Project Development Phase

Sprint 3

Date	18 November 2022
Team ID	PNT2022TMID42615
Project Name	Natural Disaster Intensity Analysis and Classification using Artificial Intelligence

Application Building:

1.Build HTML Pages:

We use HTML to create the font end part of the web page. Here we have created 3 HTML pages – home.html, intro.html, and upload.html. home.html displays the home page. Intro.html displays an introduction about the project. upload.html gives the emergency alert

2.Build Python Code:

Build the flask file 'app.py' which is a web framework written in python for server-side scripting.

Procedure for building the backend application:

1. The app starts running when the "__name__" constructor is called in main.

2. render_template is used to return HTML file.

3." GET" method is used to take input from the user

4." POST" method is used to display the output to the user.

Building an application:

1.Creating our flask application and loading our model by using load_model method.

2.Routing to the HTML page.

Here, the declared constructor is used to route to the HTML page created earlier. In this, the '/' URL is bound with the home.html function. Hence, when the home page of the webserver is opened in the browser, the HTML page is rendered. Whenever you enter the values from the HTML page the value can be retrieved used the POST method. Here, "home.html" is rendered when the home button is clicked on the UI. When "Camera" icon is clicked on the UI, predict function is executed. And the upload route is used for prediction and it contains all the codes which are used for predicting our results.

- The tasks involved are:
    - Grab the frames from the web cam
    - Loop over the frames from the video stream
    - Convert the image from BGR to RGB
    - Predicting our results
    - Displaying the result
    - Run the application

Grab the frames from the webcam

To recognize the type of disaster we have to capture the video stream. There are two ways we can capture an input video

1. Using in-built webcam
2. Using video file residing on the disk

We use the VideoCapture module in the OpenCV library to capture a live video. We create a VideoCapture object using the constructor provided in the module. The argument to the constructor can be either a device index or the path of the video file. A device index is a number used to identify the webcam and, in most cases, the value is 0. The object of the VideoStream module enables us to capture frame-by-frame video data.

Loop over the frames from the video stream: Let us grab the video frames from the video by looping over the frames and check if the frame was not grabbed, then we have reached the end of the stream.clone the output frame for showcasing an output.

Predicting the results: We then proceed to detect all type of disaster in the input image using model.predict function and the result is stored in result variable.

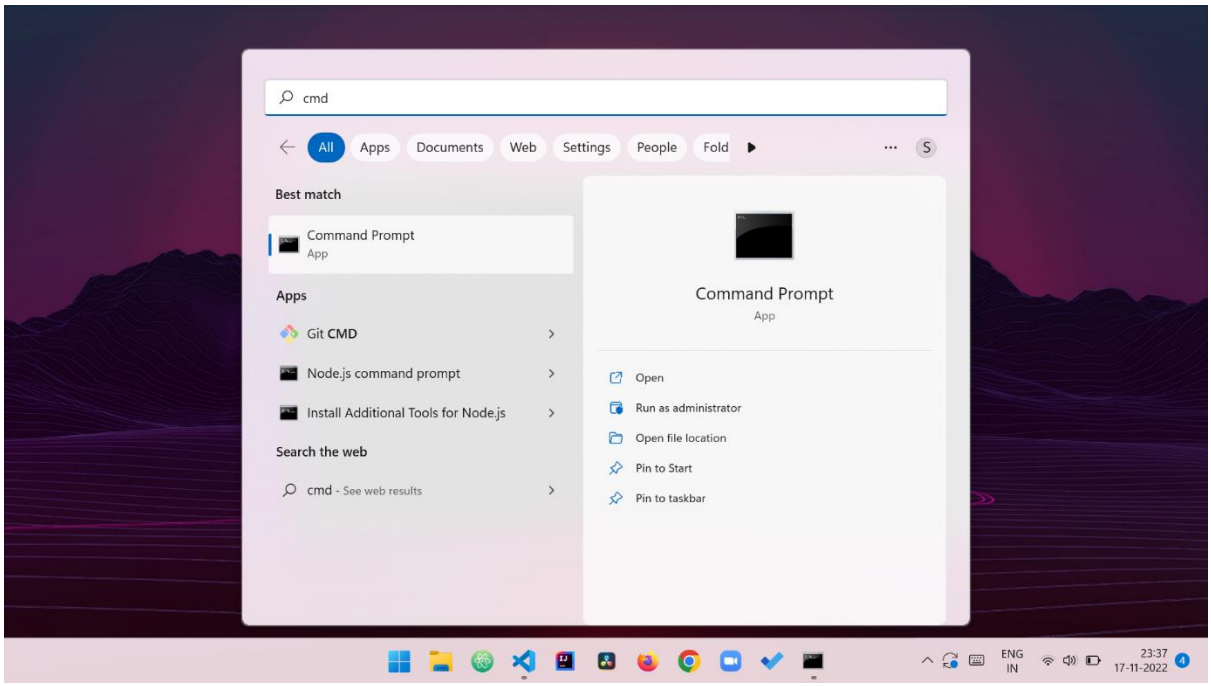
Displaying the result: After we recognise the type of disaster, we have to display the same on the live video stream for visualization. The cv2.imshow() function always takes two more functions to load and close an image. These two functions are cv2.waitKey() and cv2.destroyAllWindows(). Inside the cv2.waitKey() function, we can provide any value to close the image and continue with further lines of code.

