**CHAPTER 1**

**Introduction**

Learn Digits is an application software where the management of the recognition of numbers is computerized. The hotel management system is designed using **PYTHON** as the rich GUI the for front end and **PYTHON** Server as the secured backend database.

In this project the details are maintained like number details, dataset details, training details and dataset generation details The recognition process of numbers for the user, accuracy of the numbers, precision of the numbers, loss of the numbers, the dataset management, dataset generation process, etc all are computerized and the management is done without any difficulty.

In this software recognition of numbers make this system very flexible and convenient. The working person is a very busy and does not have the time to sit and teach the entire activities manually on paper to his/her child. This application gives him the power and flexibility to manage the entire system from a single online system. User has the power of either creating or distroying the user’s dataset.

The numbers can be viewed completely and the dataset of the digits can be review. For blind peoples it will be more useful. This Proposed System will be interactive, faster and user-friendly for the end users. Using the Learn Digits, the following activities can be performed.

* Capture Screen
* Calculate Accuracy
* Generate dataset
* Load dataset
* Train dataset
* Live prediction

## AIM

The mission is to facilitate easy learn and recognize of a number with capabilities to do predict the numbers, capture screenshots of the images, generate dataset, Load dataset, Train dataset, Live Prediction, etc. Using the automated learn digits software. One can keep detailed records or info on an unlimited amount of dataset. The system lets the user know which all number are available after he draw the number at any point of time. This makes the prediction considerably faster. And thus helps the recognition in better management and reduce a lot of paper work as well as manpower.

1.1) Project Description

Learn digits provides dataset creation, load dataset, and generation of dataset. The system will be so simple and attractive which will make the customer comfortable to use and choose their ideal number. The system allows the Owner to check the Accuracy of the number from interactive Graphics and he will be notified of each new change made in number.

**Flexibility:** Ability to add new features to the system and handle them conveniently

**Reliability**: Specify the factors required to establish the required reliability of the software system at time of delivery. Meantime between failures and mean time to recovery

## Definition

Learn Digits is a application software where the management of entire numbers(0-9) is computerized. The Learn Digits system is designed using PYTHON as the rich GUI for front end and PYSCREENSHOT as the secured backend database.

In this project the details are maintained like Load dataset details, generate dataset details, train dataset details and Live prediction details The recognitoin process of numbers for the customers, live prediction of the numbers, etc all are computerized and the management is done without any difficulty.

******

**Fig 1. Learn digits**

* 1. **Project Scope**

The software to be developed deals with creating a Learn digits which will automate the major learning operations such as generating dataset,. Admin have the authority to create and modify the dataset.

The mission is to facilitate easy management and administration of a recognition of numbers with capabilities to do predict or recognitize them, generate of the dataset, live prediction, load dataset, generate dataset, train dataset, predict dataset, etc. using the automated Learn Digits software. One can Keep detailed records or info on an unlimited amount of number recorgnition. The system lets the user Know which number is he drawing at any point of time. This makes the prediction considerably faster. And thus helps the hotel in better management and reduce a lot of paper work as well as manpower.

* 1. **Hardware & Software Requirements**
     + Processor: Multi-core – 1.8 GHz processor or better.
     + Ram: 4GB RAM.
     + Hard disk: 80GB.
     + Operating system: Windows 7, 8 and above.
     + Front END: PYTHON
     + Back end: PYTHON

What is Learn Digits?

Learn Digits is a application software where the management of entire numbers(0-9) is computerized. The Learn Digits system is designed using PYTHON as the rich GUI for front end and PYSCREENSHOT as the secured backend database.

In this project the details are maintained like Load dataset details, generate dataset details, train dataset details and Live prediction details The recognitoin process of numbers for the customers, live prediction of the numbers, etc all are computerized and the management is done without any difficulty.

The mission is to facilitate easy management and administration of a recognition of numbers with capabilities to do predict or recognitize them, generate of the dataset, live prediction, load dataset, generate dataset, train dataset, predict dataset, etc. using the automated Learn Digits software. One can Keep detailed records or info on an unlimited amount of number recorgnition.

The system lets the user Know which number is he drawing at any point of time. This makes the prediction considerably faster. And thus helps the hotel in better management and reduce a lot of paper work as well as manpower.

Learn digits provides dataset creation, load dataset, and generation of dataset. The system will be so simple and attractive which will make the customer comfortable to use and choose their ideal number. The system allows the Owner to check the Accuracy of the number from interactive Graphics and he will be notified of each new change made in number.

How does hotel management work?

In this project the details are maintained like number details, dataset details, training details and dataset generation details The recognition process of numbers for the user, accuracy of the numbers, precision of the numbers, loss of the numbers, the dataset management, dataset generation process, etc all are computerized and the management is done without any difficulty.

In this software recognition of numbers make this system very flexible and convenient. The working person is a very busy and does not have the time to sit and teach the entire activities manually on paper to his/her child. This application gives him the power and flexibility to manage the entire system from a single online system. User has the power of either creating or distroying the user’s dataset.

The mission is to facilitate easy learn and recognize of a numbers with capabilities to do predict the numbers, capture screenshots of the images, generate dataset, Load dataset, Train dataset, Live Prediction, etc.

Using the automated learn digits software. One can keep detailed records or info on an unlimited amount of dataset. The system lets the user know which all number are available after he draw the number at any point of time. This makes the prediction considerably faster. And thus helps the recognition in better management and reduce a lot of paper work as well as manpower.

## CHAPTER 2

**Feasibility Study**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

* 1. **Operational Feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development [1]. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes [2]. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters [3]. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

* 1. **Technical Feasibility**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor [4]. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs

of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system [5].

* 1. **Economical Feasibility**

Establishing the cost-effectiveness of the proposed system i.e. If the benefits do not outweigh the cost then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible [6], [7]. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide [8], [9]. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis [10].

**CHAPTER 3**

DATABASE DESIGN

ER Diagram

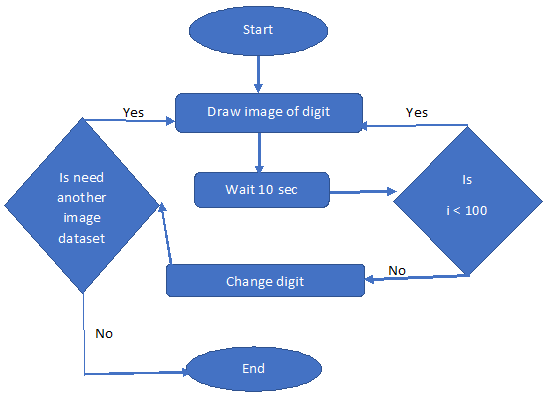


Fig 2. Flow diagram

DFD

Zero Level DFD:

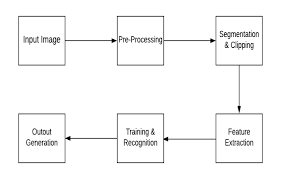


Fig 3. Zero level DFD

First Level DFD:

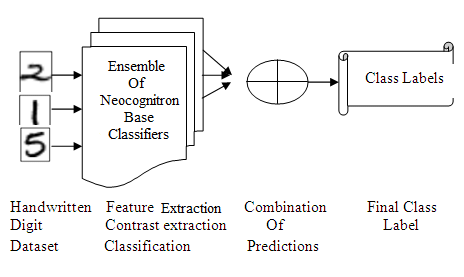


Fig 3. First level DFD

Second Level DFD:

