Team ID	PNT2022TMID41253
Project Name	A Novel Method For Handwritten Digit Recognition System

SPRINT – 4

predict.py

```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
import cv2
def predict(image):
 model=load_model('./Module/model/mnist.h5')
 image = cv2.imread(image)
 grey = cv2.cvtColor(image.copy(), cv2.COLOR_BGR2GRAY)
 ret, thresh = cv2.threshold(grey.copy(), 75, 255, cv2.THRESH_BINARY_INV)
 contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
 preprocessed_digits = []
 if not contours:
    return False
 for c in contours:
```

```
    x,y,w,h = cv2.boundingRect(c)
    cv2.rectangle(image, (x,y), (x+w, y+h), color=(0, 255, 0), thickness=2)
    digit = thresh[y:y+h, x:x+w]
    resized_digit = cv2.resize(digit, (18,18))
    padded_digit = np.pad(resized_digit, ((5,5),(5,5)), "constant", constant_values=0)
    preprocessed_digits.append(padded_digit)
 for digit in preprocessed_digits:
    prediction = model.predict(digit.reshape(1, 28, 28, 1))
    best= np.argmax(prediction)
 return best
```

Uploadphoto.html

```
{% extends 'base.html'%}
{%block title%}
A Novel Method For Handwritten Digit Recognization System
{%endblock%}
body{
        overflow:hidden;
}
{%block style%}
{%endblock%}
{%block uploadform%}
<div id="upload-sector">
        <div style="display: flex; align-items: center;">
```

```
<div class="uploadform">
    <form method="POST" action="{{url_for('uploadpage')}}" enctype="multipart/form-data">

        <input type="hidden" name="type" value="Upload">
        <div class="custom-file">
        <label class="form-label">UPLOAD IMAGE</label><br>
        <input type="file" accept="image/*" name="image" required class="form-control-sm upinputbox" id="formFileLg">
        </div>
        <br><br>
        <input type="submit" value="Submit" class="btn btn-success float-left select ">
    </form>
</div>
<div class="line" id="bline"></div>
</div>
<div id="resultbox">
    <div id="numbers">
        <span id="result">Result</span>
        <div class="numberbox" id="1"><span id="one" class="number">{{bodyele}}</span></div>
    </div>
</div>
<div id="up-image">
</div>
```

```
</div>
{%endblock%}
{{msg}}
```

app.py

```
@app.route('/uploadpage',methods = ['POST','GET'])
def uploadpage():
    if not session['user']:
        return render_template('Authentication.html', msg="Login First")
    if request.method == "POST":
        if request.form["type"]=="Upload":
            image = request.files['image']
            filesrc = './uploads/.predict.jpg'
            image.save(filesrc)
            best = predict(filesrc)
            if best == False:
                return render_template('debug.html',msg="Cannot able to Recognize, reupload with
        correct image")
            user = session.get('user',None)
            user[0] = list(user[0])
            user[0][4] = ','+str(best)+user[0][4]
            db.updateHistory(user[0])
            return render_template('Uploadphoto.html', bodyele = int(best))
    return render_template('Uploadphoto.html', bodyele = '-')
```