Note: Press ‘q’ on the keyboard to close the webcam which is opened after we grab an input and an application recognise an input image.

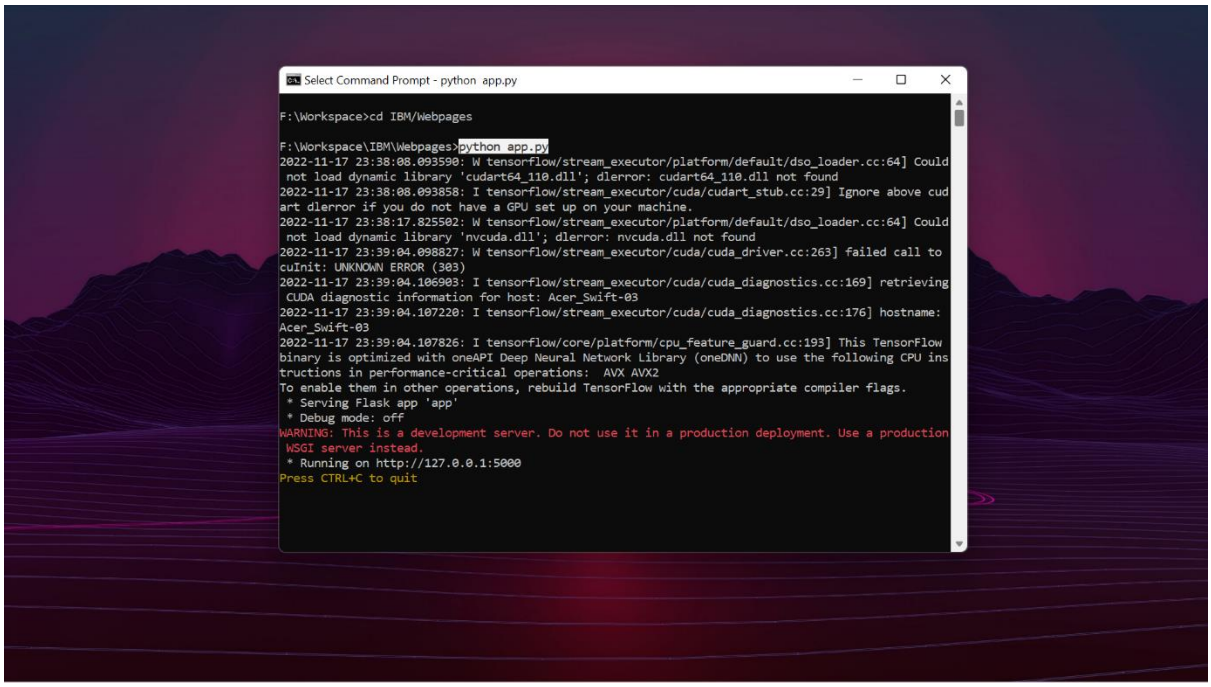
Finally, Run an application. This is used to run the application in a local host. The local host runs on port number 5000.

Running an application:

First of all, open an Anaconda prompt or Command Line from Start Menu. Then Navigate to the folder where ‘app.py’ exists.



