

Project Development Phase

Sprint-3

Date	11 November 2022
Team ID	PNT2022TMID40411
Project Name	A Novel Method for Handwritten Digit Recognition System

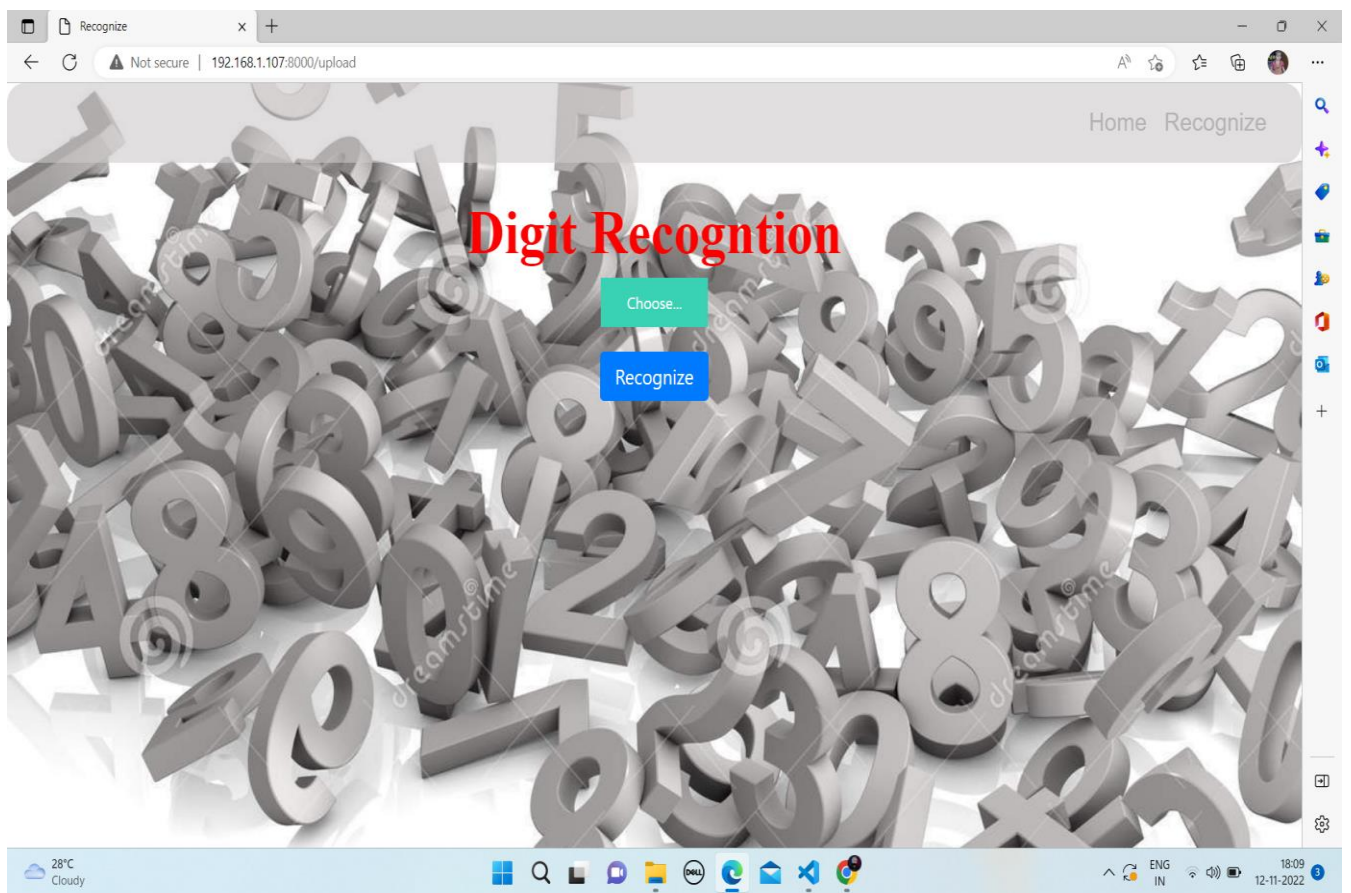
Home Recognize

Handwritten Recognition System

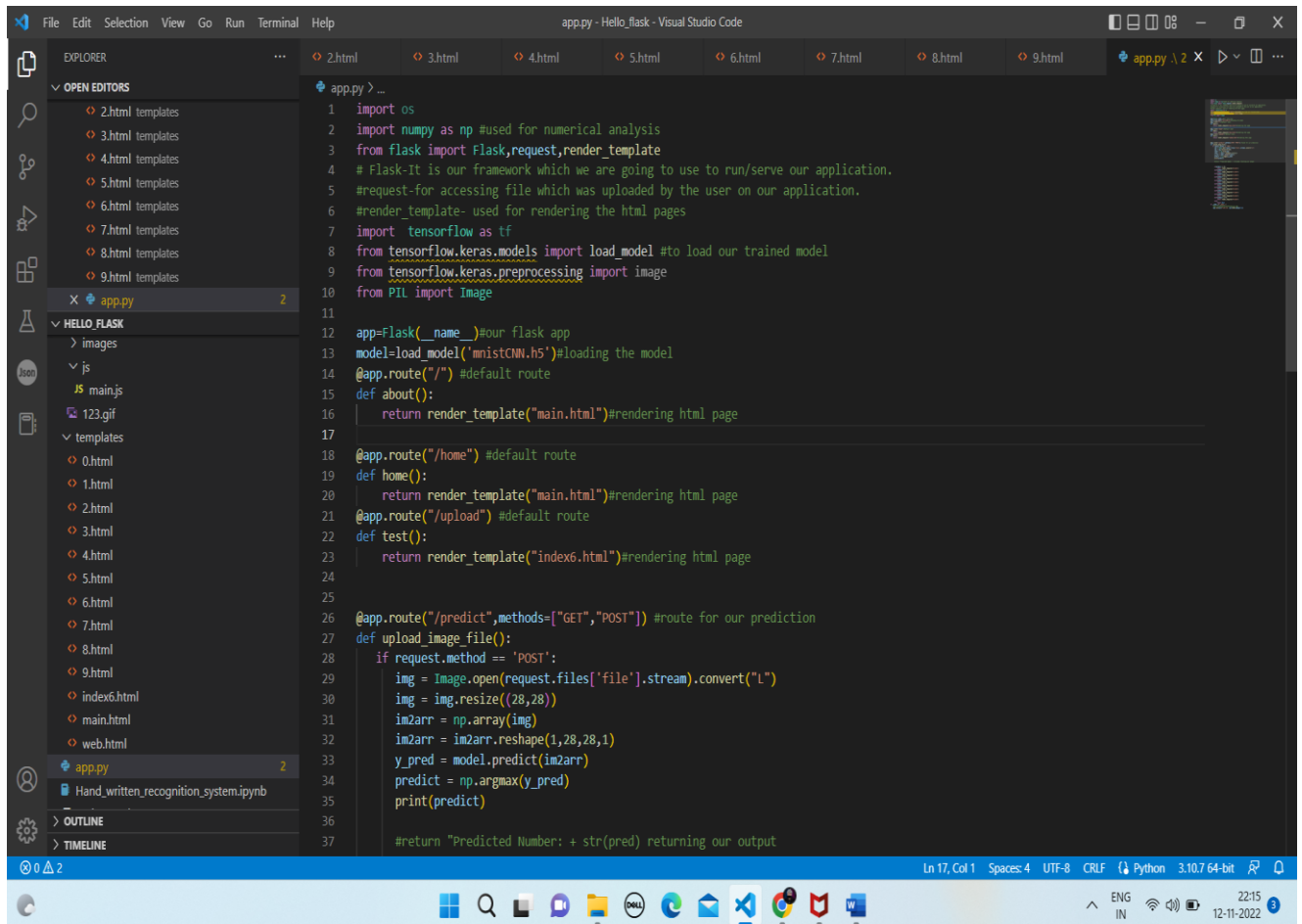
Handwritten Text Recognition is a technology that is much needed in this world as of today. This digit Recognition system is used to recognize the digits from different sources like emails, bank cheque, papers, images, etc. Before proper implementation of this technology we have relied on writing texts with our own hands which can result in errors, It's difficult to store and access physical data with efficiency. The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset. Here we will rising artificial neural networks/convolution neural network

