HANDWRITTEN DIGIT RECOGNITION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements of Professional Communication Skills II

Prepared By:

Name	Roll No
Arpita Nikam	525
Vaishnavi Pol	527
Riya Stanly	538
Susmit Walekar	545

Submitted to:

Prof. Darshana Pachkawade

(Department of Humanities)

Department Of Computer Engineering

Pillai College of Engineering

New Panvel – 410 206

Academic Year (2022-2023)

Acknowledgement

We have implemented our ideas in a unique way in this project successfully. For this success, we would like to thank our parents and all our respected teachers for their encouragement. We would like to express our special thanks to Principal Dr. Sandeep Joshi for giving us this platform to showcase our relevant skills. We would like to express our gratitude to The Department of Computer Engineering and our Head of Department Prof. Dr. Sharvari Govilkar for encouraging us throughout and also the PCS Co-ordinator Prof. Darshana Pachkawade who gave us the opportunity to do this project as a part of Professional Communication Skills Learning in the topic Handwritten Digit Recognition System, which helped us in applying the knowledge that we have acquired during the semester and to learn new concepts and for supporting and helping us throughout the project time period. We would also like to thank the lab assistants for helping us with various equipment. We would also like to thank all our colleagues for supporting each other. We are immensely grateful to all of them for sharing their pearls of wisdom with us during this course.

Summary

It is a hard piece of work for the machine to identify the handwritten digits because handwritten digits are not perfect and can be made with many different flavors. Many times, it becomes very difficult to recognize a handwritten character or the digits because every people;s handwriting varies from one-to-one. Thus to overcome this major problem, we are introducing the handwritten digit recognition system which is created by implementing the machine learning algorithms. Additional features are planned to be implemented in the future. The handwritten digit recognition system is the solution to this problem which uses the image of a digit and recognizes the digit present in the image. This project mainly introduces a handwritten digit recognition system for handwritten digits based on machine learning algorithms. When a handwritten digit is written live, it is recognized on the spot by the machine. Hence making it advanced and suitable for every problem of digit recognition. This machine could be further modified by implementing the handwritten character recognition features too. Handwritten digit recognition is the ability of the computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc and classify them into ten predefined classes(0-9).

Table Of Contents

Chapter 01

Introduction

Many times, handwritten digits are not recognized by the systems. Earlier various systems were made to overcome this problem but every time it ended in failure. It thus became a very big task for the machine to identify the human handwritten digits as the handwriting are of various flavors. Thus, a handwritten digit recognition system is made using machine learning algorithms to identify the digits easily even if it is of various flavors. The main objective of this system is to ensure effective and reliable approaches for recognition of handwritten digits and make various online transactions and economic well being easier and error free. The significance of this project is to introduce the recognition of various digits using machine learning algorithms through a relatively easy-to-understand application to the general public. This has a great importance in online handwriting too. Recently handwritten digit recognition has become a vital scope and it is appealing to many researchers because of its use in a variety of machine learning and computer vision applications. However, there are insufficient works accomplished on Arabic pattern digits because Arabic digits are more challenging than English patterns.

Chapter 02

Literature Review

Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9). Handwritten digit recognition has become an issue of interest among researchers. There are a large number of papers and articles being published these days about this topic. In the research paper by Fathima Siddique et al, it is shown that Deep Learning algorithms like multilayer CNN using Keras with Theano and Tensorflow gives the highest accuracy in comparison with the most widely used machine learning algorithms like SVM, KNN & RFC. In this paper, the variations of accuracies for handwritten digits were observed for 15 epochs by varying the hidden layers. The accuracy curves were generated for the six cases for the different parameter using CNN MNIST digit dataset.The main disadvantage of the method is that it requires much more computation than more standard OCR techniques.[1]

But in the research paper by D.C Ciresan et al "Flexible high performance convolutional neural networks for image classification in Twenty-Second International Joint Conference on Artificial Intelligence 2011", it is pretty challenging to get good performance as more parameters are needed for the large-scale neural network but it used much more computation than more standard OCR techniques. Many Researchers are trying to increase the accuracy with less error on CNN . In another research, they have shown that deep net performs better when they are trained by simple back propagation .Their architecture results in the lowest error rate on MNIST compare to NORB and CIFAR10[2] There also exists some other resulting lowest error that is, according to Vijaylaxmi R Ridraswamimath, digit recognition is a core problem as the manually written digits varies in in various characteristics.This research paper proposes a reliable approach to implement a classification algorithm to recognize the handwritten digits using Deep Learning

methods, few algorithms and MNIST dataset. As a result, compared to other research methods, this method classifier works better by improving the accuracy by more than 99%. Also,CNN gives an accuracy of 98.72%, KNN gives 96.67%, while RFC and SVM are not that Outstanding.[3]

