A PRELIMENERY REPORT ON

"Hand Written Digit Recognition using Machine Learning"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

BACHELOR OF ENGINEERING (INFORMATION TECHNOLOGY)
SUBMITTED BY

Bidgar Vivek Vinayak	**(Exam Seat No:)**
Patil Kunal Dnyandev	**(Exam Seat No:)**
Lodhavat Ameya Vijaysingh	**(Exam Seat No:)**
Jadhav Rinku Gopal	**(Exam Seat No:)**

THE MET LEAGUE OF COLLEGES
MET Bhujbal Knowledge City
AS SHARP AS YOU CAN GET

Department of Information Technology

MET's Bhujbal Knowledge City, Institute of Engineering
Adgaon, Nashik-422003

SAVITRIBAI PHULE PUNE UNIVERSITY
2021-22

THE MET LEAGUE OF COLLEGES
MET Bhujbal Knowledge City
AS SHARP AS YOU CAN GET

Certificate

This is to Certify that the project report entitles

"Hand Written Digit Recognition using Machine Learning"

Submitted by

Bidgar Vivek Vinayak	**(Exam Seat**
Patil Kunal Dnyandev	**(Exam Seat**
Lodhavat Ameya Vijaysingh	**(Exam Seat No:)**
Jadhav Rinku Gopal	**(Exam Seat No:)**

are bonafide students of this institute and the work has been carried out by them under the guidance of Prof. Kanchan Dhomse and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Information Technology).

Project Guide	H.O.D	Principal
(Prof. Kanchan Dhomse)	(Dr. S. V. Gumaste)	(Dr. V. P. Wani)

Place: Nashik

Date: / /

Acknowledgements

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individual and organizations. We would like to extend our sincere thanks to all of them. It gives us proud privilege to complete the project on **“Hand Written Digit Recognition using Machine Learning”**. We are highly indebted to our internal guide **Prof. Kanchan Dhomse** for her guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

The project is not sponsored . We would like to express our special gratitude and thanks to **Technical Staff** for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciation also goes to our beloved parents for their blessings, all the staff members, friends and colleagues for providing help and support in our work.

Abstract

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image.

Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc, and classify. them into 10 predefined classes (0-9). This has been a Content of bottomless- exploration in the field of deep literacy. Number recognition has numerous operations like number plate recognition, postal correspondence sorting, bank check processing, etc. In Handwritten number recognition, we face numerous challenges . because of different styles of jotting of different peoples as it . is not an Optic character recognition. This exploration provides a comprehensive comparison between different machine literacy and deep literacy algorithms for the purpose of handwritten number recognition. Testing- training time corroborated by plots and maps that have been constructed using matplotlib for visualization. We are not using any pre-collected images of digit. But we will predict images live drawn in paint, so for this we have to collect images from paint by our own.

Contents

List of Figures

Chapter 1

Introduction

Recognition is identifying or distinguishing a thing or an individual from the past experiences or learning. Similarly,Digit Recognition is nothing but recognizing or identifying the digits in any document. Digit recognition framework is simply the working of a machine to prepare itself or interpret the digits.

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc.

Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits. Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions.

Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources. Handwriting recognition of characters has been around since the 1980s.

The task of handwritten digit recognition, using a classifier, has extraordinary significance and use such as – online digit recognition on PC tablets, recognize zip codes on mail, processing bank check amounts, numeric sections in structures filled up by hand (for example - tax forms) and so on.

There are diverse challenges faced while attempting to solve this problem. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins.

The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0-9).

1.1 Aims and Objective

- Our aim to create GUI (graphical user interface) model for handwritten digit recognition.
- To recognize handwritten digits correctly.
- To improve the accuracy of detection.
- To develop a method which is independent of digit size and writer style/ink independent.

1.2 Motivation

The general problem we predicted we would face in this digit classification problem was the similarity between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 etc. Also people write the same digit in many different ways-the digit ‘ 1 ’ is written as ‘ 1 ’, ‘1’ ’ , or ‘ 1 ’. Similarly 7 may be written as ‘ 7 ’, ‘ 7 ’, or ‘ 7 ’.