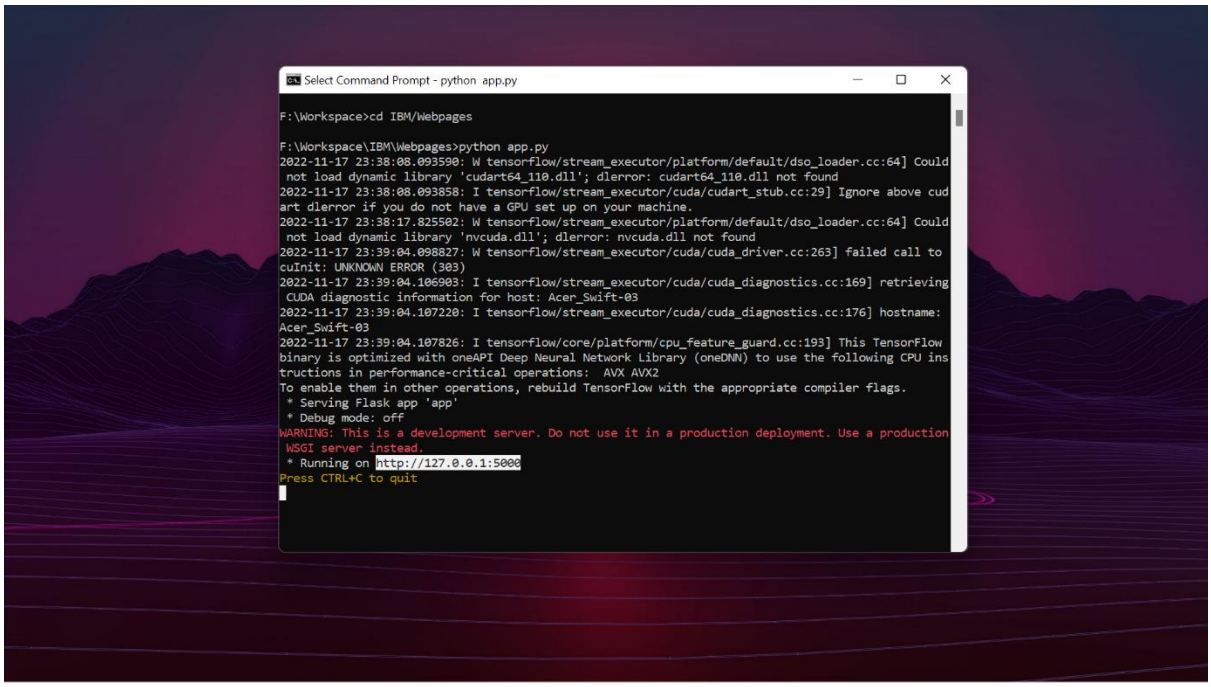
*Open CMD (in Windows) in Start Menu



*Navigate to the folder where ‘app.py’ exists

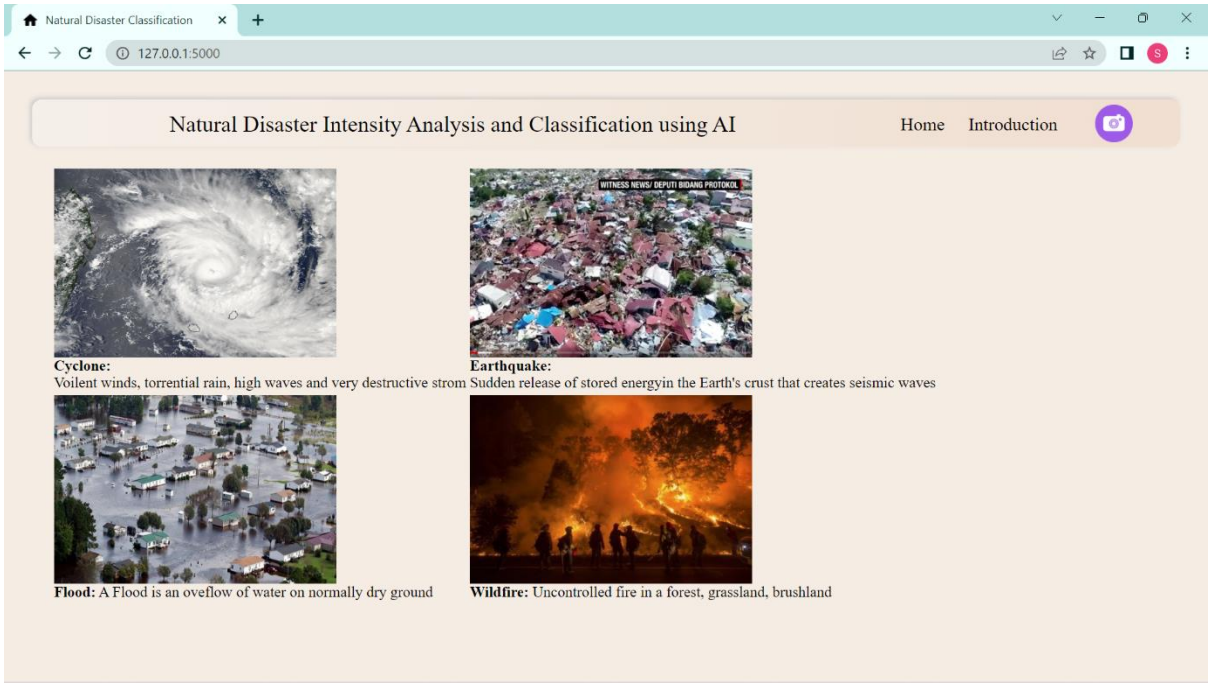
Run the command: ‘python app.py’