Understanding CNN and applying it to the handwritten digit recognition system is the target of the proposed mode.To implement our CNN architecture, MatConvNet was used. It exposes the building blocks of CNN as easy-to-use MATLAB functions, providing routines for computing linear convolutions with filter banks, feature pooling and many more.The results are pretty good for such a simple model with CPU training and less training time.Testing accuracy 99.15% implies that the model is trained well for prediction. Training set size affects the accuracy and accuracy increases as the number of data increases. The more data in the training set, the smaller the impact of training error and test error, and ultimately the accuracy can be improved.[4]mBut according to Pranit Patil and Bhupinder Kaur, handwritten digit recognition is a pivotal concern in today's computer world. This research paper hence suggests machine learning algorithms that can help in reducing the complexity and improve the productiveness of digit recognition using different models. Based on various models used, three algorithms were found to yield a high accuracy score which was above 99%. Among the three algorithms the highest accuracy acquired was by Convolution Neural Network for handwritten digit recognition[5]

Akanksha Gupta et al suggests that handwritten digit recognition has numerous applications in different sectors. This research paper suggests machine learning techniques like K Nearest Neighbors, Support Vector Machine, Convolutional Neural Network can be used for handwritten digit recognition. It was found that Convolutional Neural Network performed better than other mentioned algorithms[6] But in Anuj Dutt and Aashi Dutt's study, "Handwritten Digit Recognition Using Deep Learning," compares the performance of some of the most popular Machine Learning algorithms, such KNN and RFC, with Deep Learning algorithms, like using Keras, Theano, and Tensorflow to create a multilayer CNN. They were able to achieve an accuracy of 98.70% by employing these and CNN. (Keras+Theano) as opposed to 96.67% for

methods, few algorithms and MNIST dataset. As a result, compared to other research methods, Applications of Neural Network Chips and Automatic Learning," the authors used neural network techniques to tackle a significant real-world task.[7]

This research paper prepared by Viragkumar N Jagtap et al proposes the fast efficient artificial neural network for handwritten digit recognition on GPU using standard BP algorithm with MLP classification. As a result, compared to previous research method approaches, this time a better accuracy of 98% was acquired. Also the training time was reduced too. It was found that if the input dataset is larger then GPU based parallelization is suitable to reduce training time. The research paper further expects to explore the capabilities of GPU and multicores in cloud environments and offering them as services, where the users can query and select them, depending on respective service level agreements.[8] SM Shamim et al proposes that in pattern recognition applications, handwritten digit recognition is one of the most important problem to be tackled. This research paper suggests an efficient and reliable approach for the handwritten digit recognition with the use of eight machine learning algorithms. Multilayer perceptron, which is a machine learning algorithm, had the highest accuracy for handwritten digit recognition. The other used had more amount of incorrectly classified instances.[9]

Ritik Dixit performed handwritten digit recognition with the help of MNIST datasets using SVM , MLP , CNN Models. The main objective is to compare the accuracy of the models stated along with their execution time to get the best possible model for digit recognition. After implementing all 3 algorithms (SVM , MLP , CNN) their accuracies and execution time with the help of experimental graphs were compared and it was found that SVM has highest accuracy. The limitations of research paper are that by comparing execution time of algorithms, the increasing then number of epochs without changing the configuration of algorithm is useless.[10] Thus the highest accuracy, Convolutional Neural Network (CNN) is being used on a large scale in image classification, video analysis, etc. Many researchers are trying to make sentiment recognition in a sentence. CNN is being used in natural language processing and sentiment recognition by varying different parameters. Other ML algorithms are also used but the best way to get the highest accuracy is using Convolutional Neural Network Algorithm.

Chapter 03

Methodology

Primary Sources: The primary sources for the handwritten digit recognition system needed are the handwritten digits of different people from different regions. The various varieties of human handwritten digits are collected. For this the dataset used was the Modified National Institute of Standards and Technology dataset. It includes 60 thousand squares filled with the digits. Each square is of 28x28 pixel size. Thus this dataset helped to find out various handwriting. Also it was needed for testing and training of the machine designed.

Secondary sources: The language used to implement this idea was python. Also the use of a jupyter notebook was identified. To learn those things, some books and the internet was considered to be the source. From this source, the way of implementing and developing a machine was understood.

Chapter 04

Data Collection

The dataset is to be analyzed and collected for the process throughout.

