PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

ABSTRACT

The project work is a practical experience of the knowledge one has.The documentation leads a way to the concept to present the thinking and the upgradation of various techniques into the project .This project entitled **“HANDWRITTEN DIGIT RECOGNITION”** is a practical project based on some trends of computer science.Every day the world is searching new techniques in the field of computer science to upgrade the human limitations into machines to get more and more accurate and meaningful data.The way of machine learning and artificial intelligence has no negative slop it has only the slop having positive direction.This project is a very basic idea of those concepts .This project deals with the very popular learning process called Neural Network. There are various ways by which one can achieve the goal to a desired output,but in machine learning Neural network gives a way that machine learns the way to reach the output.

This project has come through the concepts of statistical modeling,the computer vision and machine learning libraries which includes a lot of study about these concepts.I tried to lead these project to the end of some updated techniques, upgradation and application of some new algorithms . This project has a good explanation and this project can be enhanced further into some complex applications of machine learning.

INDEX

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW :

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

- ☞ Training part.
- ☞ 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.

1.2 PURPOSE :

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 by the software provided by the Google Colab free version.

Apart from choosing Anaconda Navigator, Spyder or Jupyter Notebook, Google Colab or Colabotory have been chosen because this provide more speed and accurate compilation as is known. Creation of the machine learning algorithms deals with data and bigger size programs.

This project deals with the End users with some functionalities.The major functionality can be the Recognition of digit.User can write a handwritten digit 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 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.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM :

These days, a growing number of people are using images to transfer data. In addition it is the main distribution to separate the important data from the images. Image Recognition is an important research area for your most used apps. In general, in the field of pattern recognition, one of the most difficult tasks is the precise computerization of human handwriting. Without a doubt, this is a very difficult subject because there are so many variations of handwriting from person to person. Despite the fact that, this difference does not cause problems for humans, however, it is becoming increasingly difficult to instruct computers to interpret common handwriting. In the case of image recognition, for example, classification by hand, it is important to know how the information is displayed in the pictures.

The Handwritten Recognition from the MNIST database is well known to scientists as through the use of different parameters separators, the error rate is reduced.for example, from the line phase (1 layer NN) and 12% to 0.23% with a board of 35 convolution neural systems. The scope of this is to use the Handwritten Digital Awareness Framework and think of different categories and strategies by focusing on how to achieve closeness to personal performance. In the task of naming different digits (0-9) for different people the most common issue to be dealt with is the issue of digit order and the closeness between digits such as 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so on.

In addition, people create the same digit with different ideas, the diversity and diversity in the handwriting of different people also contributes to the development and existence of digits.

2.2 REFERENCES :

[1] Fischer, A., Frinken, V., Bunke, H.: Hidden Markov models for off-line cursive handwriting recognition, in C.R. Rao (ed.): Handbook of Statistics 31, 421 – 442, Elsevier, 2013.
[2] Frinken, V., Bunke, H.: Continuous handwritten script recognition, in Doermann, D., Tombre, K. (eds.): Handbook of Document Image Processing and Recognition, Springer Verlag, 2014.
[3] S. Günter and H. Bunke. A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In Proc. 16th Int. Conf. on Pattern Recognition, volume 2, pages 332–337. IEEE, 2002.
[4] U.-V. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In Proc. 6th Int. Conf. on Document Analysis and Recognition, pages 159– 163.
[5] M. Liwicki and H. Bunke, "Iam-ondb - an on-line English sentence database acquired from the handwritten text on a whiteboard," in ICDAR, 2005.
[6] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in Advances in neural information processing systems, 2009, pp. 545– 552.
[7] Nafiz Arica, and Fatos T. Yarman-Vural, —Optical Character Recognition for Cursive Handwriting,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113, June 2002.
[8] Anita Pal and Davashankar Singh,"Handwritten English Character Recognition Using Neural Network", International Journal of Computer Science and Communication, pp: 141- 144, 2011.
[9] Sandhya Arora, "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.
[10] Om Prakash Sharma, M. K. Ghose, Krishna Bikram Shah, "An Improved Zone Based Hybrid Feature Extraction Model for Handwritten Alphabets Recognition Using Euler Number", International Journal of Soft Computing and Engineering (ISSN: 2231 - 2307), Vol. 2, Issue 2, pp. 504- 508, May 2012.

2.3 PROBLEM STATEMENT 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.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.

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS :

