NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Offered by **Smart Internz**

Karri Jyothi Sree (Team Leader)

Karri Sravanthi

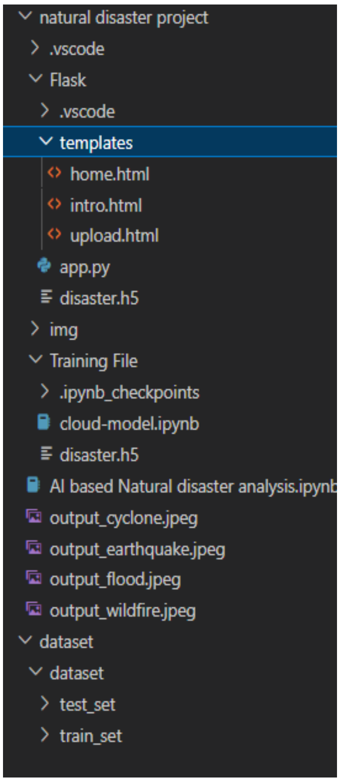
Kannedi Jahnavi

Team ID: LTVIP2023TMID08051

Introduction:

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, flood and wildfire. Many deep learning techniques have been applied by various researchers to detect and classify natural disaster to overcome losses in ecosystems. But detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem, we developed a multilayered classifies the natural disaster and tells the intensity of disaster of nature. The model uses ab integrated webcam to capture the video frame and the video frame is compared with the pre-trained model and the type disaster is identified and showcased on the OpenCV window.

Project Structure:

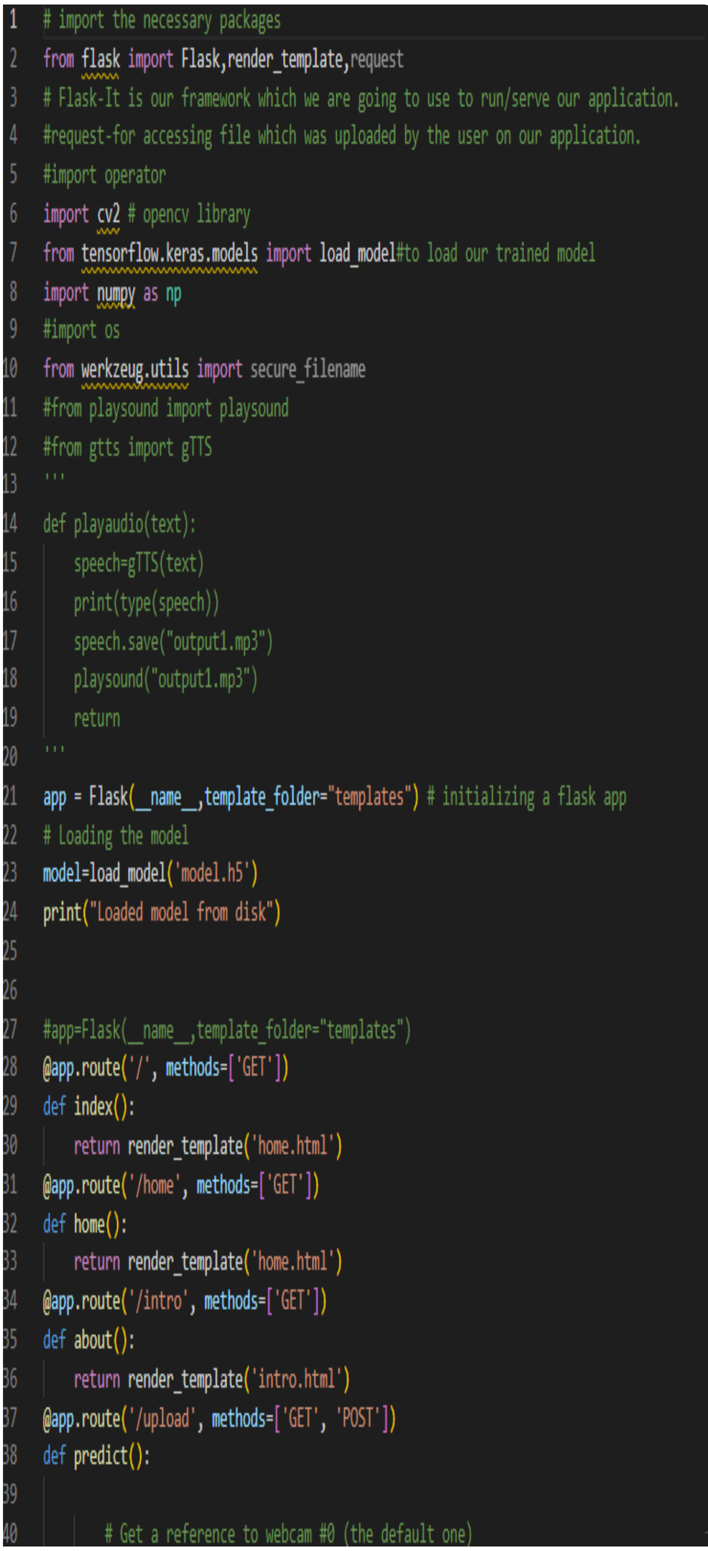


```
natural disaster project
  .vscode
  Flask
    .vscode
    templates
      home.html
      intro.html
      upload.html
    app.py
    disaster.h5
  img
  Training File
    .ipynb_checkpoints
    cloud-model.ipynb
    disaster.h5
  AI based Natural disaster analysis.ipynb
  output_cyclone.jpeg
  output_earthquake.jpeg
  output_flood.jpeg
  output_wildfire.jpeg
dataset
  dataset
    test_set
    train_set
```