4.1 MNIST Dataset:

MNIST stands for Modified National Institute of Standards and Technology dataset.

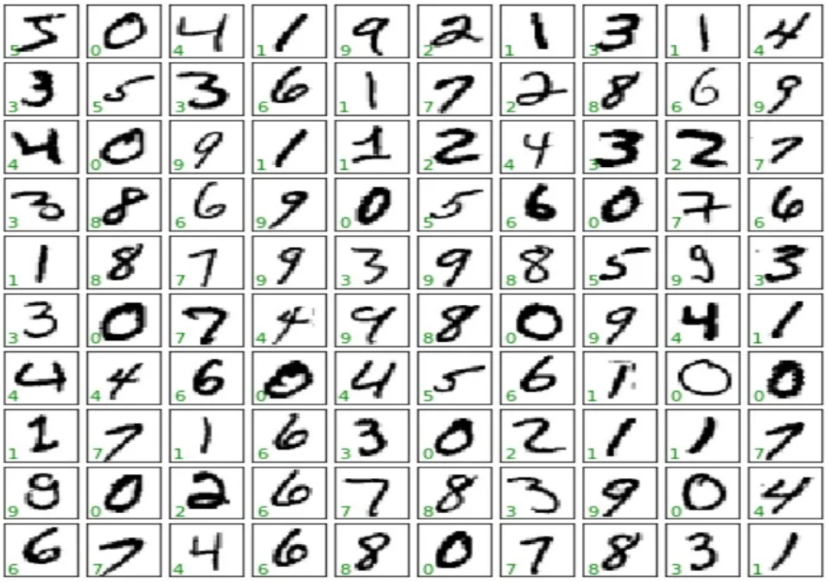


Fig 1.1

The MNIST dataset consists of 60 thousand squares of 28x28 pixel size. Fig 1.1 shows this dataset is a collection of the human handwritten digits. It is a simple dataset which can be easily used to understand different patterns and implementations. This MNIST was created by re-mixing the NIST dataset. It consists of digits from 0-9. The Kaggle site provides this dataset. In this project the different human handwritten digits were needed and thus this MNIST dataset was collected and analyzed and thus implemented. This dataset was downloaded from kaggle. The number of rows and columns required was

analyzed. While coding, the first step is to import the necessary libraries and the downloaded Modified National Institute of Standards and Technology dataset. MNIST is a large database and thus it was possible to train and test the machine successfully.

Various digits were tested using the MNIST dataset for getting the appropriate accuracy.

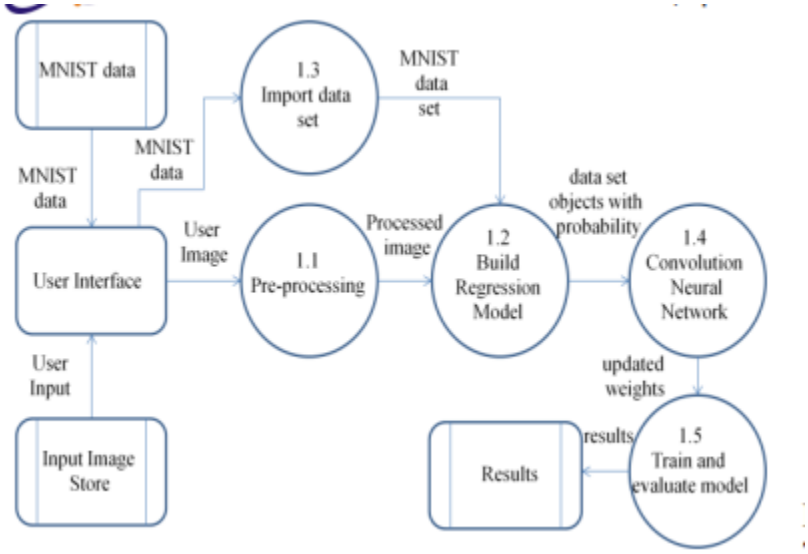


Fig 1.2

Fig 1.2 shows how the flow goes by using the MNIST dataset.