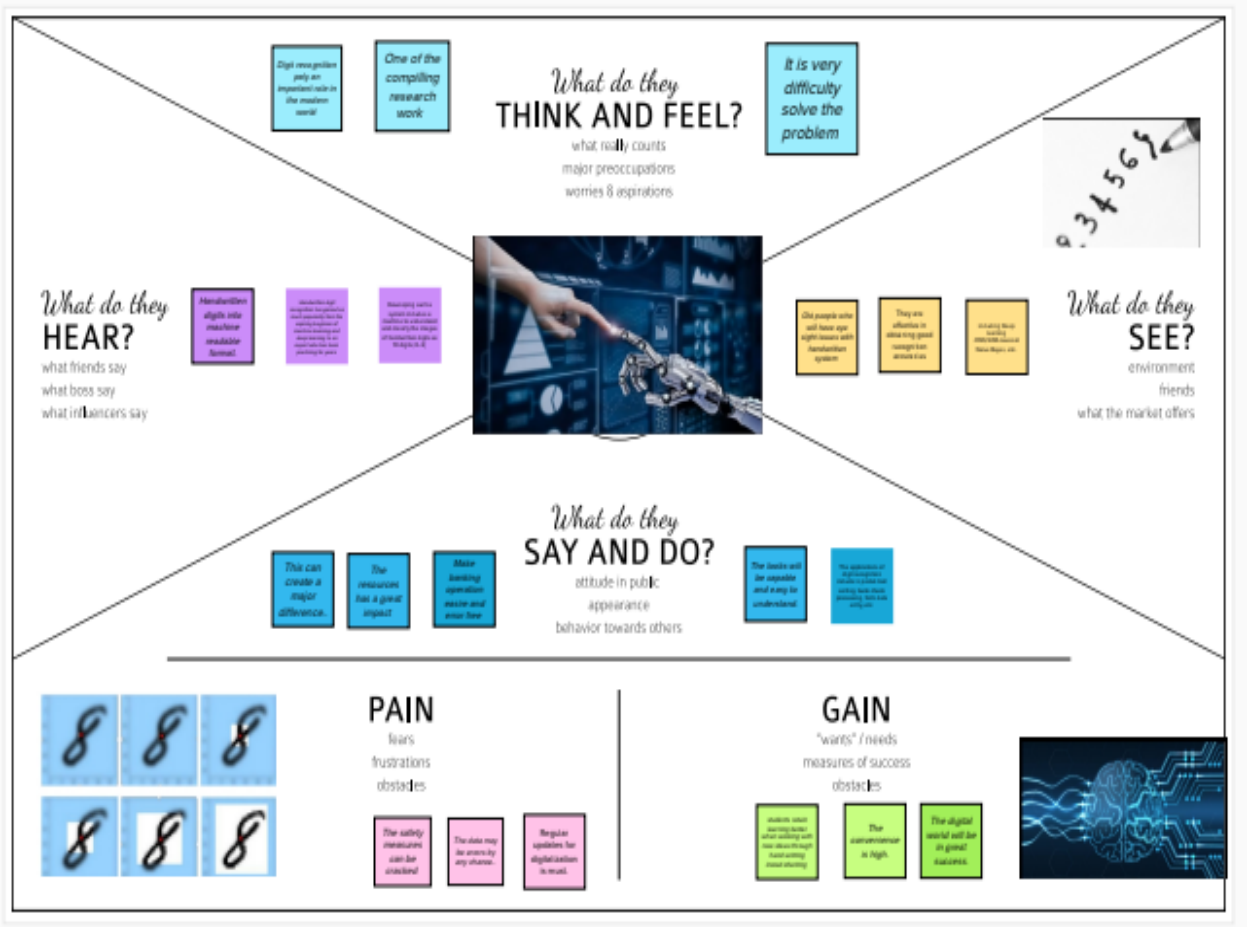

Fig 1.1 - Empathy Map Canvas for
Handwritten Digit Recognition System

3.2 IDEATION & BRAINSTORMING :

Step-1: Team Gathering, Collaboration and Select the Problem Statement :

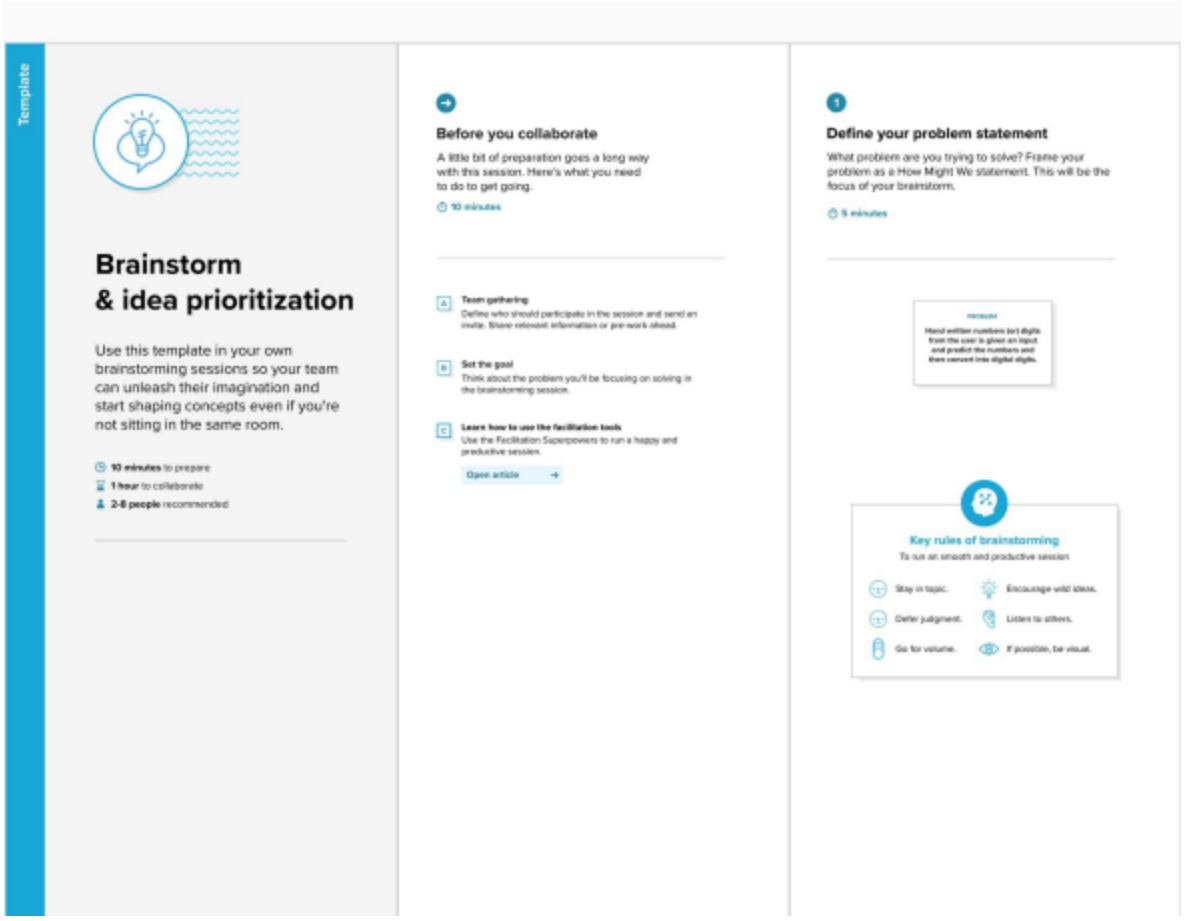

Fig 1.2 - Team Gathering, Collaboration
and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping :

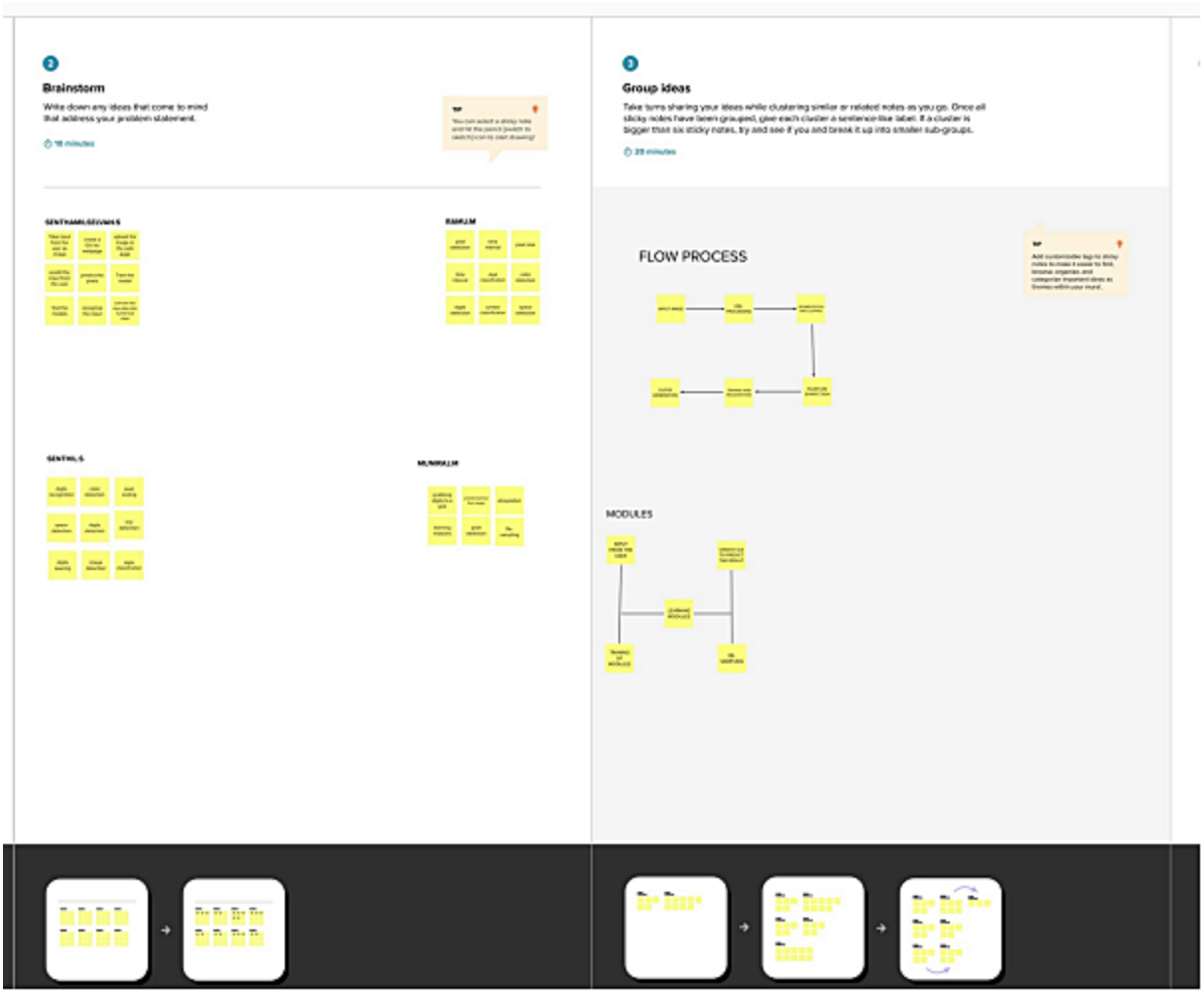

Fig 1.3 - Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization :

Fig 1.4 - Idea Prioritization

3.3 PROPOSED SOLUTION :

It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyse images easily. Also, recognize the different elements present in the images.We presented a system for dealing with such problem. The system started by acquiring an image containing digits, this image was digitized using some optical devices and after applying some enhancements and modificationsto the digits within the image it can be recognized using feed forward back propagation algorithm.Handwritten digits are made with many different flavours.The recommendation system and user Interface of the platform will satisfies the customers in detection of banksforreading cheques, vehicle numbers, post offices for arranging letters.We are building a platform for the dealer so that he may sell hisitemsthrough it and we will pay the dealer based on how well the consumer satisfied with the product.We will charge dealers based on the quantity of goods they sell through our platform.

Key rationale toward optical character recognition (OCR) from handwritten image includes features extraction technique supported by a classification algorithm for recognition of characters based on the features. Previously, several algorithms for feature classifications and extraction have been utilized for the purpose of character recognition. But, with the advent of CNN in deep learning, no separate algorithms are required for this purpose. However, in the area of computer vision, deep learning is one of the outstanding performers for both feature extraction and classification. However, DNN architecture consists of many nonlinear hidden layers with a enormous number of connections and parameters. Therefore, to train the network with very less amount of samples is a very difficult task. In CNN, only few set of parameters are needed for training of the system. So, CNN is the key solution capable to map correctly datasets for both input and output by varying the trainable parameters and number of hidden layers with high accuracy [49]. Hence, in this work, CNN architecture with Deeplearning4j (DL4J) framework is considered as best fit for the character recognition from the handwritten digit images. For the experiments and verification of system’s performance, the normalized standard MNIST dataset is utilized.

The performance of a CNN for a particular application depends on the parameters used in the network. In general, CNN architecture comprised of two main units or parts: (a) feature extractor and (b) feature classifier. In the feature extraction unit, every layer of network collects the output from the immediate previous layer (as input) and forward the current output to the immediate next layer as inputs; contrarily, classification unit generates the predicted outputs.

3.4 PROBLEM SOLUTION FIT :

Deep learning has been widely used to recognise handwriting. In offline handwriting recognition, text is analysed after being written. The only information that can be analysed is the binary output of a character against a background. Although shifts towards digital stylus for writing gives more information, such as pen stroke, pressure and speed of writing, there is still a necessity for offline methods, when online is inaccessible. It is particularly necessary for historical documents, archives, or mass digitisation of hand-filled forms.Extensive research into this field has resulted in significant progress from classical methods, right up to human-competitive performance. This article serves as an overview of that journey, and the potential future of the field.

We captured the images and converted it into 16x16 pixels. We are considering 100 samples of each image i.e. 1000 samples as a dataset. These images are then converted into gray scale.Captured images may not be necessarily in a form that can be used by image analysis routines. We may need to reduce noise or may need to simplify, enhanced, and altered, filtered etc. for the improvement. In this the first step is to transfer the coloured image into gray scale image which will result to noisy gray scale image.In this stage, the filtering is done to cancel out the presented noise. It is necessary to remove the noise from the image since it may cause difference between the captured image and actual palm. Edge detection will be difficult in noisy image as noise and the edges contain high frequency content. The noise produced is mainly due to device we use for capturing images, or atmospheric conditions or surroundings etc.To obtain edge of the noiseless gray scale image, edge detection algorithm is applied. Edge detection operators can’t detect every edges like gray level edges, color edges, texture edges etc. Each of different operation has specialty in edges and thus for better edge detection more complex will the operation be.Here, we check for the maximum connected components with the properties of each component that is in the box form. Then individual characters are cropped into different sub images, which is the raw data for the feature extraction. Once the extraction of the character is done, we pass it to another stage for further classification and training purpose of the neural network. For Logistic Regression, We are loading the training data in the work space. Images shown in section Fig 1 have 1000 handwritten digits (100 for each digit). Each digit image is a 16x16 pixel image. So we get 1000 dataset and these dataset are used to train and test the neural network. By using logistic regression algorithm we get the best probability of a scanned image and based on that we classify the image from 0 to 9.

