Handwritten Character Recognition App

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Ludhiana College of Engineering & Technology)

Submitted By:
Gurinder Sandhu (1807472)

Submitted To:
Er. Tejinder Kaur

Department of Computer Science Engineering Ludhiana College of Engineering & Technology, Katani Kalan, Ludhiana, 141006

ABSTRACT

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to handwritten character(both alphabets and digits) recognition based on Convolution Neural Network (Deep learning) technique. The main objective is to ensure effective and reliable approaches for recognition of handwritten characters.

ACKNOWLEDGEMENT

I am highly grateful to, Dr. Pawan Kumar, Principal, Ludhiana College of Engineering and Technology, katani kalan (LCET), Ludhiana, for providing this opportunity to carry out the Six-Month industrial training.

The constant guidance and encouragement received from Prof. Tejinder Kaur, LCET, Katani Kalan has been of great help in carrying out the project work and is acknowledged with reverential thanks.

They would like to express a deep sense of gratitude and thanks profusely to Mr. MANJINDER SINGH, CEO of Company. Without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

I also express gratitude to other faculty members of Computer Science Engineering department of LCET for their intellectual support throughout the course of this work. Finally, I express my indebtedness to all who have directly or indirectly contributed to the successful completion of my training.

GURINDER SANDHU
1807472
B. Tech {CSE}

TABLE OF CONTENTS

CONTENTS	PAGE NO.

CHAPTER-1
INTRODUCTION

INTRODUCTION TO PROJECT

The project comes with the technique of OCR (Optical Character Recognition) which includes various research sides of computer science .The project is to take a picture of a character and process it up to recognize the image of that character like a human brain recognize the various digits.The project contains the deep idea of the Image Processing techniques and the big research area of machine learning and the building block of the machine learning called Neural Network.There are two different parts of the project 1) Training part 2) Testing part. Training part comes with the idea of to train a child by giving various sets of similar characters but not the totally same and to say them the output of this is “this”. Like this idea one has to train the newly built neural network with so many characters.This part contains some new algorithm which is self-created and upgraded as the project need. The testing part contains the testing of a new dataset .This part always comes after the part of the training .At first one has to teach the child how to recognize the character .Then one has to take the test whether he has given right answer or not. If not, one has to train him harder by giving new dataset and new entries. Just like that one has to test the algorithm also. There are many parts of statistical modeling and optimization techniques which come into the project requiring a lot of modeling concept of statistics like optimizer technique and filtering process, that how the mathematics (How to implement a neural network intermezzo 2 Peter Roelants (2016)) and prediction (Kaiming He et al)) behind that filtering or the algorithms comes after or which result one actually needs to and ultimately for the prediction of a predictive model creation.Machine learning algorithm is built by concepts of prediction and programming.

CHAPTER - 2

REQUIREMENT ANALYSIS & SYSTEM SPECIFICATION

2.1 PROBLEM DEFINITION

The total world is working with the various problems of the machine learning.The goal of the machine learning is to factorize and to manipulate the real life data and the real life part of the human interaction or complex ideas or the problems in the real life.The most curious of those is Handwritten Character Recognition because it is the building block of the human certified and the classification interaction between other humans. So, the goal was to create an appropriate algorithm that can give the output of the handwritten character by taking just a picture of that character. If one asks about Image processing then this problem can't be solved because there can be a lot of noises in that taken image which can't be controlled by human.The main thing is when human write a handwritten character or for our case digit he has no single idea whether he has to draw it in the circulated pixels or just same as a standard image given .A machine can do that but not the human.So by matching only the pixels one can't recognize that. The idea of machine learning lies on supervised data.Machine learning algorithm fully dependent on modeled data .If someone models the Image directly, the model will get a lot of flatten values because that picture can be drawn with various RGB format or with various pixels which can't be modeled accurately due to noise. So, for this project one has to create a model by image processing and the machine learning. Both the techniques will be needed because these two techniques will enhance the technique of the machine learning and that can shape this project

2.2 NEED OF THE SOFTWARE

The total project lies with a great computation speed and by a online server where run and compilation done quickly. All the packages were imported that were needed for the software online. We need the tools to be imported also.

This project at first is in need of the software of python.The total code is written in python so it needs Python3. Python2 was not chosen because python3 has some additional upgrade over python2. The packages have been imported and the algorithm created which is done by installing the new packages from online in python3.

Apart from that the total project is online compiled or ran and done using Jupyter-notebook, Pycharm .There are two apps available one to run on the localhost i.e. my machine and another is deployed using flask and Heroku and can be accessed by anyone. Creation of the machine learning algorithms deals with data and bigger size programs.

2.3 BETTER FUNCTIONALITIES

This project deals with the End users with some functionalities.The major functionality can be the Recognition of character(Alphabet or digit) .User can write a handwritten character and this project will recognise it accurately. Edge detection can be set in the process of image processing.ML algorithm can differentiate the various digits from another by recognising it. The better functionality where the building block can be this project is Mathematical Model solver. One can take any picture of a mathematical problem and by this project one can recognise the digit inside it and then computer can compute that problem on its own.If a wrong answer comes, it can be checked through a step by step process by the computer and if it recognized the answer wrongly, it must be trained again. One has to train the model in various extents to recognise the various digits not only 0 to 9, alphabet A to Z but also more and more figure, like derivative integration and others. The better functionality of this project can be license plate verification.Car license plate can be checked and one can set the record rightfully that which car is passing the gate and when by the recognition of characters.

2.4 FEASIBILITY ANALYSIS

There are a lot of sides of feasibility. Let it be discussed one by one.

2.4.1 Technical Feasibility

The software which has been used in this project is fully open source and one can connect to it whenever he or she wish. The concept of python and open CV another side the research concept of image processing and machine learning is a very trending topic nowadays .Apart from that all the software running environment Jupyter notebook is fully open source and easily can be accessed in the presence of internet.The user can also be a non-programmer and by clicking the run button he or she can draw the character on drawing screen and can see the output.

2.4.2 Seasonal Feasibility

This project is feasible in time that means this project has been started in a particular date and completed in the available time fully. It was an efficient effort which resulted in completion of the project in time.

2.4.3 Economic Feasibility

This project is economically free because all the open source softwares have been used that is why no money was charged or given.Only the study materials in the devoloper side or the designer are not available in free of cost.The software or the program is fully free of cost.