4.2 Model Selection:

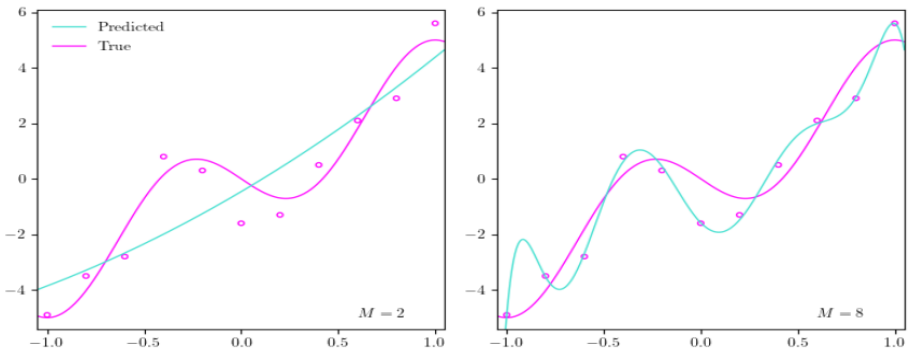


Fig 1.3

Fig 1.3 illustrates the selection of a model should be accurate. Model selection is the process of choosing the best suited model as a solution for a particular problem. The selection of a particular model depends upon various factors like the dataset we have selected, the nature of the machine, and the task to be implemented. Also the models can be selected on the basis of type of data available. If the data is of images and videos then Convolutional Neural Network is used. If it is text data or speech data, then Recurrent Neural Network is used. If the data is numerical, then Support Vector Machine, Logistic Regression and Decision Trees are used to find the solution to a particular problem.

Based on the task the model can be selected.If the task is to classify then support vector machines, logistic regression and decision trees are used. If the task is on regression, then hierarchical clustering methods are used to solve a clustering problem. This project mainly focuses on identification of the human handwritten digits. Thus the model uses the Convolutional Neural Network. By using this the recognition of a handwritten digit is done easily through various processes and coding. A real time handwritten digit recognition is also achieved. Cross validation should also be considered

as a factor to select the model. The accuracy for each model is found out then the best model with more accuracy is selected after cross validation. Cross validation is the process of validating if the actual value and the value we have got is more close or not. Also to check which model is appropriate after the model performance. 0. 98% accuracy was achieved and hence we selected the best model.

Chapter 05

Machine Learning Algorithms

The machine learning algorithms are implemented in this project to land in a proper accuracy.

5.1 Convolutional Neural Network:

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:

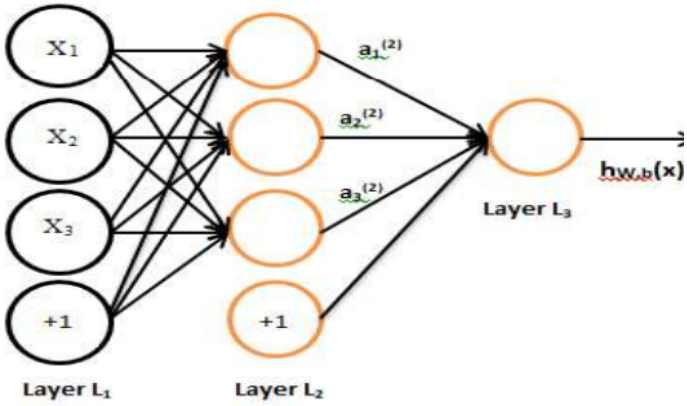


Fig 1.4

Fig 1.4 illustrates how the neurons are hooked up together to get an output of another neuron.A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification

algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

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 output 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.

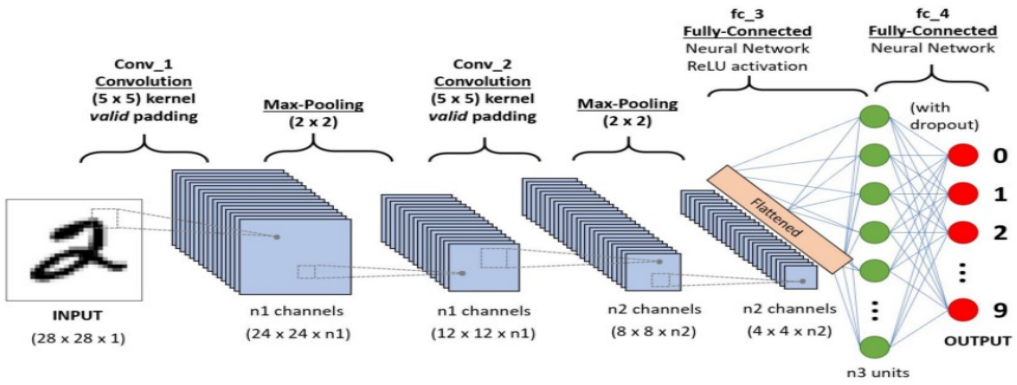


Fig 1.5

Fig 1.5 illustrates how input is taken and through valid padding, max-pooling, kernel valid pooling the accuracy is dropped out. Thus we have taken the Convolutional Neural Network Algorithm to get the best accuracy after testing. We decided to select this after cross validating this algorithm with other algorithms. Many times, the accuracy is not achieved by using this algorithm. But after implementing it in a proper way the right accuracy is achieved. After using the Convolutional Neural network the accuracy received was 0.98%. Thus it proves to be one of the best algorithms.