CHAPTER 4

4. REQUIREMENT ANALYSIS :

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.

4.1 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) Webcam
- connectivity: Yes

4.2 SOFTWARE REQUIREMENTS :

SOFTWARE	TYPE/PLATFORM	VERSION
Operating System	Windows, Linux ubuntu,fedora or 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
Tensorflow	--	Tensorflow 2.1.6 or above
Ipython	--	Ipython 7.10 or above
Pil	--	Pil 1.1.5 or above

CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS :

A flowchart is a type of diagram that represents a workflow or process.A flowchart can also be defined as diagrammatic representation of an algorithm, a step by step approach.

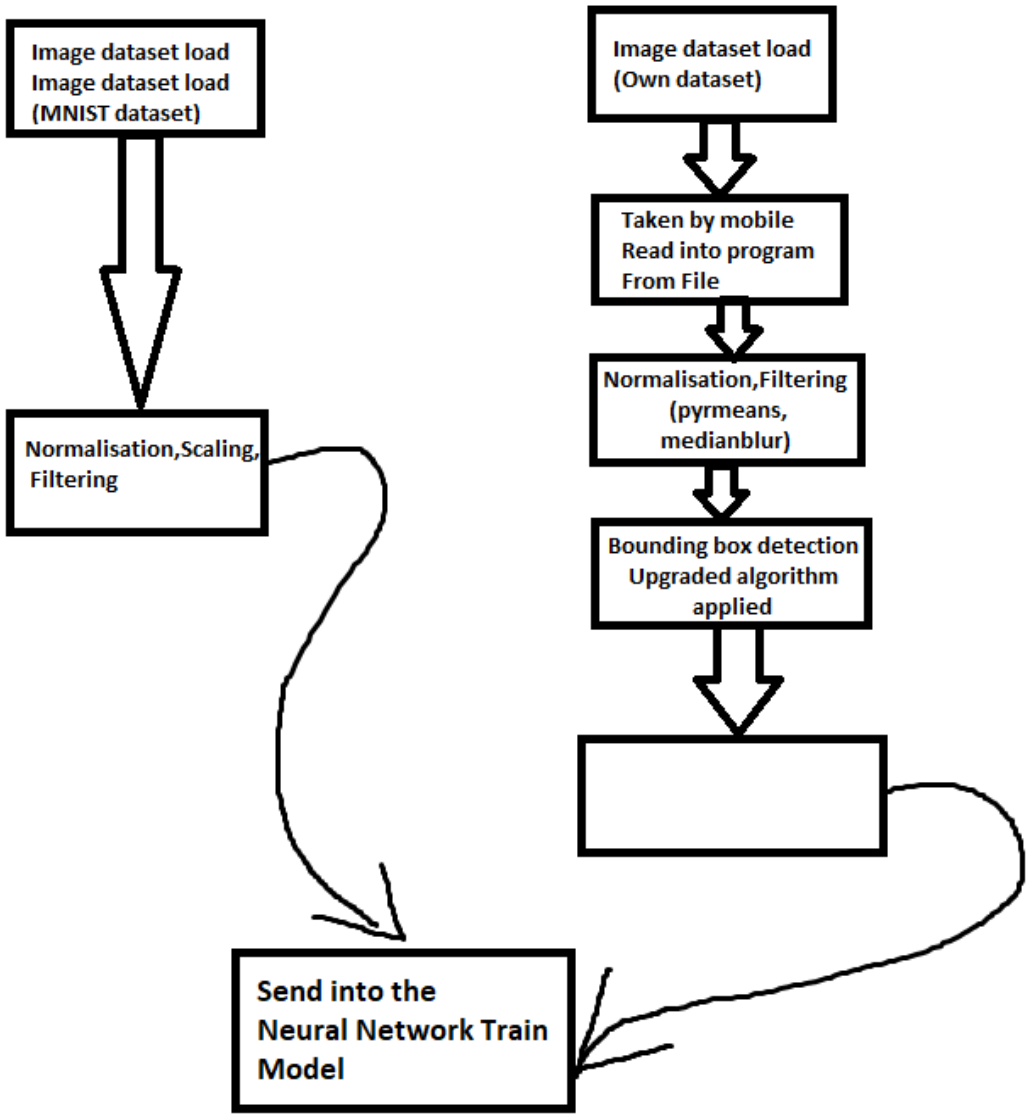

Fig 1.5 - Image processing for Training Data

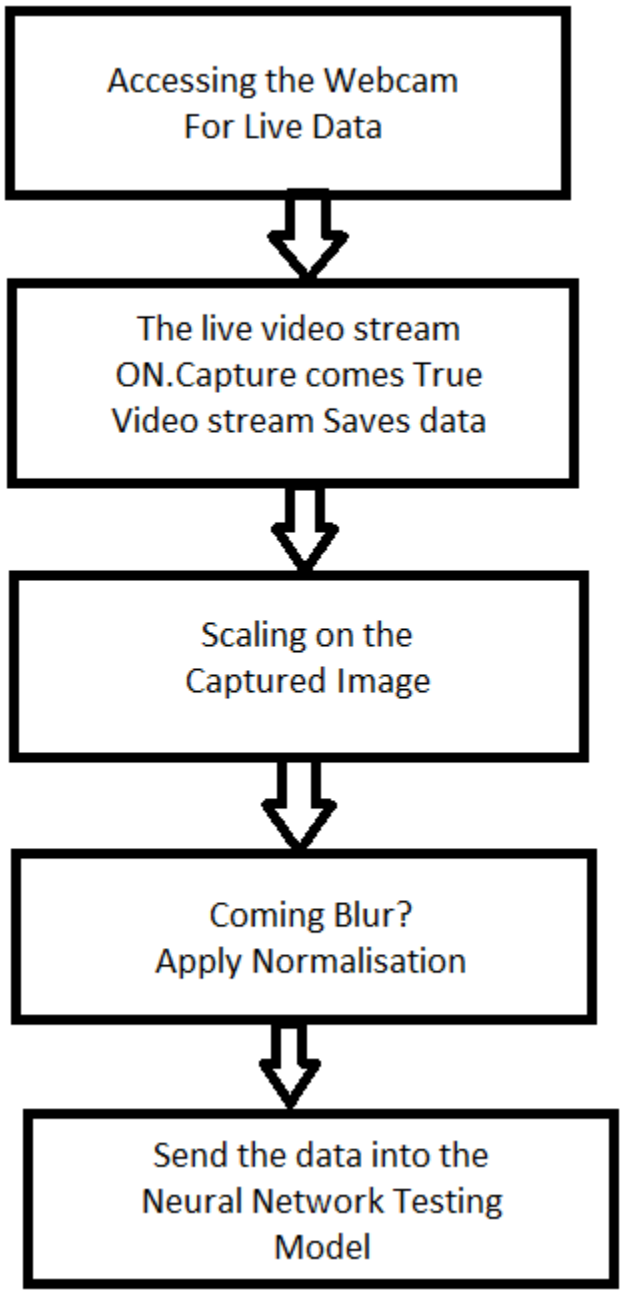

Fig 1.6 - Image Processing for Testing data

5.2 SOLUTION & TECHNICAL ARCHITECTURE :

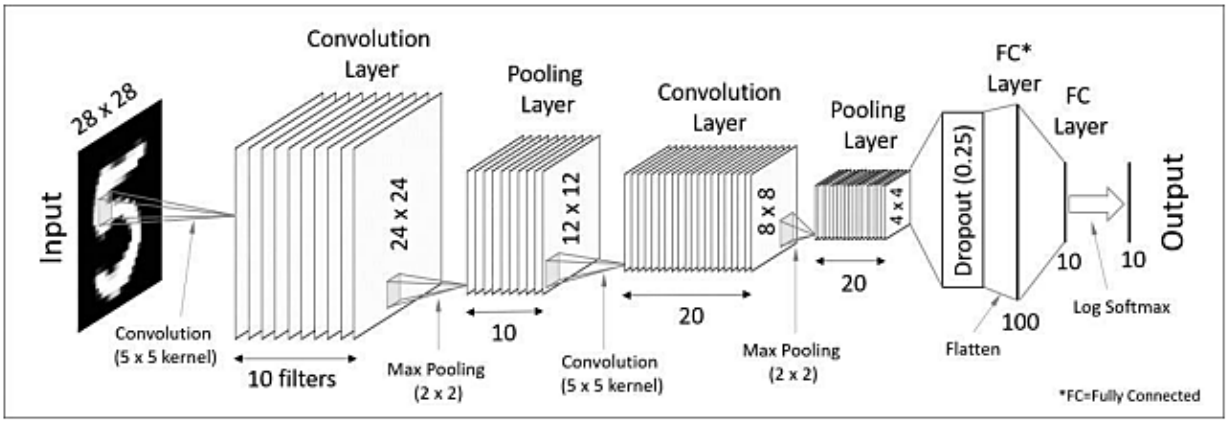

Fig 1.7 - CNN Architecture For Handwritten Digit Recognition

MOBILE PICTURES OF CHARACTER WRITTEN ON A PAPER
VARIOUS ALGORITHMS APPLIED FOR NOISE FILTERING
CONTOUR DETECTION

Fig 1.8 - Image Processing parts

Hierarchy
Bounding Box
Putting Contours

Fig 1.9 - Contour detection upgradation

5.3 USER STORIES :

The aim of our feasibility study is to select the system that fits the most our performance requirements. We would like to determine if it is possible to develop our product in terms of resources and technicalities. Thus, we will analyze the problem and collect information for the product, including the data we will input to our system, how we will carry our process on that data, and the output we wish to obtain following our process, as well as the constraints applied on how the system behaves.

