HAND-WRITTEN DIGIT RECOGNITION

A PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF COMPUTER APPLICATION

BY

ANISHA GHOSH (26601219043)

APARNA MUNSHI (26601219042) RINKI DAS (26601219027)

Under the Esteemed Guidance of

Prof. Kaushik Mukherjee

Prof. Sanket Dan

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION

BENGAL SCHOOL OF TECHNOLOGY & MANAGEMENT

Sugandha, Delhi Road, Hooghly

2019-2022

BENGAL SCHOOL OF TECHNOLOGY & MANAGEMENT

Sugandha, Delhi Road, Hooghly

Code: 266

CERTIFICATE

This is certified that Major project titled "HAND-WRITTEN DIGIT RECOGNITION" which is based on machine learning programming is submitted for qualifying in the paper BCAN-591: Major Project in partial fulfilment of the degree BCA, during the academic year 2019-2022 done by ANISHA GHOSH (26601219043), APARNA MUNSHI (26601219042), RINKI DAS (26601219027) is an authentic work carried out by them.

(Signature of Examiner)

DECLARATION

We hereby declare that project titled "HAND-WRITTEN DIGIT RECOGNITION" which is based on machine learning programming is bonafide original record done by us at Bengal School Of technology & Management, Hooghly affiliated to MAKAUT towards the partial fulfilment of requirement for the award of degree of Bachelor of Computer Application during the period of 2019-2022 in BSTM, Hooghly and also we state that this project has not been submitted anywhere in the partial fulfilment for any degree of this or any other University.

APARNA MUNSHI (26601219042)

ANISHA GHOSH (26601219043)

RINKI DAS (26601219027)

ACKNOWLEDGEMENT

It is indeed with a great pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Principal Sir, Bengal School of Technology and Management for the facilities provided to accomplish this main project.

Apart from the efforts of all the team members, the section of this project report largely depends on the encouragement and guidance of our teachers. We take this opportunity to express our gratitude to our project guide Prof Sanket Dan sir and Prof. Kaushik Mukherjee sir for providing able guidance and his generosity and advice extended to me throughout my project and my thesis. We cannot think them enough for their tremendous support and help. They motivated and encouraged we very time while selecting the proper project topic. Without their encouragement and guidance, we would not have been able to select proper topic.

I am also thankful to our team members including Rinki Das and Aparna Munshi.

Project Associates:

RINKI DAS (26601219027)

APARNA MUNSHI (26601219042)

ANISHA GHOSH (26601219043)

TABLE INDEX

ABSTRACT

Handwritten Numeral recognition plays a vital role in postal automation services. This is an important but very hard practical problem. Digit recognition is used in post offices, in banks for reading cheques, for license plate recognition, for street number recognition. The digit recognition can be divided into two groups, printed digit recognition and handwritten digit recognition. Recognition of printed digits is easier compared to the handwritten digit recognition. On the other hand, there are numerous handwriting styles for the same digit; hence more effort is required to find the accurate handwritten digit. In this project, we propose using SVM for recognition of handwritten digit. SVM is Machine Learning Technique. Support Vector Machine (SVM) is one of the most successful classifiers. Many applications use SVM for solving the classification problem, especially those for handwritten digit recognition. The SVM is used to improve classification accuracy. Our proposed algorithm will be tested on standard MNIST dataset for handwritten digit recognition. Total dataset size is of 70,000 datapoints. Among 70,000 datapoints, 60,000 datapoints are for train dataset and 10,000 datapoints are for test dataset. Each image size in 28*28 pixels.

Keywords: SVM, dataset, digit recognition, etc.

CHAPTER 1: INTRODUCTION

1.1 OBJECTIVES

This paper presents a technique developed for recognition images of handwritten digits. This is a task of paramount importance for businesses and enterprises of most diverse areas. A practical example would be automatic bank check processing. This type of business requires a solution that provides high successful rates in handwritten digit recognition in order to correctly process the values of the checks. The need for a method with low error rates is clear, since an error in interpreting the value of a check can cost to the client or to the bank high amount of money. Handwritten digit recognition is a task which presents an elevated level of difficulty due to the high variety of different writings. The constant quest for classifiers with smaller error rates is justified by the high costs associated with misclassifying a character. Support Vector Machine (SVM) is a supervised machine learning algorithm. SVM is one of the powerful techniques for Classification, Regression & Outlier detection with an intuitive model. Drawing hyperplanes only for linear classifier was possible. Many applications use SVM for solving the classification problem, especially these for handwritten digit recognition. SVM is widely used for classification objectives. The objective of the support vector machine algorithm is to find the best hyperplane in an N dimensional space that classifies the data points distinctly.

1.2 PURPOSE

This project aims to meet the following objectives:

i. To develop handwritten digit recognizing system that enables users to automate the process of digit recognition using this deep learning model.
ii. To test the accuracy of the model 10
iii. Efficient model which is less computation intensive

1.3 MOTIVATION

The task of handwritten digit recognition using a classifier has great importance and use such as online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand and so on. There are different 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. Our goal is to implement a pattern classification method to recognize the handwritten digits provided by the user. 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, 8 and 8 etc. Also people write the same digits in many different ways. Finally the uniqueness and variety in the handwriting of different individuals also influences the formation and appearance of the digits.

1.4 BACKGROUND

Handwritten digit recognition system has played a dominant role in modern society. A human can easily solve and recognize any problem, but this is not the same in the case of a machine. Many techniques or methods should be implemented to work as a human. Apart from all the advancements that have been made in this area, there is still a significant research gap that needs to be filled. Consider, for example, online handwriting recognition vs offline recognition. In online handwriting recognition of letters, an on-time compilation of letters is performed while writing because stroke information is captured dynamically. Whereas, in offline recognition, the letters aren't captured dynamically. Online handwriting recognition is more accurate when compared to offline handwriting recognition because of the lack of information. Therefore, there can be research done in this area to improve offline handwriting recognition.

Initially, a dataset is given as input. This is followed by pre-processing, where an image is subjected to various operations like noise reduction, document skew correction, slant correction, normalization, smoothing, and skeletonization. The result of this pre-processing can be given as an input to feature generation. Segmentation of an image is done to isolate the digits of an image into different sub images. Each sub-image is considered as one individual digit.

The next phase is feature generating, in which various extraction techniques are used to represent an image as a vector feature in the feature generator. To keep it clear without any noise, an algorithm is implemented to reduce the size of the image, which in turn reduces the noise in that image. The feature generation is followed by classification. There are a large number of classifiers, which reduces the performance of classification at each step. There are also classifiers for recognition including the statistical, the structural, the stochastic classifiers and finishing on a combination of classifiers. At each step, selecting the appropriate parameters could affect the final classification performance, which results in the complete recognition of an image.