5.2 Support Vector Machine:

SVMs are used in applications like handwriting recognition, intrusion detection, face detection, email classification, gene classification, and in web pages. This is one of the reasons we use SVMs in machine learning. It can handle both classification and regression on linear and non-linear data. Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

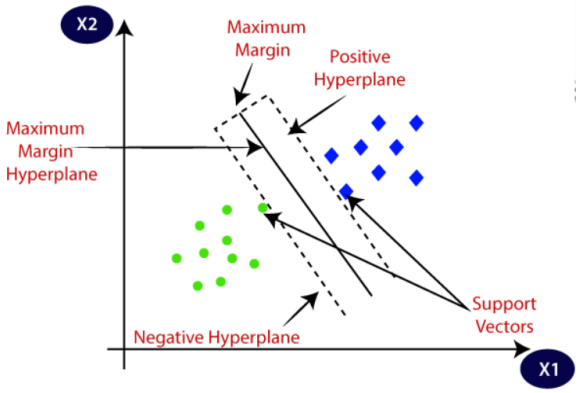


Fig 1.6

Fig 1.6 gives the picture of SVM,the hyperplane and also how the hyperplanes are

identified. SVM algorithms can be used for Face detection, image classification, text categorization, etc.

Types of SVM

SVM can be of two types:

- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Chapter 06

Implementation

The ideas are implemented after all the above discussions.

6.1 Training Dataset:

The dataset is trained to make the machine understand the various factors of the dataset. 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 datasets (for our case 60500 datasets) for better results and to upgrade the weights into the neural network.

In this project the max relu and softmax because this gives really good results tested and better than others. Total experiment is based on result no extra theory applied but the convergence of the sigmoid and the tangential curve to the verified output scenario.

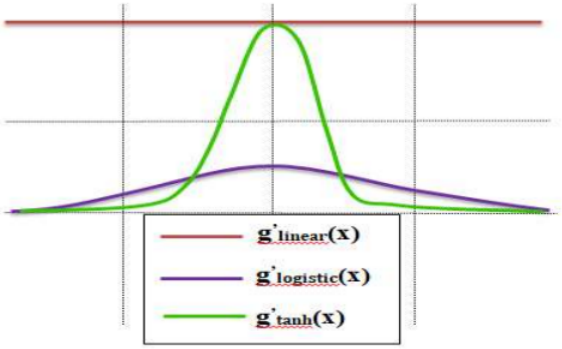


Fig 1.7 Training Part

Fig 1.7 describes the training part. The training part implemented through the code could

be further recognized as below in fig 1.8

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt

In [ ]: import keras
        from keras.datasets import mnist

In [ ]: #get the data and preprocess it
        (X_train, y_train), (X_test, y_test) = mnist.load_data()

In [ ]: X_train.shape , y_train.shape , X_test.shape , y_test.shape

        def plot_input_img(i):
            plt.imshow(X_train[i] , cmap='binary')
            plt.title(y_train[i])
            plt.show()

In [2]: for i in range(10):
            plot_input_img(i)
```

Fig 1.8

6.2 Testing Dataset:

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 the figure from the webcam 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. Results of the model shall be discussed in no time because the model has been trained more than 23 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 self created data and MNIST dataset 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.

6) Model check with checksum value removal and biased value removal after setting the weight value by the neural network on its set and set the dataset into the unbiased.

7) Access the webcam and capture the image using python javascript and google colab display.

8) Check the parameter of probability of the figure after normalisation and noise removal and scaling.

9) Got highest predicted value?match the figure with the detected from the neural.

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 contrast 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 Unit Testing.It is often done by a programmer to test the part of the program he or she has implemented. Unit Testing is successful means all the modules have 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.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)Verifying the input image

ii)Verifying the work flow

iii)Correct recognition and calculate the error . 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 checking 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.

System 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.

Acceptance Testing 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.

Using Kera's backend and TensorFlow as the software, a CNN model is able to give accuracy of about 99.5 percent in the training and 97.6 percent in testing.