28°C Cloudy

ENG IN 18:08 12-11-2022

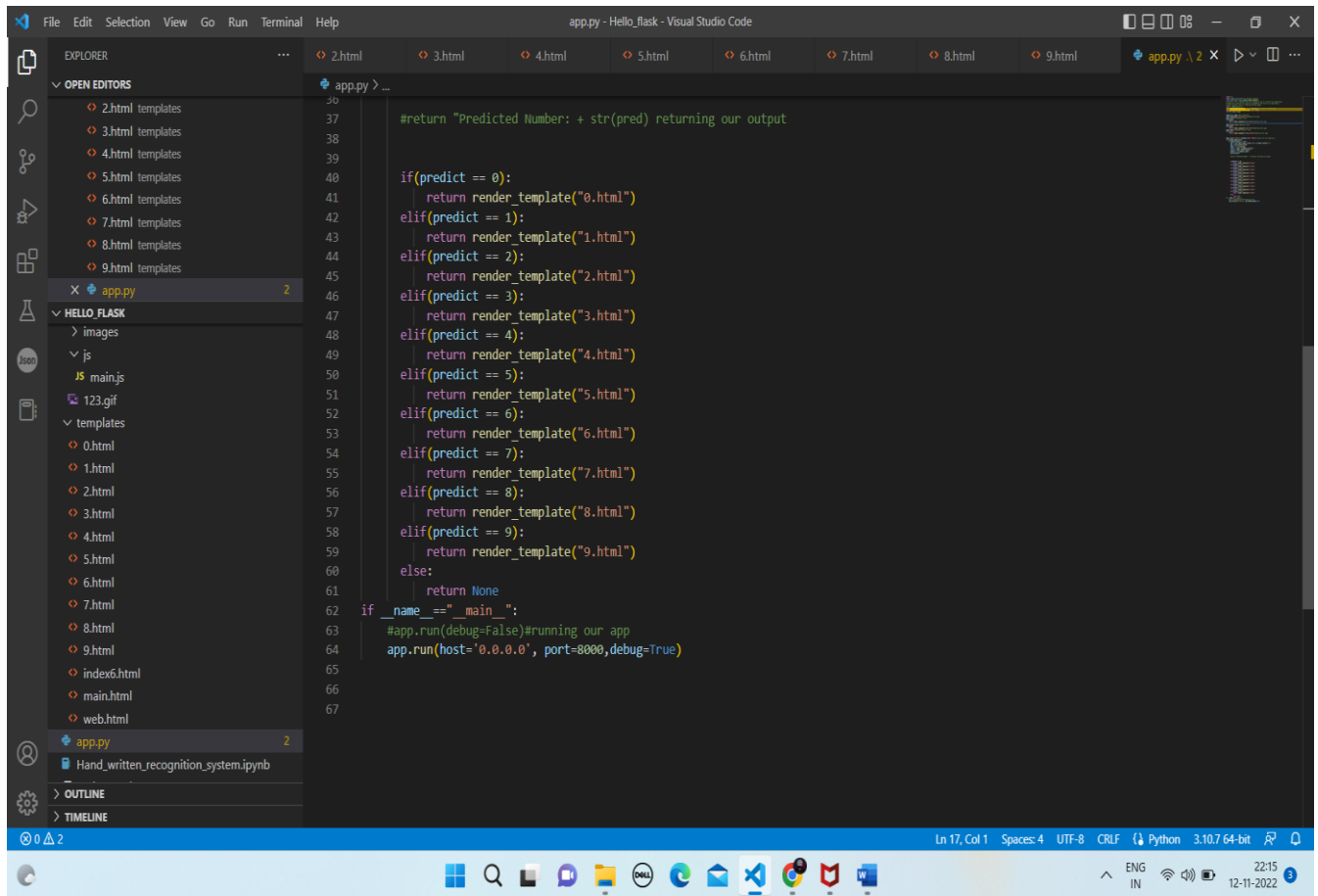