Our main purpose is to find out the rules to be used in the automatic handwritten digit recognition for document images using machine learning methods. The field studied in this thesis work is to recognize the corrupted handwritten digits. Since handwritten digits recognition contains a wide variety of options the way the digits are written, the main concern of this study is to recognize handwritten digits. Although the performance of machine learning algorithms on handwritten digits are very good when the digits are segmented well, the segmentation performance of the existing algorithms is poor, which in turn reduces the recognition performance when digit strings are considered. Therefore, reliable handwritten digit string recognition methods are necessary in order to increase the recognition rate of handwritten digit strings. The success of water reservoir method in handwritten digit recognition motivated us to use them also to decide vertical cuts for segmentation of digit strings by training the regions between the digits.

1.5 HARDWARE AND SOFTWARE USE

Hardware Requirements:

1) Intel Core i5

2) 2.4. GHZ processor

3) 4 GB RAM

4) 4 GB of free hard disk space

5) Keyboard, mouse

Software Requirements:

1) Operating system – windows 8/8.1/10
2) Jupyter notebook

CHAPTER 2: LITERATURE REVIEW

"Literature review (also referred to as a systematic review). A form of secondary study that uses a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable". The motivation behind adopting the literature review is to gain knowledge towards the data sets and the implementation of different types of classifiers to recognize the handwritten digits. A systematic literature review does not opt for this research as the results gathered through this were not used as the results. Once the required data has been obtained from the literature review, then data analysis is performed. Narrative synthesis is adopted as our data analysis method for our literature review. During the literature review, the data collected through the articles were gathered together and they summarized in a paragraph. The results gathered through this data analysis were documented and these were used for the experimental research method. While conducting the literature review, towards recognition of handwritten digits a critical analysis is taken for the methods used solving this problem.

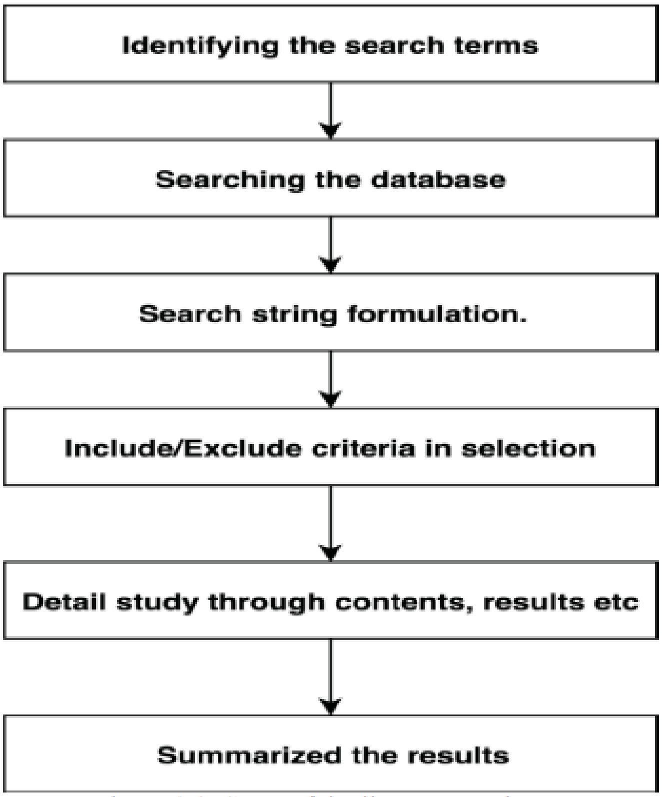


Figure Steps of the literature review

Automatic handwritten digits extract from images is a crucial role for creating documents and processing the systems. The main purpose is to find out the rules to be used in the AHDR for document images using machine learning methods. The field studied in this work is to recognize the corrupted handwritten digits and increase the reliability of the result of the recognition process and to speed up the collecting training and test data from handwritten digit strings. The overall recognition process consists of pre-processing, segmentation, classification and finally recognition of given input data.

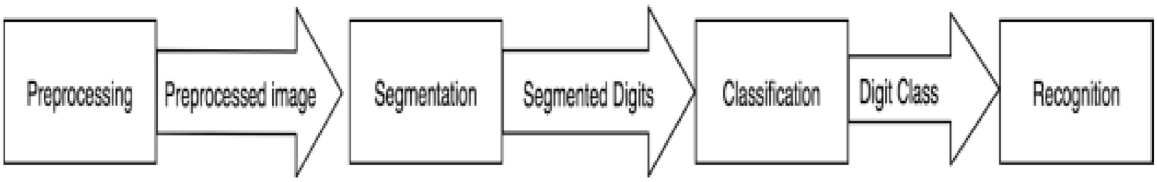


Figure Modules of Handwritten Digit Recognition

2.1 Pre-Processing Method

Pre-processing and feature extraction are very important steps in automatic handwritten digit recognition (AHDR) for documented images. The basic step is to improve the discriminating nature of the pixels or raw features being computed from input images. It has been taken a lot of work improving the pre-processing. One of the problems in the recognition process is skew/slant detection and correction in the documented images, which introduces challenges for segmentation.

However, we can generally categorize the tasks into noise reduction, normalization, smoothing and skeletonization as the Fig.

Figure Pre-Processing Method

At first, the Input image is in RGB format and huge saturation are discarded and intensity is used to obtain a grayscale image. Then the grayscale image should be turned into a binary form.

After binarization, the skew correction could be done in order to correct the angle of the digits and the X-axis. Knerr et al determined the angle by computing the pixel densities between ±5 degrees with the help of horizontal guidelines. With the help of the histogram, we created the pixel densities, histogram with the longest peak is chosen as the angle of the text in the image. The problem with this method relies on the fact of horizontal guideline to realize the actual angle.

2.1.2 Normalization

Normalization method is the most popular method used in character recognition. Because to reduce all types of variations and to obtain standardized data and it also gives excessive shape. The characters for normalizing methods in the following:

- Skew normalization:

It is used due to different types of writing style; the skew can hurt the effectiveness of recognition and therefore it is easy to detect and correct the baseline. Various methods have been used, which are the projection profile of the image, the Hough transform or the shape of the nearest neighbor clustering. After skew detection, the character or word is translated to the origin and rotated until the baseline is horizontal.

- Slant normalization:

The character inclination typically found in cursive script is called slant. Formally, it is defined as the angle between the longest stroke in a word and the vertical direction referred to the word slant. Slant normalization is used to normalize all characters to a standard form with no slant. Many methods have been proposed to detect and to correct the slant of cursive words. One of the used methods is based on the center of gravity, another method uses the projection profiles and some used a variant of the Hough transform.

● Size normalization:

It is used to adjust the size, position, and shape (dimension) of the character image. This step is required for reducing the shape variation between images of the class to facilitate the feature generation and improve their classification.