On the technical level, our two main concerns will be finding the right data and finding and coding the right algorithms. As for the input data, we will use a famous handwritten digits dataset assembled by the National Institute of Standards and Technology and arranged by Yann Lecun, professor at NYU, available at http://yann.lecun.com/exdb/mnist/. It contains a training set of 60,000 examples and a testing set of 10,000 examples.

Then, to input the data, the images will be scanned pixel by pixel by our program. Each pixel will have a value saying how dark its color is, and the value for each pixel will be put in an array. Then, the value of each element from the array will be fed to the input layer of our neural network. Each pixel value will be given to a unit from the input layer.

We will then use different types of neural network algorithms and select the one that gives us the most precise predictions. Part of our work will consist of adapting the different codes for neural networks available to our program. Another part will consist in building an interface for the user to input a handwritten number with a touchscreen.

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION :

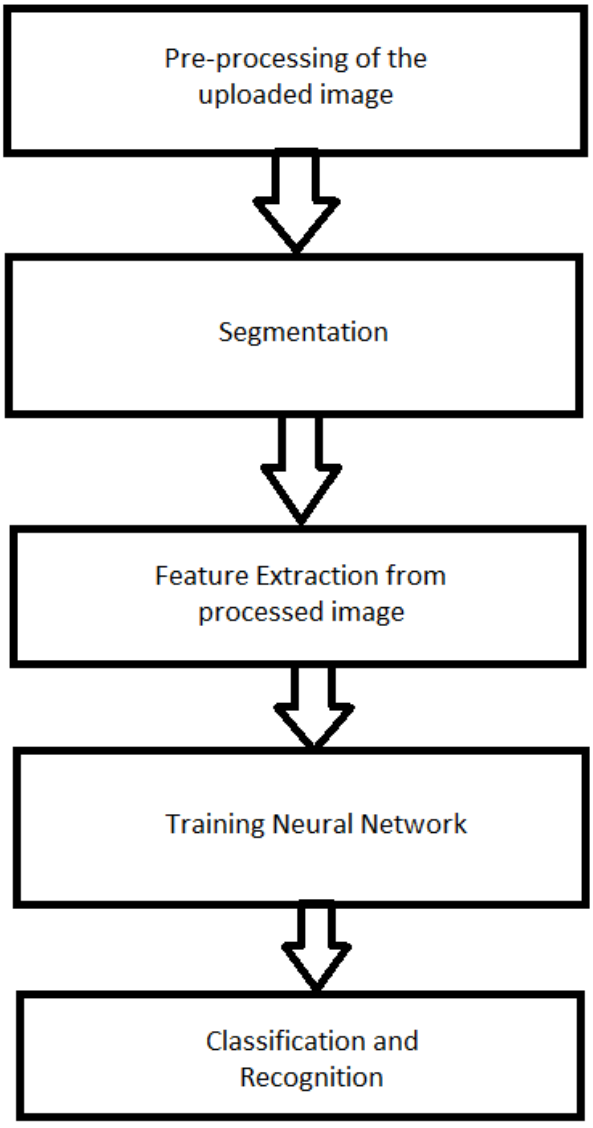

Fig 1.10 - Sprint Planning & Estimation

CHAPTER 7
CODING & SOLUTIONING

7.1 FEATURE 1

First, we are going to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easily import the dataset and start working with it. The mnist.load_data() method returns us the training data, its labels and also the testing data and its labels.