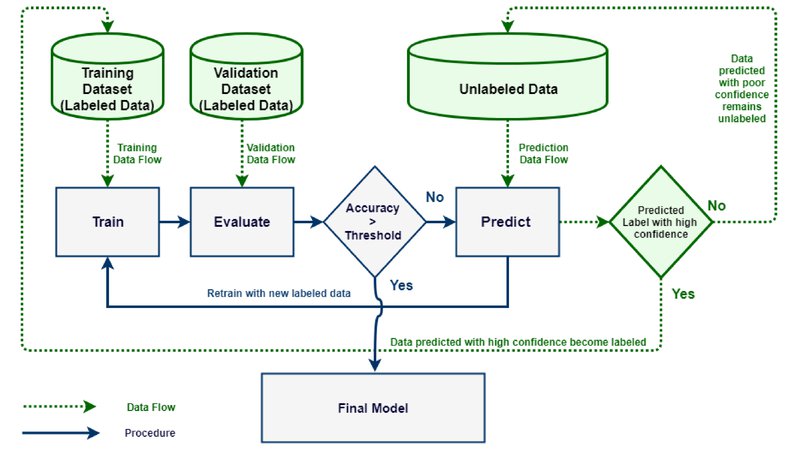


Fig 4. Second level DFD

* 1. **Database Tables**

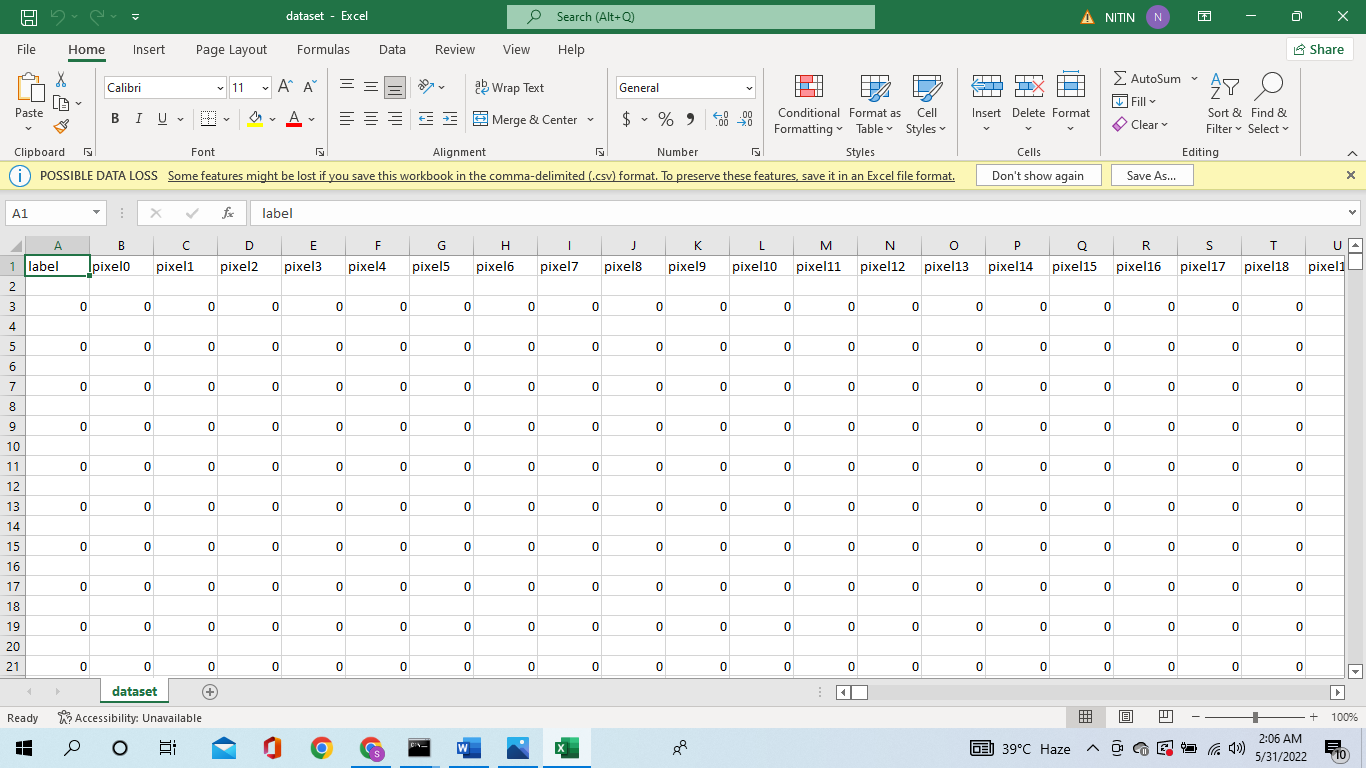


Fig 5. Dataset image 1

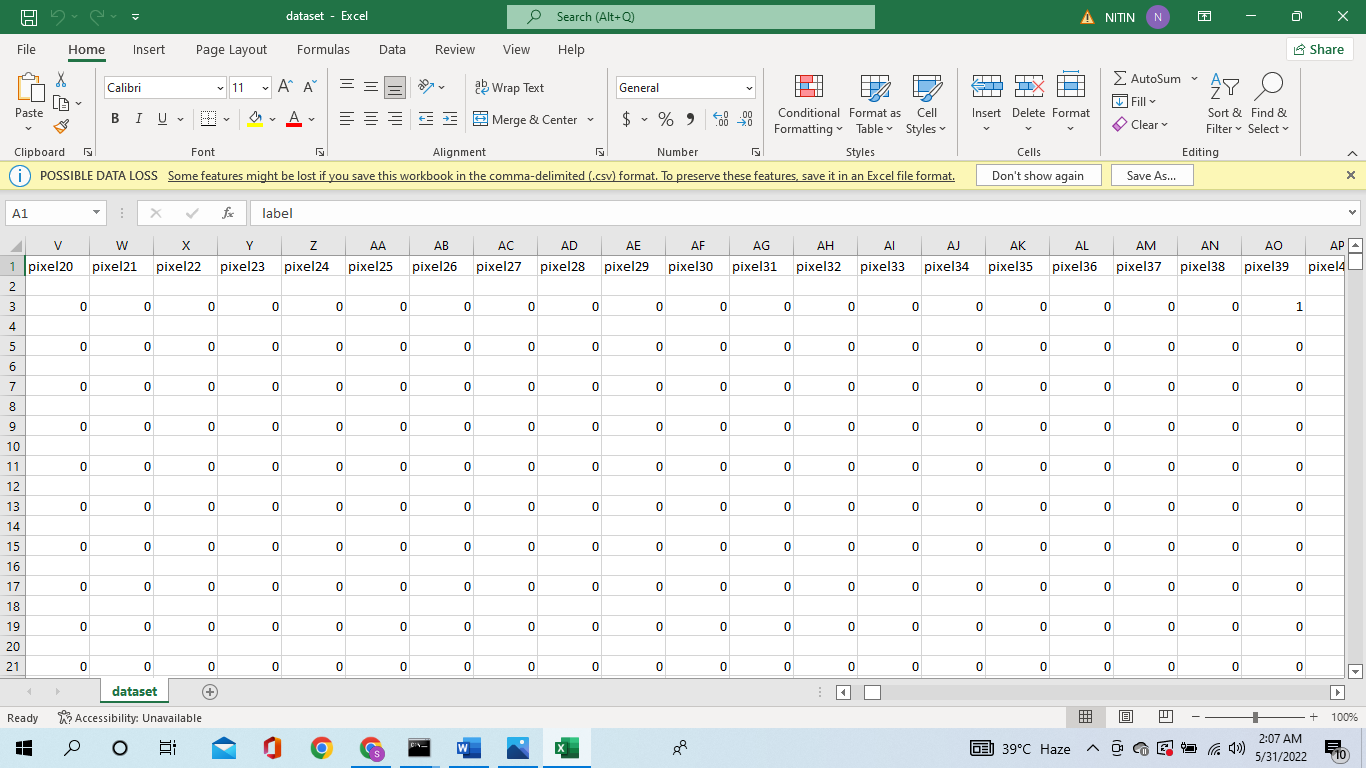