2.1.3 Smoothing

Smoothing operation is done to regularize the edges in the image, to remove small bits of noise and to reduce the high-frequency noise in the image. Furthermore, different pre-processing methods are used for the smoothing image to acquire a more accurate output image.

CHAPTER 3: MODEL ARCHITECHTURE

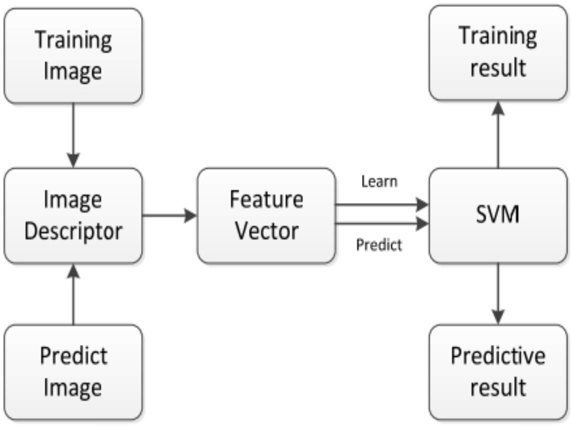


Fig .: The flow chart of recognition using Support Vector Machine

CHAPTER 4: METHODOLOGY

4.1 DATASET INFORMATION

Collecting handwritten digits dataset is a very time-consuming task. The dataset is a hand-written digit’s datasets. The digits dataset consists of 1000 images, in which each image is an 8*8 pixel one representing a handwritten digit. Those images in dataset are divided into 10 classes, and each class refers to a digit, is a number in the range of 0 to 9 (fig). The test set was used for writer-independent testing and is the actual quality measure.

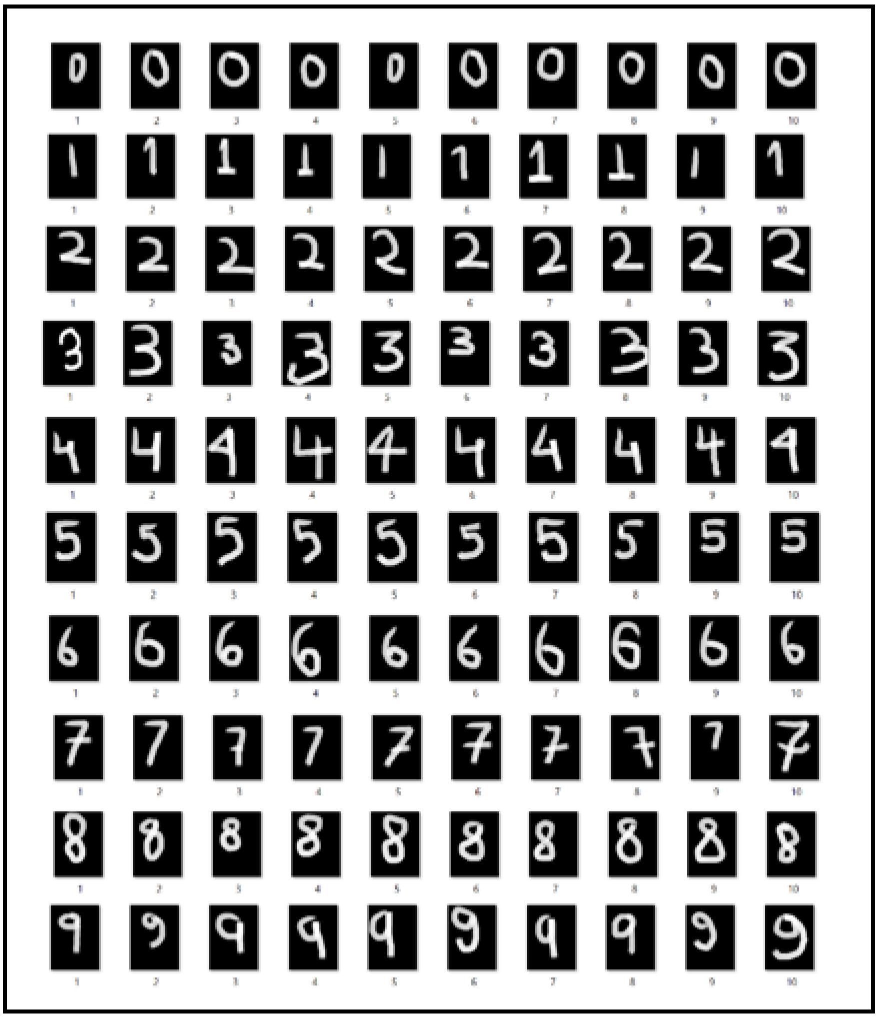


Fig : DATASET

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	
1	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	pixel10	pixel11	pixel12	pixel13	pixel14	pixel15	pixel16	pixel17	pixel18	pixel19	pixel20	pixel21	pixel22	pixel23	pixel24	pixel25	pixel26	pixel27	pixel28	pixel29	pixel30	pixel31	pixel32	pixel33	pixel34	pixel35	pixel36	pixel37	pixel38	pixel39	pixel40	pixel41	pixel42	pixel43	pixel44	pi
2																																															
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4																																															
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6																																															
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8																																															
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10																																															
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12																																															
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14																																															
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16																																															
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18																																															
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20																																															
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22																																															
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
24																																															
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26																																															
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
28																																															
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30																																															
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
32																																															
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
34																																															
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
36																																															
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
38																																															
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
40																																															
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
42																																															
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
44																																															
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
46																																															
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
48																																															
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
50																																															
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
52																																															
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
54																																															
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
56																																															
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
58																																															
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
60																																															
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig 2.: After converting from pixel to binary image

```
<matplotlib.image.AxesImage at 0x1aed9a246d0>
```

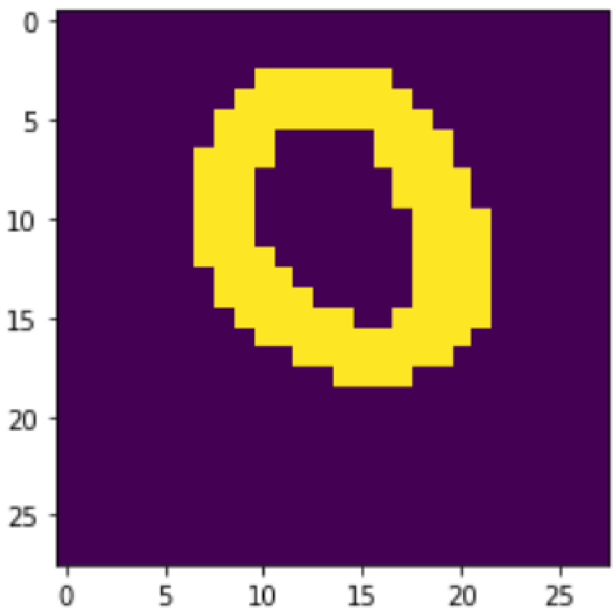
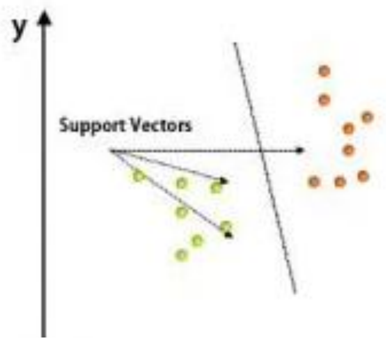