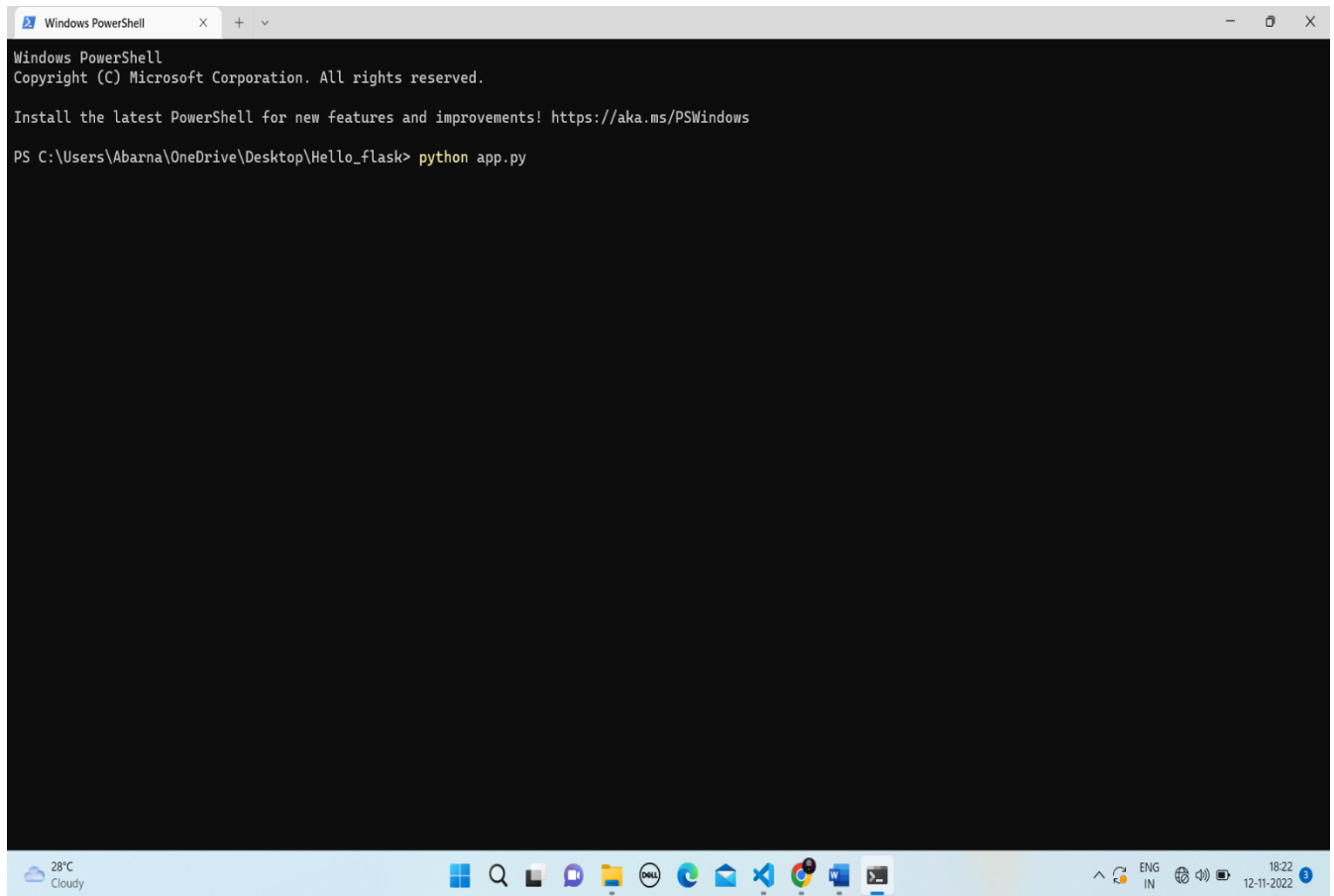
Python code(app.py)