Our goal was to implement a pattern classification method to recognize the handwritten digits provided in the MNIST data set of images of hand written digits (0-9).

Chapter 2

Literature Survey

In a paper published by Saeed AL-Mansoori, Multilayer Perceptron (MLP) Neural Network was implemented to recognize and predict handwritten digits from 0 to 9.The proposed neural system was trained and tested on a dataset achieved from MNIST. A. Existing System These days, an ever-increasing number of individuals use pictures to transmit data. It is additionally main stream to separate critical data from pictures. Image Recognition is an imperative research area for its generally used applications. In general, the field of pattern recognition, one of the difficult undertakings is the precise computerized recognition of human handwriting. Without a doubt, this is a very difficult issue because there is an extensive diversity in handwriting from an individual to another individual. In spite of the fact that, this difference does not make any issues to people, yet, anyway it is increasingly hard to instruct computers to interpret general handwriting. For the image recognition issue, for example, handwritten classification, it is essential to make out how information is depicted onto images. Handwritten Recognition from the MNIST dataset is well known among scientists as by utilizing different classifiers for various parameters, the error rate has been decreased, for example, from linear classifier (1-layer NN) with 12% to 0.23% by a board of 35 convolution neural systems. The scope of this is to implement a Handwritten Digit Recognition framework and think about the diverse classifiers and different techniques by concentrating on how to accomplish close to human performance. For an undertaking of composing diverse digits (0-9) for various people the general issue confronted would be of digit order issue and the closeness between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so forth. Additionally, individuals compose a similar digit from various perspectives, the uniqueness and assortment in the handwriting of various people likewise impact the development and presence of the digits .

Chapter 3

Architecture

The reason behind this document is to look into the design possibilities of the proposed system, such as architecture design, block diagram, sequence diagram, data flow diagram and user interface design of the system in order to define the steps such as pre-processing, feature extraction, segmentation, classification and recognition of digits.

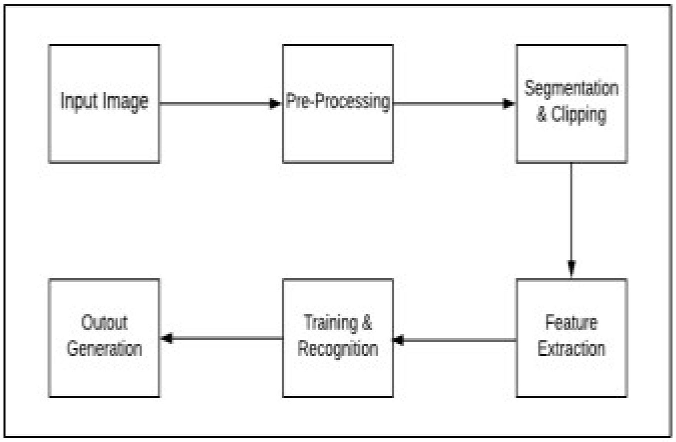


Fig 1. Architecture of the proposed system

The above Figure 1 illustrates the architecture diagram of the proposed system. The proposed model contains the four stages in order to classify and detect the digits: A. Pre-processing B. Segmentation C. Feature Extraction D. Classification and Recognition.

A. Pre-Processing:

The role of the pre-processing step is it performs various tasks on the input image. It basically upgrades the image by making it reasonable for segmentation. The fundamental motivation behind pre-processing is to take off a fascinating example from the background. For the most part, noise filtering, smoothing and standardization are to be done in this stage. The pre-processing additionally characterizes a smaller portrayal of the example. Binarization changes over a gray scale image into a binary image. The initial approach to the training set images that are to be processed in order to reduce the data, by thresholding them into a binary image. The Figure 2 shows a sample of images taken from the MNIST database.