2.4.4 Profitability

This project deals with two trending topics.Image Processing and Machine learning.The total machine learning concept has not used here but we have dealt with the building block of the machine learning called Neural Network. These two topics are an appropriate research topic and many scholars and teachers also are working with it every day to upgrade the techniques or the algorithms or to create some new algorithms. The extension of the project can be used in large scale to detect the written character from images and to extract them in no time.Or in banking signature recognition or in license plate verification, Real time image chaining or filtering or object detection etc.This project is fully profitable because many sides of this project is in research nowadays and government is funding those researches.

CHAPTER 3
SOFTWARE REQUIREMENT SPECIFICATION (SRS)

3.1. SYSTEM REQUIREMENT

This project needs the help of hardware and software requirements to be fit in the computer or the laptop Pc.The user and the toolkits and hardware and software requirements are required also.

3.1.1 Hardware Requirements

RAM: At least 4 GB.

Processor: Intel(R) core (TM) i3 or more. 2.00 Ghz.

Internet connectivity: Yes.(Broadband or wi fi)

3.1.2 Software Requirements

SOFTWARE	TYPE/PLATFORM	VERSION
Operating System	Windows,Linux,ubuntu,fedora of similar	Windows 7 or More, Ubuntu 16 or above, Fedora 20 or above
Python	--	Python 2.7 or above
Opencv	--	Opencv 3.2.0 or above
Numpy	--	With Python 2.7 3.5
Keras	--	Keras 2.2 or above
Tensoflow	--	Tensorflow 2.1.6 or above
Pil	--	Pil 1.1.5 or above
Flask	--	Flask 1.1.2
Pip	--	Pip 20.2.3

3.2 TOOLS AND TECHNOLOGIES

3.2.1 OpenCV

The Opencv(Open Source Computer Vision Library) is a python DIL library which is very updated in the work of Image processing .One image file and pixel values can easily come into surface by this library.This library provides a common infrastructure and module related to computer vision technologies.The most important thing about this tool is it is totally free and can be easily modified and changed respective to input by the programmer.

3.2.2 Numpy

Numpy is a library for the python programming language, adding support for large multidimensional array and metrices.This package contains a large collection of high level mathematical functions to operate on those arrays. Numpy is open source and has many contributors.

3.2.3 Tensorflow

TensorFlow is an end-to-end open source package in python for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

3.2.4 PIL

PIL is a library of advanced image tools having full name Pillow.It is used for noise cancellation and to draw modified pixels by noise.It is useful to Image crop and various subjectial techniques.

3.2.5 Jupyter Notebook

The Jupyter Notebook App is a server-client application that allows **editing and running notebook documents via a web browser**. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed

through the internet.

3.2.6. Keras

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

3.2.7 Flask

Flask is a small and lightweight Python web framework that provides useful tools and features that make **creating web applications in Python easier**. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file.

3.2.8 Heroku

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

3.2.9 Pycharm

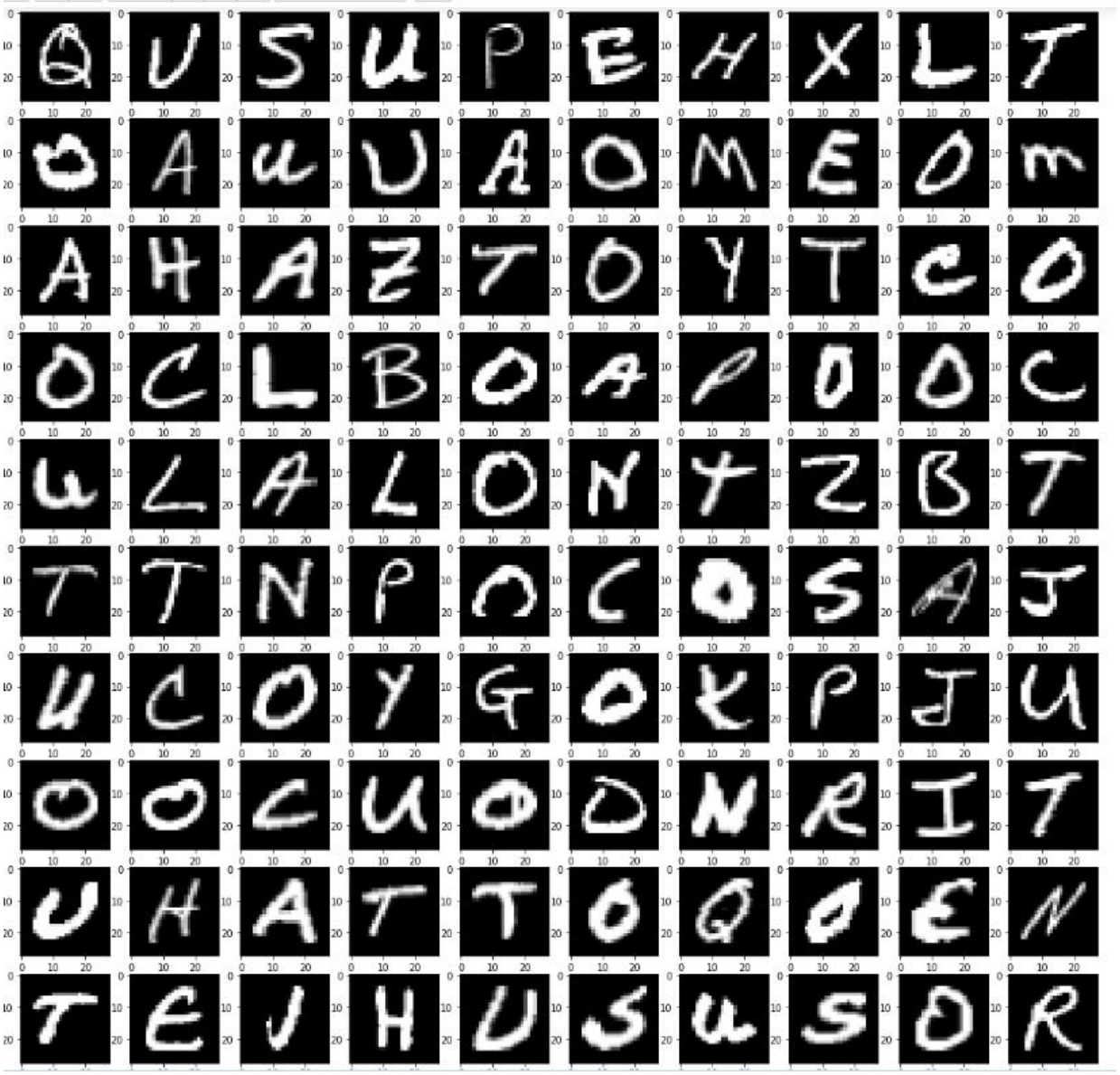
PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated **to create a convenient environment for productive Python, web, and data science development**.