It will show the local host where our app is running on http://127.0.0.1:5000/



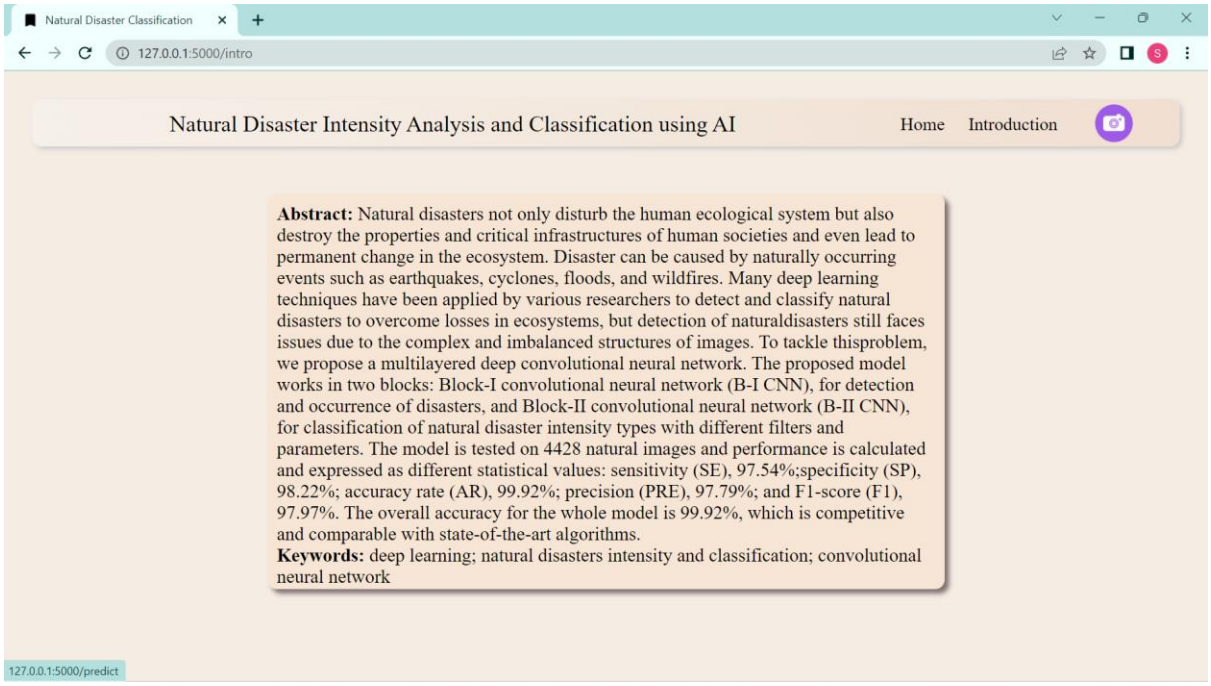
*Running the flask application on localhost:5000