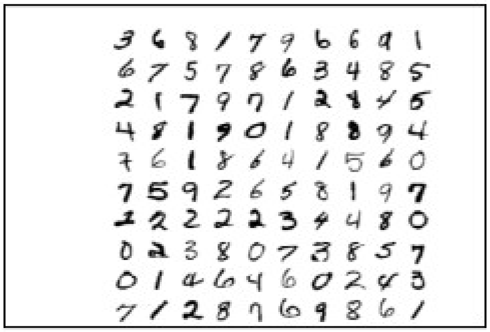


Fig 2: Sample images taken from MNIST dataset

B. Segmentation:

Once the pre-processing of the input images is completed, sub-images of individual digits are formed from the sequence of images. Pre-processed digit images are segmented into a sub-image of individual digits, which are assigned a number to each digit. Each individual digit is resized into pixels. In this step an edge detection technique is being used for segmentation of dataset images.

C. Feature Extraction:

After the completion of pre-processing stage and segmentation stage, the pre-processed images are represented in the form of a matrix which contains pixels of the images that are of very large size. In this way it will be valuable to represent the digits in the images which contain the necessary information. This activity is called feature extraction. In the feature extraction stage redundancy from the data is removed.

D. Classification and Recognition:

In the classification and recognition step the extracted feature vectors are taken as an individual input to each of the following classifiers. In order to showcase the working system model extracted features are combined and defined using following three classifiers:

- K-Nearest Neighbor
- Random Forest Classifier
- Support Vector Machine

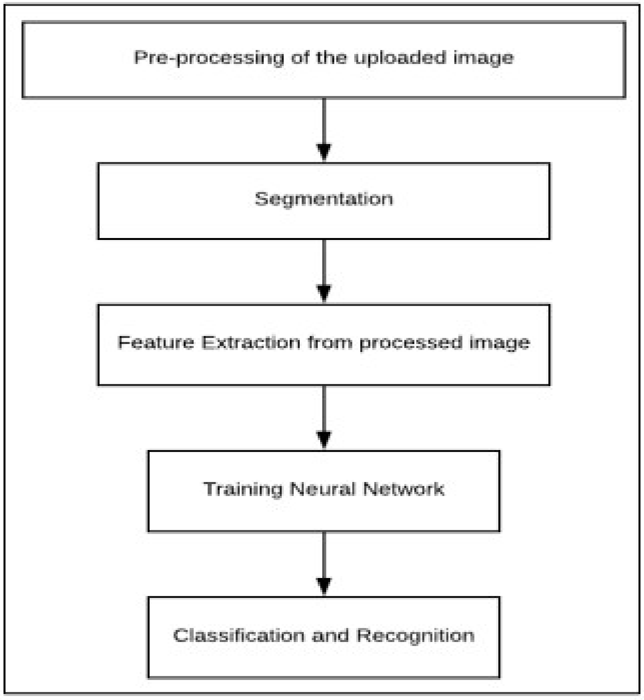

Pre-processing of the uploaded image
Segmentation
Feature Extraction from processed image
Training Neural Network
Classification and Recognition

Fig 3: Block digram of proposed model

1) K-Nearest Neighbor:

KNN is an instance based learning algorithm. There are two main benefits of using KNN algorithm, that is, it is robust to noisy training data and it is very efficient if the data is very large in size. To perform admirably, this algorithm requires a set of training datasets which includes perfectly labeled data points. KNN is also a non-parametric classifier. The algorithm considers new data point as its input and performs classification by calculating distance between new and labeled data points using the Euclidean or Hamming distance formulas. The Euclidean distance is calculated using the following formula:

2) Random Forest Classifier:

RFC is a supervised learning method. It infers that there is an immediate connection between the total number of trees and the result it gets: the bigger the number of trees, the more precise the outcome will be. This classifier can be used for regression as well as classification. For RFC algorithm if there are sufficient trees then the classifier will not over fit the model, instead it avoids the over fitting issues. This classifier can deal with the missing quantities. Once the training is done, predictions are taken from each individual tree and the average is calculated using the following formula:

3) Support Vector Machine:

SVM is also a supervised learning method. It is also used for both classification and regression tasks. In this type of algorithm, there are data items which are considered as points in an n-dimensional space. This classifier finds the hyper plane by performing classifications between the two classes. One of the main advantages of this algorithm is that it provides a regularization parameter which avoids the over fitting problems. The block diagram shown below in the Figure 3 describes all these above steps.

