

# Real Time Handwritten Digit Recognition

Aditya Nikam<sup>1</sup>, Jatin Lanje, Bhavesh Chaturvedi<sup>2</sup>, Harshit Kumar<sup>3</sup>, Prof. Monali Chinchamatpure<sup>4</sup>  
<sup>1,2,3</sup>B. Tech Students Department of Computer Science and Engineering, MIT ADT University, Pune  
<sup>4</sup>Assistant Professor, Department of CSE, MITSOE, ADT University, Pune

**Abstract - A Recognition of handwritten characters is one of the most important issues in recognition application of patterns. The use of digital recognition includes postal planning, processing of bank checks, etc. The main problem is the ability to create an efficient algorithm that can detect handwritten numbers sent by users in the form of scanners, tabs, and other digital devices. This paper introduces a method in which offline handwritten digits are recognized based on different machine learning techniques. The main intention of this paper is to ensure constructive and authentic methods of recognizing handwritten digits.**

**Index Terms - Machine Learning, Pattern Recognition, Handwritten Recognition, Digit Recognition.**

## 1.INTRODUCTION

In artificial intelligence, image analysis is an appealing research sector and it is crucial for a variety of till date open research problems. Handwritten Digit Recognition is a very widely searched topic within the sectors or fields which are concerned about the learning model to differentiate pre-segmented handwritten digits. Also, it is the most important problem in ML, pattern analysis, data mining and AI. The uses of these machine learning methods are competing with the accuracy and decision taking performance of humans.

Not every feature of those models was inspected or were given digit on but a big amount of people is working on these data mining and machine learning algorithms in order to make those recognition models more accurate.

The HDR system has set its standards in the 21st Century and has been widely used in daily life scenarios. Among all the challenges the particularly important challenge is to verify and sort the patterns of the handwritten characters or digits.

All these challenges are mainly the cause of various and different types of patterns and strokes made by a person which changes from person to person. In

today's world the Handwritten pattern recognition has made a focal point towards the field of pattern recognition systems and sorting data.

The main goal of all these character or digit recognition systems is to bring more efficiency and accuracy in determining and recognizing the patterns/numbers/characters.

This project gives the idea to recognize the handwritten digits by using tools from ML i.e., libraries, classifiers, to produce high recognition performance. Also, the use of tools from CV is explored to investigate/search the effect of selection of classifiers, features and image pre-processing on the entire error rate.

Basically, the dataset used is the SELF-CREATED dataset by capturing images from paint by using Pyscreenshot library. According to the image drawn / digit drawn it captures the images first and then analyze it and its dimension, pattern, pixels etc. and then finally the model which we create (by importing ML libraries, training the images, forming of dataset and forming ML-model) and then finally the DIGIT is detected.

This whole project process can be broken into several parts:

1. Collecting sample images for a dataset captured from a paint tool.
2. Pre-processing
3. Feature-extraction
4. Classification
5. Training images
6. Evaluation and verification

## 2.PROBLEM DEFINITION

The main objective of this project is implementation of a classification algorithm to recognize handwritten digits from (0-9) and this is one of the powerful researches works. As we know there is rapid growth

of new documents and multimedia which has created new challenges in this field. Because of the progress in the field of technology and science, in order to reduce human effort everything is being digitalized. Hence there comes a need of HANDWRITTEN-DIGIT RECOGNITION in many real time APPLICATIONS for ex: Administration and finance, postal zip codes for postcard, bank check processing and also for form data entry. This is a powerful research going on because every individual in this world has their own style / way of writing. So, because of this AI model, it reduces the human effort and brings much efficiency in work and also time taken is so less as compared to human, as the model is trained in that way. So, in simple words, this project performs digit recognition i.e., analysis of accuracy of algorithms. The implementation and generation of data involves 2 stages,

- One is to capture image and getting done with dataset which is self-created.
- And another one is of forming ml model (training the dataset as it is already created and then using libraries like OpenCV, CSV, GLOB)

Then to preview one image, we used Matplotlib. After that we import train test split function in order to train our model. Then we fit the model using SVC and Joblib. After that we calculated the accuracy of our project.

And then finally, live prediction of our handwritten digits is done.

### 3. LITERATURE SURVEY

Velappa Ganapathy, and Kok Leong Liew [1], they deduced that the first multi-scale neural training with modifications in the input training vectors method is adopted to acquire its advantage in training the greater resolution character images and then finical threshold using minimum distance technique is deduced to improve the level of accuracy in recognizing the character. A GUI was designed in such a way that the characters can be spotted on the blank paper in which the characters are written. The results show that these methods with average level of training stage can generate accuracy of at least 85% and more for handwritten numbers and upper-case letters in English.