Fig 3.: Image preview

4.2. Algorithms and methods used

4.2.1 SVM

SVM stands for Support Vector Machine. SVM is a supervised machine learning algorithm that is commonly used for classification and regression challenges. Common applications of the SVM algorithm are Intrusion Detection System, Handwriting Recognition, Protein Structure Prediction, Detecting Steganography in digital images, etc.
In the SVM algorithm, each point is represented as a data item within the n-dimensional space where the value of each feature is the value of a specific coordinate.
After plotting, classification has been performed by finding hype plane, which differentiates two classes. Refer to the below image to understand this concept.



Support Vector Machine algorithm is mainly used to solve classification problems. Support vectors are nothing but the coordinates of each data item. Support Vector Machine is a frontier that differentiates two classes using hyper-plane.

4.2.2 SVC

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations.

CHAPTER 5: RESULT AND DISCUSSION

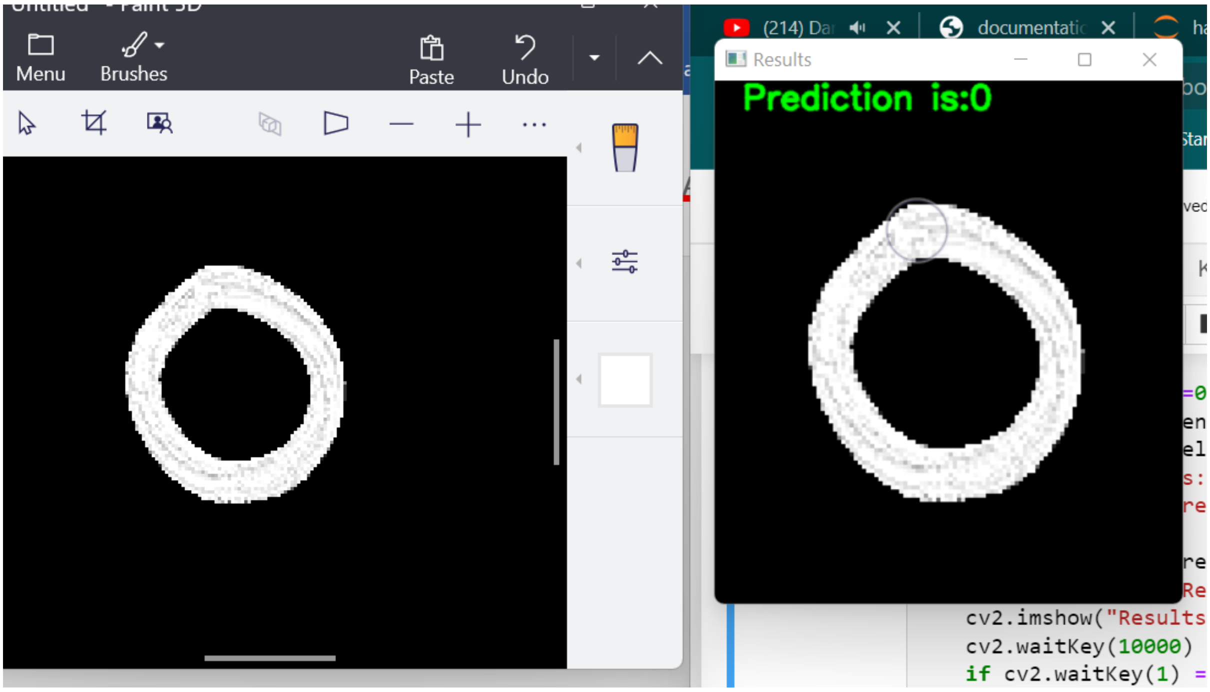


Fig 4.: Prediction of image.

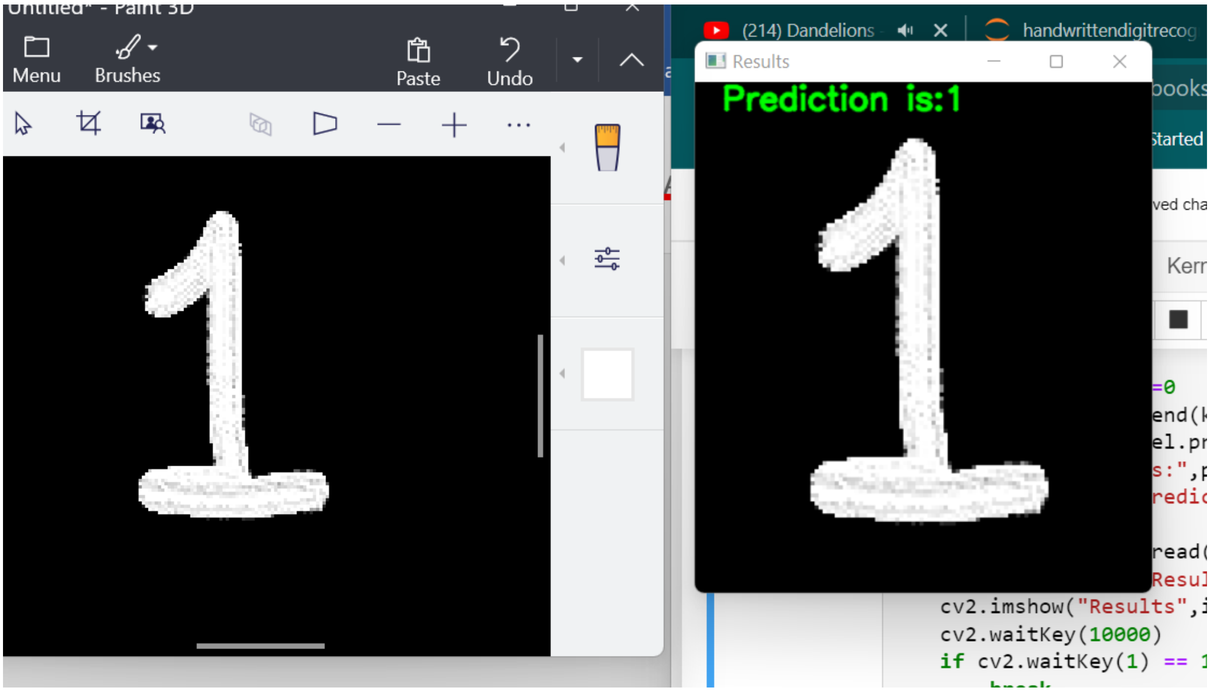


Fig 5.: Prediction of image.

