of the image to compare with other sets of images. Color can be defined in three different ways. The RGB,CMY and YIQ models. Another vital image processing application includes HSL(Hue,Saturation and Light). Using these components of the colour, the image is translated into the digital domain **reference13**.

2.5.5 Wavelets and Multi resolution

Wavelets represents an image into multiresolution forms. It has the ability to reduce the size of the image so that it can be processed faster through compression. It can further display the data into a pyramidal representation **reference13**.

2.5.6 Compression

Compression is a technique that reduces the size of the picture by reducing the pixels from it. The image will still be understandable but the quality will take a hit. Even though the quality will degrade it is necessary to compress the data for a storage that is limited. Lower image resolution would mean greater number of images can be transferred through a particular bandwidth. This also saves a huge load of time during processing as the read time will be much lower. In the following figure the picture lost some of the resolution in order to accommodate with the size available **reference14**.

2.5.7 Morphological processing

Morphology describes the structure and the shape of an image. It uses various non-linear methods to extract and determine morphology. It refers to a mathematical tool that finds the geometric structure of binary and gray scale image. It is especially suitable for binary images. Although it can be applied in gray scale images in a way that their light transfer functions of light transfer are oblivious and their absolute values of pixel are of no use or a very tender interest. It is based on set theory. Morphological image processing probes an image with structural element which is directed in all the corners of an image. These elements are compared with their neighboring elements. Some of the elements check whether it fits in the neighborhood whereas others examine whether they intersect with the neighborhood or not. The goal of this of this image processing is to differentiate information of shape from an irrelevant one. Erosion followed by dilation are the two most elementary executions of image processing in morphological arena . The following figure represents the

original image which then goes through several processes to achieve its structure. The processes are erosion, dilation, opening and finally closing **reference12**.

2.5.8 Segmentation

Segmentation divides the image into several constituents or objects. If we are automatically generate a segmentation of image then it would be quite hard to achieve in digital processing of image. After an in-depth segmentation of the image, the image is a lot closer to being successfully solved from imaging problems. This way individual objects in an image can be determined. In this figure each of the fruit are separately determined to distinctively differentiate between objects **reference15**.

2.5.9 Representation and description

This stage determines how are we going to define the data that we have processed through various techniques. On the basis of the requirement the necessary information will be used while the other ones will be of no use. Here specific values are given to all the objects of the image to describe them. This way objects will have different class labels and a different function altogether **reference14**.

2.5.10 Object recognition

In this process, objects are identified from either an image or a video. Then those objects are grouped with a particular label. For instance, there is an image of two kinds of fruits. After recognizing the objects, one group is labeled "guava" while the other one is labeled "apple." There are algorithms that recognize patterns using appearance or features **reference12**. It involves:

- Identifying different objects.
- Providing positional and geometrical descriptions in 3-dimensional space.
- Classifying the object into an already determined class.
- Recognizing the specific example.
- Understanding the similarities of the objects in terms of spatial construction.

2.5.11 Knowledge base

The image has quite a few regions. Knowledge is knowing which region is important and then choose it in order to reduce the search and therefore saving cost. In other times applying knowledge could be difficult as the final output might not translate into something that occurred before. For instance, finding correlation between defective materials might not be easy.

2.6 Algorithms

2.6.1 Support Vector Machine (SVM)

Support vector machine is differentiation of classes which finds out a hyperplane in N-dimensional space and also a supervised method of classification and regression[2]. So, SVM is mainly finding a line or separating two things with a line. If we want to draw a line between the dots and squares it would be easy, just draw a line between the dots and square.reference9