K. Gaurav, Bhatia P. K. [2], this project deals with the various character recognition with different kind of

images ranging from a simple handwritten based form and docs containing colored and complex background involves various preprocessing techniques. In this project, different preprocessing techniques like image upliftment techniques of contrast stretching, skew detecting and correction, binarization, normalization and segmentation, noise removal techniques, morphological processing technique are discussed. It was the deduced that we can't get a complete image using a single above technique for preprocessing. However, it is not accurate to say that we will get full accuracy in the preprocessing system even after applying all the said techniques. In this paper for recognizing unconstrained offline handwritten texts, hybrid Hidden Markov Model (HMM) model was proposed. In this model, the structural part of the optical model has been modeled with Markov chains, and a Multi-layer Perceptron was used to calculate the emission probabilities.

R. Bajaj, L. Dey, S. Chaudhari [6], engaged 3 different types of attributes(feature), namely, the density feature, moment feature and descriptive component feature for devnagri numerals classification. They proposed that for increasing the recognition reliability multi classifier connectionist architecture was introduced and they obtained 89.6% accuracy for handwritten Devanagari numerals.

### 2.1 SURVEY CARRIED OUT FOR FINDING SOLUTIONS WITH DIFFERENT CLASSIFIERS/ALGORITHMS AND APPROACHES

Several strategies that are recorded for these handwritten digit recognition systems formed on various ML algorithms and the techniques. However, timely accuracy in finding solutions and classification is very necessary and it makes the work of upcoming generation people's task in particular Institute very easy and efficient. Data entry can include different types of data entering process related to digits plus student related data i.e., roll number, finance, postal zip codes, etc. So, all these activities can be performed very smoothly and effectively for predicting the digits and entering the above-mentioned data. So out of all these handwritten digit images which are grouped together in a data set in number of times that is from 0 to 9 to predict the resultant image and to enter a particular data. It is important to note that out of all the Machine Learning techniques, SVC and Artificial

Neural Network are used by most of the proposed algorithms to help make prediction faster and more accurate. There are several different techniques and algorithms used in these handwritten digits recognition system for machine learning technique so choosing an appropriate technique has been a challenge for researchers.

#### 4. PROPOSED METHODOLOGY

As we know the technology evolving day-by-day, but Handwritten digit recognition is one of the precisely important problems in recognizing the patterns.

The problem is the ability to make an algorithm or a model that can recognize and memorize the different and specific patterns of different humans as every human handwriting is different as strokes of the handwriting varies person to person.

So, we have used some datasets and libraries in the project in order to track the pattern and to memorize it for effective prediction of the numbers/digits.

Here we are using a set of data; that we created in order to train the data and for which we have used a module named pyscreenshot.

we have used various libraries for data processing such as OpenCV, Numpy, Pandas, Sklearn.

And also, a library for data visualization such as Matplotlib.

#### 5. SYSTEM ARCHITECTURE

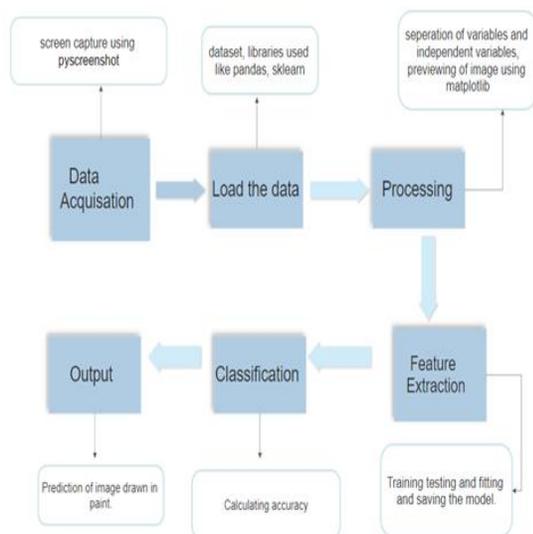
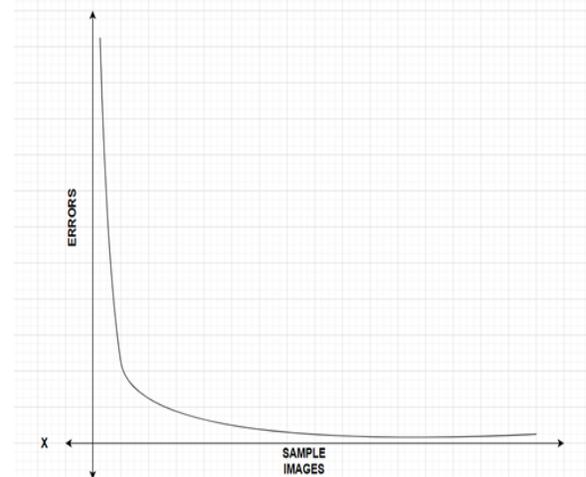


Figure 1. System Architecture

In this Handwritten Digit Recognition ML model, we have only used software. No hardware is used or required in forming this model.