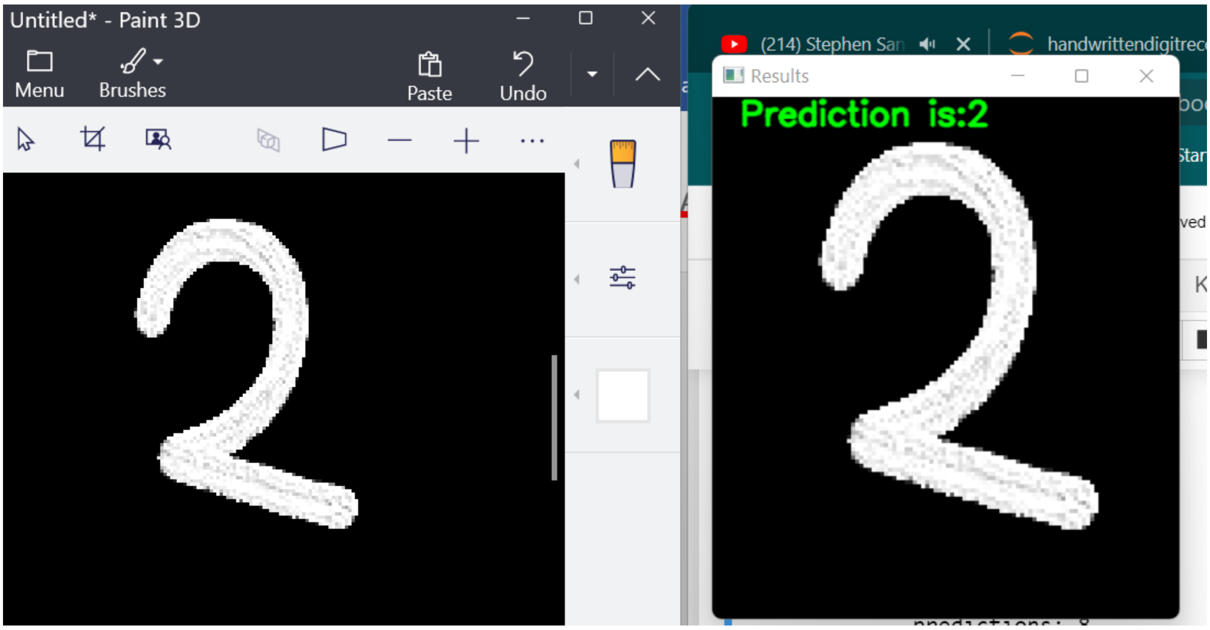


Fig 5.: Prediction of image.

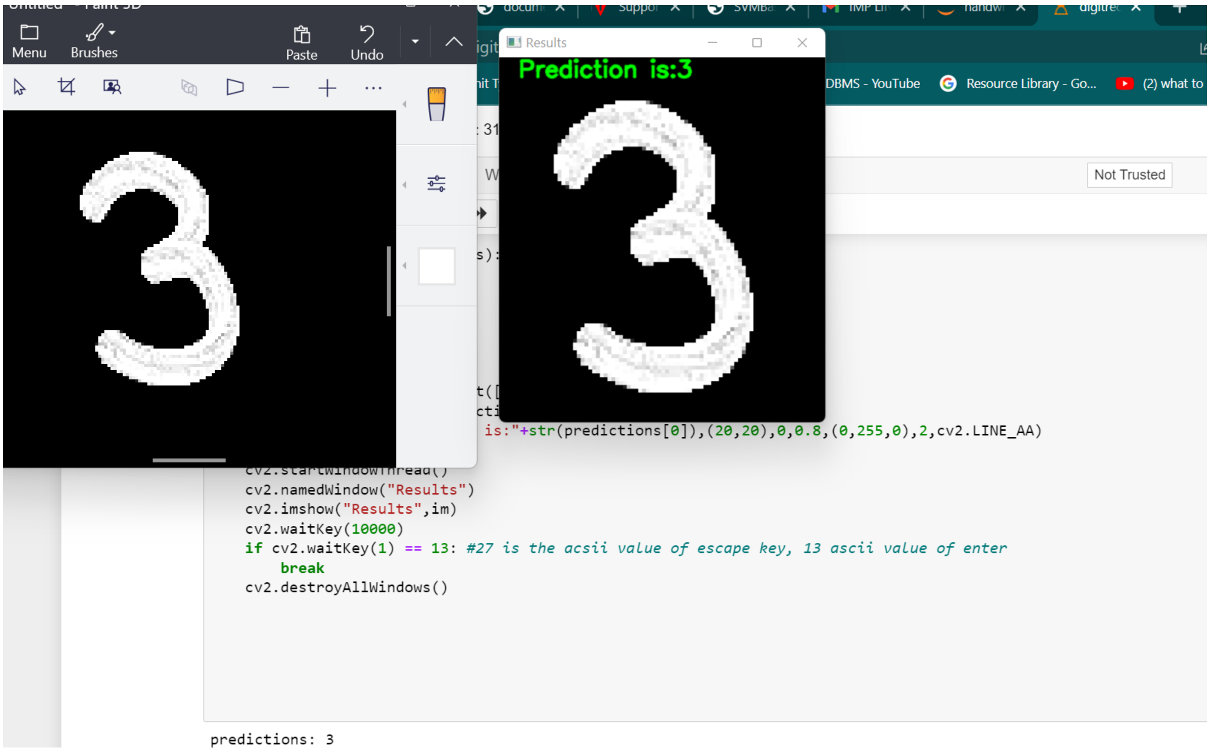


Fig 5.: Prediction of image.