Fig 6. Dataset image 2

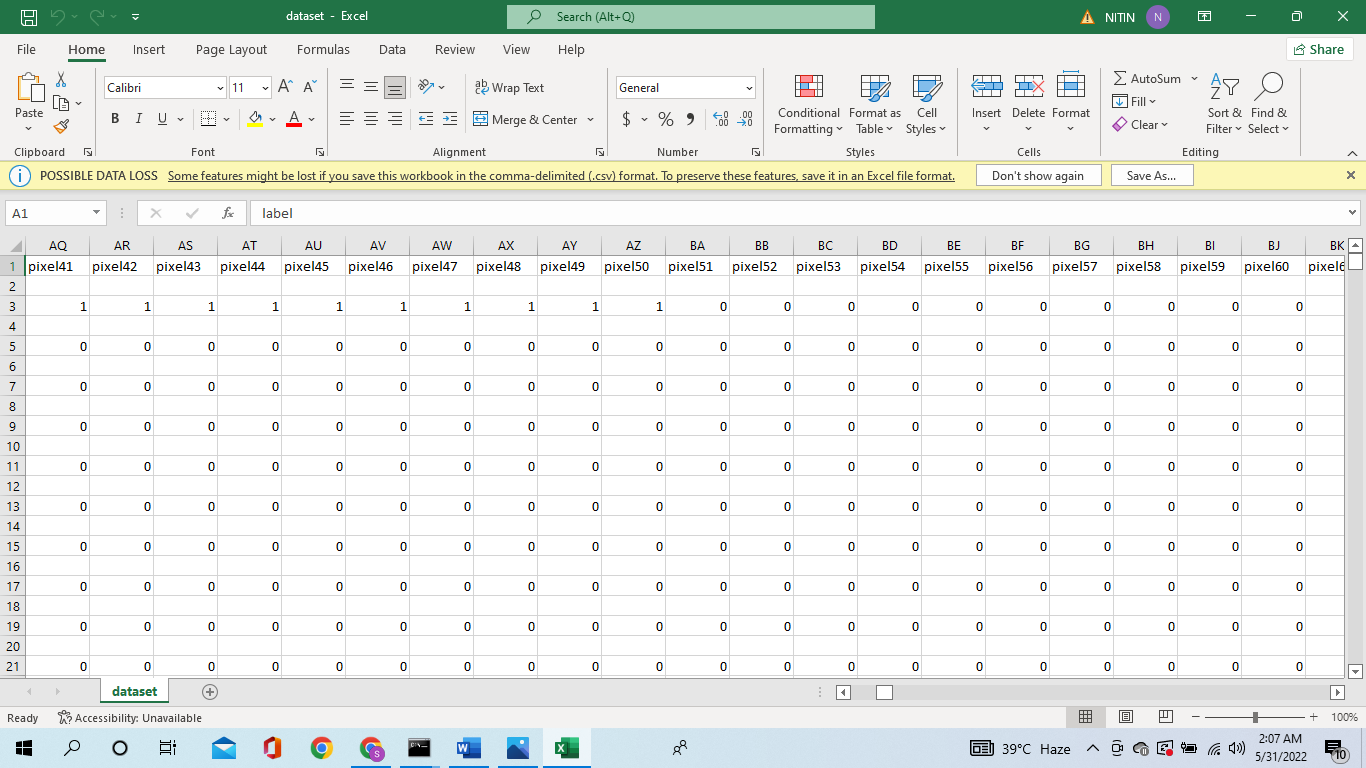


Fig 7. Dataset image 3

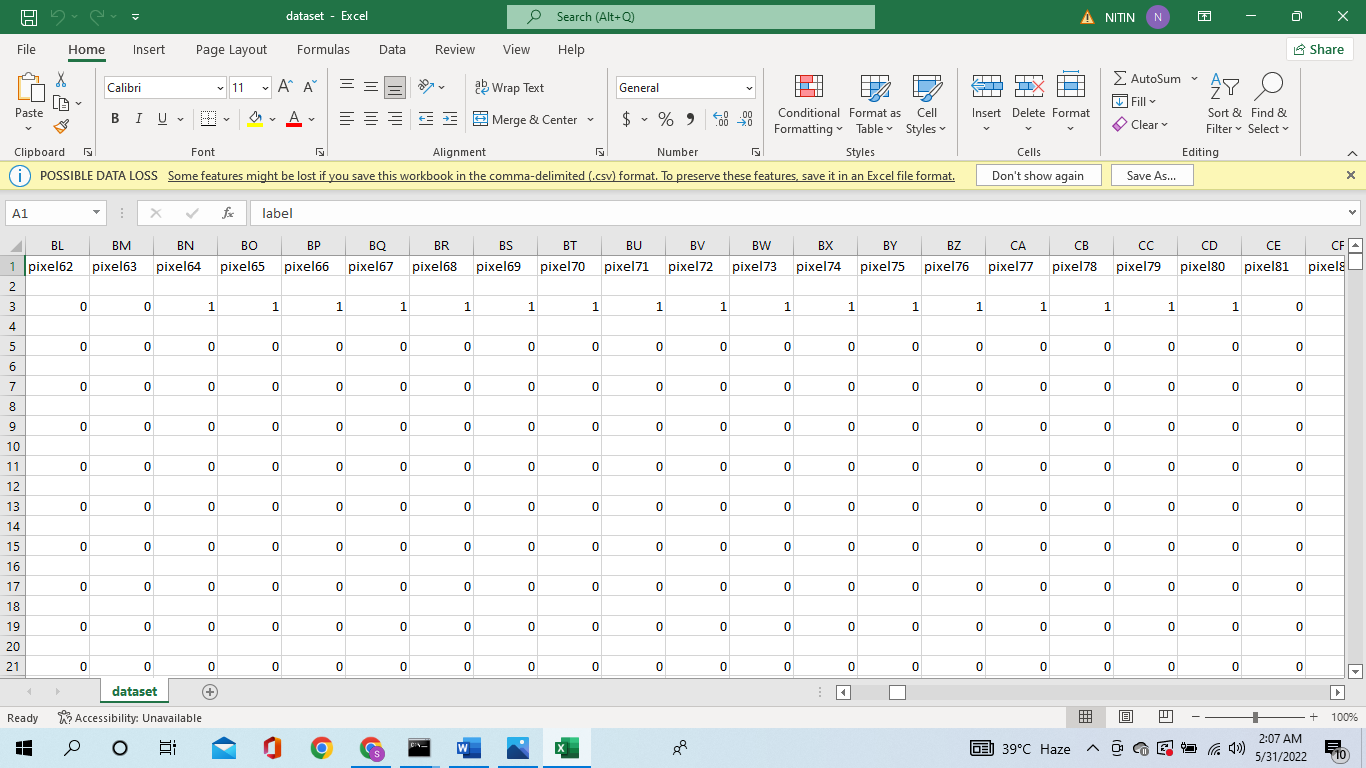


Fig 8. Dataset image 4

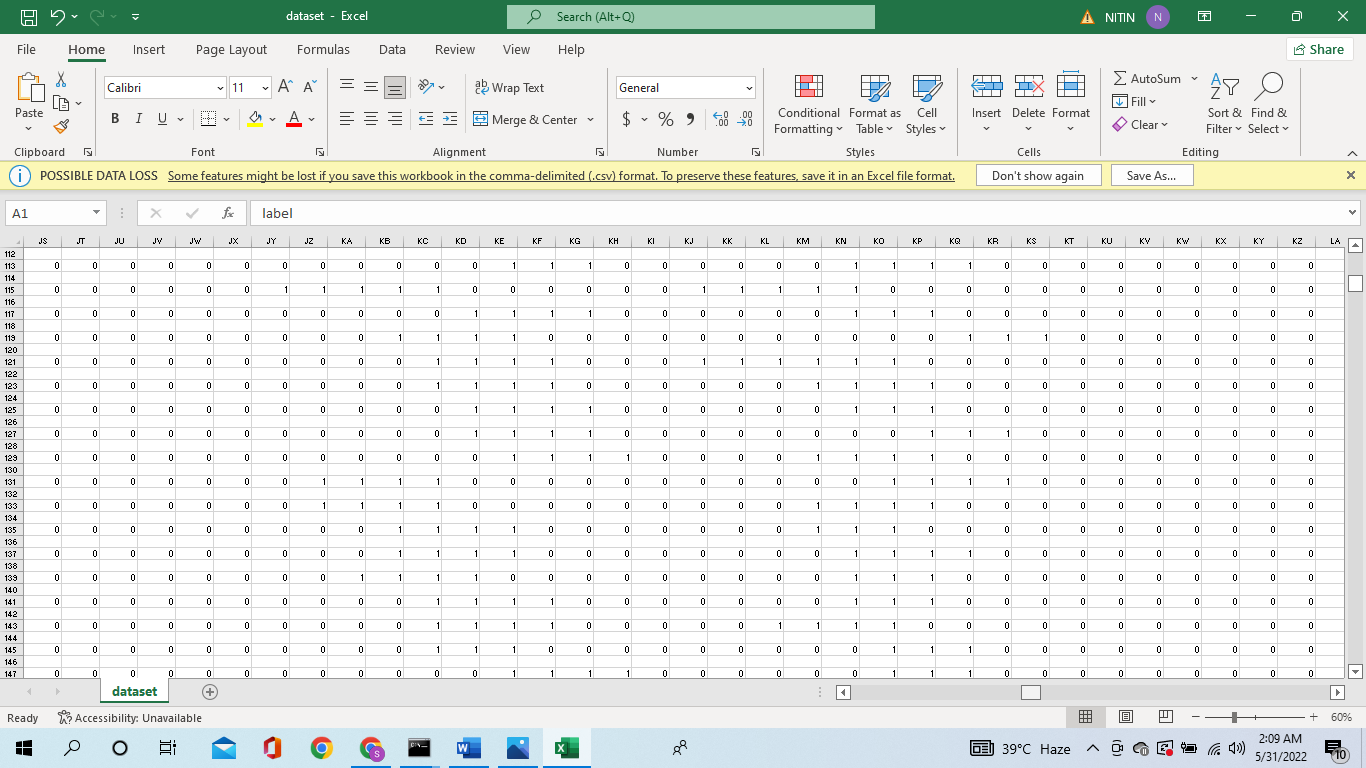