We have used python libraries, modules and a default paint tool application for the formation of this project. Our model can be explained in simple words by showing a graph of our model.



Graph -1: The accuracy and error graph

The accuracy of the model depends on the number of sample images we store in it. In other words, if the model has stored a smaller number of sample images of the handwritten digits, the more probability of the model increases in producing an error or a false prediction of the digit. If the model has more than enough or sufficient data of stored sample images, the model will have less error generating capacity and will predict the handwritten digits with more accuracy.

#### 4.1 SOFTWARE COMPONENTS

The only components you need for this project is software and no hardware.

In software we have used various python modules, libraries and classifiers.

Here we are using a dataset; that we created in order to train the data and for which we have used a module named pyscreenshot. We have used various libraries for data processing such as OpenCV, Numpy, Pandas, and Sklearn. And also, a library for data visualization such as Matplotlib. A python classifier like SVC.

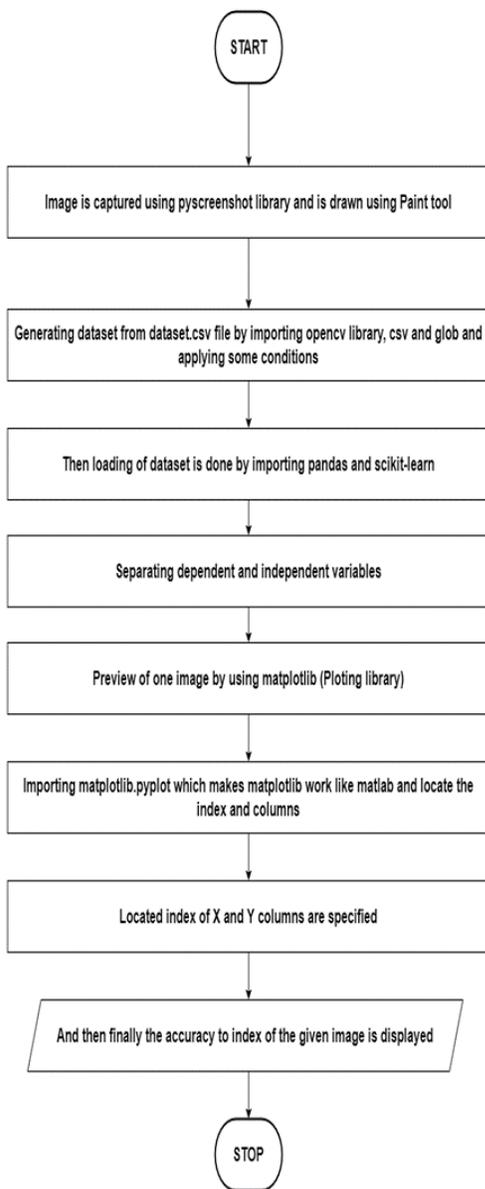
#### 4.2 ALGORITHM USED

Step1: Start

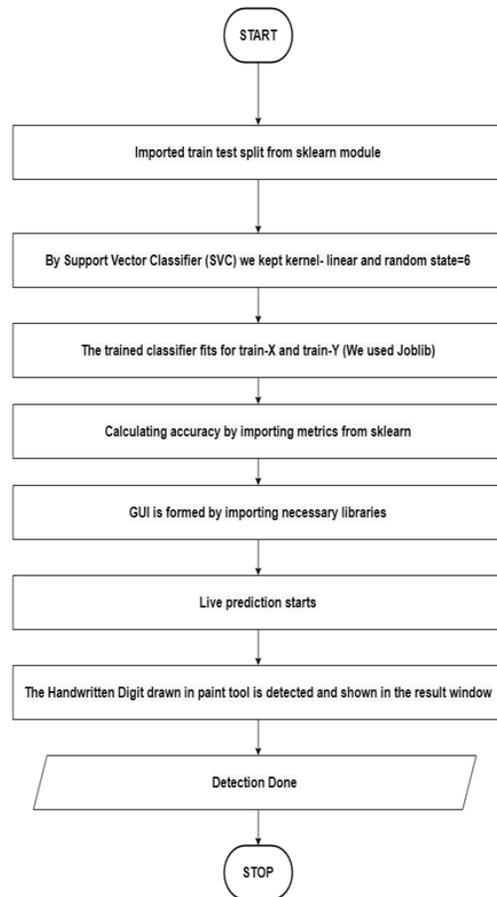
Step2: Collecting image using the paint tool