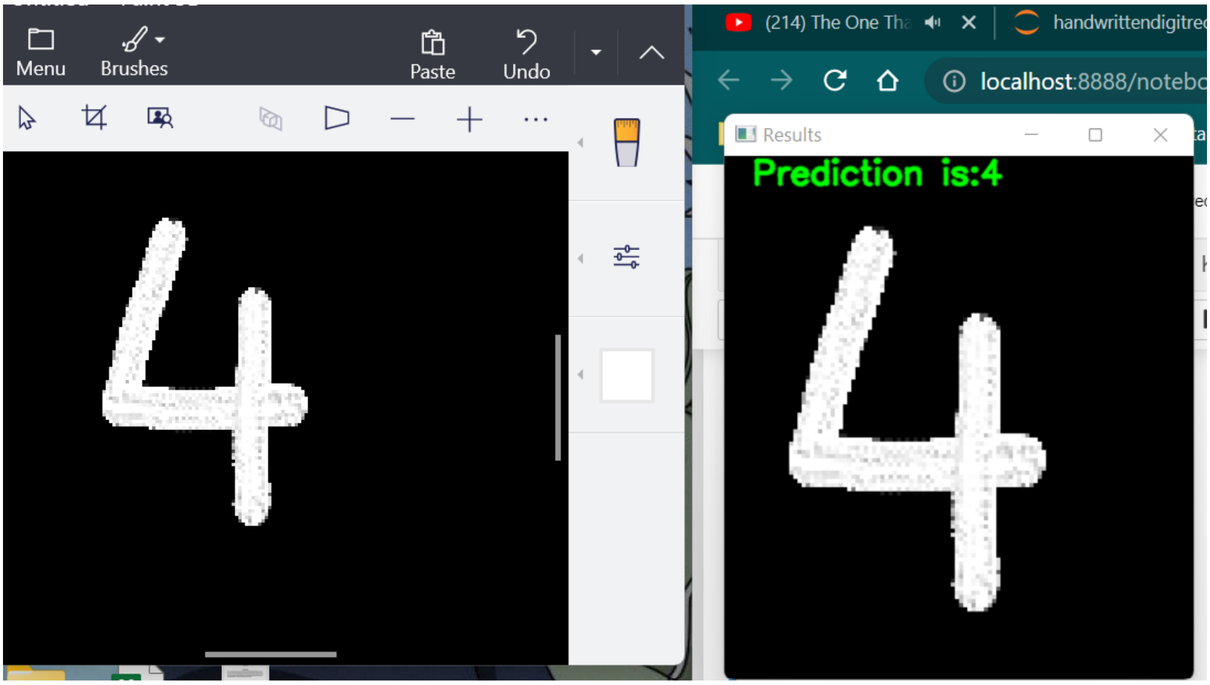


Fig 5.: Prediction of image.

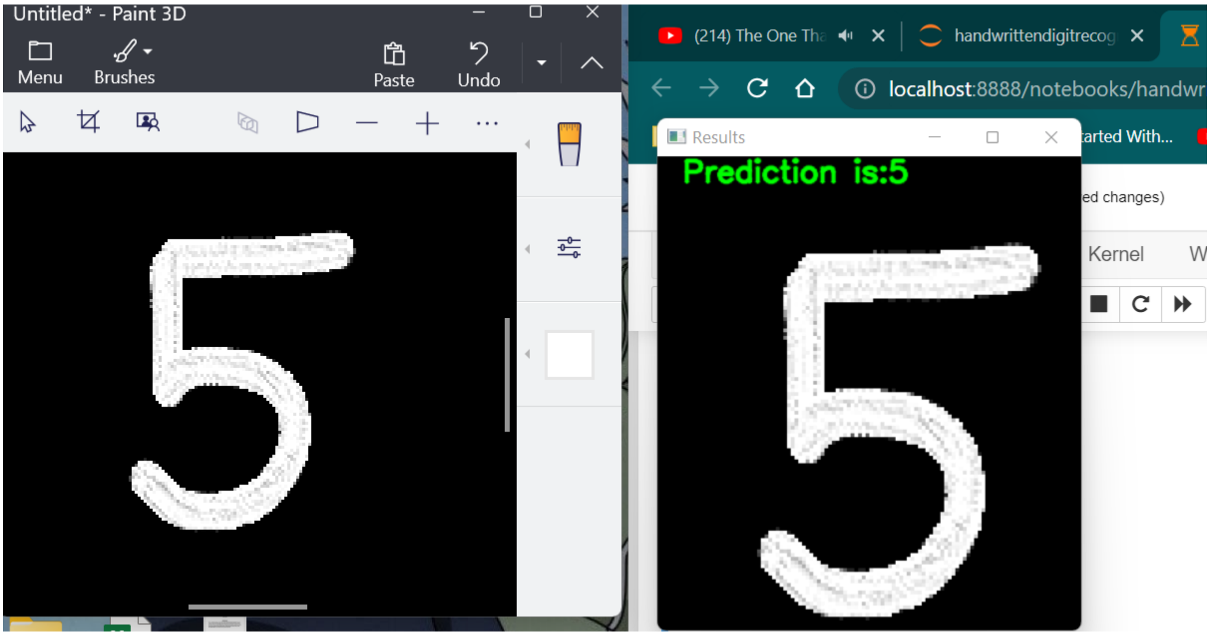


Fig 5.: Prediction of image.

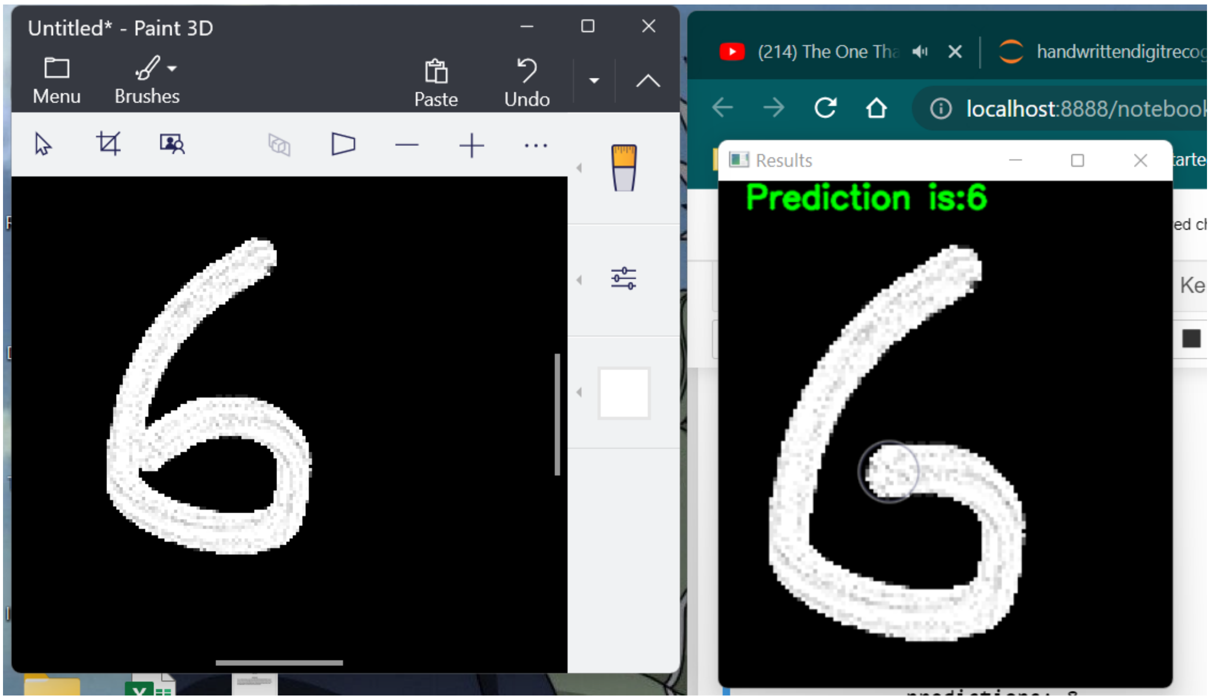


Fig 5.: Prediction of image.