CHAPTER 4
MODEL DETAILS

4.1 DATASETS USED:

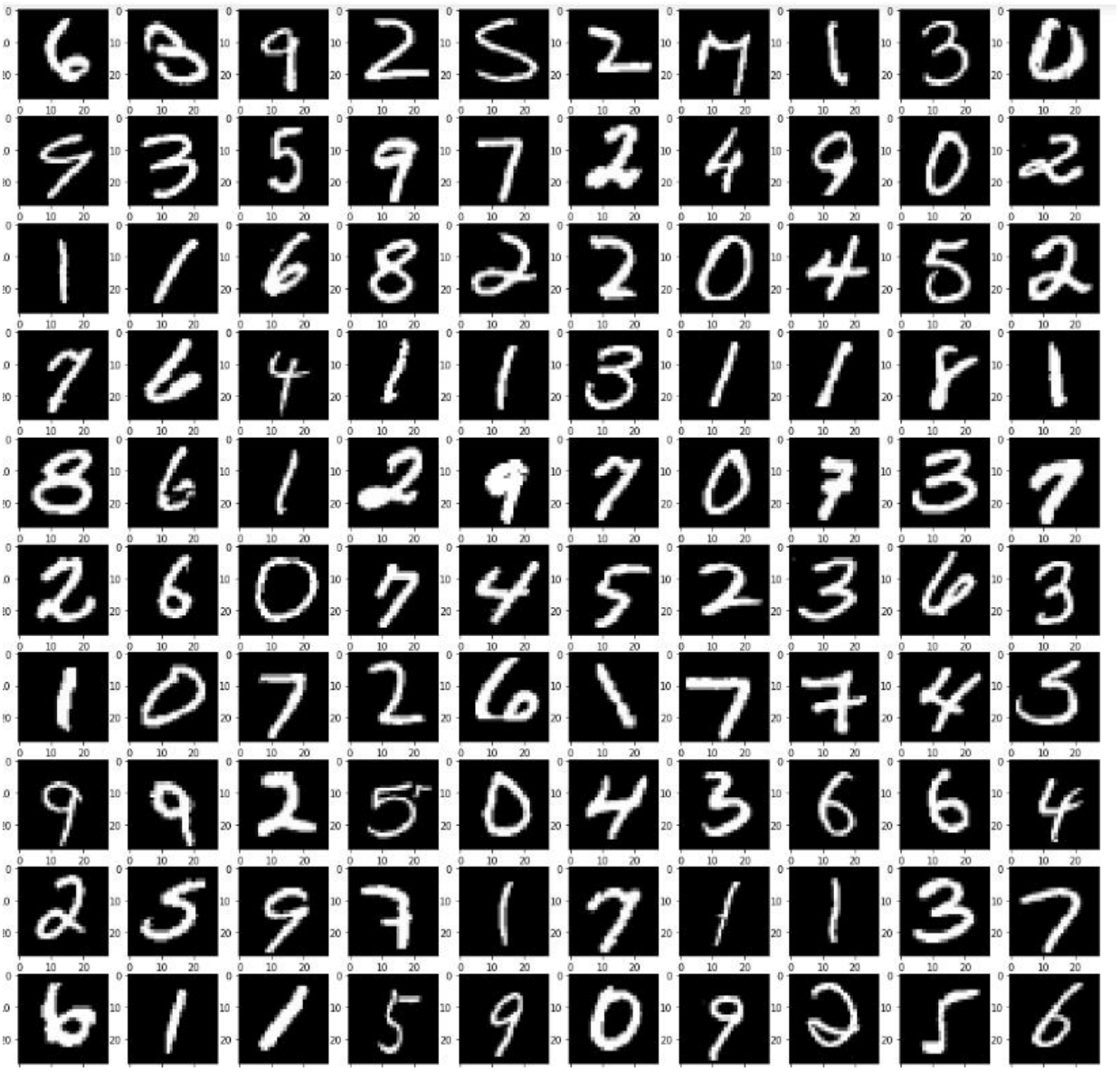
1. Handwritten A-Z Alphabets:

https://www.kaggle.com/sachinpatel21/az-handwritten-alphabets-in-csv-format



2.Handwritten Digits from 0-9

http://yann.lecun.com/exdb/mnist/



4.2 MERGING BOTH THE DATASETS:

It was really a big task to merge both the datasets for taring and testing purpose. Both the datasets were not familiar. MNIST dataset had images of shape (28,28) and their corresponding labels, whereas A-Z kaggle dataset had 785 columns, last column represented the label but first 784 columns were the features. So the initial task was to combine these 784 features to build an image of shape (28,28) and only then we could move further to join the ttwo datasets and perform operations.

4.3 IMAGE PROCESSING

Image processing technique has been implemented extensively at the very first part of the project.

So, what is Image Processing? Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or features or characteristics associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

1) Importing the image via image acquisition tools

2) Analysing and manipulating the image

3) Output in which result can be altered image or report that is based on image analysis.

In this project at the last part of detection to let the user draw the handwriiten character, to reshape that image and in the very first part of training merged A-Z and MNIST dataset image reshaping and real life dataset written reshaping, cutting, filtering all requires the idea of the image processing. One by one the concepts of image processing in this project will be covered.

4.3.1 Image/Data processing

In the training dataset the neural network model has been trained with two different dataset merged into one final dataset.

I) MNIST (Modified National Institute of Standards and Technology database)dataset

ii) A-Z Kaggle dataset

This final dataset almost had 442450 entries in the total, out of which 80% of data was used as training data: 353960,remaining 20% (88490)is used as test data and trains the model with excellence with 10 epochs in total.

i) The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits. The database is also widely used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets. The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments. Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels.