- Step3: Generating dataset using CSV, cv2, Glob
- Step4: Loading dataset using pandas
- Step5: Train-Test model using train\_test\_split function
- Step6: Fit the model using SVC and also save the model using joblib
- Step7: Calculate accuracy by using accuracy score function
- Step8: Live prediction
- Step9: Making GUI for Handwritten digit recognition
- Step10: Stop

4.3 FLOWCHART



Flowchart no. 1



Flowchart no. 2

4.4 SYSTEM DESCRIPTION

1. First, we capture the screen/take a screenshot of the digit that we want to predict. For this we use the pyscreenshot module. This module can be used to capture or copy the contents of the screen by converting it into a Pillow image memory by using different back-ends and then saves it. This is a replacement for image grab module which only works on MacOS and Windows.
2. By using the pyscreenshot we train the model to predict the digits by drawing same digit in different ways again and again for many times and then save them in a folder named of the respected digit that we trained it.
3. We train numbers from Zero (0) to Nine (9) and save all the zeros in folder named Zero (0). We do the same with all the numbers from 0 to 9 and save them in their respective folders. This is how we made a Dataset of the mentioned numbers from (0 to 9).

4. Then we import OpenCV and different libraries to our model in order to get the path of our created Dataset and to load the Dataset.

5. We import another Dataset which has all the Labels and Positions of pixels of the numbers/digits.

6. We separate the dependent and independent variables and then by using the Dataset, we preview one image using the Matplotlib as it is a library for data visualization.

7. Then by using Sklearn model selection we apply the train-test-split function for splitting the data into two subsets, one to train data and another to test data.

8. Then we make our model memorize the patterns of the digits in order to recognize the number/digit. (joblib)

And finally, we draw a number in the paint software of our system and then our model captures the screenshot of the number and then it further recognizes the number and pops a window to give us the predicted number which we draw.

#### 4.5 LIBRARIES USED IN THE ABOVE-MENTIONED DESCRIPTION

1. Pyscreenshot (For taking screenshot of the screen): In order to take a screenshot using Python, we will be using a module. For this task, Python provides a module called pyscreenshot. For this library, interactivity and performance are not important.

2. Generating Dataset: In order to generate dataset, we have used OpenCV, CSV, Glob.

OpenCV: To develop a real-time computer vision application, we will use OpenCV which is a cross-platform library. Image processing, video capture and analysis including features such as object detection and face detection is what this OpenCV actually focuses on.

CSV: CSV allow the data to be in a tabular form as it is a comma-separated value file. It looks like cut-and-dried spreadsheet but with a .csv add-on at end. These CSV files can be used with any spreadsheet program, such as Google Spreadsheet or Microsoft Excel. As only one sheet can be present in a file, cell, column, or row cannot be saved, so they are different from other spreadsheet files.

Glob: In order to retrieve files/pathnames matching a specified pattern, this glob module is used.

3. Loading the Dataset: we have used Pandas and Scikit-learn libraries.

Pandas: In data science, pandas are python library which is most extensively used. It provides data analysis tools, great-performance and easy to use structures. It is fast, flexible, powerful and manipulation tool which is built on top of the Python language and is easy to use open-source data analysis. Scikit-Learn: For machine learning and statistical modeling including regression, classification clustering and dimensionality reduction, the Sklearn library contains a lot of efficient tools. It is been widely used for machine learning models.

4. Matplotlib.Pyplot is used to Preview the image: Matplotlib.Pyplot is a cluster of functions that makes the Matplotlib work like MATLAB. A figure is modified by every pyplot function and makes some changes: e.g., plotting few lines in a plotting area, decorating the plot with labels, creating a figure, creating a plotting area in a figure, etc.

5. The Train-Test-Split: The train-test-split function is a very speedy and easy procedure to perform and as the results permits us to contrast the performance of a ML algorithms for the predictive modeling issues. Even though it is very easy to use and explain but there come times when the procedure cannot be or should not be used, such as when you have small situations and small set of data where additional configuration is required, such as when the dataset is not balanced it is used for classification.