Fig 9. Dataset image 9

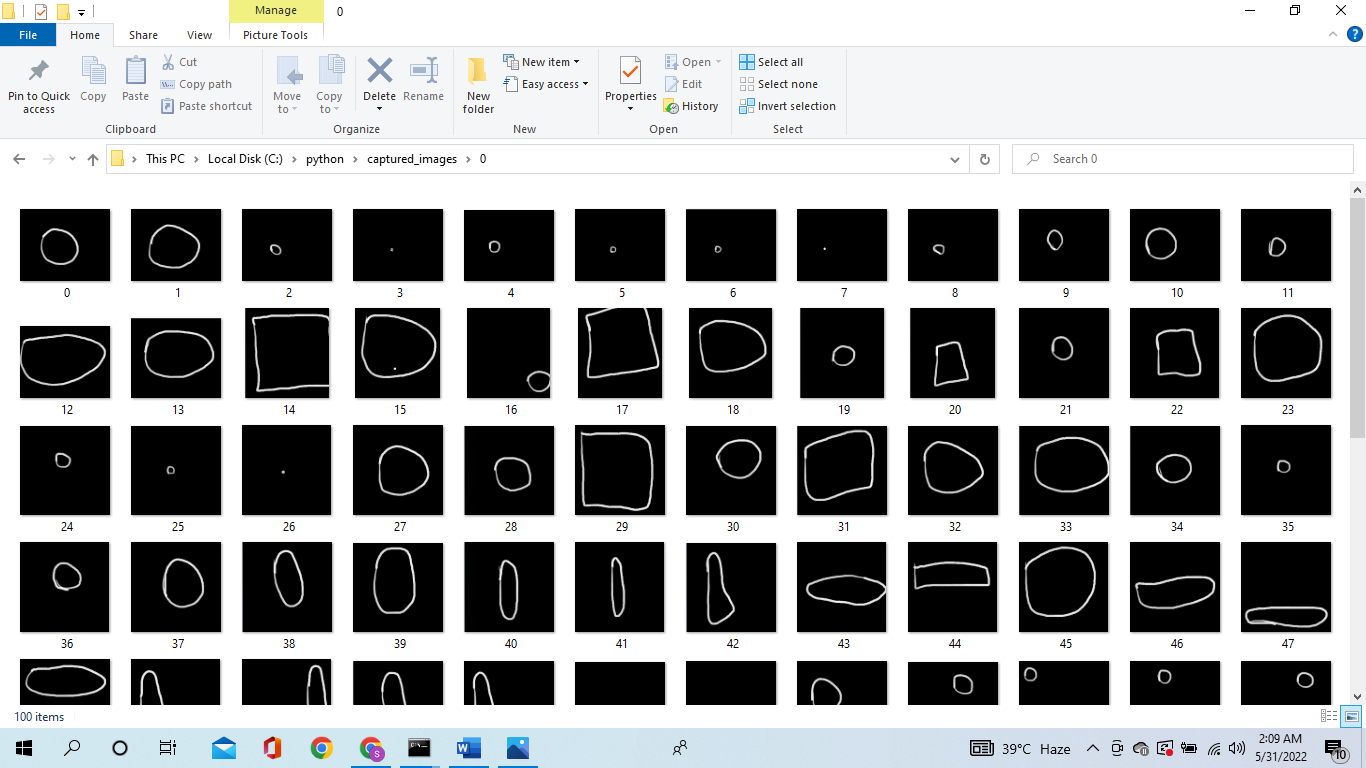


Fig 10. Sample image 1

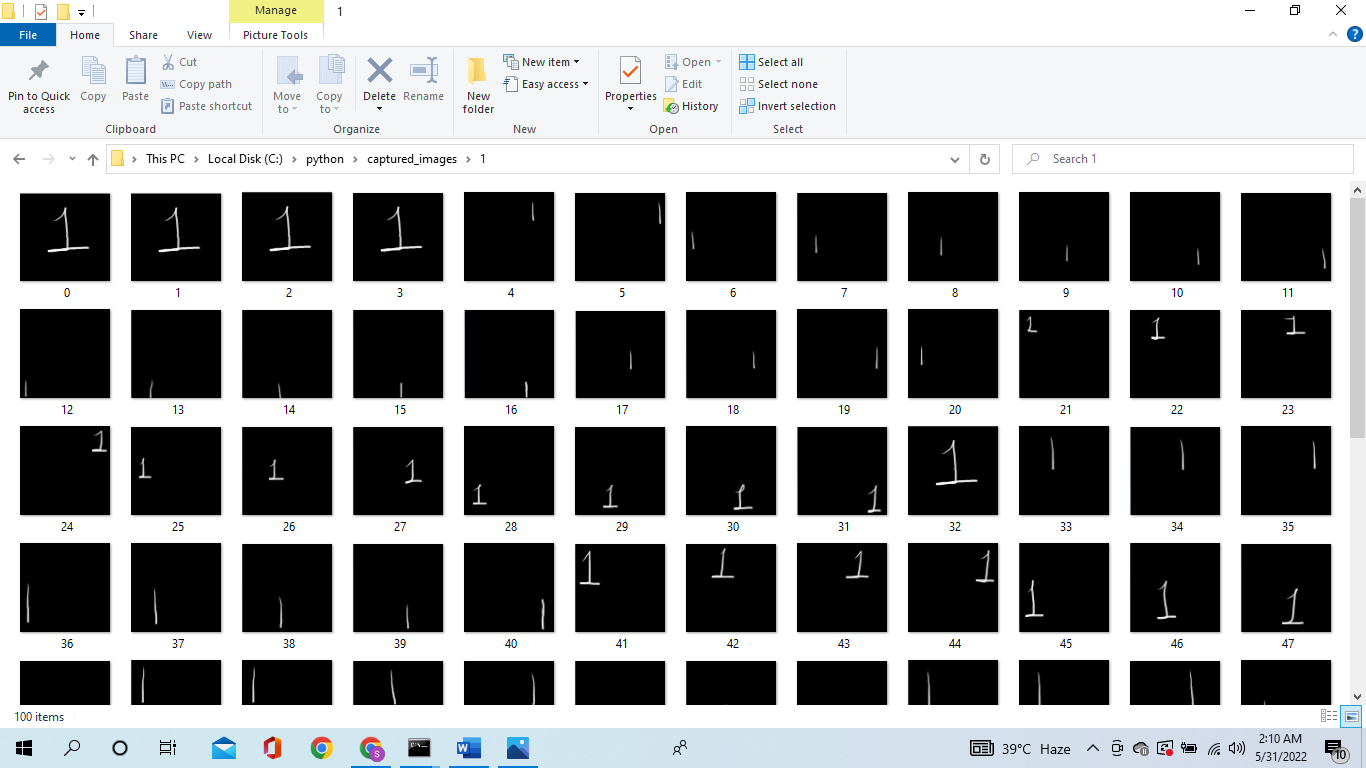


Fig:11 Sample images

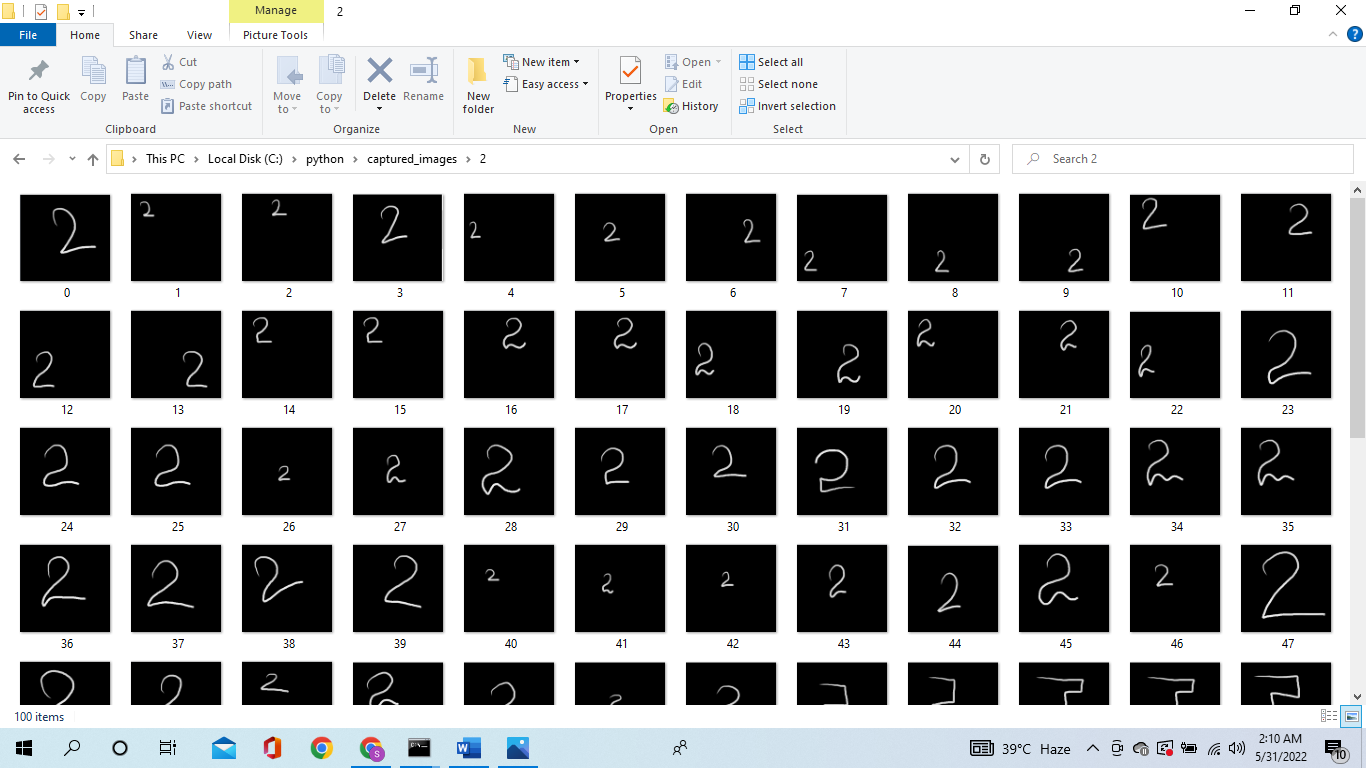


Fig12: Sample Images