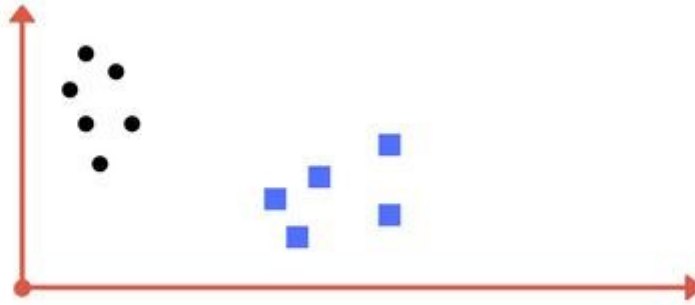


Figure 2.1: Finding a line to separate

Drawing a line we get

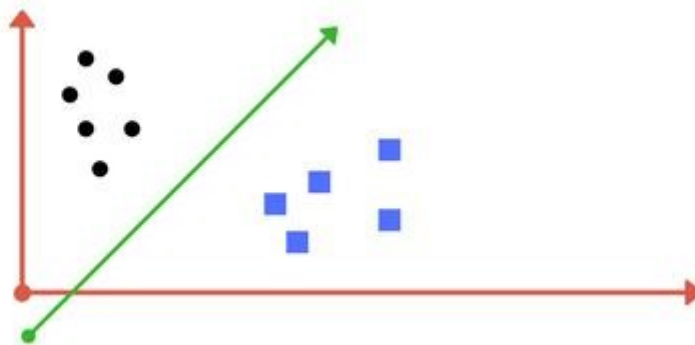


Figure 2.2: Green line separates these classes

But there are so many hyperplanes we can give in this diagram to detached data point. Earlier we said that, the given data portrayed by N- dimensional spaces and that means illustrates the number of data we use in our algorithm, which also represent as a particular coordinate. So, SVM takes input which is our data and giving us the output by giving a line which separates the data in particular numbers. So, this line is actually hyperplane which differentiates the two classes.reference10. In left diagram there are many lines and red squares and blue circles. So we have to differentiate an optimal hyperplane between those two things. To find an optimal hyperplane, the line should be far from those red squares and blue circles because it will not give us correct answer and it will be noisy. So, the more the distance the more we get the outcomes from all data points[10]. Also, there are data points

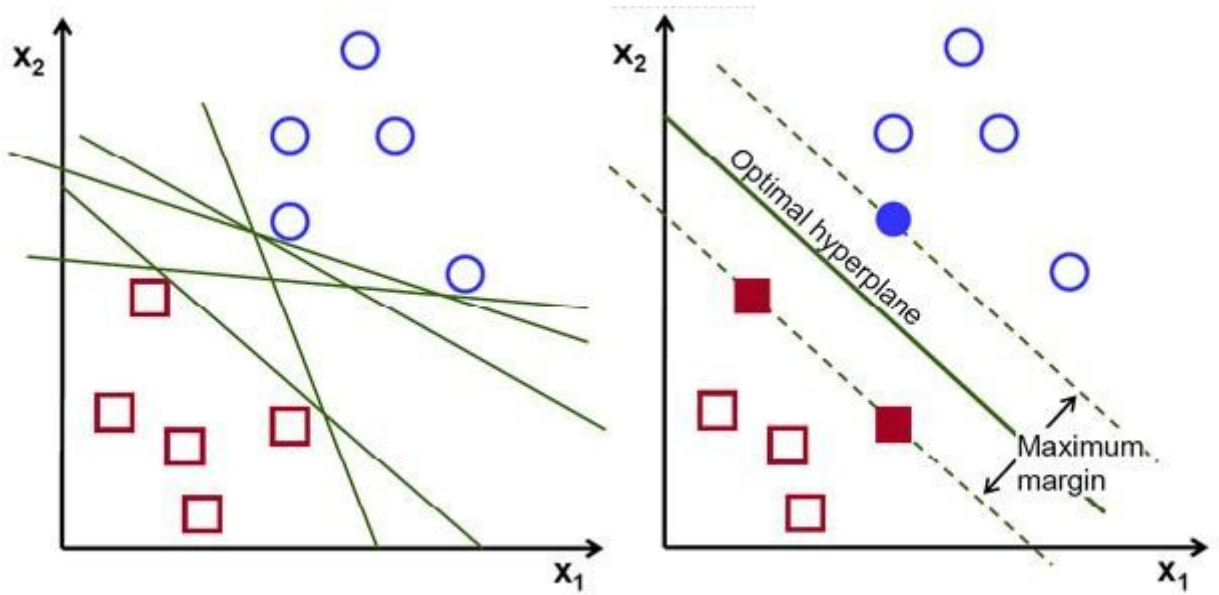


Figure 2.3: Optimal hyperplanes

which influence the position and orientation of the line and nearer to the hyperplane known as support vectors. It is used to maximize the margin of the data we are training.