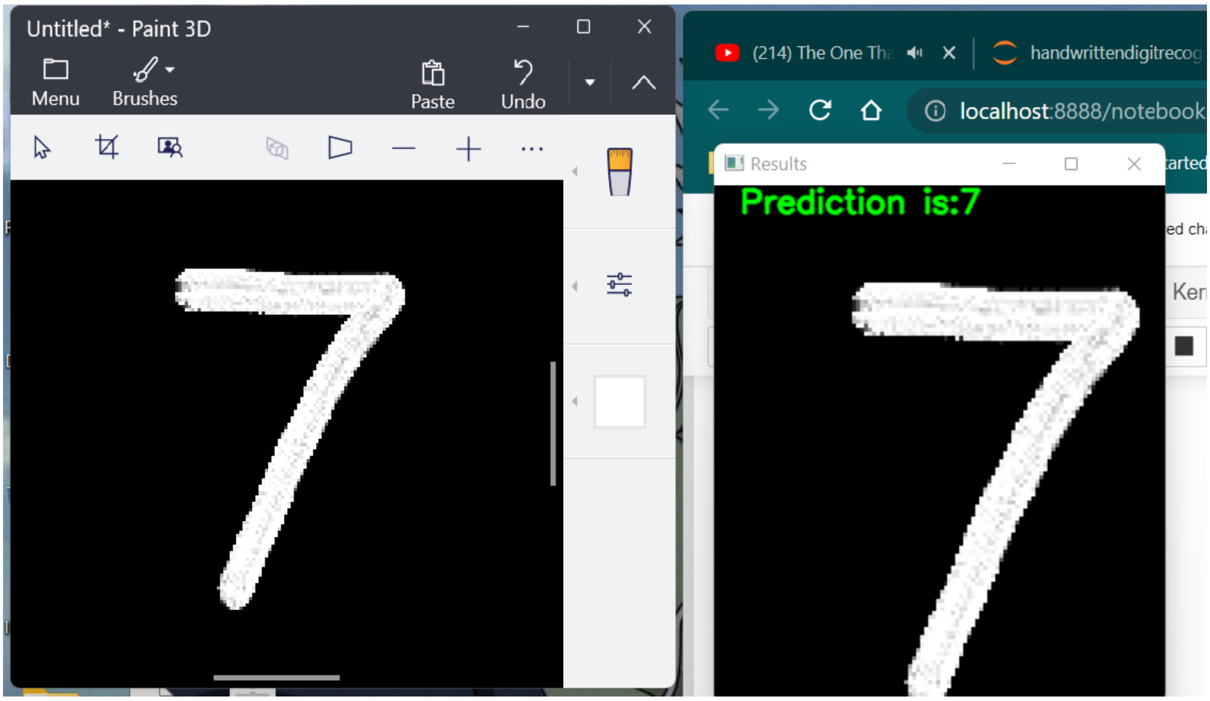


Fig 5.: Prediction of image.

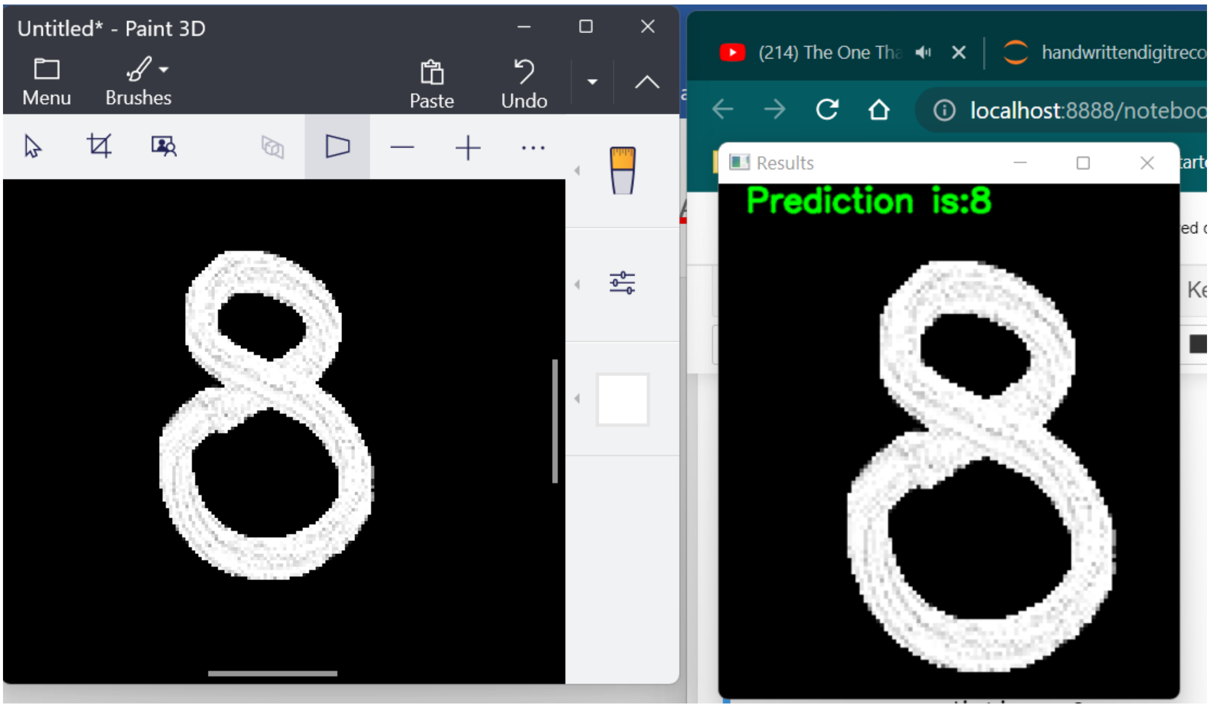


Fig 5.: Prediction of image.