Chapter 4

Methodology

Each research work needs some estimation, to measure the accuracy and performance of handwritten digits, MNIST dataset is being used for such reasons.

MNIST is the most broadly utilized standard for handwritten digit recognition. MNIST is a huge and a standard database of handwritten digits.

MNIST dataset has been commonly used as a standard for testing classification algorithms in handwritten digit recognition frameworks. The initial step to be carried out is to place the dataset, which can be effectively done through the Keras programming interface.

The images in the MNIST dataset are available in type of a cluster comprising of 28x28 values constituting to an image along with their labels. This is equivalent if there could be an occurrence of the testing images.

The pixels are given as a variety of 784-d pixels and the range extends from 0 to 255 for example 0 implies Black and 255 implies White.

A. Algorithm Used

The following Figure 4 describes the Data flow diagram of the proposed system model. There are two ways to provide input to the system. The user can either upload the image of the digit he wants to detect or the data from the MNIST dataset. The input images are pre-processed. Using the different classifiers the recognized digits' accuracy is compared and the result is obtained. The results obtained are displayed along with the accuracy

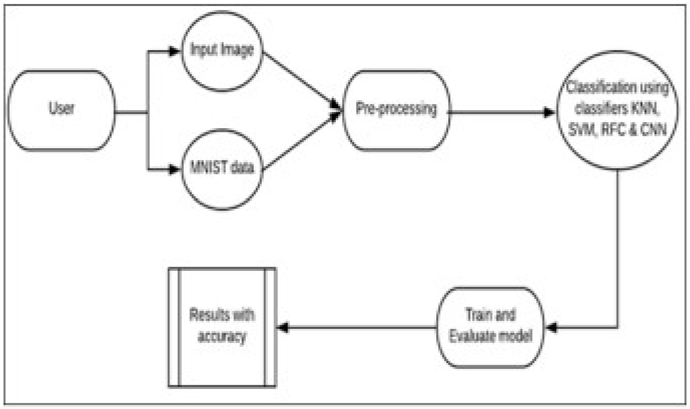


Fig 4: Dataflow Diagram of the system model

A 1: Super Vector Classifier algorithm

We used Super Vector Classifier algorithm for classification.

Support Vector Classifier (SVC) method applies a kernel function to perform classification and it performs well with a large number of samples.

SVC is a nonparametric clustering algorithm that does not make any assumption on the number or shape of the clusters in the data. In our experience it works best for low-dimensional data, so if your data is high-dimensional, a preprocessing step, e.g. using principal component analysis, is usually required.

Chapter 5

Implementation

A. Software Platform

Tensorflow

TensorFlow is an amazing information stream in machine learning library made by the Brain Team of Google and made open source in 2015.It is intended to ease the use and broadly relevant to both numeric and neural system issues just as different spaces. Fundamentally, TensorFlow is a low level tool for doing entangled math and it targets specialists who recognize what they're doing to construct exploratory learning structures, to play around with them and to transform them into running programs. For the most, it can be considered as a programming framework in which one can entitle to calculations as graphs. Nodes in the graph speak the math activities, and the edges contain the multi-dimensional information clusters (tensors) related between them.

Python

Python is broadly utilized universally and is a high-level programming language. It was primarily introduced for prominence on code, and its language structure enables software engineers to express ideas in fewer lines of code. Python is a programming language that gives you a chance to work rapidly and coordinate frameworks more effectively

Sci kit-learn

Sci kit-learn is probably the most useful library for machine learning in Python.

The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

open cv

The use of open cv is to convert images into grayscale images.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

B. Implementation steps

We divided implementation in four parts.

- Screen Capture
- Generate dataset and Load it
- Fit the model using SVC and calculate accuracy
- Prediction of image drawn in paint & converting our project in GUI.

B1. Screen Capture

We have to collect images of digits (from 0 to 9). To collect images, pyscreenshot package can be used.