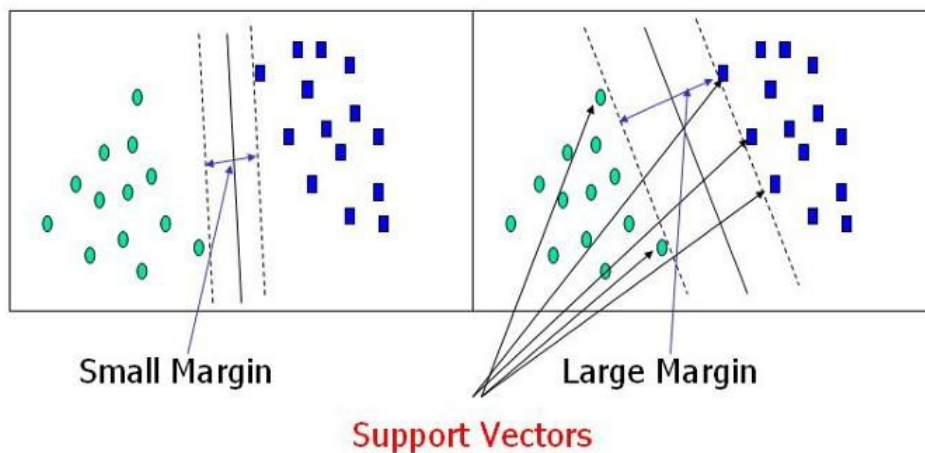


Figure 2.4: Support vectors

Hyperplanes helps to categorize data points and works as a decision boundaries. There can be many possible hyperplanes to separate the data points also it depends on the number of the features. Suppose, the number of feature is 2, it gives us a line. On the other hand, if the number of feature is 3,[10] it gives us a two dimensional plane as shown in the Fig 2.4

These are the data that can be linearly separable, but there are also data which are not linearly separable. To deal with this, we have to add an axis known as the z-axis because we can see the translucent separation of the lines which are more visible **reference9**. So, the points on the z-plane are given by:

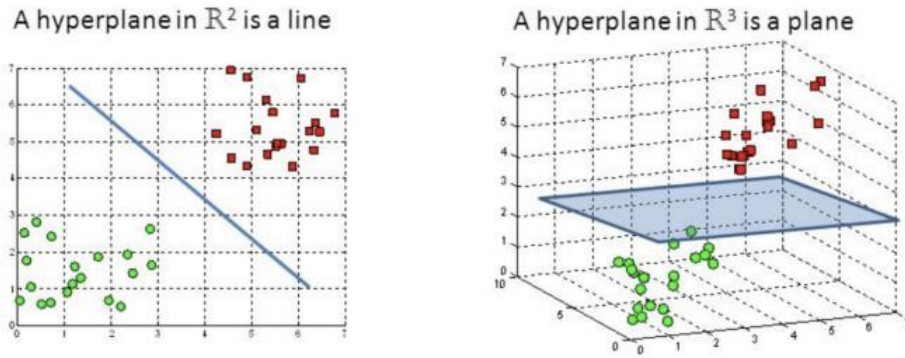


Figure 2.5: 2D and 3D feature hyperplane

$$z = x^2 + y^2$$

Here z- origin is manipulating the data as a distance of point shown in the Fig below

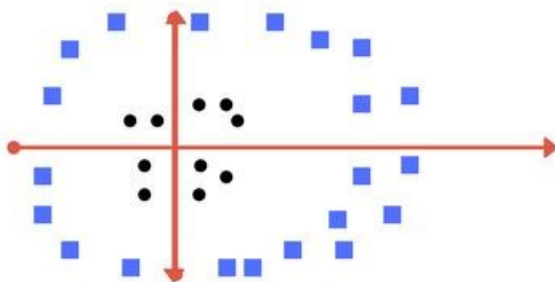


Figure 2.6: Finding a line

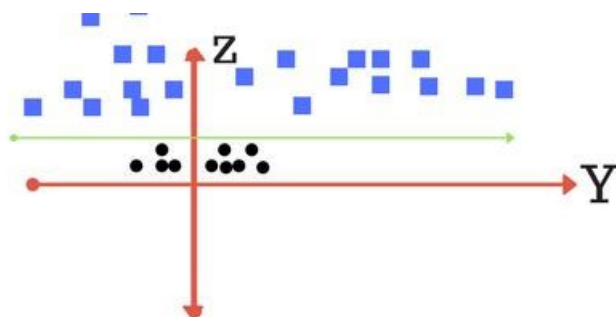


Figure 2.7: Line separation

Now, we get a line which is separating those things and also is mapping a circular bound-ary which known as Kernels or the name of the techniques is Kernel trick. It converts the low dimensional input spaces into higher dimensional input spaces. In non-linear separation we used this kernel trick. In microarray data this algorithm gives us better performance and it is a powerful classifier[9].

Thus we can get maximum margin to find a plane and which is our moto. It can also give us more confidence to classified data point which we can use in future.