The image data cannot be fed directly into the model so we need to perform some operations and process the data to make it ready for our neural network. The dimension of the training data is (60000,28,28). The CNN model will require one more dimension so we reshape the matrix to shape (60000,28,28,1).

Now we will create our CNN model in Python data science project. A CNN model generally consists of convolutional and pooling layers. It works better for data that are represented as grid structures, this is the reason why CNN works well for image classification problems. The dropout layer is used to deactivate some of the neurons and while training, it reduces offer fitting of the model. We will then compile the model with the Adadelta optimizer.

The model.fit() function of Keras will start the training of the model. It takes the training data, validation data, epochs, and batch size.It takes some time to train the model. After training, we save the weights and model definition in the ‘mnist.h5’ file.

We have 10,000 images in our dataset which will be used to evaluate how good our model works. The testing data was not involved in the training of the data therefore, it is new data for our model. The MNIST dataset is well balanced so we can get around 99% accuracy.

Now for the GUI, we have created a new file in which we build an interactive window to draw digits on canvas and with a button, we can recognize the digit. The Tkinter library comes in the Python standard library. We have created a function predict_digit() that takes the image as input and then uses the trained model to predict the digit.Then we create the App class which is responsible for building the GUI for our app. We create a canvas where we can draw by capturing the mouse event and with a button, we trigger the predict_digit() function and display

the results.

CHAPTER 8
TESTING

8.1 TEST CASES :

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.

- ☞ Being non-parametric, the algorithm does not make any assumption about the underlying data assumption.

- ☞ Select the parameter K based on data.

- ☞ Requires a distance metric to define proximity between any two data points. This distance can be calculated from the Euclidean distance, Mahalanobis distance, Hamming distance, etc.

8.2 USER 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.

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.

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.Functional Testing of this project mainly involves

below things. All of these are tested successfully and errors are also calculated.

- ☞ Verifyng the input image
- ☞ Verifying the work flow
- ☞ Correct recognition and calculate the error

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.

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.

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.

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS :

Our model is built to work on real-world data, and real-world images are not even close to MNIST raster images, a lot of pre-processing was done to make a real image to look like a raster image.

Our model stopped training at the 2 nd epoch as it reached 98.21% training accuracy and 98.51% validation accuracy with 5% training loss and 4% validation loss.From the above curve, we can observe that the loss and accuracy are cooperatively changed at every fold during k-fold cross-validation. Before two folds, efficiency almost reached 98%, and that's why the number of iterations stopped at the 2 nd epoch. Stability inaccuracy score can be observed from 2nd iteration.

Our model is able to recognize computer-generated digits as well as handwritten digits. Computer-generated digit prediction is more accurate compared to real-world digit prediction.

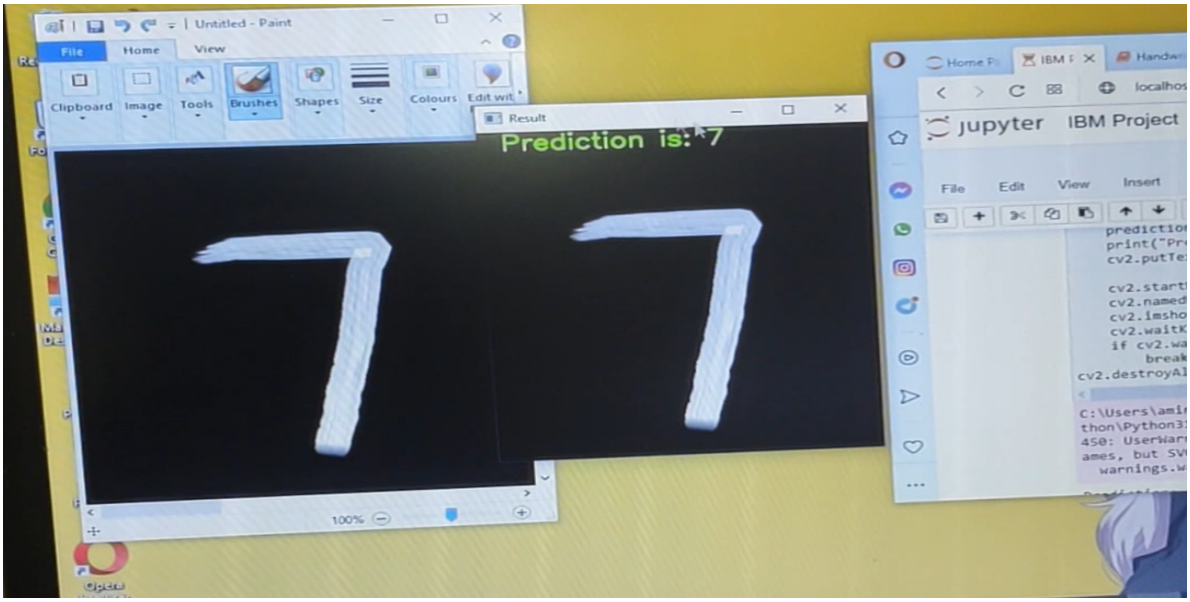

Fig 1.11 - Output

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- ☞ The system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style;

- ☞ The generative models can perform recognition driven segmentation;

- ☞ The method involves a relatively small number of parameters and hence training is relatively easy and fast; and

- ☞ Unlike many other recognition schemes, it does not rely on some form of pre-normalization of input images, but can handle arbitrary scalings, translations and a limited degree of image rotation. We have demonstrated that our method of fitting models to images does not get trapped in poor local minima. The main disadvantage of the method is that it requires much more computation than more standard OCR techniques.

DISADVANTAGES :

- ☞ The disadvantage is that **it is not done in real time as a person writes and therefore not appropriate for immediate text input**. Applications of offline handwriting recognition are numerous: reading postal addresses, bank check amounts, and forms.