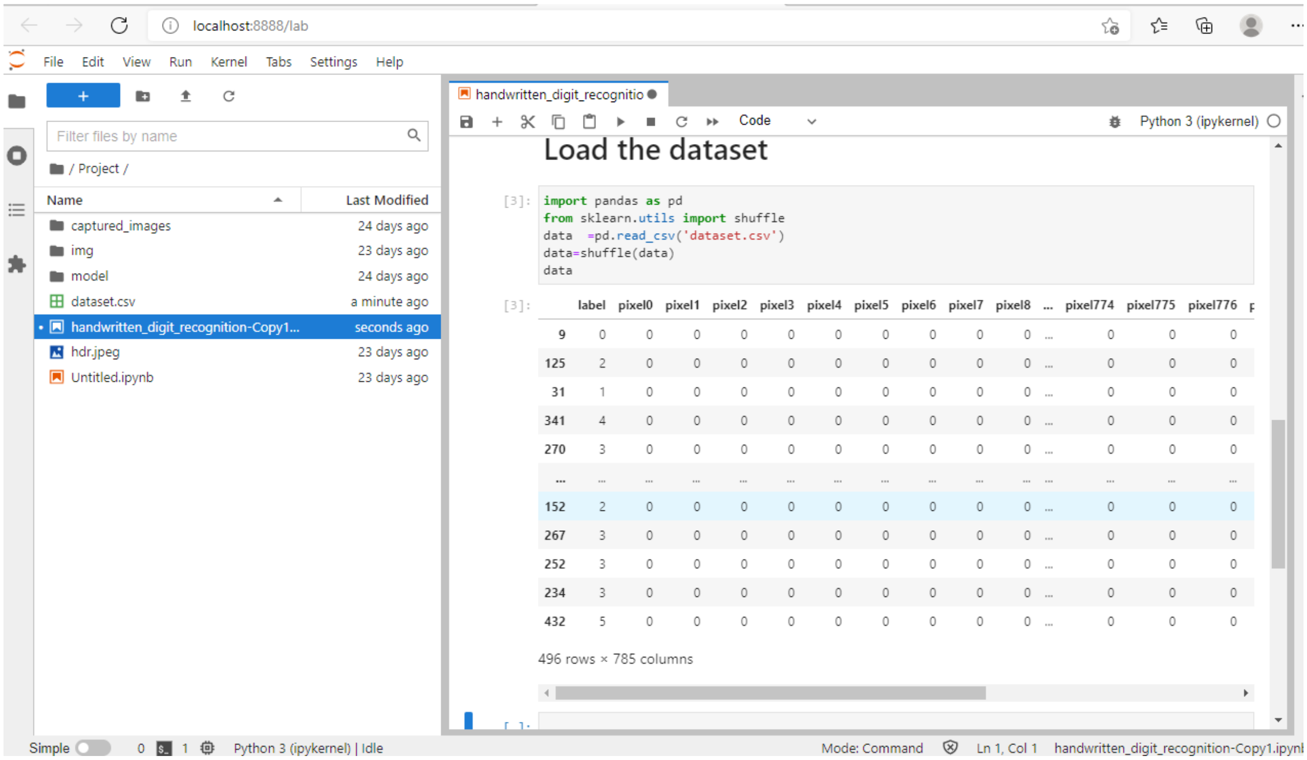
This package can be downloaded using pip which is a package management tool that are written in python and used to install python packages.

B2. Generate dataset and Load it

We generated our dataset using images that we have collected in 1st step. To generate dataset, we assign 1 to the drawn region and 0 to the background. That means, in our dataset, we will be having only two values i.e., 0 and 1.

Pixel value ranges from 0 to 255. Typically, 0 represent black and 255 represent white. We will assigning 0 to pixel value from 0 to 100 and 1 to pixel value from 100 to 255. Now our pixel value is not from 0 to 255, it is only 0 and 1. In this way, We will generating dataset (csv file).