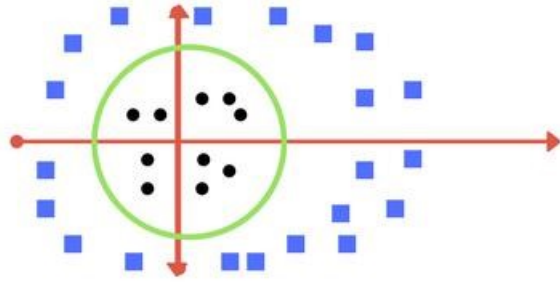


Figure 2.8: Line changing into circle

2.6.2 K-nearest Neighbor (KNN)

One of the most widely used supervised learning techniques in machine learning and data mining is K-Nearest Neighbor (KNN). KNN is simple, easy to understand, useful, and efficient. This algorithm is primarily instance-based, non-parametric, and lazy learning. In instance-based learning, it records the training data, which is used for knowledge in the prediction phase. In non-parametric learning, it makes no assumptions about the underlying data distribution. Although it is called a lazy algorithm, it does not use generalization techniques for training data and keeps all training data, making its progress very fast.

KNN is particularly useful when there is no knowledge about the data distribution. For example, consider a dataset of capsicums, including red capsicums (R), green capsicums (G), and capsicums of unknown color. To determine the color of the unknown capsicum, we analyze its characteristics, comparing them to the characteristics of red and green capsicums. This is done through mathematical calculations to measure the distance between classifications, known as feature similarity.

To find the nearest neighbor in KNN, we follow these steps:

1. Take the unclassified data.
2. Find the distance between known and unknown data using methods such as the Euclidean method, Manhattan method, Minkowski method, or weighted method.
3. Specify a parameter K to find the smallest distance.
4. Create a list of the shortest distances and count their occurrences.
5. Choose the classification that occurs most frequently.
6. Categorize the new data based on the last step.

In the example, if we calculate the distances of the unknown capsicum, we can determine its color based on the smallest distance. Here, $K = 3$, and the closest data points ($k=3$) consist of two green capsicums and one red capsicum, so the previously unclassified capsicum is classified as green.

Selecting the appropriate value for K in the KNN algorithm is crucial for achieving accuracy. There is a process called "Parameter Selection" to find the optimal K value, including these steps:

1. Find the best value of "K" because there is no mathematical way to determine it.
2. Smaller values of K may introduce noise and detachment from the main body of the system.
3. Larger values provide smoother decision boundaries.
4. Cross-validation can be used to find K from the training dataset, by predicting labels using different values of K and selecting the best one.
5. One formula to estimate K is $k = \sqrt{N}$, where N is the number of samples in the training dataset.
6. Using odd values for K can help avoid confusion between classes.

KNN is used for both classification and regression problems. It classifies objects based on the smallest distance and similarities to the nearest neighbors. In our paper, we apply this algorithm for classification and emphasize the importance of selecting an appropriate K value for successful classification.

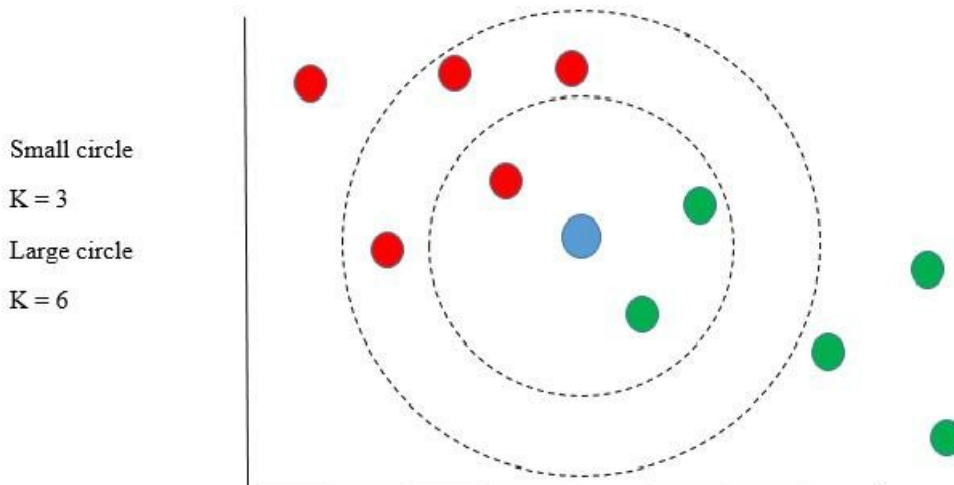


Figure 2.9: KNN example

2.7 Convolutional Network Architectures

2.7.1 LeNet-5 (1998)

In 1998, LeCun et al first introduced this 7 convolutional network which can classify digits. Later on, many banks used this to recognize hand written checks digitized in 32x32 pixel greyscale. For processing images with higher resolution, it needs more convolutional layers which are also larger in size. There are 3 main ideas behind building this network. They are local receptive fields, shared weights and special subsampling.