A-Z Kaggle dataset -

The dataset contains 26 folders (A-Z) containing handwritten images in size 28*28 pixels, each alphabet in the image is centre fitted to 20*0 pixel box.

Each image is stored as Gray-level

Kernel **CSV***To***Images** contains script to convert .CSV file to actual images in .png format in structured folder.

Note: Might contain some noisy image as well

Some operations of data preprocessing:

1. Normalization
2. Converting labels to categorical labels
3. Reshaping the images to be fed into model

1. **Normalization:**

Why do we normalize image?

the point from normalization comes behind calibrating the different pixels intensities into a normal distribution which makes the image looks better for the visualizer. Main purpose of normalization is **to make computation efficient by reducing values between 0 to 1**.

```
train_data = train_data / 255.0
test_data = test_data / 255.0
```

2. **Converting labels to categorical labels:**

Why is to_categorical used?

You use to_categorical **to transform your training data before you pass it to your model**. If your training data uses classes as numbers, to_categorical will transform those numbers in proper vectors for using with models. You can't simply train a classification model without that.

```
train_labels = np_utils.to_categorical(train_labels)
test_labels = np_utils.to_categorical(test_labels)
```

3. **Reshaping the images to be fed into model:**

Why do we reshape images in CNN?

Most convolutional neural networks are designed in a way so that they can only accept images of a fixed size. This creates several challenges during data acquisition and model deployment. The common practice to overcome this limitation is to reshape the input images **so that they can be fed into the networks**.

Reshape is **used to change the shape of the input**. For example, if reshape with argument (2,3) is applied to layer having input shape as (batch_size, 3, 2), then the output shape of the layer will be (batch_size, 2, 3)

```
train_data = np.reshape(train_data, (train_data.shape[0], train_data.shape[1], train_data.shape[2], 1))
test_data = np.reshape(test_data, (test_data.shape[0], test_data.shape[1], test_data.shape[2], 1))
```

4.4 THE NEURAL NETWORK MODEL

The neural network lies with the concept of machine learning.At first it creates a model like brain unit and then like child it trains that model like brain with many and many datasets,for my project it is the digits.There are two parts of a neural network model.

1) Training Part
2) Testing Part

Before going into the deep concept of these two parts in the project some ideas on the building blocks and structural parts of neural network (C.C.Jay.Kuo – 2016, Adit Deshpande – 2016) have been shared.

Each and every neural network can be structured in three different parts

1) Input layer
2) Hidden layer
3) Output layer

A neural network is put together by hooking together many of our simple "neurons," so that the output of a neuron can be the input of another. For example, here is a small neural network:

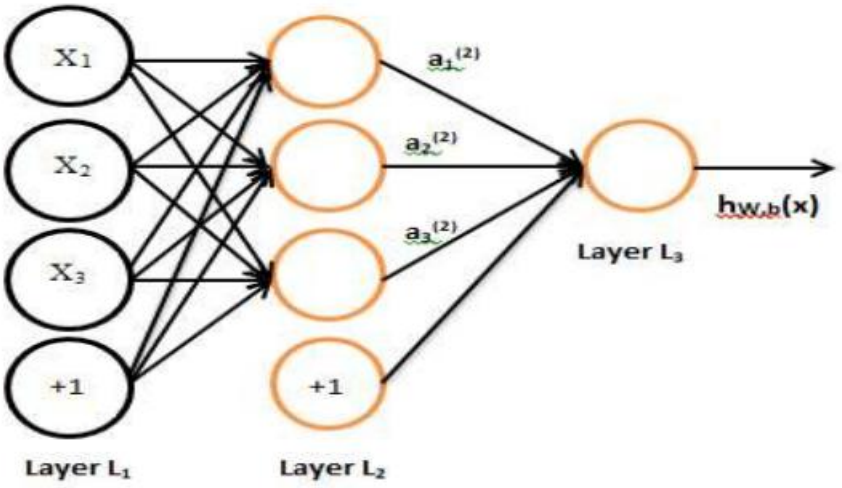


Fig-4.1 Neural Network Model

In this figure 4.1, circles are used to denote the inputs to the network. The circles labeled "+1" are called bias units, and correspond to the intercept term. The leftmost layer of the network is called the input layer, and the rightmost layer the

o

utput layer (which, in this example, has only one node). The middle layer of nodes is called the hidden layer, because its values are not observed in the training set. Say, that our example neural network has 3 input units (not counting the bias unit),3 hidden units, and 1 output unit.

The types of layers in a neural network can be summarized as follows:

1) Input Layer

Input variables, sometimes called the visible layer.This layer can be of features or the direct input from the dataset.

2) Hidden Layer

Layers of nodes between input and output layers . There may be one or more of these layers.More one create the hidden layer more operation will be there,more weights will be there and there will be less change due to the value of difference in this layer.This layer is the most important part of the neural network.Weights change in these layers varies in the returning input.

3) Output Layer

A layer of nodes that produce the output variables and then based on the targets it can perform iteration.Finally, there are terms used to describe the shape and capability of a neural network; for example:

Size: The number of nodes in the model.

Width: The number of nodes in a specific layer.

Depth: The number of layers in a neural network.

Capacity: The type or structure of functions that can be learned by a network configuration. Sometimes called “representational capacity“. Architecture: The specific arrangement of the layers and nodes in the network.

4.4.1 TRAINING PART

In the training part of the neural network a neural network model designed with various factors and determining other statistical learning model that can differ based on the dataset given has been created.This neural network then can be trained with a lot of datapoints (for our case 353960 datapoints) for better results and to upgrade the weights into the neural network.

4.4.2 CREATING THE NEURAL NETWORK MODEL

Now comes the main part of the project ,so how to create a neural network model that can take an image and run some probabilistic calculation on it which can implement the model with better accuracy

1) Input Layer

The total model trains on the dataset 28*28 pixels .The neural network model depends upon the number of inputs and the intensity in input layer and the filtered target output calculation in output layer. One dimensional array of the intense dataset is needed into this model and then sequentially give the total MNIST dataset and out retrieved dataset into the neural. One needs to flatten and sequential the total input into the neural.