Final step is to open the dataset, shuffle it i.e., change the position of each row of data and display it.



localhost:8888/lab
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ Project /
handwritten_digit_recognitio
Code
Python 3 (ipykernel)
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
handwritten_digit_recognition-Copy1.ipynb

Name	Last Modified
captured_images	24 days ago
img	23 days ago
model	24 days ago
dataset.csv	a minute ago
handwritten_digit_recognition-Copy1...	seconds ago
hdr.jpeg	23 days ago
Untitled.ipynb	23 days ago

Load the dataset

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

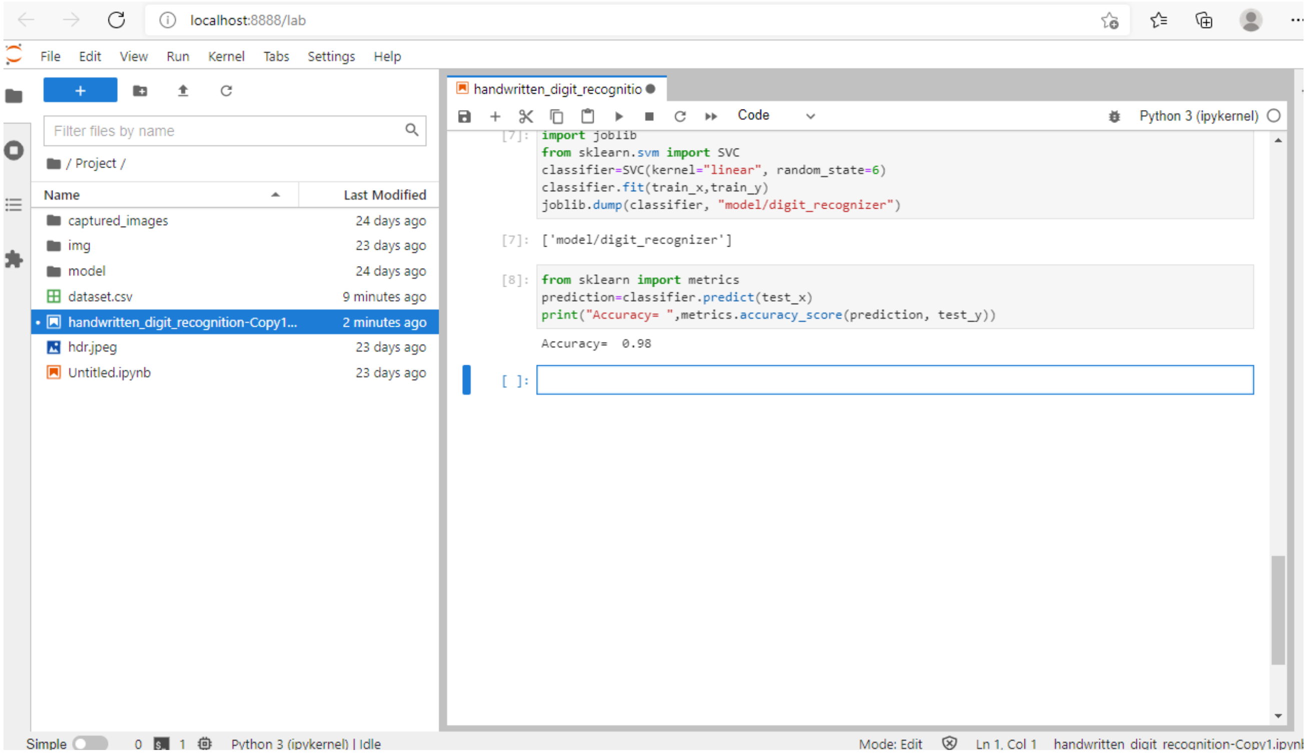
```
[3]:      label  pixel0  pixel1  pixel2  pixel3  pixel4  pixel5  pixel6  pixel7  pixel8  ...  pixel774  pixel775  pixel776  p
      9      0       0       0       0       0       0       0       0       0       0  ...         0         0         0
    125      2       0       0       0       0       0       0       0       0       0  ...         0         0         0
     31      1       0       0       0       0       0       0       0       0       0  ...         0         0         0
    341      4       0       0       0       0       0       0       0       0       0  ...         0         0         0
    270      3       0       0       0       0       0       0       0       0       0  ...         0         0         0
    ...    ...     ...     ...     ...     ...     ...     ...     ...     ...     ...  ...       ...       ...       ...
    152      2       0       0       0       0       0       0       0       0       0  ...         0         0         0
    267      3       0       0       0       0       0       0       0       0       0  ...         0         0         0
    252      3       0       0       0       0       0       0       0       0       0  ...         0         0         0
    234      3       0       0       0       0       0       0       0       0       0  ...         0         0         0
    432      5       0       0       0       0       0       0       0       0       0  ...         0         0         0

496 rows × 785 columns
```