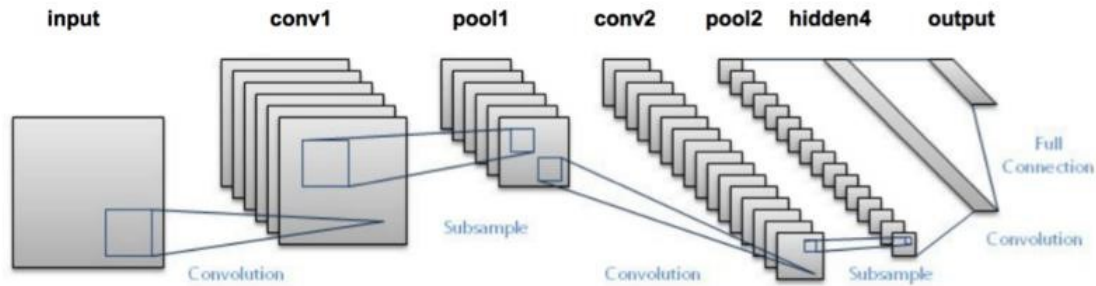


Figure 2.10: LeNet5

2.7.2 AlexNet

AlexNet was developed by Alex Krizhevsky, Geoffrey Hinton and Liya Sutskever in 2012. This network is very similar framework to LetNet 5 but it was deeper because it has more filters per layer along with stacked convolutional layers. It has more data and it is a much bigger model than LetNet7 hidden layers, 650K units, 60M params). In 2012 , AlexNet got massive success by outperforming all the prior competitors as they won the challenge by reducing the top 5 error from 26 percent to 15.3 percent.

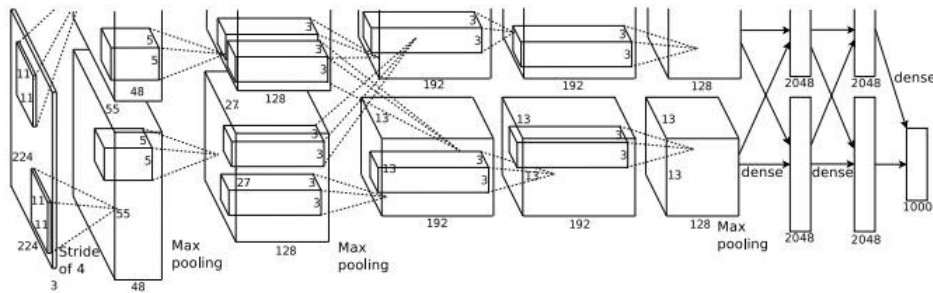


Figure 2.11: AlexNet

2.7.3 ZFNet

ZFNet was designed by Dr Fergus and his PhD student at the moment, Dr. Mathew D. Zeiler in New York University. Hence, this network is called ZFNet, based on surname, Zeiler and fergus. The image classification error rate was improved massively in ZFNet at time of comparing with Alexnet. This network maintained almost same structure and framework like AlexNet and got great achievement by tweaking the hyper parameters. This network successfully won the ILSVRC 2013.

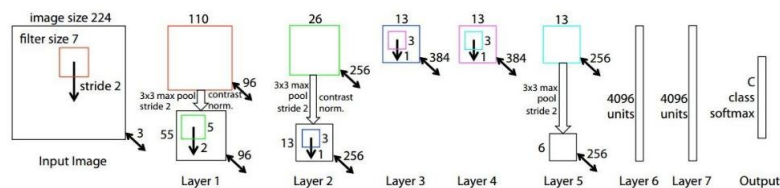


Figure 2.12: ZFNet

2.7.4 GoogLeNet

From the word “GoogLeNet” we can realize that this is from Google. Here, “LeNet” used for paying tribute to Prof. Yan LeCun’s Letnet. GoogLeNet was the winner of ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2014. This was a big improvement over ZFNet and AlexNet and it had a very lower rate of error than the first runner-up VGGNet. This was also known as Inception V1 but later on, V2, V3 and V4 are also designed by them. In GoogleNet the top 5 error rate was only 6.67 percent which was a great achievement. GoogLeNet architecture consisted of 22 layers CNN and reduced number of parameters significantly from 60 million (AlexNet) to 4 million.