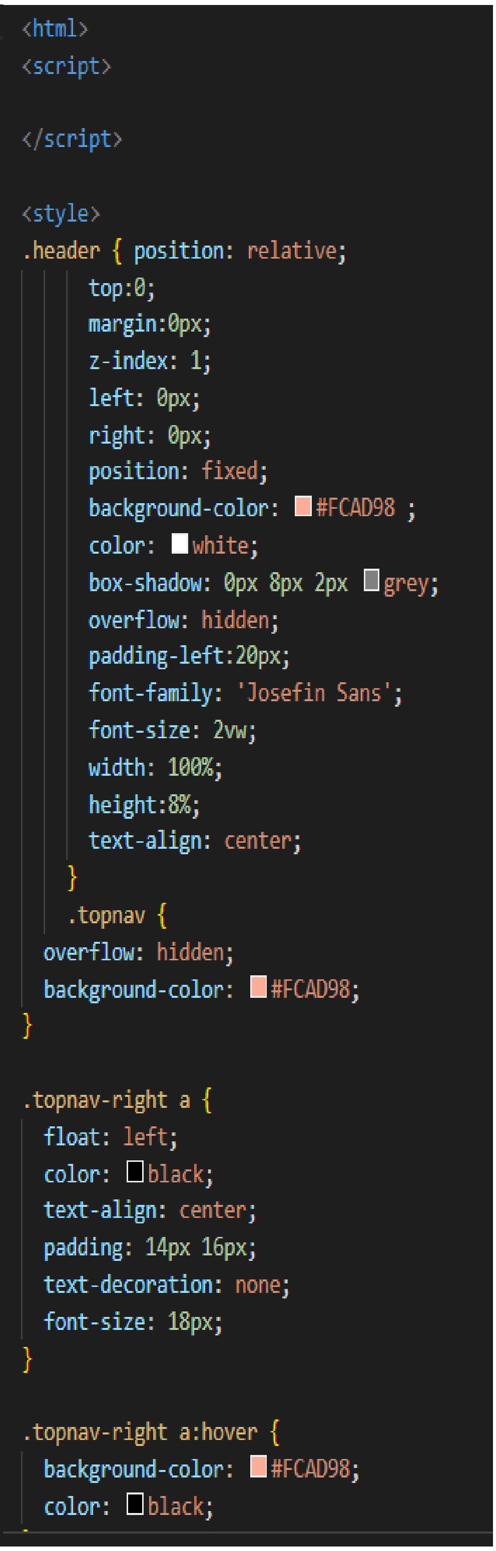
The Dataset folder consists of two different folders. One is train_set and other one is test_set, where train_set is used to train the model and test_set is used to test the model.