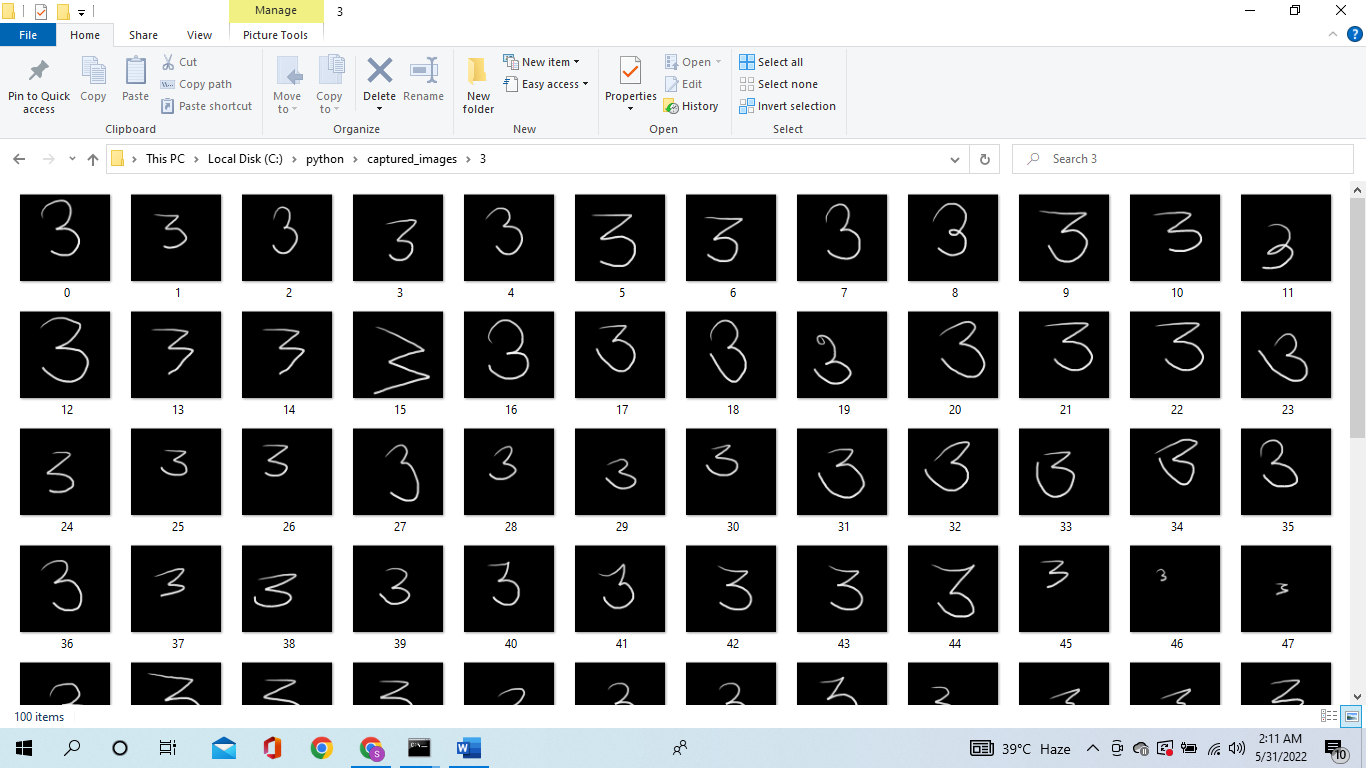
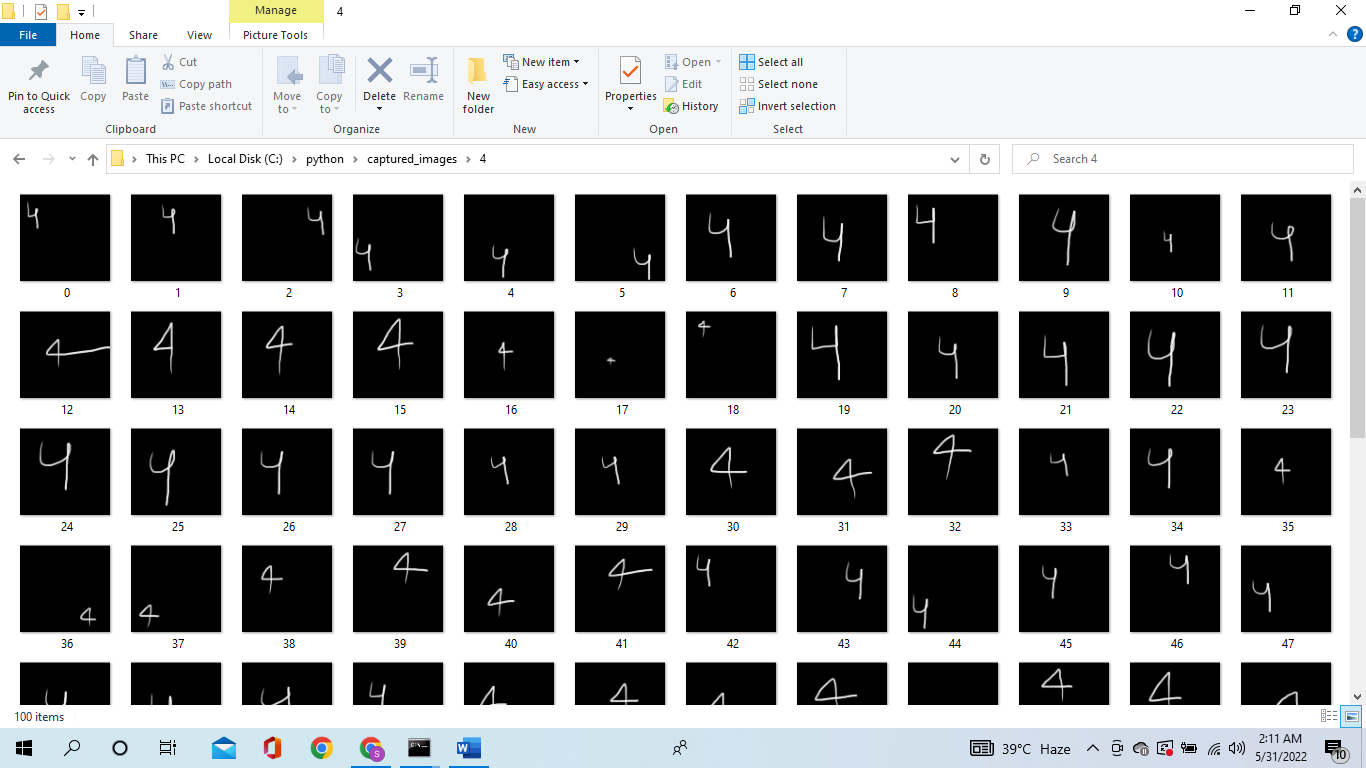
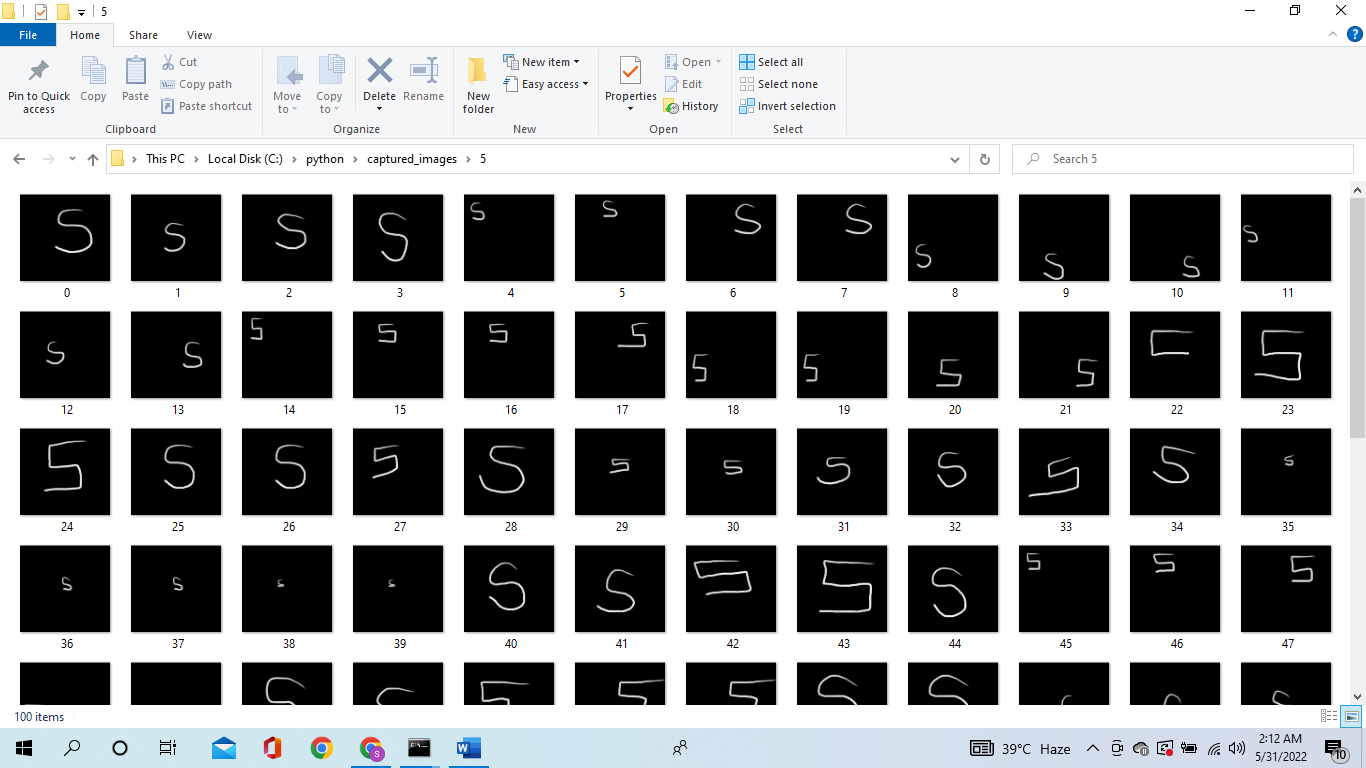
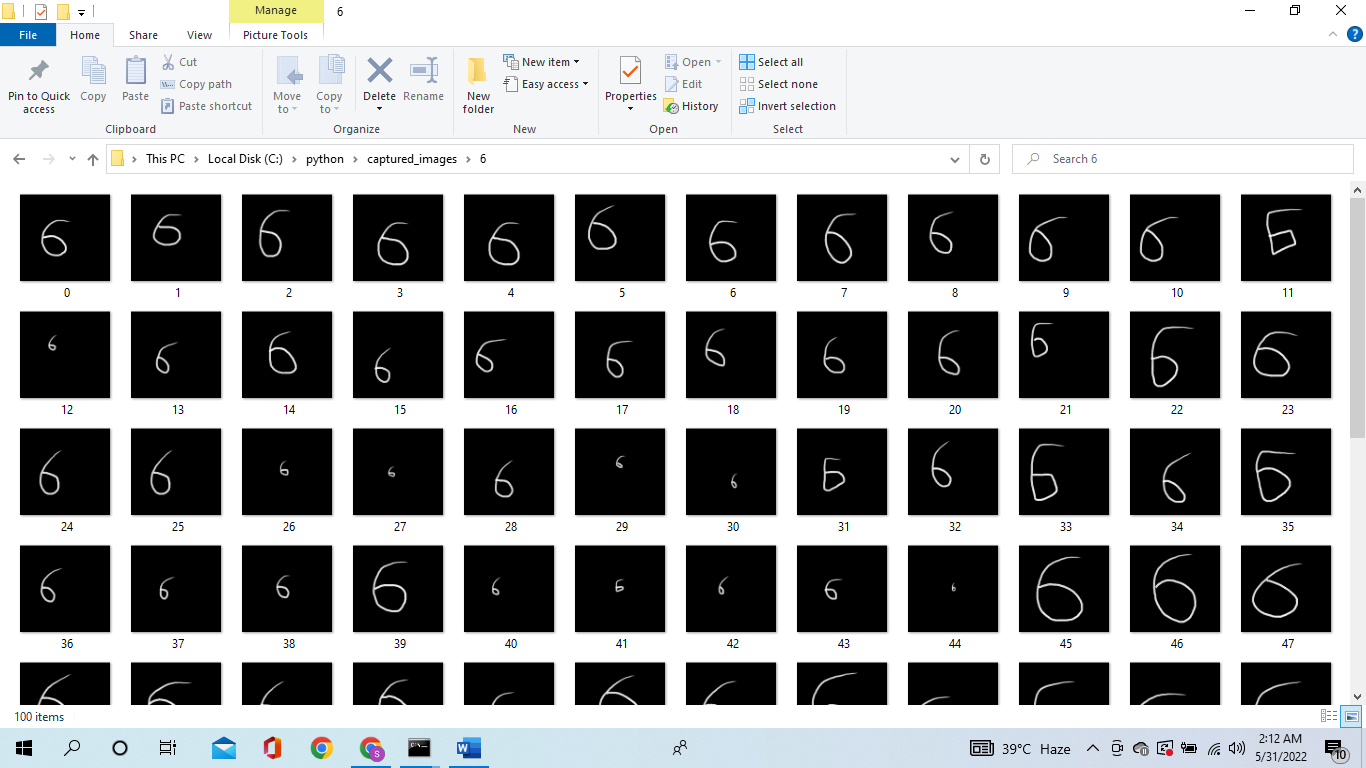
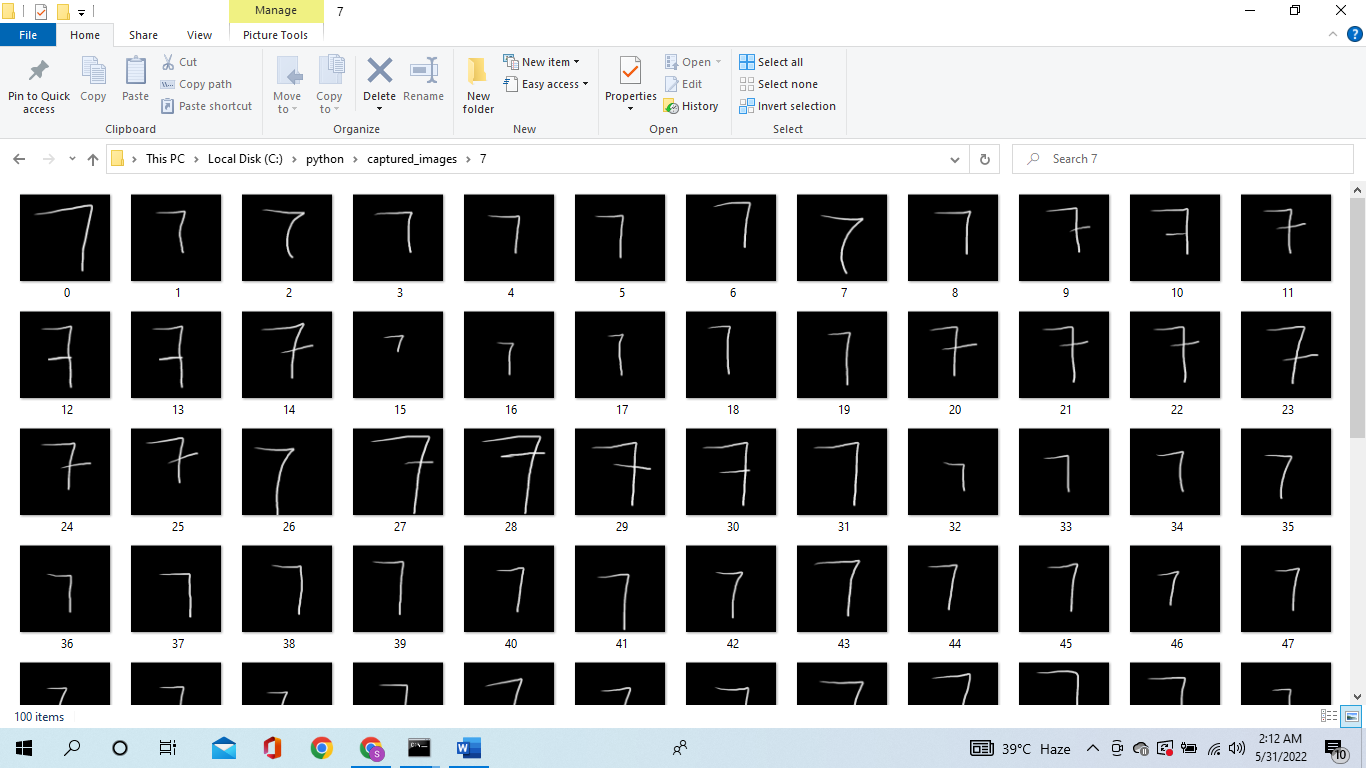