home.html



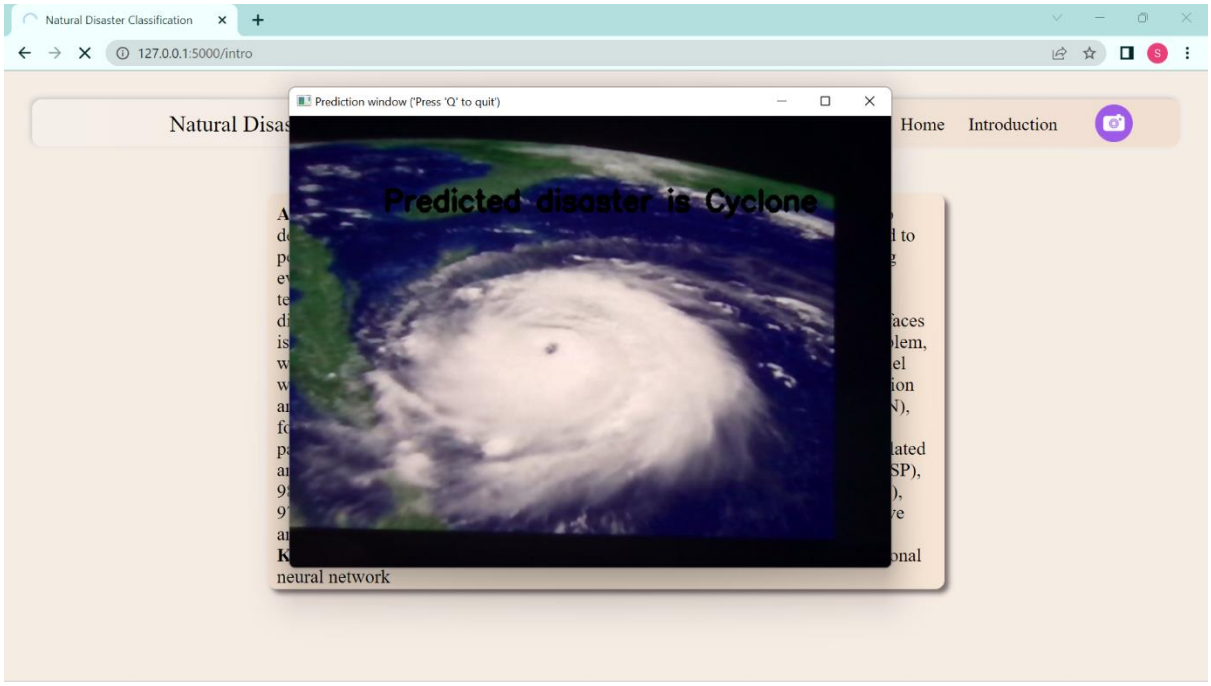
*home.html

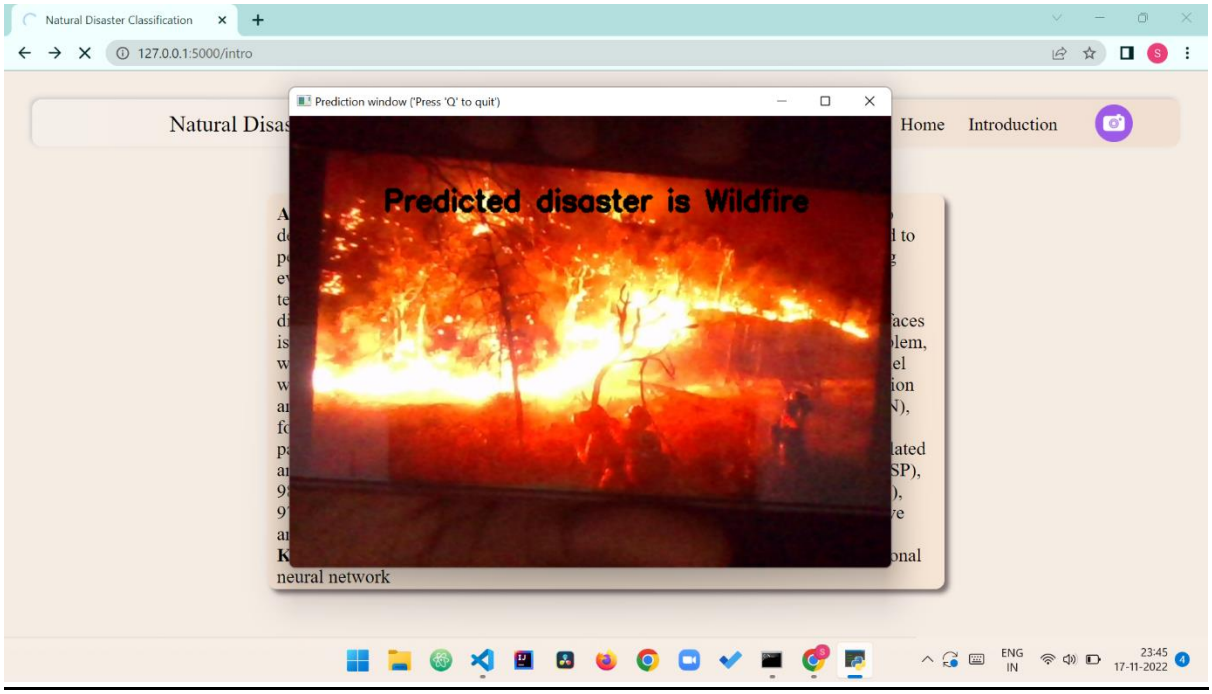
Intro.html



upload.html

Webcam:


Natural Disaster Classification
127.0.0.1:5000/intro
Prediction window ('Press 'Q' to quit')
Predicted disaster is Cyclone
Home
Introduction
neural network


Natural Disaster Classification
127.0.0.1:5000/intro
Prediction window ('Press 'Q' to quit')
Predicted disaster is Wildfire
Home
Introduction
neural network
ENG
IN
23:45
17-11-2022