2) Hidden Layer

The hidden layer part is the most efficient part in the neural network.It is the part where the total calculation and the backpropagation like learning algorithm takes place.Inthis project 128 hidden layers have been applied by taking the data from the input layer. In inner input layer and dense hidden layer MAXRELU function has been applied as an activation function and in the output layer the softmax activation function has been used (Softmax output function, Jefferey Hinton)

What Activation Function actually does?

Activation function controls the boundary of the data like normalisation .The normalisation function are rejected in the hidden layer because in the hidden layer the data or the values gets dynamically changed.There are various types of Activation functions.

a) Sigmoid- Y=1/(1+exp(-x))

b) Tangent- y=(exp(x)-exp(-x))/(exp(x)+exp(-x))

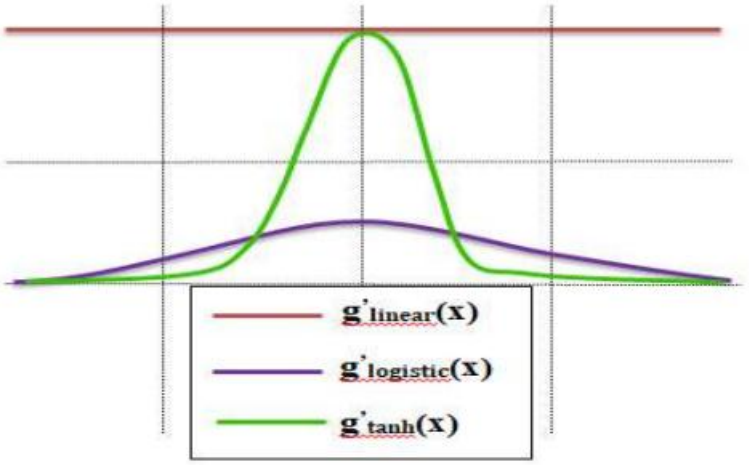
c) MaxReLu- y=max(0,x)

d) Softmax- y=exp(x)/(summation(exp(x)) etc.

In this project the max relu and softmax because this gives really good result tested and better than others like fig 4.2.Total experiment is based on result no extra theory applied but the convergence of the sigmoid and the tangential

c

urve to the verified output scenario



- Types of activation functions over graph

As the linear,logistic(sigmoid) and tangential is too much deflected for stagnant inputs and touches the curve in 1 and 0 immediately and never comes back practically ,these 3 are rejected.Sigmoid gives a very good result but practically max relu and softmax is better for character recognition.

Bias input are rejected to contain the actual shaped data from the main flatten and sequential data. Learning rate has been given very low 0.002 to make the model more effective.

OPTIMIZER

Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate in order to reduce the losses.

There are many types of optimizers.Each and every optimiser has its own method to deal with the weights and bias inputs.In this project the backpropagation algorithm is modified with the adam optimizer instead of the gradient descent optimiser. At first gradient descent was set to be final but then the adam or stochastic gradient descent optimiser is better than anyone as the inputs from the various pixels are varying in points

4.4.3 MODEL OVERVIEW

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 24, 24, 32)        832

 batch_normalization (BatchN  (None, 24, 24, 32)       128
 ormalization)

 conv2d_1 (Conv2D)           (None, 20, 20, 32)        25632

 batch_normalization_1 (Batc  (None, 20, 20, 32)       128
 hNormalization)

 max_pooling2d (MaxPooling2D  (None, 10, 10, 32)       0
 )

 dropout (Dropout)           (None, 10, 10, 32)        0

 batch_normalization_2 (Batc  (None, 10, 10, 32)       128
 hNormalization)