6. Using SVC and Joblib: SVC: SVC stands for (Support Vector Classifier). To fit the data, you provide the support vector classifier is used, which returns a "best fit" hyperplane that divides, or categorizes, your data. After getting the hyperplane, to see what the "predicted" class is, you can feed some features to your classifier. Because of which it makes this specific algorithm suitable for our uses.

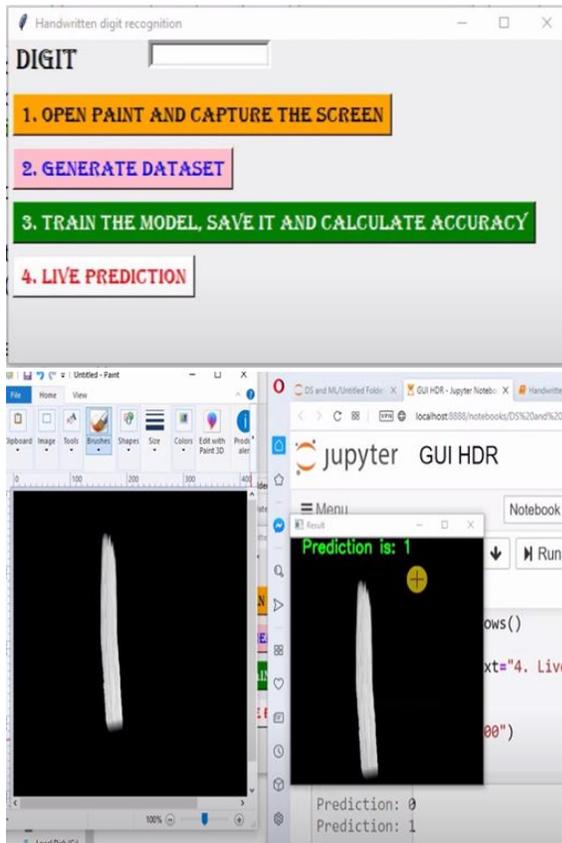
Joblib: Lightweight pipelining is provided by a set of tools which is called as joblib. Specifically said, joblib is used to memorize the pattern with easy simple and parallel computing. We have used it here to memorize the patterns of our handwritten digits.

7. Calculating accuracy of our model: To calculate the accuracy of our model we are using the sklearn implementation of accuracy score function.

8. Prediction of our handwritten digit: in order to predict the handwritten digit, we have used Joblib, OpenCV, Numpy, and Pyscreenshot.

Numpy: Numpy is used for working with arrays as a python library. It has few functions for the working of linear algebra domain, matrices and Fourier transform.

## 5. RESULTS



## 6. CONCLUSION

The primary focus of this work is to build a handwritten digit recognition model on document images; hence our model has recognized the digit which we provided. In this way human effort is reduced, and the desired model is obtained. Hence, we have successfully built a handwritten digit recognition GUI app in python based on machine learning.

## REFERENCES

- [1] Virag Jagtap, Ompriya Kale, Ishani Patel, "A Survey on Feature Extraction Methods for HDR", 12, Dec (2015).
- [2] Bhatia P. K., K. Gaurav, "Analytical Review of Preprocessing Techniques for offline HDR", ICETEM, 2014.

- [3] Micheal Revow, Christopher K.I., Williams, "Using Generative Models for HDR", vol 18, no. 6, June 1996.
- [4] K. Deepthi Krishna, Sathvika. R, K. Kavya Sree, Raga Deepika K, Satya Priyanka, "Handwritten Digit Recognition and Text Conversion using MNIST Dataset", IJCRT, Volume 8, Issue 5 May 2020
- [5] Hayder M. Albehadili, Haider A. Alwzwayz, "Handwritten Digit Recognition Using Convolutional Neural Networks", IJRCCE Vol. 4, Issue 2, February 2016
- [6] "Devnagari Numeral Recognition Using Multiple Neural Classifiers" by R. Bajaj, L. Dey, S. Chaudhari.
- [7] B. Zhang and S. N. Srihari, "Fast k-nearest neighbor classification," IEEE, vol. 26, no. 4, 2004.
- [8] Velappa Ganapathy, and Kok Leong Liew, "Handwritten Character Recognition Using Multi scale Neural Network Training Technique", Proceedings of World academy of Science, Engineering and Technology, vol. 29, ISSN 1307-6884, May (2007)