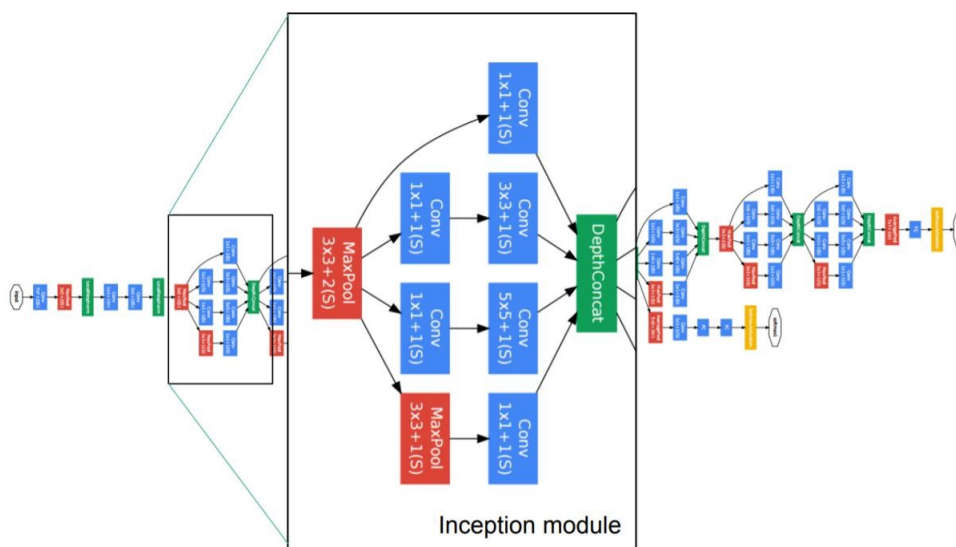


Figure 2.13: GoogLeNet

2.7.5 VGGNet

VGGNet was designed by Simonyan and Zisserman in 2014. This network became runners-up at the ILSRC 2014 competition. Like AlexNet, this network has only 3×3 convolutions but it has a lot of filters. VGGNet has 16 convolutional layers. VGGNet has around 138 million parameters which is a bit difficult to handle since GoogLeNet which was also invented on the same year, had only 4 million parameters.

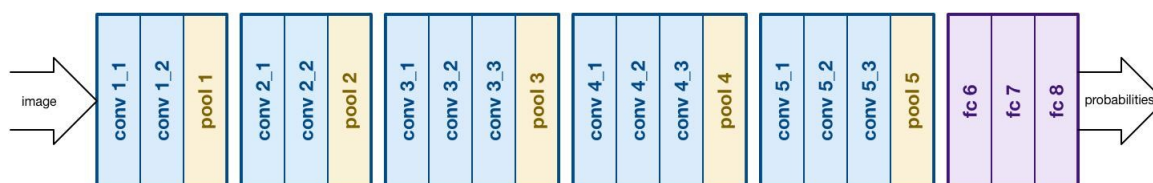


Figure 2.14: VGGNet

2.7.6 ResNet

ResNet was developed by kaiming He and was first introduced in ILSVRC 2015. ResNet is also called Residual Neural Network. They used around 152 layers while

still having lower complexity than VGGNet. It won ILSVRC 2015 with a top 5 error rate of 3.57 percent which beats human-level performance on dataset. While GoogLeNet used inception modules, ResNet used residual connections.