Templates folder contains all the HTML pages related to the project. The home.html, intro. Html, and upload.html are the three html pages. App.py is a python code written to work on the background of the project.

Source code:



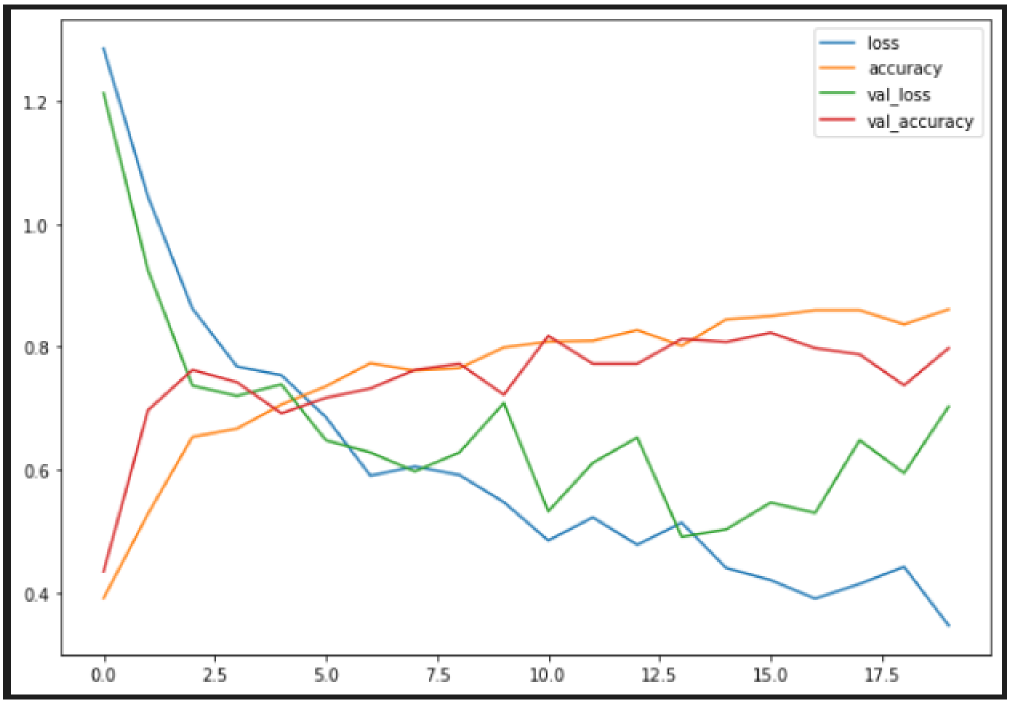
```python
# import the necessary packages
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
#import operator
import cv2 # opencv library
from tensorflow.keras.models import load_model#to load our trained model
import numpy as np
#import os
from werkzeug.utils import secure_filename
#from playsound import playsound
#from gtts import gTTS
'''

def playaudio(text):
    speech=gTTS(text)
    print(type(speech))
    speech.save("output1.mp3")
    playsound("output1.mp3")
    return
'''
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('model.h5')
print("Loaded model from disk")


#app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def home():
    return render_template('home.html')
@app.route('/intro', methods=['GET'])
def about():
    return render_template('intro.html')
@app.route('/upload', methods=['GET', 'POST'])
def predict():

        # Get a reference to webcam #0 (the default one)
```