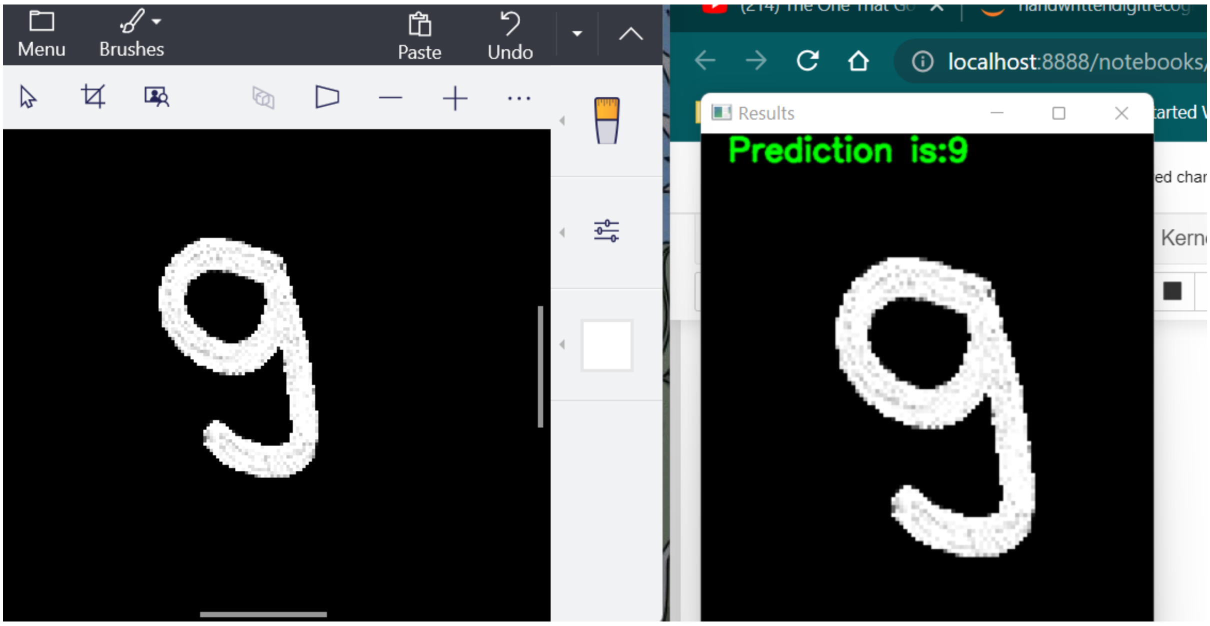


Fig 5.: Prediction of image.

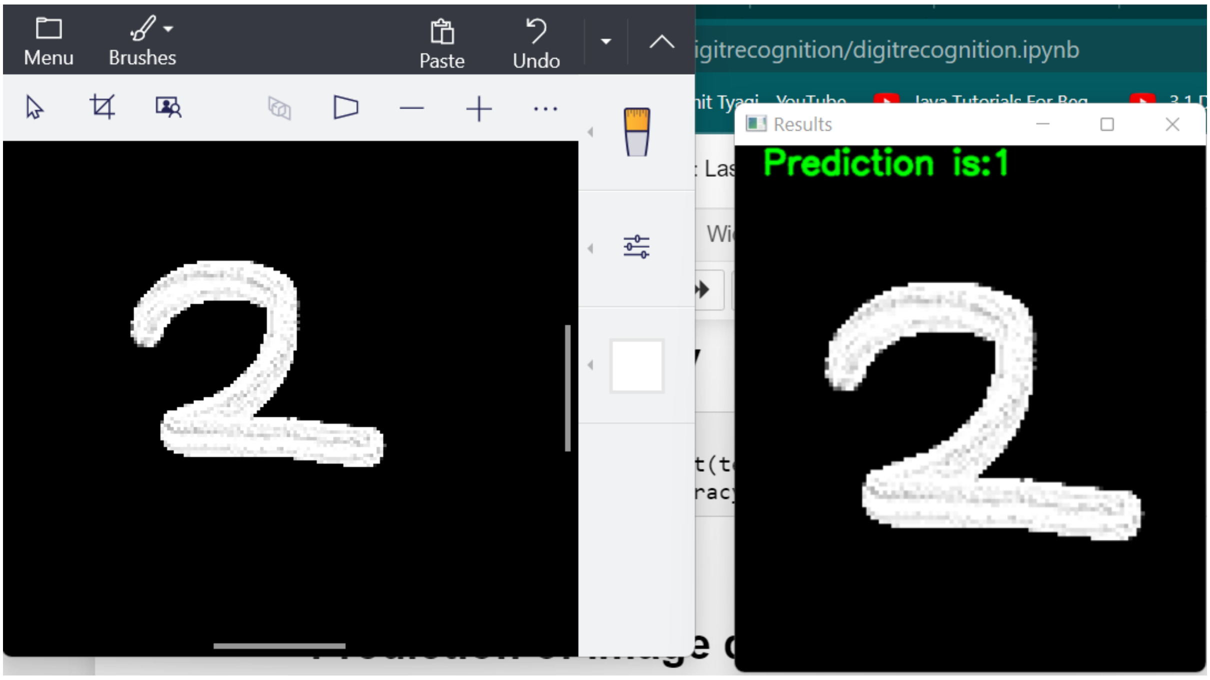


Fig 5.: Prediction of image.

5.1 TEST ACCURACY

We gave 10 image input of ONE where 6 time the prediction was correct so the test accuracy is 60%. We gave 10 image input of TWO where 5 time the prediction was correct so the test accuracy is 50%. We gave 10 image input of THREE where 7 time the prediction was correct so the test accuracy is 70%. We gave 10 image input of FOUR where 4 time the prediction was correct so the test accuracy is 60%. We gave 10 image input of FIVE where 6 time the prediction was correct so the test accuracy is 60%. We gave 10 image input of SIX where 5 time the prediction was correct so the test accuracy is 50%. We gave 10 image input of SEVEN where 6 time the prediction was correct so the test accuracy is 60%. We gave 10 image input of EIGHT where 5 time the prediction was correct so the test accuracy is 50%. We gave 10 image input of NINE where 3 time the prediction was correct so the test accuracy is 30%.The test accuracy is 49%.

CONCLUSION

This paper proposed hand written digit recognition using support vector machine. In the existing system, spatial information is lost which is important for classification. Using, Support Vector Machine, we overcome this problem which provides best classification. So, with Support Vector Machine Technique we can recognize the digit more accurately.

FUTURE SCOPE

The proposed recognition system is implemented on handwritten digits taken from handmade database. Handwritten digit recognition system can be extended to a recognition system that can also able to recognize handwritten character and handwritten symbols. The proposed work can be extended to work on broken digits. The proposed work can be further extended to improve the digit recognition accuracy.

Reference

[1] David Cournapeau "scikit-learn Machine Learning in Python," scikit-learn,Vangie Beal. "Android SDK. Internet: https://scikit-learn.org/stable/,[8,March,2022].

[2] Ponty "Pyscreenshot," pypi, "Android SDK.Internet:https://pypi.org/project/pyscreenshot/,[10,March,2022].

[3] Sam Patterson "Python Documentation," python docs, "Android SDK. Internet: https://docs.python.org/3/,[11,March,2022].

[4] Javatpoint technical team " Machine Learning support vector machine algorithm," javatpoint, "Android SDK. Internet: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm,[10,March,2022].