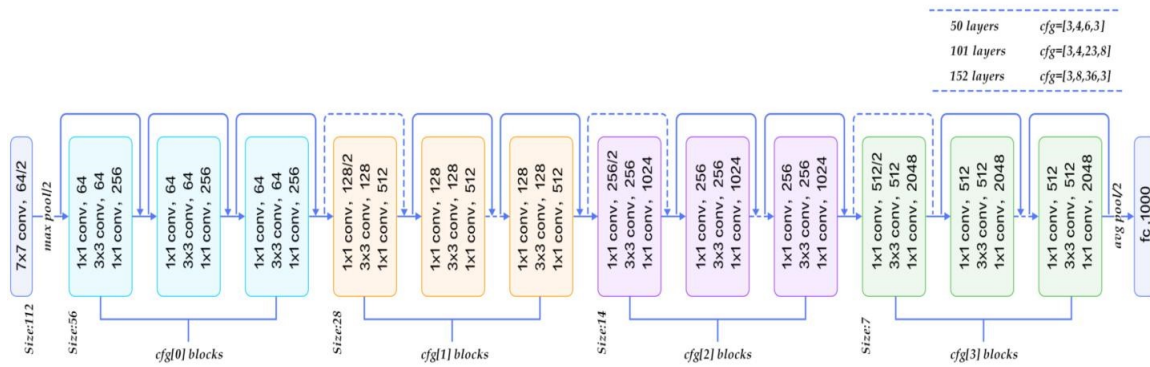


Figure 2.15: ResNet

Chapter 3

Proposed method

The aim of our project is to make an interface that can be used to recognize user handwritten characters. We approached our problem using Convolutional neural Networks in order to get a higher accuracy. Several researches have been undertaken to improve the accuracy of alphanumeric character prediction. Our research will include that to some extent. But our main focus will be providing a GUI that can be used to easily predict characters for further use. We plan to do so using tensorflow [2] and keras. Firstly, we will define a model that will be trained with the Emnist dataset which contains over 690,000 train images and will be validated using the test dataset provided by Emnist again.

The flow chart of the proposed text line segmentation method is given below is given below:

3.1 Data Training

After collecting the data we converted those data into an attributed relation file format (.arff) and then we have used Weka for classification. After classification using some algorithm we got some result and later we have analyzed those result. Here is the simple work flow chart given.

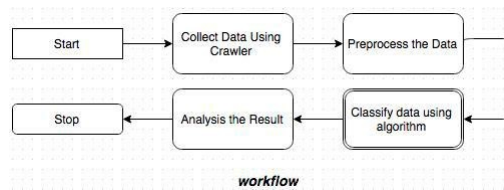


Figure 3.1: Workflow

Chapter 4

Experimental tests and results

4.1 Dataset

The EMNIST dataset is a collection of handwritten alphanumeric derived from the NIST Special Database 19. Each image is converted to a 28x28 format and dataset structure that directly matches the dataset is used. The training set has 697932 images and test set has 116323 of uppercase and lowercase alphabets and numerals from 0-9 which are mapped to their corresponding classes. The test set and training set is available in the form of list within list. Each item of outer list represents an image and inner list represents the intensity values of 784 pixels (because size of image is 28 x 28 pixels) ranging from 0-255. The test images as well as train images have a white foreground and black background. Both the test images as well as train images are horizontally flipped and rotated 90 degrees clockwise. Y train and Y test both are arrays which contain number ranging from 0 to 61 as there are 10 numerals from 0-9 and 26 uppercase and lowercase alphabets each which adds up to 62 **reference1**.

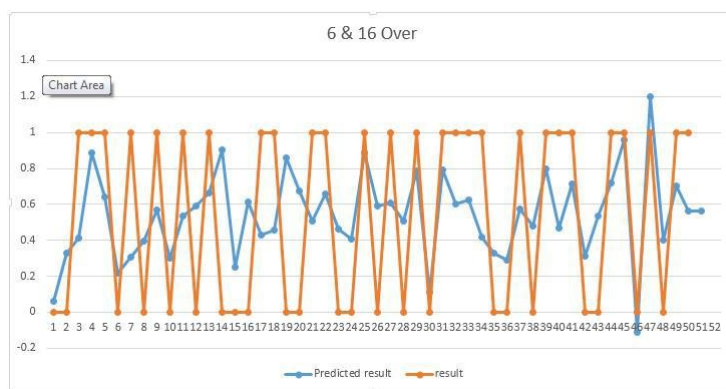


Figure 4.1: First and Second Segment Prediction (Bat First)

4.2 Training

We trained our models with different optimizers available for keras **reference3**. As the bar chart illustrates, the models we used included RmsProp, Adam, Adadelta, Adamax, SGD. The highest accuracy obtained was when we used Adamax for our

Attributes	Coefficients
Intercept	0.092219
Venue	0.242112
M6ORN	0.039573
M6OW	-0.12872
M16ORN	0.05121
M16OW	-0.07214

Table 4.1: First and Second Coefficient (Bat First)

Attributes	P-value
Intercept	0.87596
Venue	0.088577
M6ORN	0.323375
M6OW	0.084286
M16ORN	0.246463
M16OW	0.254117

Table 4.2: First and Second Segment P-value (Bat First)

experiment. So, we decided to use it for the purpose of our research on the Emnist dataset. Other optimizers gave accuracies somewhat very close to Adamax, but Adamax substantially reduced our training time as well. The training on a personal computer (RAM - 16 GB) took about 20 hours with 512 units in our dense layer. This training time can be reduced in a manifold way by adopting the usage of GPU. The model trained with our Emnist dataset which had been pre-processed and optimized for training beforehand. We have categorically distributed the train and test labels and flattened out train and test arrays for easy input into our model and android application. The Adamax optimizer is extremely popular for training large models and has provided us with robust results.