Fig 5 : Loaded Dataset

B3. Fit the model using SVC and calculate accuracy

We have to train our model and calculate the accuracy. For this, we separate dependent (Y) and independent variable (X). Our independent variable will be the pixel value (i.e. 0 and 1). Each digit is represented by huge number of 0 and 1. And our dependent variable will be our digit (i.e. from 0 to 9).

We used scikit-learn to train our model.



Name	Last Modified
captured_images	24 days ago
img	23 days ago
model	24 days ago
dataset.csv	9 minutes ago
handwritten_digit_recognition-Copy1...	2 minutes ago
hdr.jpeg	23 days ago
Untitled.ipynb	23 days ago

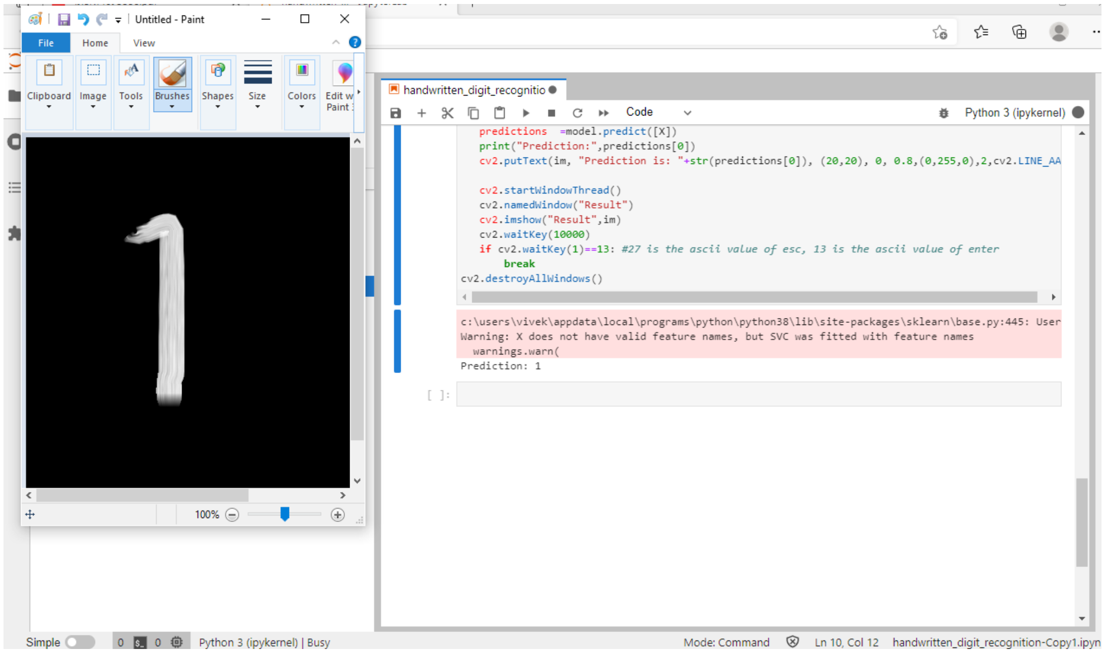
```
[7]: import joblib
     from sklearn.svm import SVC
     classifier=SVC(kernel="linear", random_state=6)
     classifier.fit(train_x,train_y)
     joblib.dump(classifier, "model/digit_recognizer")

[7]: ['model/digit_recognizer']

[8]: from sklearn import metrics
     prediction=classifier.predict(test_x)
     print("Accuracy= ",metrics.accuracy_score(prediction, test_y))

     Accuracy=  0.98

[ ]:
```

Fig 6: Accuracy

B4. Prediction of image drawn in paint & converting our project in GUI.