```
In [3]: #preprocess the image

        #Normalizing the image to [0,1] range

        X_train = X_train.astype(np.float32)/255
        X_test = X_test.astype(np.float32)/255

In [4]: #Reshape / expand the dimension of images to (28,28,1)
        X_train = np.expand_dims(X_train, -1)
        X_test = np.expand_dims(X_test, -1)

In [5]: X_train.shape

Out[5]: (60000, 28, 28, 1, 1)

In [6]: #convert classes to one hot vectors
        from keras.utils import np_utils
        y_train =keras.utils.np_utils.to_categorical(y_train)
        y_test =keras.utils.np_utils.to_categorical(y_test)
```

Fig 1.9

Fig 1.9 shows the way we have implemented the testing of the MNIST dataset.

6.3 Flowchart: Below fig 1.10 illustrates the process of implementation.

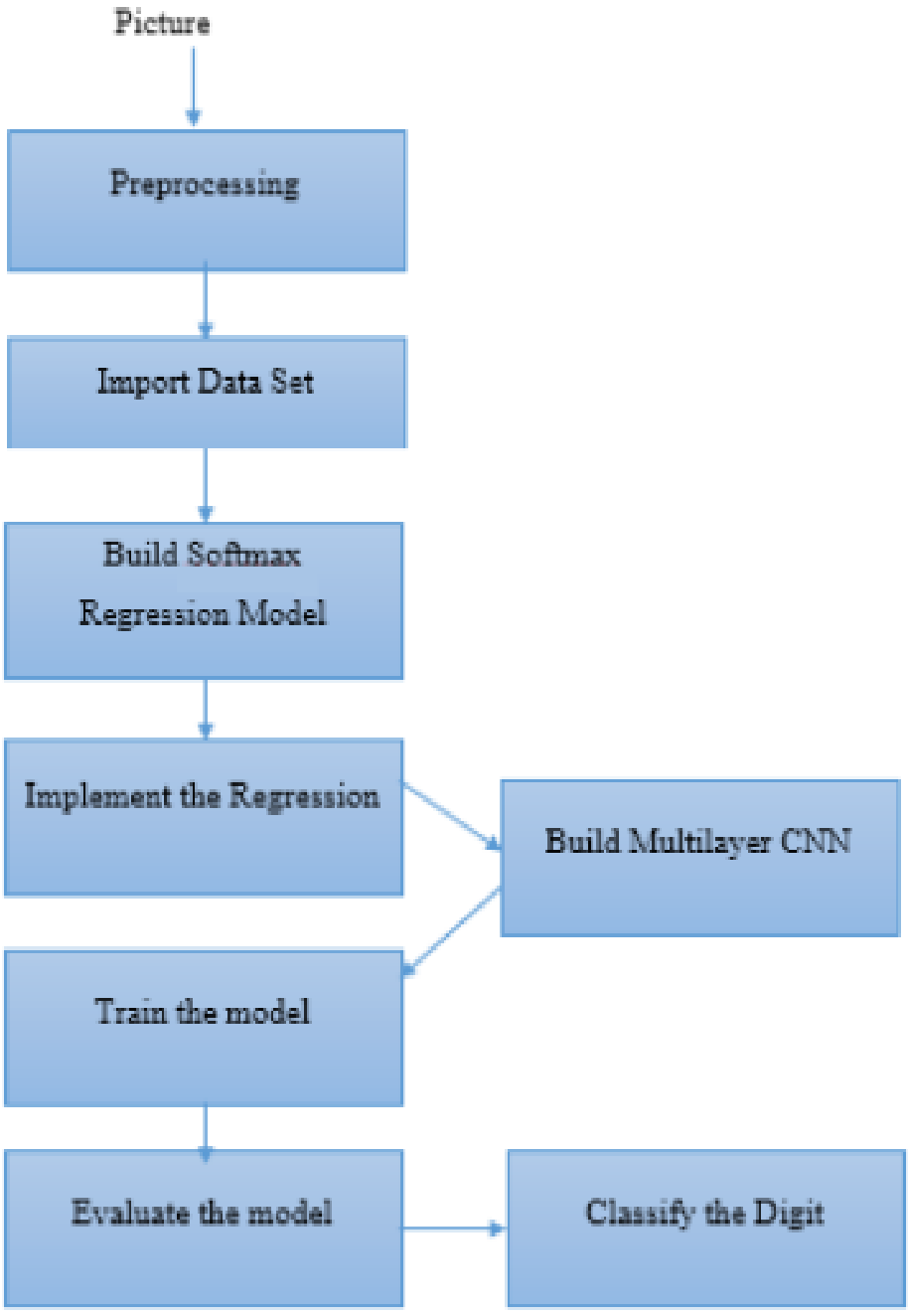


Fig 1.9

Chapter 07

Analysis

It is mandatory to carry out the analysis of the prediction accuracy and also the experiment results. The detailed examination of the machine learning algorithms is carried out here. It is important to check if the algorithm and the model chosen gives the correct accuracy.

7.1 Prediction Accuracy:

The accuracy of the prediction is checked and it is observed as 0.928 which is a fair accuracy or the result.