 flatten (Flatten)           (None, 3200)              0

 dense (Dense)               (None, 256)               819456

 dense_1 (Dense)             (None, 36)                9252

=================================================================
Total params: 855,556
Trainable params: 855,364
Non-trainable params: 192
_________________________________________________________________
```

Layers:

1. Conv2D
2. Batch Normalization
3. Max Pooling2D
4. Dropout
5. Flatten
6. Dense

1. Conv2D :

Such layers are also represented within the Keras deep learning framework. For two-dimensional inputs, such as images, they are represented by `keras.layers.Conv2D`: the Conv2D layer!

In more detail, this is its exact representation (Keras, n.d.):

keras.layers.Conv2D(filters, kernel_size, input_shape, activation)

Now, what does each attribute mean?

- **Filters** represents the number of filters that should be learnt by the convolutional layer. From the schematic drawing above, you should understand that each filter slides over the input image, generating a "feature map" as output.
- The **kernel size** represents the number of pixels in height and width that should be summarized, i.e. the two-dimensional width and height of the filter.

- **Input_shape** represents the input shape of the image to be fed as input to the model.
- The **activation function** to use. If you don't specify anything, no activation is applied

2. **Batch Normalization:**

- Deep neural networks are challenging to train, not least because the input from prior layers can change after weight updates.
- Batch normalization is a technique to standardize the inputs to a network, applied to ether the activations of a prior layer or inputs directly.
- Batch normalization accelerates training, in some cases by halving the epochs or better, and provides some regularization, reducing generalization error.

3. **Max Pooling2D:**

Max pooling operation for 2D spatial data. Downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool_size) for each channel of the input. The window is shifted by strides along each dimension.

4. **Dropout:**

The Dropout layer randomly sets input units to 0 with a frequency of `rate` at each step during training time, which helps prevent overfitting. Inputs not set to 0 are scaled up by 1/(1 - rate) such that the sum over all inputs is unchanged.

Note that the Dropout layer only applies when `training` is set to True such that no values are dropped during inference. When using `model.fit`, `training` will be appropriately set to True

automatically, and in other contexts, you can set the kwarg explicitly to True when calling the layer.

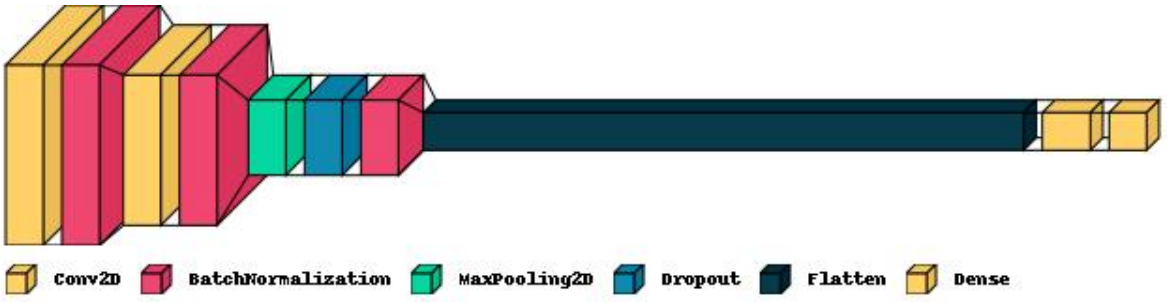
5. **Flatten :**

Flatten layer is used to make the multidimensional input one dimensional, commonly used in the transition from the convolution layer to the full connected layer.

For example, if flatten is applied to layer having input shape as (batch_size, 2,2), then the output shape of the layer will be (batch_size, 4) Flatten has one argument as follows keras.layers.Flatten(data_format = None)

6. **Dense :**

Dense Layer is **simple layer of neurons in which each neuron receives input from all the neurons of previous layer**, thus called as dense. Dense Layer is used to classify image based on output from convolutional layers. Working of single neuron. A layer contains multiple number of such neurons.



Layered view of model

4.4.4. TESTING PART

The testing part comes with some latest real time update in the machine learning part .The image processing work like webcam access and the other techniques has been discussed before that is why it has not been discussed here. So this part comes with some ransom upgrade of the technique which is very small to say but not that much easy to design.

Let's dive into the model after the total training and taking handwritten character from the user into that predictive modeled neural network. Set the input pixel values into the numpy array value where the argmax function will check the range and give the output figure .The range will be done by the feed forward step of the neural network.No other backpropagation algorithm will be applied because this is the test to check have the weights updated them to detect that input figure correctly? NO neural network can come with 100 % accuracy but if it can't detect most of the test figures then it must need more training. Result of the model shall be discussed in no time because the model has been trained more than 10 times to make it better and of course not with the same dataset every time. Datasets include ones from web and pictures and also handwritten and trained the neural model to give the correct result in the testing part.

To make things shorter what has been done in this model

1) Loaded dataset(A-Z alphabets and 0-9 digits) into the model

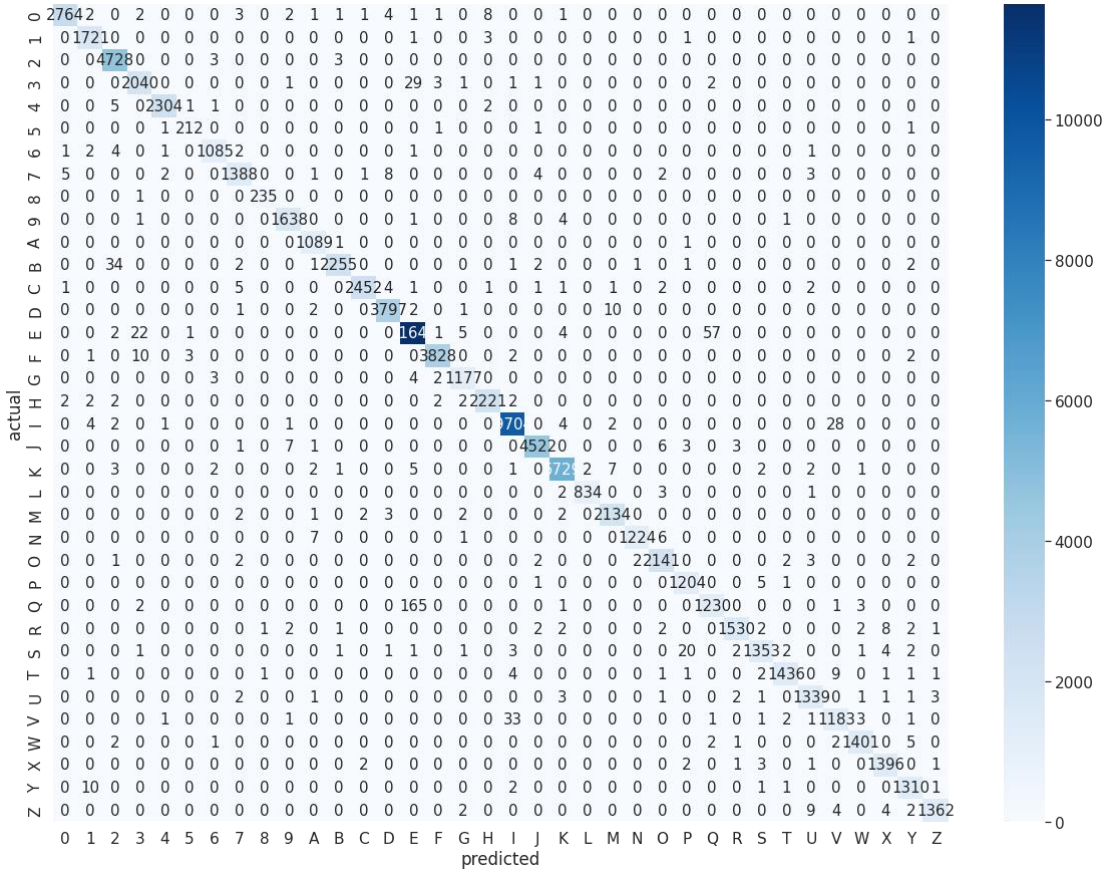
2) Set the numpy array system to take input the kernel along with the data.

3) Create the neural network model setting the input layer and the number of hidden layers and the output layers along with the activation functions used in different layers.

4) Set the probabilistic statistical value into the biased dataset into unbiased.

5) Check the target and the output predicted value every time while training the dataset and set the number of epochs corresponding to the error.

4.4.5 Experimental Testing Result :



Confusion Matrix