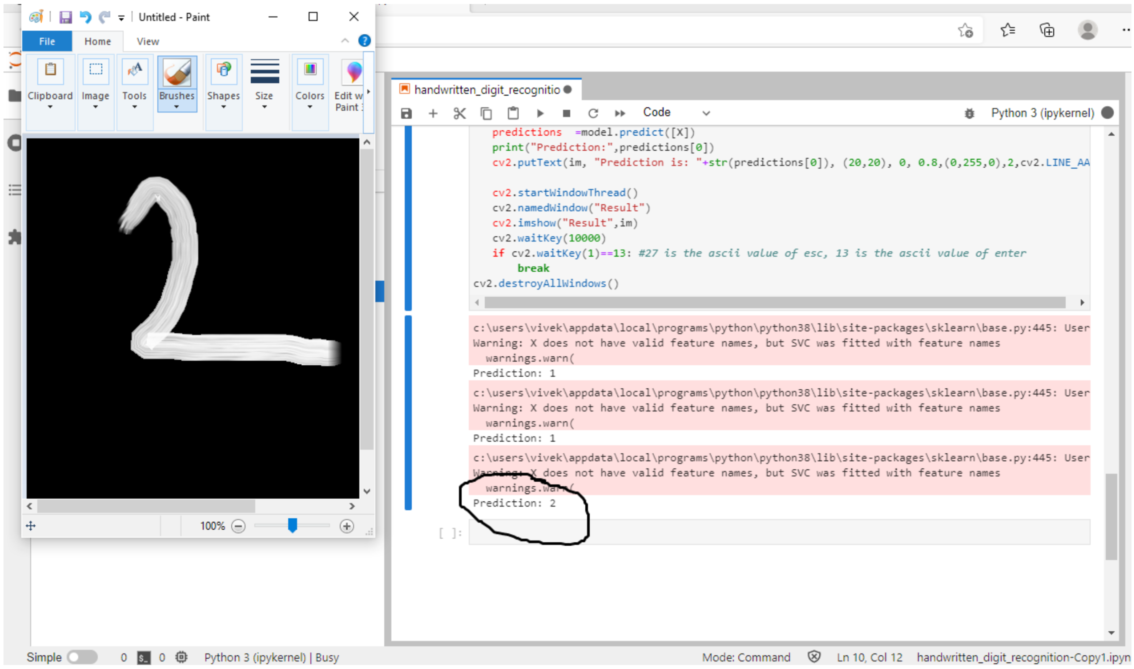
We have to predict digit drawn in paint. Now, we don't provide 20% images (testing images) to our model but we provide new images. We draw digit in paint and at the same time, pass that digit to model, our model has to predict that digit.

Result



```
    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()
```

```
c:\users\vivek\appdata\local\programs\python\python38\lib\site-packages\sklearn\base.py:445: User
Warning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
Prediction: 1
```



```
    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()
```

```
c:\users\vivek\appdata\local\programs\python\python38\lib\site-packages\sklearn\base.py:445: User
Warning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
Prediction: 1
c:\users\vivek\appdata\local\programs\python\python38\lib\site-packages\sklearn\base.py:445: User
Warning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
Prediction: 1
c:\users\vivek\appdata\local\programs\python\python38\lib\site-packages\sklearn\base.py:445: User
Warning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
Prediction: 2
```

Conclusion

In this paper, the Handwritten Digit Recognition using Deep learning method SVM has been implemented.

Utilising these deep learning techniques, a high amount of accuracy can be obtained. Compared to other research methods, this method focuses on better by improving the accuracy of classification models by more than 99%. Using Keras as backend and Tensorflow as the software.