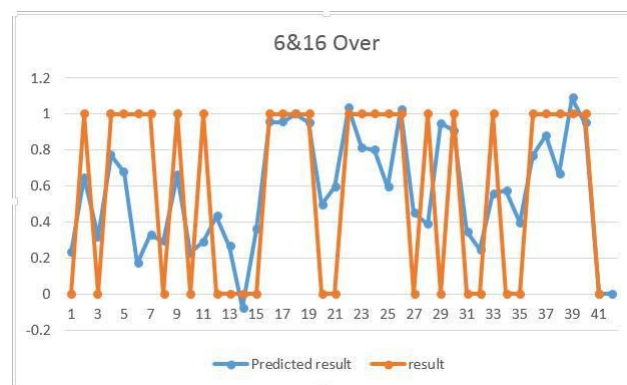


Figure 4.2: First and Second Segment Run Prediction (Bat Second)

Coefficient: These are the coefficients values for all the attributes from Win prediction based on bat second.

Attributes	Coefficients
Intercept	2.282567
Venue	-0.04063
M6ORN	-0.02869
M6OW	-0.22694
M16ORN	-0.12705
M16OW	-0.10903
O6ORN	0.008056
O6OW	0.066748
O16ORN	0.028731
O16OW	-0.0978

Table 4.3: First and Second Segment Coefficient (Bat Second)

P-values: These are the p values for all the at-tributes from Win prediction based on bat second.

First and Second Segment P-value (Bat Second)	
Attributes	Coefficients
Intercept	0.007165
Venue	0.811504
M6ORN	0.482433
M6OW	0.019258
M16ORN	0.112474
M16OW	0.090761
O6ORN	0.878555
O6OW	0.429492
O16ORN	0.633494
O16OW	0.179298

Table 4.4: First and Second Segment P-value (Bat Second)

4.2.1 Character Segmentation and Prediction

We begin by inputting an image. The image can be of a single character or a word. We use OpenCV to work with images for this research. Using inbuilt OpenCV library functions, we find contours in the image. After finding contours, we create rectangular bounding boxes around each character in a copied image. This is done because if we create boxes in the original image, the boxes may overlap with each other and hinder the performance of the classifier. Contours can be defined in a simple manner as a curve joining all the continuous points (along the boundary), having same color or intensity. They prove to be a useful tool for shape analysis and

object detection. For better accuracy, we use binary images. `findContours()` function modifies the source image that's the reason we send a copy of image. After we create boxes around each identified character, we extract ROI's(Region of Interest) from the image. Since the size of each character might be different, we resize each image into a 28*28 image using OpenCV again so that this image can be used as an input to our model classifier. Once segmentation is completed, we provide each 28*28 ROI as an input to our model and use the converted result to display the outcome in a formatted manner.

Chapter 5

Conclusion

Using modern day techniques like neural networks to implement deep learning to solve basic tasks which are done with a blink of an eye by any human like text recognition is just scratching the surface of the potential behind machine learning. There are infinite possibilities and application of this technology. Traditional OCR used to work similar to biometric device. Photo sensor technology was used to gather the match points of physical attributes and then convert it into database of known types. But with the help of modern-day techniques like convolution neural networks we are able to scan and understand words with an accuracy never seen before in history. We used the EMNIST data set to train our model and tested different optimizers to finally select Adamax as it not only yielded a high accuracy with each epoch on our train data but also our test data. A further application of accurate text OCRs is to help the partially sighted and the blind in the absence of braille. By also integrating a simple text to speech module in the app the user can point his phone to any text which will then read out the text for the user. A dedicated device can also be built for this purpose with a more sophisticated image recognition system which can identify objects to tell the user how many steps to walk in which direction and even when to stop and turn. The EMNIST datasets, a suite of six datasets, considerably increased the challenge faced by employing only the MNIST dataset. Even though the structure of EMNIST dataset is similar to that of MNIST, it provides a higher number of image samples and output classes and an even more complex and varied classification task. It was thus obvious to use it as the backbone of our project. Without the use of EMNIST data set it would be practically impossible to achieve this accuracy.

bibliography/references.bib