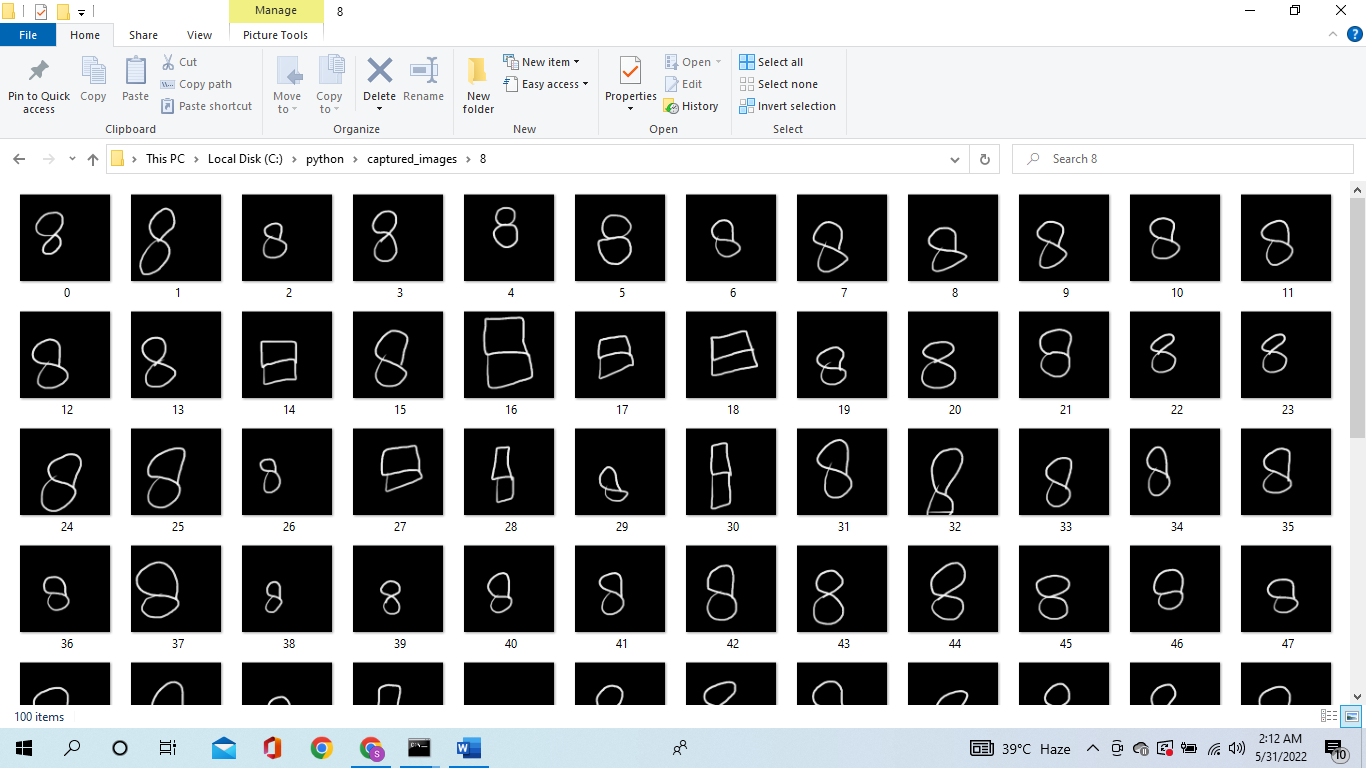
Fig13:Sample Images

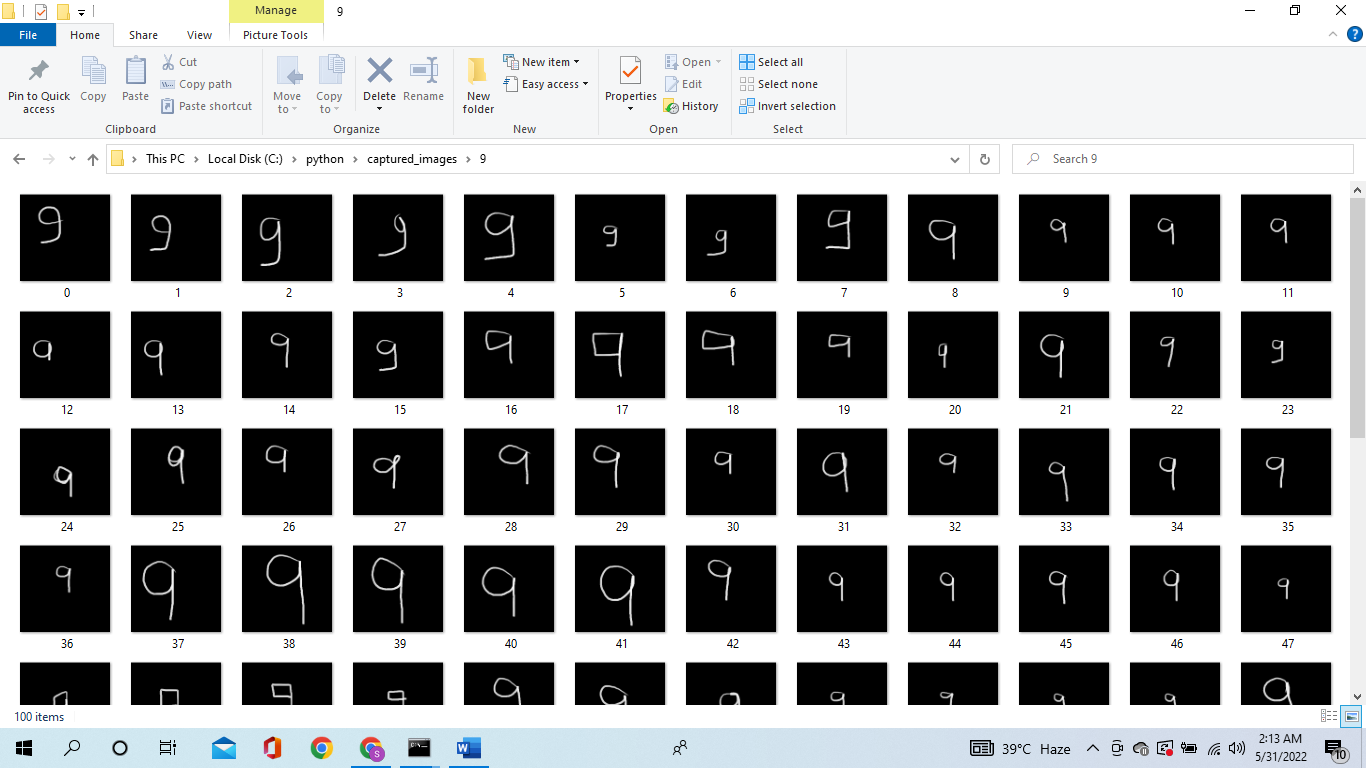






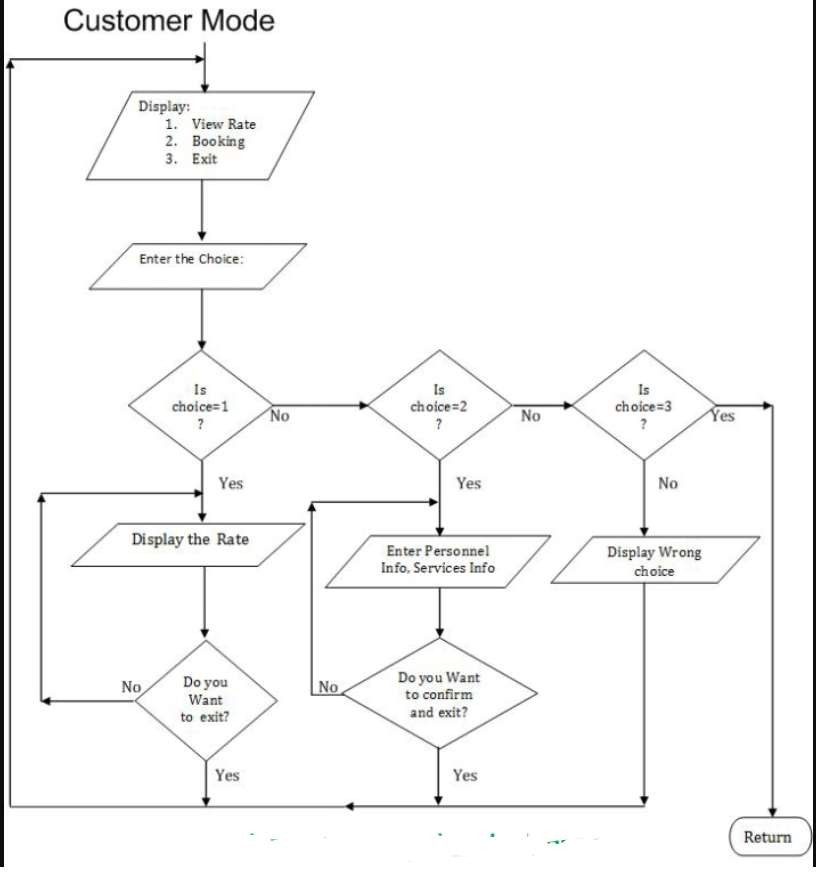






* 1. **Flow Chart**

Customer Flow Chart



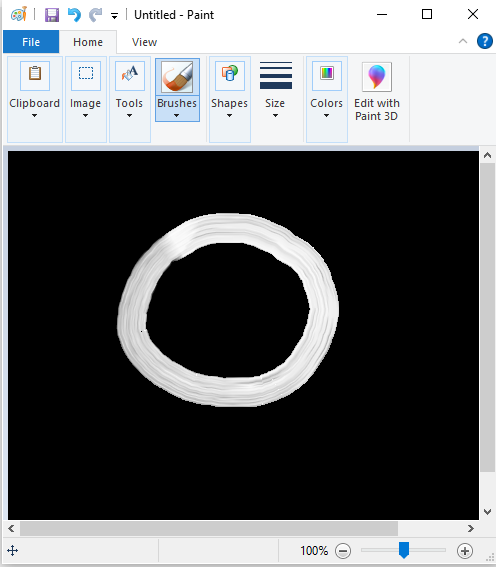
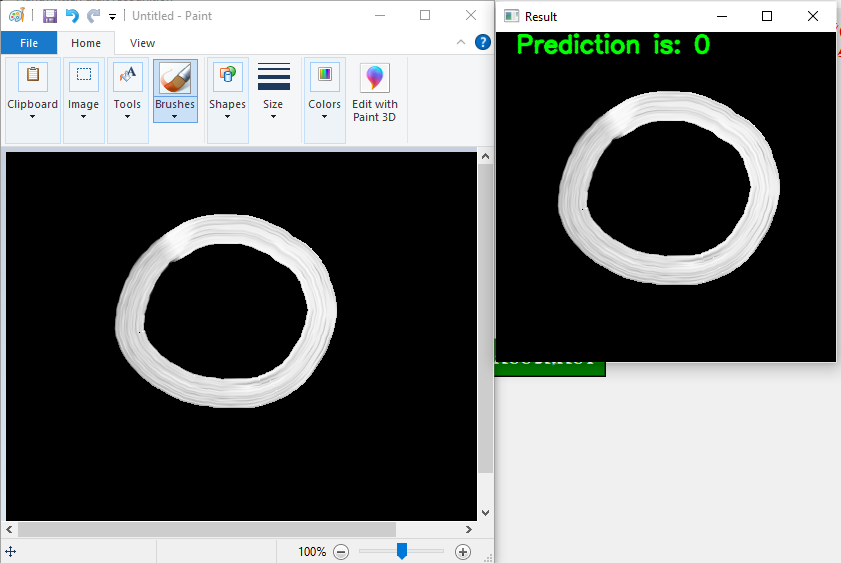
**CHAPTER 4**