```
In [7]: y_train

Out[7]: array([[[1., 0.],
                [1., 0.],
                [1., 0.],
                ...,
                [1., 0.],
                [1., 0.],
                [1., 0.]],

               [[0., 1.],
                [1., 0.],
                [1., 0.],
                ...,
                [1., 0.],
                [1., 0.],
                [1., 0.]],

               [[1., 0.],
                [1., 0.],
                [1., 0.],
                ...,
                [1., 0.],
                [1., 0.],
                [1., 0.]],
```

In [8]:
```
from keras.models import Sequential
from keras.layers import Dense , Conv2D , MaxPool2D , Flatten , Dropout
```

In [9]:
```
model = Sequential()
```

In [10]:
```
model.add(Conv2D(32, (3,3) , input_shape=(28,28,1) , activation='relu'))
model.add(MaxPool2D((2,2)))
```

In [11]:
```
model.add(Conv2D(64, (3,3) , activation='relu'))
model.add(MaxPool2D((2,2)))
```

In [12]:
```
model.add(Flatten())
model.add(Dropout(0.25))
model.add(Dense(10 , activation="softmax"))
```

In [13]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_2 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 5, 5, 64)         0
 2D)
```

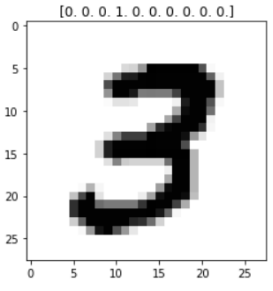
In [14]:
```
import tensorflow as tf
```

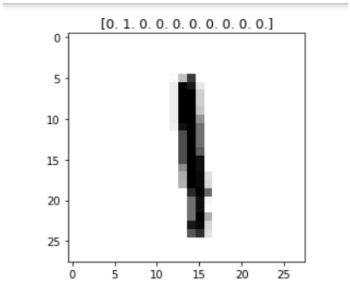
In [15]:
```
model.compile(optimizer = 'adam' , loss = keras.losses.categorical_crossentropy , metrics = ['accuracy'])
```

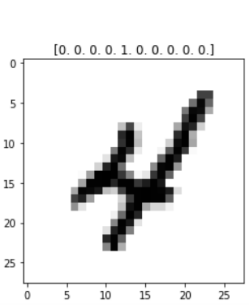
In [16]:
```
#callbacks

from keras.callbacks import EarlyStopping, ModelCheckpoint
```

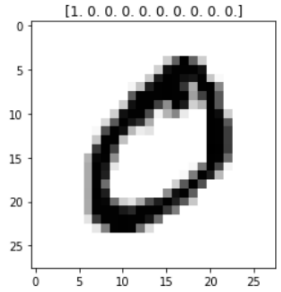
In [17]:
```
#EarlyStopping

es=EarlyStopping(monitor='val_acc' , min_delta=0.01 , patience=4, verbose=1)
```

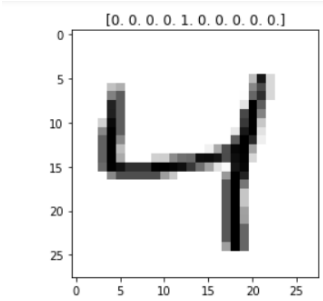
In [41]:
```
#ModelCheckpoint

mc=ModelCheckpoint("./bestmodel.hs", monitor="val_acc", verbose=1 , save_best_only=True)

cb=[es,mc]

hist = model.fit(X_train , y_train , epochs=5, validation_split = 0.3)
```

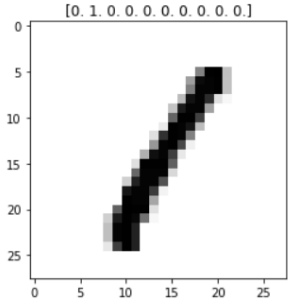


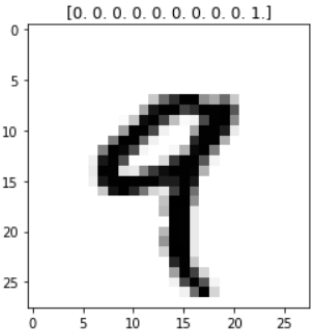


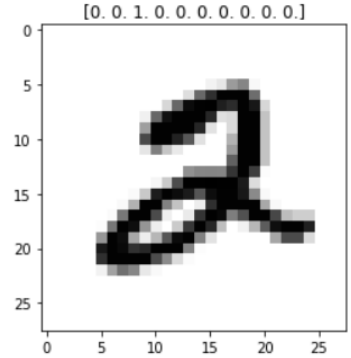

[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]