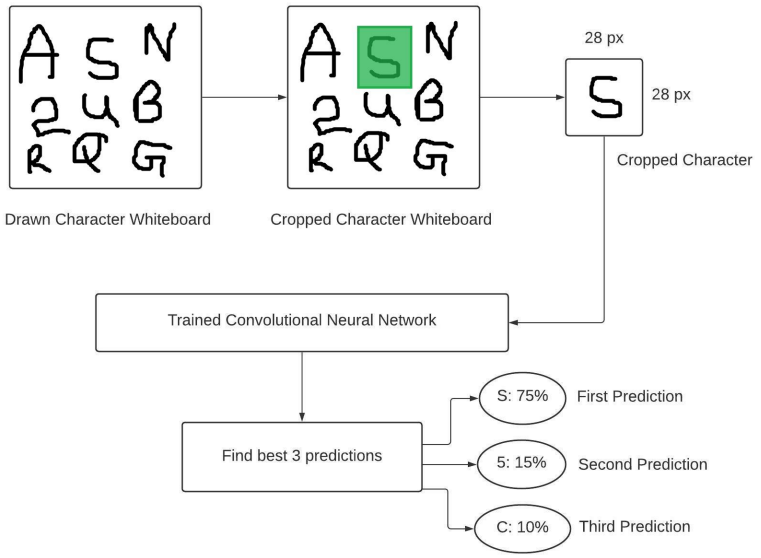
We can infer from the above data that there are some mis-predictions, but overall the model is good to work with real life examples.

CHAPTER 5
APPLICATION OVERVIEW

I have build two application using the same model and serving the same purpose of handwritten character recognition. The two aplications are:

1. Live cropped character recognition (hosted on local machine)
2. Handwritten character recognition webapp (deployed using Heroku)

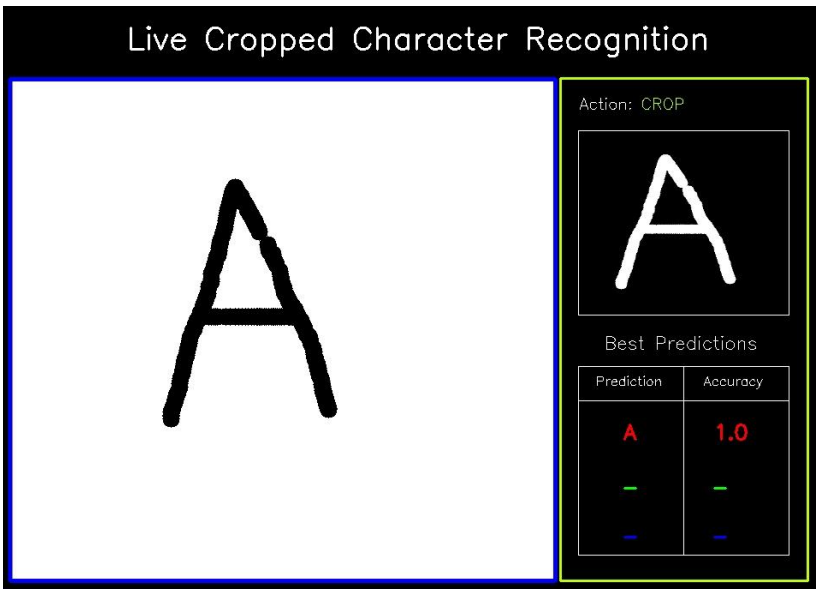
1. Live Cropped Character recognition:



Flow of the application

This app provide user an interface where he/she can write more than one character at once and then can crop the desired character to get the predicted result. And best 3 Predictions are given on the right panel with their respective probabilities.
This app is built on my local machine and is limited to my environment only. It has functionalities of Opencv.

Further I would show how the application exactly look like. I have used it many times and personally i feel satisfied as only building machine learning models is not sufficient, to make it a product to be used by people and customers it should be used in app or any interface, so that a person who is coding illiterate can even use your app and get the motive behind it.



Interface of the Live Cropped Character Recognition App

The interface is divide into two parts:

1. Right Panel
2. Whiteboard on the left

1. Right panel:

Right panel serves two tasks. Upper part show the cropped image but inverted and the lower half of panel shows the prediction with their accuracy rate.

2. Whiteboard on the left:

It is the interactive screen where people can write the character and crop them out to get the results.

There are few keyboard shortcuts which should be used. Initially, when the app starts the action is set to N/A , to draw something or to write you need to press D key and action will be selected as draw. And to crop C key is used and image can be cropped using mouse itself, as soon as the mouse is released we get to see the output on the right side of the application. Lastly, E key should be pressed if you want to clear out the entire whiteboard.

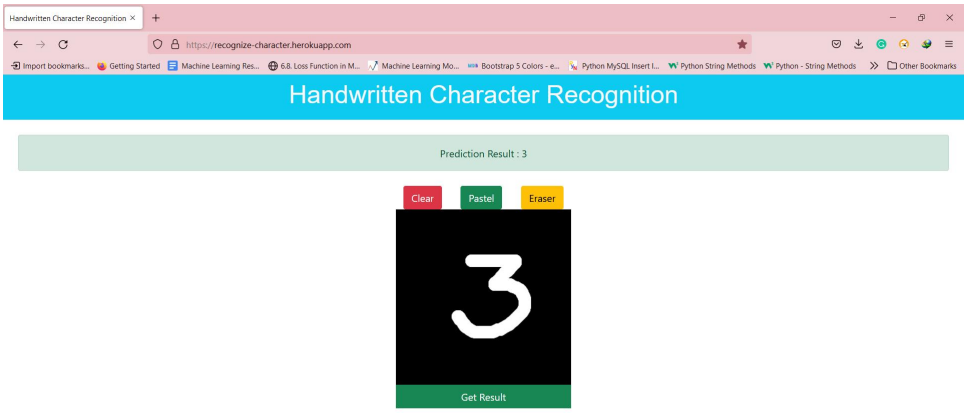
2.Handwritten character recognition web application:

This app provide user an interface where you can write one character at once and press the “get result” button to get the predicted result. And best prediction with the highest accuracy will be shown.

Unlike previous app, this application isn’t limited to my local machine, I have built a web application using Flask asd deployed it on the internet using Heroku (It is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps.). Anyone can use my application worldwide, although it might not be an great app but yes it is available to everyone.

My application is served at:

https://recognize-character.herokuapp.com/



“Recognize-character” Web Application interface

The interface of this application is really simple, you are provide with black screen where you can draw/ write the character (A-z or 0-9) and press the green button on bottom “Get Result” to get the predicted result by model. Three button are provide on the top of black window to serve different tasks: Red button “Clear” is used to entirely clear the screen when you are done using it, Green button “Pastel” is used when you need to draw/write a character, and Yellow button “Eraser” is used to erase something or make some changes to the drawn character.

CHAPTER 6
SYSTEM TESTING

6.1 **TESTING**

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements.

6.1.1 **Unit Testing**

When the testing happens for some individual group or some related units then that type of testing is called as Unit Testing.It is often done by programmer to test the part of the program he or she has implemented.

Unit Testing is successful means all the modules has been successfully tested and it can proceed further.

6.1.2 **Functional Testing**

This type of testing is tested because to check the functional components or the functionality required from the system is gained or not .It actually falls under the testing of the Black Box testing of Software Engineering.This part includes the feeding of the inputs in the system or the project and to check if that system or the project is getting the same value or not as expected if not then calculate the error as wanted and check for more.Functional Testing of this project mainly involves below things. All of these are tested successfully and errors are also calculated.

i) Verifyng the input image

ii) Verifying the work flow

iii) Correct recognition and calculate the error

6.1.3 **Integration Testing**

In a total project or the system, many groups of components

are getting added or summed up in the purpose of the project query. Integration testing is about to check the interaction between various modules of the project or the system. This module also includes the hardware and the software requirements of the project.

All the individual modules are integrated and tested together.All the best and extreme cases that the modules are interacting or not are successfully checked and passed,errors are calculated for the machine learning platforms.

6.1.4 **System Testing**

This type of testing is actually meant for the system or the project and also the platform and the integrated softwares and tools,technologies are also tested.The idea or purpose behind the system testing is to check all the requirements that will be provided by the system.