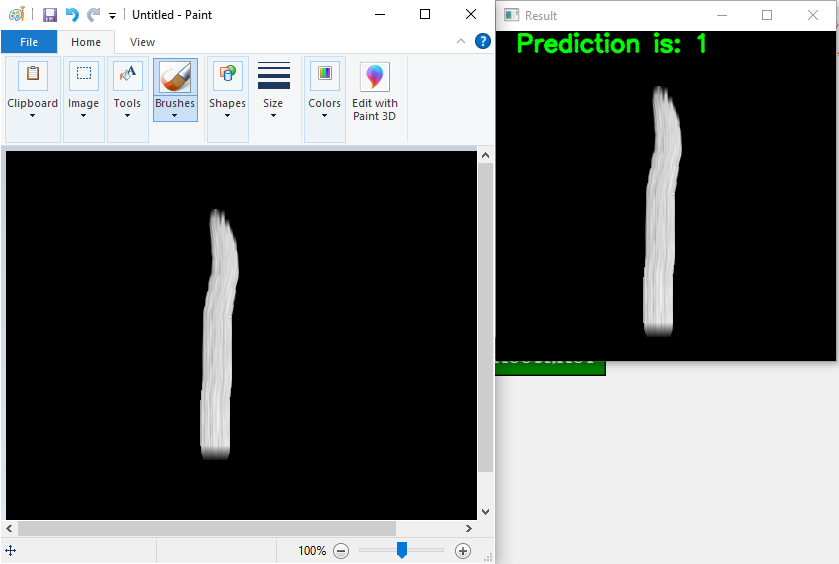
Form Design

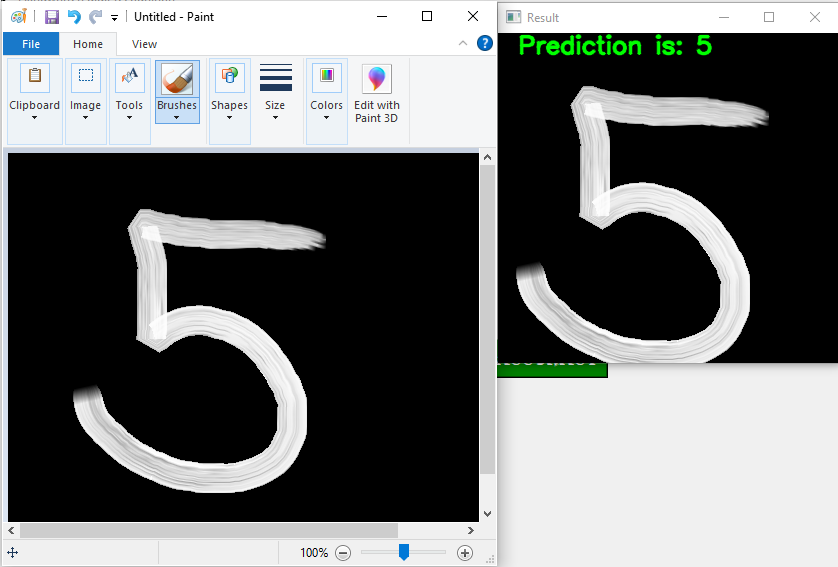
4.1) Input/Output Screenshots

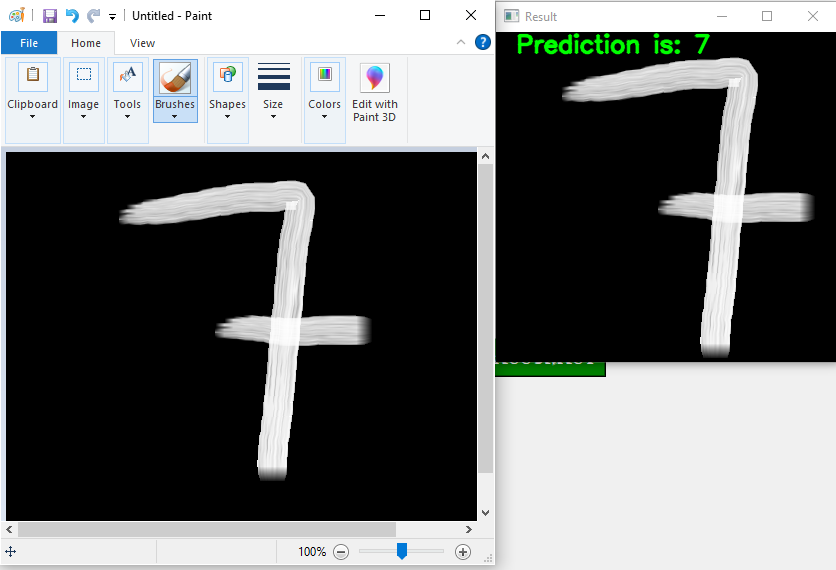
Home Page

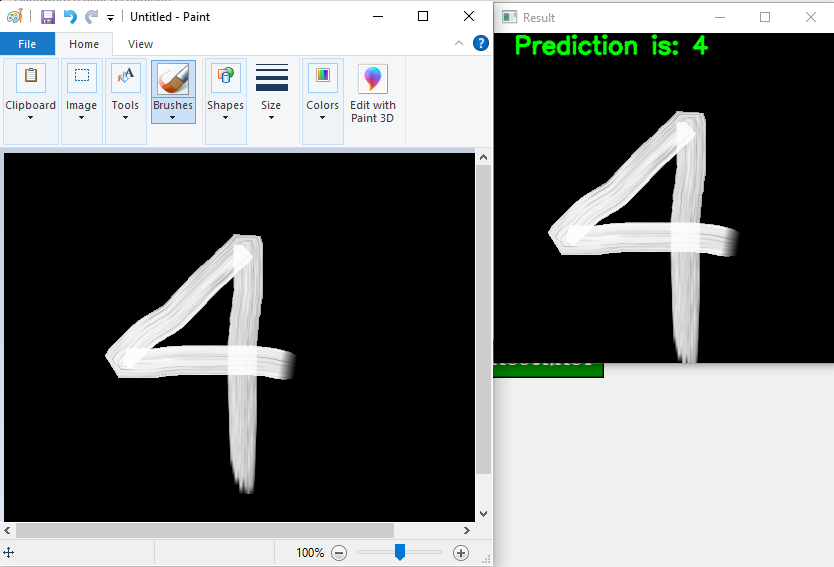
**

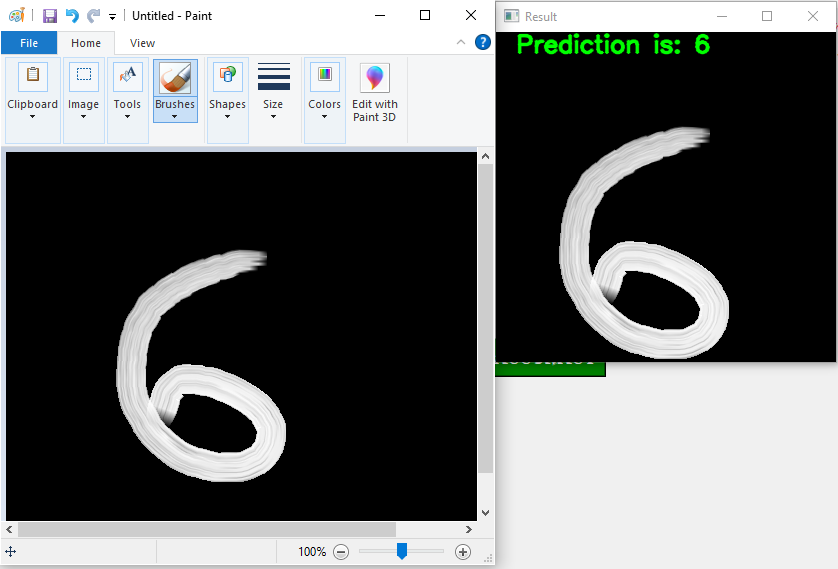


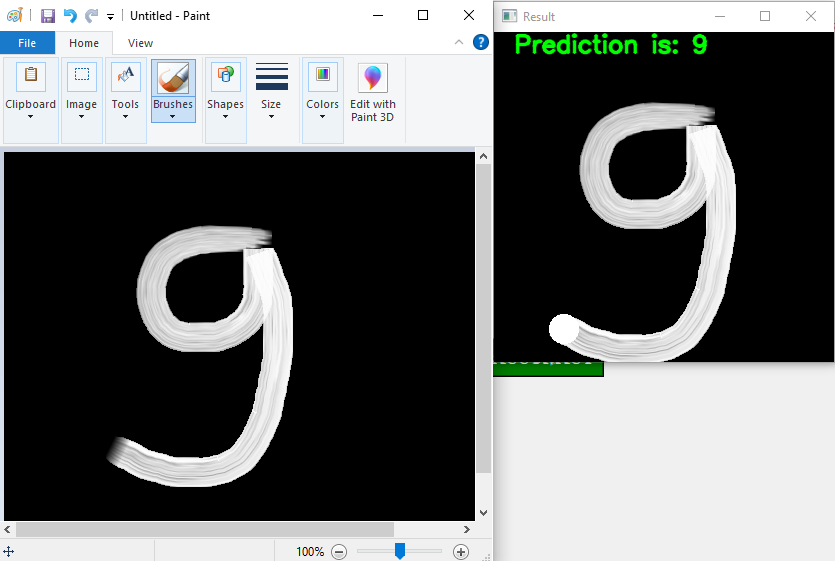


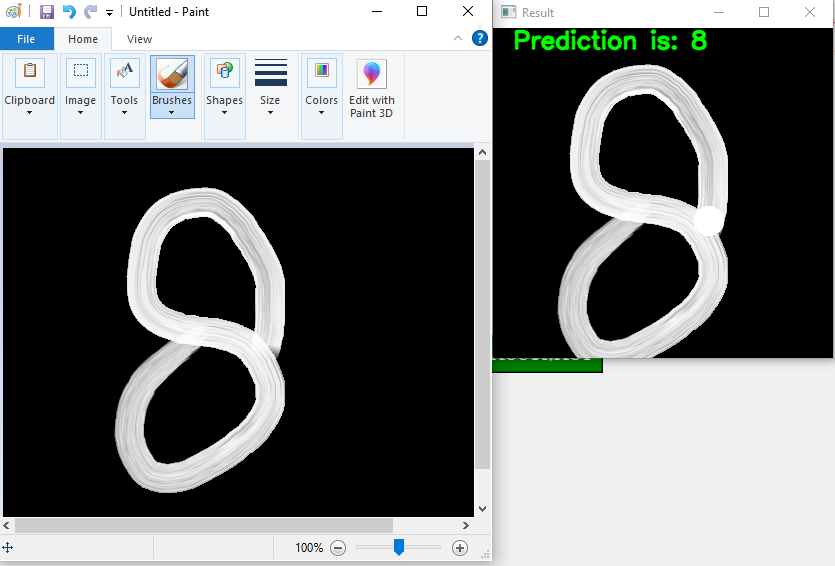


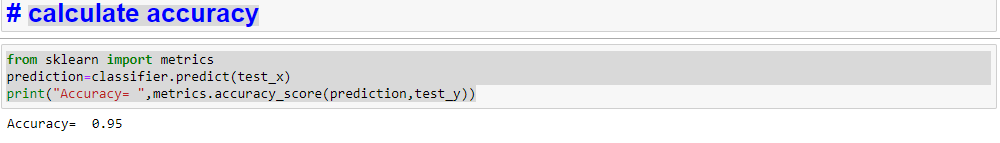


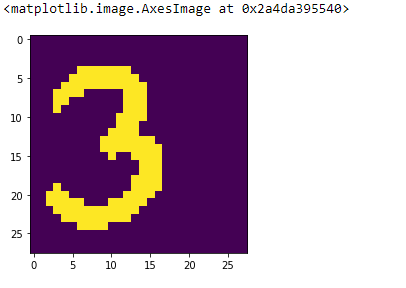


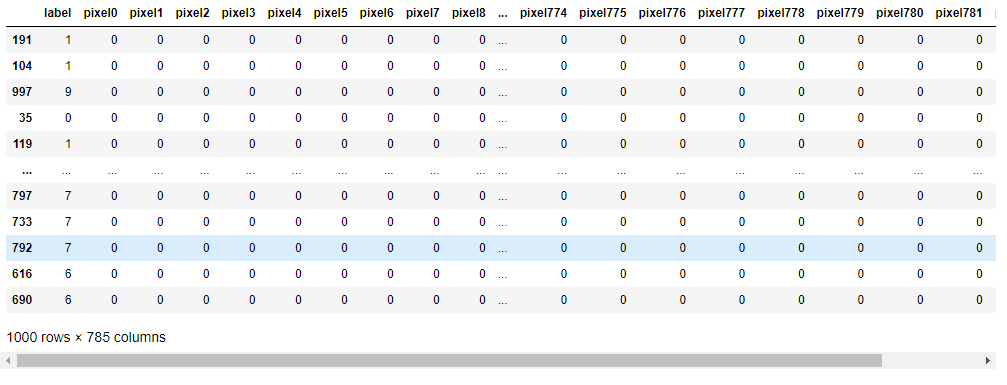


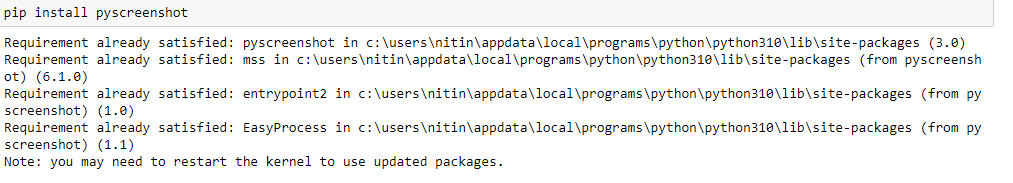












User Zone:

## CHAPTER 5

Coding

# **Screen capture**

def one\_time():

import pyscreenshot as ImageGrab

import time

images\_folder = "captured\_images/9/"

for i in range(0,100):

time.sleep(6)

im=ImageGrab.grab(bbox=(60,170,400,500))

print("saved.......",i)

im.save(images\_folder+str(i)+'.png')

print("clear screen now and redraw one........."