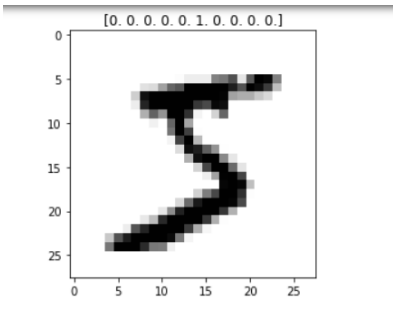

[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

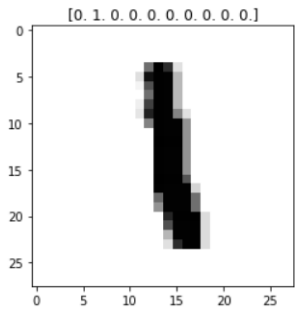

[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]


[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]


[0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]


[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]





The above images from the MNIST dataset were input to train, test and get the accuracy. Those were the variety of handwritings taken from the MNIST dataset.

7.2 Experimental results:

```
In [21]: calculate_accuracy (X_test, y_test, X_train, y_train, k=5)
Out[21]: 0.928
```

The result thus received in this project was 0.928% accuracy which is in the appropriate range of the accuracy.

Chapter 08

Conclusion and Future Scope

In this project the Handwritten Digit Recognition using Machine Learning has been implemented. The most widely used algorithm CNN has been trained and tested on the MNIST dataset. Utilizing these machine learning techniques, a high amount of accuracy is obtained. Using Kera's backend and TensorFlow as the software, a CNN model is able to give accuracy of about 99.5 percent in the training and 97.6 percent in testing. The goal of this project was to explore the field of Machine Learning and try to come up with some techniques that could be used without going into deep computations, and even if the final result is not so reliable, it still provides an accuracy way better than random. In future this system could be developed further by making this system a real-time handwritten digit recognition system and could be made more advanced using artificial intelligence.

References

[1] ResearchGate.[Online]. Available:

https://www.researchgate.net/publication/335908757_Recognition_of_Handwritten_Digit_using_Convolutional_Neural_Network_in_Python_with_Tensorflow_and_Comparison_of_Performance_for_Various_Hidden_Layers

[2] People idsia.[Online]. Available:

https://people.idsia.ch/~masci/papers/2011_ijcai.pdf

[3] Ijisrt.[Online]. Available:

https://ijisrt.com/wp-content/uploads/2019/06/IJISRT19JU358.pdf

[4] ResearchGate.Online. Available:

https://www.researchgate.net/publication/333191386_Recognition_of_Handwritten_Digit_using_Convolutional_Neural_Network_CNN

[5] Ijircst. [Online]. Available:

https://www.ijircst.org/DOC/16-handwritten-digit-recognition-using-various-machine-learning-algorithms-and-models.pdf

[6] Ijrte. [Online]. Available:

https://www.ijrte.org/wp-content/uploads/papers/v9i5/E5287019521.pdf

[7] ResearchGate.Online. Available:

https://www.researchgate.net/publication/343054680_Handwritten_Digit_Recognition_using_Machine_and_Deep_Learning_Algorithms

[8] Ijcsit .[Online] . Available:

http://ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502303.pdf

[9]ResearchGate.Online. Available:

https://www.researchgate.net/publication/326408524_Handwritten_Digit_Recognition_Using_Machine_Learning_Algorithms

[10] Arxiv. [Online]. Available:

https://arxiv.org/abs/2106.12614

Appendix

K-Nearest Neighbor (KNN) Algorithm

KNN is one of the simplest forms of machine learning algorithms mostly used for classification. It classifies the data point on how its neighbor is classified. KNN classifies the new data points based on the similarity measure of the earlier stored data points.

Natural language processing(NLP)

Natural Language Processing is a very broad term that encompasses all techniques related to the processing of human communications, both oral and written language. Traditionally, NLP analysis was based on lexicographic rules. With the rise of Machine Learning, they can be combined with new AI tools like Deep Learning. Among them, we can highlight the LSTM networks.

Neural Network(NN)

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.

Index