- ☞ Perhaps the most obvious problem when processing handwritten forms during the data capture process is poor quality or illegible handwriting. We all know the old stereotype about doctors' handwriting, so trying to perform accurate data capture and validation on this type of form-filling may result in little meaningful data being extracted.A study in 2012 by the Educational Summit found that "25-35% of students at a secondary school level have still not gained competency in the skill of handwriting" meaning that forms filled out by hand could present an on-going challenge to data collection processes.

CHAPTER 11
CONCLUSION

All in all, we were able to program a handwritten digit recognizing program, made using machine learning and neural networks, more precisely. We needed to make a program that, first, could as input the Modified National Institute of Science and Technology dataset, then use neural networks to learn from that dataset how to recognize handwritten digits, and then apply what it learned on a testing set, first, with a precision greater than 95%. Then, we were able to implement an interactive part of the program, which implies making the program read a number that would be input through a touchscreen.

That first part was, however, just as interesting as actually understanding in depth the algorithm and how it works. That means, first, understanding what makes up a neural network, starting from a simple model (the one neuron perceptron) and exploring a more complic

ated one (with more neurons, layers and activation functions). We were also able to explore the two steps of the neural network algorithm (feedforward neural network algorithm) we used: the forward propagation and backward propagation. In the forward propagation we learned about the activation function (Sigmoid, Softmax, Rectifier Linear Unit). As for the Backward propagation, we understood the loss function and the mathematical logic that makes it a good solution to solving our machine learning problem.

Finally, we were able to learn more about different libraries in Python. That includes Tensorflow for machine learning and Keras (contained in Tensorflow) for machine learning for image recognition. But it also includes the library Image that allowed us to convert and filter our own input image into one that can be read and used by our Neural Network. Finally, we learned about the Tkinter library, which is the standard graphical user library interface for Python.

CHAPTER 12
FUTURE SCOPE

The proposed recognition system is implemented on handwritten digits taken from MNIST database. Handwritten digit recognition system can be extended to a recognition system that can also able to recognize handwritten character and handwritten symbols. Future studies might consider on hardware implementation of recognition system.

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems. In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people, we can use these algorithms in hospitals application for detailed medical diagnosis, treatment and monitoring the patients, we can use it in surveillances system to keep tracks of the suspicious activity under the system, in fingerprint and retinal scanners, database filtering applications, Equipment checking for national forces and many more problems of both major and minor category. The advancement in this field can help us create an environment of safety, awareness and comfort by using these algorithms in day to day application and high-level application (i.e. Corporate level or Government level). Application-based on artificial intelligence and deep learning is the future of the technological world because of their absolute accuracy and advantages over many major problems.

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.

CHAPTER 13
APPENDIX

Python:

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, objectoriented, and functional programming.

Keras :

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep larning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Steps for creating a keras model:

- ☞ First we must define a network model.
- ☞ Compile it, which transforms the simple sequence of layers into a complex group of matrix operations.
- ☞ Train or fit the network. To import: from keras.models import Sequential fromkeras.layers import Dense, Activation, Dropout

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by

using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow tutorial is designed for both beginners and professionals. Our tutorial provides all the basic and advanced concept of machine learning and deep learning concept such as deep neural network, image processing and sentiment analysis. 62 TensorFlow is one of the famous deep learning frameworks, developed by Google Team. It is a free and open source software library and designed in Python programming language, this tutorial is designed in such a way that we can easily implements deep learning project on TensorFlow in an easy and efficient way. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems. It can run on single CPU systems, GPUs as well as mobile devices and largescale distributed systems of hundreds of machines.

Numpy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. Numpy which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. It is an open source project and you can use it freely. NumPy stands for Numerical Python. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Neural Networks:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Source Code :

```
import pyscreenshot as ImageGrab
import time
images_folder = "captured_images/0/"
for i in range(0,5):
    time.sleep(8)
    ImageGrab.grab(bbox=(60,170,400,550))
    print("Saved...",i)
    im.save(images_folder+str(i)+'.png')
    print("clear screen and redraw again...")

#Generate dataset
import cv2
import csv
import glob

header  =["label"]
for i in range(0,784):
   header.append("pixel"+str(i))
with open('dataset.csv', 'a') as f:
   writer = csv.writer(f)
   writer.writerow(header)

for label in range(10):
   dirList = glob.glob("captured_images/"+str(label)+"/*.png")

   for img_path in dirList:
      im= cv2.imread(img_path)
      im_gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
      im_gray = cv2.GaussianBlur(im_gray,(15,15), 0)
      roi= cv2.resize(im_gray,(28,28), interpolation=cv2.INTER_AREA)

      data=[]
      data.append(label)
      rows, cols = roi.shape
```

```
    ## Add pixel one by one into data array
    for i in range(rows):
        for j in range(cols):
            k =roi[i,j]
            if k>100:
                k=1
            else:
                k=0
            data.append(k)
    with open('dataset.csv', 'a') as f:
        writer = csv.writer(f)
        writer.writerow(data)
```

#load the dataset

```
import pandas as pd
from sklearn.utils import shuffle
data  =pd.read_csv('dataset.csv')
data=shuffle(data)
data
```

#separation of dependent and independent variable

```
X = data.drop(["label"],axis=1)
Y= data["label"]
```

#preview of one image using matplotlib

```
%matplotlib inline
import matplotlib.pyplot as plt
import cv2
idx = 314
img = X.loc[idx].values.reshape(28,28)
print(Y[idx])
plt.imshow(img)
```