# **Generate dataset**

import cv2

import csv

import glob

header =["label"]

for i in range(0,784):

header.append("pixel"+str(i))

with open('dataset.csv', 'a') as f:

writer = csv.writer(f)

writer.writerow(header)

for label in range(10):

dirList = glob.glob("captured\_images/"+str(label)+"/\*.png")

for img\_path in dirList:

im= cv2.imread(img\_path)

im\_gray = cv2.cvtColor(im,cv2.COLOR\_BGR2GRAY)

im\_gray = cv2.GaussianBlur(im\_gray,(15,15),0)

roi= cv2.resize(im\_gray,(28,28),interpolation=cv2.INTER\_AREA)

data=[]

data.append(label)

rows,cols = roi.shape

##Add pixel one by one into data array

for i in range(rows):

for j in range(cols):

k =roi[i,j]

if k>100:

k=1

else:

k=0

data.append(k)

with open('dataset.csv','a') as f:

writer = csv.writer(f)

writer.writerow(data)

Load the dataset

import pandas as pd

from sklearn.utils import shuffle

data =pd.read\_csv('dataset.csv')

data=shuffle(data)

data

# **seperation of dependent and independent variable**

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

Y = data["label"]

**preview of one image using matplotlib**

%matplotlib inline

import matplotlib.pyplot as plt

import cv2

idx = 306

img = X.loc[idx].values.reshape(28,28)

print(Y[idx])

plt.imshow(img)

**Train-test split**

from sklearn.model\_selection import train\_test\_split

train\_x,test\_x,train\_y,test\_y = train\_test\_split(X,Y,test\_size=0.2)

import joblib

from sklearn.svm import SVC

classifier=SVC(kernel="linear",random\_state=6)

classifier.fit(train\_x,train\_y)

joblib.dump(classifier,"model/digit\_recognizer")

**calculate accuracy**

from sklearn import metrics

prediction=classifier.predict(test\_x)

print("Accuracy= ",metrics.accuracy\_score(prediction,test\_y))

**Predict digits**

import joblib

import cv2

import numpy as np #pip intall numpy

import time

import pyscreenshot as ImageGrab

model=joblib.load("model/digit\_recognizer")

images\_folder="img/"

while True:

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

img.save(images\_folder+"img.png")

im = cv2.imread(images\_folder+"img.png")

im\_gray =cv2.cvtColor(im,cv2.COLOR\_BGR2GRAY)

im\_gray =cv2.GaussianBlur(im\_gray, (15,15),0)

#Threshold the image

ret,im\_th = cv2.threshold(im\_gray,100,255,cv2.THRESH\_BINARY)

roi = cv2.resize(im\_th, (28,28), interpolation =cv2.INTER\_AREA)

rows,cols=roi.shape

X = []

## Add pixel one by one into data array

for i in range(rows):

for j in range(cols):

k = roi[i,j]

if k>100:

k=1

else:

k=0

X.append(k)

predictions =model.predict([X])

print("Prediction:",predictions[0])

cv2.putText(im, "Prediction is: "+str(predictions[0]), (20,20), 0, 0.8,(0,255,0),2,cv2.LINE\_AA)

cv2.startWindowThread()

cv2.namedWindow("Result")

cv2.imshow("Result",im)

cv2.waitKey(10000)

if cv2.waitKey(1)==13: #27 is for esc, 13 in for enter

break

cv2.destroyAllWindows()

**Interface**

import tkinter as tk

from tkinter import \*

from tkinter import messagebox

window=tk.Tk()

window.title("Handwritten digit recognition")

l1=tk.Label(window,text="Learn Digits",font=('Algerian',40),fg="red")

l1.place(x=530,y=0)

#t1=tk.Entry(window,width=20, border=5)

#t1.place(x=150, y=0)

def screen\_capture():

import pyscreenshot as ImageGrab

import time

import os

os.startfile("C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Accessories/Paint")

s1=t1.get()

os.chdir("E:/DS and ML/Untitled Folder/Untitled Folder/captured\_images")

os.mkdir(s1)

os.chdir("E:/DS and ML/Untitled Folder/Untitled Folder/")

images\_folder="captured\_images/"+s1+"/"

time.sleep(15)

for i in range(0,5):

time.sleep(8)

im=ImageGrab.grab(bbox=(60,170,400,550)) #x1,y1,x2,y2

print("saved......",i)

im.save(images\_folder+str(i)+'.png')

print("clear screen now and redraw now........")

messagebox.showinfo("Result","Capturing screen is completed!!")

b1=tk.Button(window,text="Open paint and capture the screen", font=('Algerian',15),bg="orange",fg="black",command=screen\_capture)

b1.place(x=75, y=130)

def generate\_dataset():

import cv2

import csv

import glob

header =["label"]

for i in range(0,784):

header.append("pixel"+str(i))

with open('dataset.csv', 'a') as f:

writer = csv.writer(f)

writer.writerow(header)

for label in range(10):

dirList = glob.glob("captured\_images/"+str(label)+"/\*.png")

for img\_path in dirList:

im= cv2.imread(img\_path)

im\_gray = cv2.cvtColor(im,cv2.COLOR\_BGR2GRAY)

im\_gray = cv2.GaussianBlur(im\_gray,(15,15), 0)

roi= cv2.resize(im\_gray,(28,28), interpolation=cv2.INTER\_AREA)

data=[]

data.append(label)

rows, cols = roi.shape

## Add pixel one by one into data array

for i in range(rows):

for j in range(cols):

k =roi[i,j]

if k>100:

k=1

else:

k=0

data.append(k)

with open('dataset.csv', 'a') as f:

writer = csv.writer(f)

writer.writerow(data)

messagebox.showinfo("Result","Generating dataset is completed!!")

b2=tk.Button(window,text="Generate dataset", font=('Algerian',15),bg="pink",fg="blue",command=generate\_dataset)

b2.place(x=75, y=230)

def train\_save\_accuracy():

import pandas as pd

from sklearn.utils import shuffle

data =pd.read\_csv('dataset.csv')

data=shuffle(data)

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

Y= data["label"]

from sklearn.model\_selection import train\_test\_split

train\_x,test\_x,train\_y,test\_y = train\_test\_split(X,Y, test\_size = 0.2)

import joblib

from sklearn.svm import SVC

classifier=SVC(kernel="linear", random\_state=6)

classifier.fit(train\_x,train\_y)

joblib.dump(classifier, "model/digit\_recognizer")

from sklearn import metrics

prediction=classifier.predict(test\_x)

acc=metrics.accuracy\_score(prediction, test\_y)

messagebox.showinfo("Result",f"Your accuracy is {acc}")

b3=tk.Button(window,text="Train the model, save it and calculate accuracy", font=('Algerian',15),bg="green",fg="white",command=train\_save\_accuracy)

b3.place(x=75, y=330)

def prediction():

import joblib

import cv2

import numpy as np #pip install numpy

import time

import pyscreenshot as ImageGrab

import os

os.startfile("C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Accessories/Paint")

model=joblib.load("model/digit\_recognizer")

images\_folder="img/"

time.sleep(15)

while True:

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

img.save(images\_folder+"img.png")

im = cv2.imread(images\_folder+"img.png")

im\_gray = cv2.cvtColor(im,cv2.COLOR\_BGR2GRAY)

im\_gray =cv2.GaussianBlur(im\_gray, (15,15), 0)

#Threshold the image

ret, im\_th = cv2.threshold(im\_gray,100, 255, cv2.THRESH\_BINARY)

roi = cv2.resize(im\_th, (28,28), interpolation =cv2.INTER\_AREA)

rows,cols=roi.shape

X = []

## Add pixel one by one into data array

for i in range(rows):

for j in range(cols):

k = roi[i,j]

if k>100:

k=1

else:

k=0

X.append(k)

predictions =model.predict([X])

print("Prediction:",predictions[0])

cv2.putText(im, "Prediction is: "+str(predictions[0]), (20,20), 0, 0.8,(0,255,0),2,cv2.LINE\_AA)

cv2.startWindowThread()

cv2.namedWindow("Result")

cv2.imshow("Result",im)

cv2.waitKey(10000)

if cv2.waitKey(1)==13: #27 is the ascii value of esc, 13 is the ascii value of enter

break

cv2.destroyAllWindows()

b4=tk.Button(window,text="Live prediction", font=('Algerian',15),bg="white",fg="red",command=prediction)

b4.place(x=75, y=430)

window.geometry("600x300")

window.mainloop()

## CHAPTER 6

Test Cases

* **Black box Testing**: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

* **White box Testing**: is the testing process in which tester can perform testing on an application with having internal structural knowledge.

Usually The Developers are involved in white box testing.

* **Gray Box Testing**: is the process in which the combination of black box and white box techniques are used.
  + **Smoke Testing**: is the process of initial testing in which tester looks for the availability of all the functionality of the application in order to perform detailed testing on them. (Main check is for available forms)
  + **Sanity Testing:** is a type of testing that is conducted on an application initially to check for the proper behavior of an application that is to check all the functionality are available before the detailed testing is conducted by on them.
  + **Regression Testing:** is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.
  + **Re-Testing:** is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.
  + **Static Testing:** is the testing, which is performed on an application when it is not been executed. ex: GUI, Document Testing
  + **Dynamic Testing:** is the testing which is performed on an application when it is being executed. ex: Functional testing.
  + **Alpha Testing:** it is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.
  + **Beta-Testing:** it is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.
  + **Monkey Testing:** is the process in which abnormal operations, beyond capacity operations are done on the application to check the stability of it in spite of the users abnormal behavior.
  + **Compatibility testing:** it is the testing process in which usually the products are tested on the environments with different combinations of databases (application servers, browsers…etc) In order to check how far the product is compatible with all these environments platform combination.
  + **Installation Testing:** it is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.
  + **Adhoc Testing:** Adhoc Testing is the process of testing in which unlike the formal testing where in test case document is used, with out that test case document testing can be done of an application, to cover that testing of the future which are not covered in that test case document. Also it is intended to perform GUI testing which may involve the cosmotic issues.

**REFERENCES**

1. LeCun, Yann, et al. "Comparison of learning algorithms for handwritten digit recognition." *International conference on artificial neural networks*. Vol. 60. 1995.
2. Torgesen, J.K. and Houck, D.G., 1980. Processing deficiencies of learning-disabled children who perform poorly on the Digit Span Test. *Journal of Educational Psychology*, *72*(2), p.141.
3. Qiao, Junfei, et al. "An adaptive deep Q-learning strategy for handwritten digit recognition." *Neural Networks* 107 (2018): 61-71.
4. Qiao, Junfei, Gongming Wang, Wenjing Li, and Min Chen. "An adaptive deep Q-learning strategy for handwritten digit recognition." *Neural Networks* 107 (2018): 61-71.
5. Diehl, Peter U., and Matthew Cook. "Unsupervised learning of digit recognition using spike-timing-dependent plasticity." *Frontiers in computational neuroscience* 9 (2015): 99.
6. LeCun, Yann, Lawrence D. Jackel, Léon Bottou, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon et al. "Learning algorithms for classification: A comparison on handwritten digit recognition." *Neural networks: the statistical mechanics perspective* 261, no. 276 (1995): 2
7. LeCun Y, Jackel LD, Bottou L, Cortes C, Denker JS, Drucker H, Guyon I, Muller UA, Sackinger E, Simard P, Vapnik V. Learning algorithms for classification: A comparison on handwritten digit recognition. Neural networks: the statistical mechanics perspective. 1995 Jan;261(276):2
8. Hamsher, K. D. S., Benton, A. L., & Digre, K. (1980). Serial digit learning: Normative and clinical aspects. *Journal of Clinical and Experimental Neuropsychology*, *2*(1), 39-50
9. Ruiter M, Loyens S, Paas F. Watch your step children! Learning two-digit numbers through mirror-based observation of self-initiated body movements. Educational Psychology Review. 2015 Sep;27(3):457-74
10. Torgesen JK, Houck DG. Processing deficiencies of learning-disabled children who perform poorly on the Digit Span Test. Journal of Educational Psychology. 1980 Apr;72(2):141