This application of the project along with the tools and technologies has been tested in both windows and linux platform and also unicertified online apple mac platform to check the requirements.It passed successfully

6.1.5 **Acceptance Testing**

This is a type of system or software testing where a system has been tested for availability.The purpose of this test is to check the business requirements and assess whether it will be accepted for delivery.In this part ADRIAN of pyrimagsearch has been referred to, who worked with the same platform and to check this project accepted by the delivery partner or not.

CHAPTER 7
CONCLUSION AND FORESEEABLE ENHANCEMENT

7.1 CONCLUSION

This is a project on OCR (Optical Character Recognition).This project is non-fundable project and designed with full of interest which also includes some outer concept on statistical modeling and optimiser technique. In these days of real time analysis, the data has been increased too much(Table 7.1).A small analysis says the size of increase of all of the data in real world.

Table 7.1 Range of Data

Year Range	Size of data(exabyte)
Upto 2005	130
2005-2010	1200
2010-2015	7900
2015-2020	40,900

Machine learning is an approach to get the real life data into the action over human analysis .This project has an aim to achieve that much goal because all machine learning algorithms intends to go to the better way than a human.

This project is a very much preliminary project based on those .This world entitles the work of Google everyday who himself hasn't achieved that much data also. This project entitles some different new ideas on

1. Image Processing
2. Machine learning
3. Activation Functions
4. Statistical predictive modeling
5. Optimiser into the programing
6. Text analysis
7. Digit extraction Features.

7.2 FORESEEABLE ENHANCEMENT

This project can be enhanced with a great field of machine learning and arfiticial intelligence. The world can think of a software which can recognise the text from a picture and can show it to the others, for example a the shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total english alphabet becuase there will be more and many more training sets and testing values that the neural network model will not be enough to detect. Think of a AI modeled car sensor going with a direction modeling in the roadside, user shall give only the destination.All of these enhancement is an application of the texture analysis where advanced image processing,Neural network model for training and advanced AI concepts will come.These applications can be modeled further .As this project is fully done by free and available resources and packages this can be also a limitation of the project.

The fund is very important because all machine learning libraries and advanced packages are not available for free.Unless of those the most of the visualizing platforms like on which developers are doing some works like Watson Studio or Aws.These all are mainly paid platforms where a lot of ML projects are going on.

CHAPTER 8

REFERENCES

1. LeCun et al., "Handwritten digit recognition with a backpropagation network," in Advances in neural information processing systems, 1990, pp. 396-404

2. Xiaofeng Han and Yan Li (2015), “The Application of Convolution Neural Networks in Handwritten Numeral Recognition” in International Journal of Database Theory and Application, Vol. 8, No. 3, pp. 367-376.

3. Caiyun Ma, Hong Zhang (2015), “Effective Handwritten Digit Recognition Based on Multi-feature Extraction and Deep Analysis”, 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 297-301.

4. Dan ClaudiuCiresan, Ueli Meier, Luca Maria Gambardella, JurgenSchmidhuber, “Deep big simple neural Nets Excel On Handwritten Digit Recognition”, MIT Press, 1st March 2010.

5. Xiaofeng Han and Yan Li (2015), “The Application of Convolution Neural Networks in Handwritten Numeral Recognition” in International Journal of Database Theory and Application, Vol. 8, No. 3, pp. 367-376.

6. Yoshihiro Shima, Meisei, Yumi Nakashima, Michio Yasuda, Meisei (2017), “Pattern Augmentation for Handwritten Digit Classification based on Combination of Pre-trained CNN and SVM”, 6th international Conference on informatics, Electronics and vision (ICIEV) and 7th International Symposium n Computational medical and health technology (ISCMHT).

7. Matthew Y.W. Teow Artificial Intelligence Lab (21 October 2017), “Understanding Convolutional Neural Networks Using A Minimal Model for Handwritten Digit Recognition”, 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS 2017), Kota Kinabalu, Sabah, Malaysia, pp. 167-172.

8. Dan ClaudiuCires¸an, Ueli Meier, Luca Maria Gambardella, Jurgen Schmidhuber (March 2010), ¨Deep, Big, Simple Neural Nets for Handwritten Digit Recognition”, arXiv, pp. 1-14.

9. Li Deng (November 2012), “The MNIST Database of Handwritten Digit Images for Machine Learning Research”, Best of the web series, IEEE signal processing magazine, pp. 141-142.

10. Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, W. Hub, “Handwritten Digit Recognition : Applications of Neural Network Chips and Automatic Learning” NATO ASI series F: Computer and system sciences, Vol. 68, pp. 41-46.

11. Li Deng, “The MNIST Database of Handwritten Digits images for Machine Learning Research”, MIT press, November 2012.