#Train-Test split

```
from sklearn.model_selection import train_test_split
train_x,test_x,train_y,test_y = train_test_split(X,Y, test_size = 0.2)
```

```
#Fit the model using svc and also to save the model using joblib
import joblib
from sklearn.svm import SVC
classifier=SVC(kernel="linear", random_state=6)
classifier.fit(train_x,train_y)
joblib.dump(classifier, "model/digit_recognizer")

#calculate accuracy
from sklearn import metrics
prediction=classifier.predict(test_x)
print("Accuracy= ",metrics.accuracy_score(prediction, test_y))

#prediction of image drawn in paint
import joblib
import cv2
import numpy as np #pip install numpy
import time
import pyscreenshot as ImageGrab

model=joblib.load("model/digit_recognizer")
image_folder="img/"

while True:
  img=ImageGrab.grab(bbox=(60,170,400,500))

  img.save(images_folder+"img.png")
  im = cv2.imread(images_folder+"img.png")
  im_gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
  im_gray  =cv2.GaussianBlur(im_gray, (15,15), 0)

  #Threshold the image
  ret, im_th = cv2.threshold(im_gray,100, 255, cv2.THRESH_BINARY)
  roi = cv2.resize(im_th, (28,28), interpolation  =cv2.INTER_AREA)

  rows,cols=roi.shape
```

```
  X = []

  ## Add pixel one by one into data array
  for i in range(rows):
      for j in range(cols):
          k = roi[i,j]
          if k>100:
              k=1
          else:
              k=0
          X.append(k)

  predictions  =model.predict([X])
  print("Prediction:",predictions[0])
  cv2.putText(im, "Prediction is: "+str(predictions[0]), (20,20), 0, 0.8,(0,255,0),2,cv2.LINE_AA)

  cv2.startWindowThread()
  cv2.namedWindow("Result")
  cv2.imshow("Result",im)
  cv2.waitKey(10000)
  if cv2.waitKey(1)==13: #27 is the ascii value of esc, 13 is the ascii value of enter
      break
cv2.destroyAllWindows()

import tkinter as tk
from tkinter import *
from tkinter import messagebox

window=tk.Tk()
window.title("Handwritten digit recognition")

l1=tk.Label(window,text="Digit",font=('Algerian',20))
l1.place(x=5,y=0)

t1=tk.Entry(window,width=20, border=5)
t1.place(x=150, y=0)
```

```
def screen_capture():
  import pyscreenshot as ImageGrab
  import time
  import os
  os.startfile("C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Accessories/Paint")
  s1=t1.get()
  os.chdir("E:/DS and ML/Untitled Folder/Untitled Folder/captured_images")
  os.mkdir(s1)
  os.chdir("E:/DS and ML/Untitled Folder/Untitled Folder/")

  images_folder="captured_images/"+s1+"/"
  time.sleep(15)
  for i in range(0,5):
    time.sleep(8)
    im=ImageGrab.grab(bbox=(60,170,400,550)) #x1,y1,x2,y2
    print("saved......",i)
    im.save(images_folder+str(i)+'.png')
    print("clear screen now and redraw now........")
  messagebox.showinfo("Result","Capturing screen is completed!!")

b1=tk.Button(window,text="1. Open paint and capture the screen",
font=('Algerian',15),bg="orange",fg="black",command=screen_capture)
b1.place(x=5, y=50)

def generate_dataset():
  import cv2
  import csv
  import glob

  header  =["label"]
  for i in range(0,784):
    header.append("pixel"+str(i))
  with open('dataset.csv', 'a') as f:
    writer = csv.writer(f)
    writer.writerow(header)

  for label in range(10):
```

```
    dirList = glob.glob("captured_images/"+str(label)+"/*.png")

    for img_path in dirList:
        im= cv2.imread(img_path)
        im_gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        im_gray = cv2.GaussianBlur(im_gray,(15,15), 0)
        roi= cv2.resize(im_gray,(28,28), interpolation=cv2.INTER_AREA)

        data=[]
        data.append(label)
        rows, cols = roi.shape

        ## Add pixel one by one into data array
        for i in range(rows):
            for j in range(cols):
                k =roi[i,j]
                if k>100:
                    k=1
                else:
                    k=0
                data.append(k)
        with open('dataset.csv', 'a') as f:
            writer = csv.writer(f)
            writer.writerow(data)
  messagebox.showinfo("Result","Generating dataset is completed!!")

b2=tk.Button(window,text="2. Generate dataset",
font=('Algerian',15),bg="pink",fg="blue",command=generate_dataset)
b2.place(x=5, y=100)

def train_save_accuracy():
  import pandas as pd
  from sklearn.utils import shuffle
  data  =pd.read_csv('dataset.csv')
  data=shuffle(data)
  X = data.drop(["label"],axis=1)
  Y= data["label"]
```

```
    from sklearn.model_selection import train_test_split
    train_x,test_x,train_y,test_y = train_test_split(X,Y, test_size = 0.2)
    import joblib
    from sklearn.svm import SVC
    classifier=SVC(kernel="linear", random_state=6)
    classifier.fit(train_x,train_y)
    joblib.dump(classifier, "model/digit_recognizer")
    from sklearn import metrics
    prediction=classifier.predict(test_x)
    acc=metrics.accuracy_score(prediction, test_y)
    messagebox.showinfo("Result",f"Your accuracy is {acc}")

b3=tk.Button(window,text="3. Train the model, save it and calculate accuracy",
font=('Algerian',15),bg="green",fg="white",command=train_save_accuracy)
b3.place(x=5, y=150)

def prediction():
    import joblib
    import cv2
    import numpy as np #pip install numpy
    import time
    import pyscreenshot as ImageGrab
    import os
    os.startfile("C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Accessories/Paint")

    model=joblib.load("model/digit_recognizer")
    images_folder="img/"
    time.sleep(15)
    while True:
        img=ImageGrab.grab(bbox=(60,170,400,500))

        img.save(images_folder+"img.png")
        im = cv2.imread(images_folder+"img.png")
        im_gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        im_gray  =cv2.GaussianBlur(im_gray, (15,15), 0)

        #Threshold the image
```

```
        ret, im_th = cv2.threshold(im_gray,100, 255, cv2.THRESH_BINARY)
        roi = cv2.resize(im_th, (28,28), interpolation  =cv2.INTER_AREA)

        rows,cols=roi.shape

        X = []

        ## Add pixel one by one into data array
        for i in range(rows):
            for j in range(cols):
                k = roi[i,j]
                if k>100:
                    k=1
                else:
                    k=0
                X.append(k)

        predictions  =model.predict([X])
        print("Prediction:",predictions[0])
        cv2.putText(im, "Prediction is: "+str(predictions[0]), (20,20), 0,
0.8,(0,255,0),2,cv2.LINE_AA)

        cv2.startWindowThread()
        cv2.namedWindow("Result")
        cv2.imshow("Result",im)
        cv2.waitKey(10000)
        if cv2.waitKey(1)==13: #27 is the ascii value of esc, 13 is the ascii value of enter
            break
    cv2.destroyAllWindows()

b4=tk.Button(window,text="4. Live prediction",
font=('Algerian',15),bg="white",fg="red",command=prediction)
b4.place(x=5, y=200)

window.geometry("600x300")
window.mainloop()
```

GITLAB & PROJECT DEMO LINK :

https://github.com/IBM-EPBL/IBM-Project-10054-1659089598

https://drive.google.com/file/d/1i382nqPubab5PEnlIeTuAkr8Dw8k3qEZ/view

Presentation Link:

https://drive.google.com/file/d/1iDJkh2OI6SPDQ0c5WPAzoIZii7MuqII6/view