The screenshot shows the Visual Studio Code interface with a Python file named `app.py` open. The code is a Flask application that uses TensorFlow Keras to load a pre-trained MNIST model for handwritten digit recognition. The application has several routes: a default route, a home route, an upload route, and a prediction route. The prediction route uses the loaded model to predict the digit from an uploaded image.

```
1 import os
2 import numpy as np #used for numerical analysis
3 from flask import Flask,request,render_template
4 # Flask-It is our framework which we are going to use to run/serve our application.
5 #request-for accessing file which was uploaded by the user on our application.
6 #render_template- used for rendering the html pages
7 import tensorflow as tf
8 from tensorflow.keras.models import load_model #to load our trained model
9 from tensorflow.keras.preprocessing import image
10 from PIL import Image
11
12 app=Flask(__name__)#our flask app
13 model=load_model('mnistCNN.h5')#loading the model
14 @app.route("/") #default route
15 def about():
16     return render_template("main.html")#rendering html page
17
18 @app.route("/home") #default route
19 def home():
20     return render_template("main.html")#rendering html page
21 @app.route("/upload") #default route
22 def test():
23     return render_template("index6.html")#rendering html page
24
25
26 @app.route("/predict",methods=["GET","POST"]) #route for our prediction
27 def upload_image_file():
28     if request.method == 'POST':
29         img = Image.open(request.files['file'].stream).convert("L")
30         img = img.resize((28,28))
31         im2arr = np.array(img)
32         im2arr = im2arr.reshape(1,28,28,1)
33         y_pred = model.predict(im2arr)
34         predict = np.argmax(y_pred)
35         print(predict)
36
37     #return "Predicted Number: + str(pred) returning our output
```





```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Abarna\OneDrive\Desktop\Hello_flask> python app.py
```

The screenshot shows a Windows PowerShell terminal window. The title bar at the top reads "Windows PowerShell" and includes standard window controls (minimize, maximize, close). The terminal content is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Abarna\OneDrive\Desktop\Hello_flask> python app.py
```

The command prompt shows the user is in the directory `C:\Users\Abarna\OneDrive\Desktop\Hello_flask` and has entered the command `python app.py`. The terminal background is black, and the text is white. At the bottom of the screen, the Windows taskbar is visible, showing the Start button, search icon, and several pinned application icons. The system tray on the right indicates the temperature is 28°C, the weather is Cloudy, and the date and time are 18:22 on 12-11-2022.

```
Windows PowerShell
2022-11-12 18:10:25.200148: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://192.168.1.107:8080
Press CTRL+C to quit
* Restarting with stat
2022-11-12 18:10:26.102986: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cuda64_110.dll'; dLError: cuda
rt64_110.dll not found
2022-11-12 18:10:26.103225: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine
.
2022-11-12 18:10:29.271041: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dLError: nvcuda.dll
not found
2022-11-12 18:10:29.271578: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-12 18:10:29.278613: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: DESKTOP-NA585NJ
2022-11-12 18:10:29.279279: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-NA585NJ
2022-11-12 18:10:29.280105: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debugger is active!
* Debugger PIN: 104-326-080
```