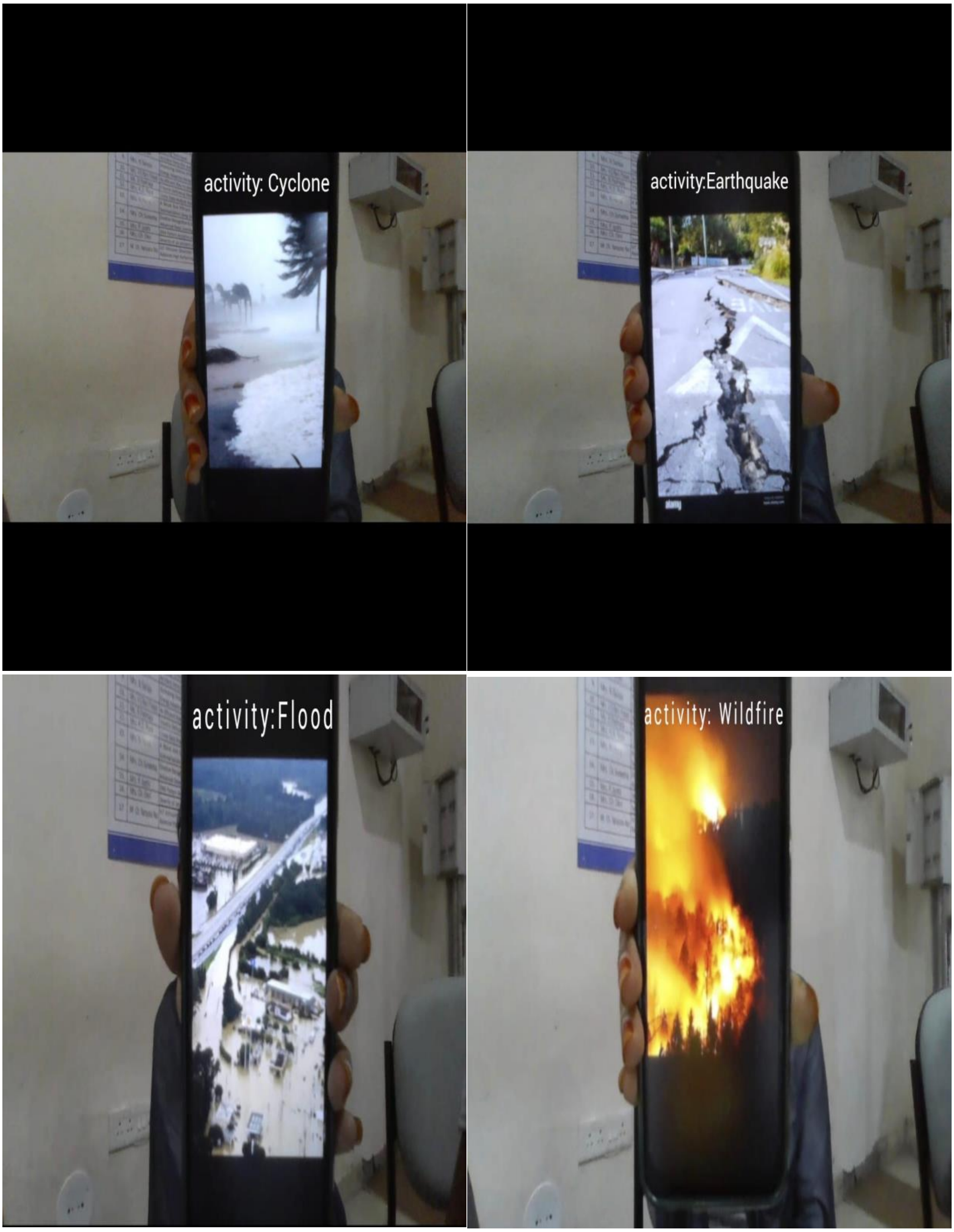
```html
<html>
<script>

</script>

<style>
.header { position: relative;
          top:0;
          margin:0px;
          z-index: 1;
          left: 0px;
          right: 0px;
          position: fixed;
          background-color: #FCAD98 ;
          color: white;
          box-shadow: 0px 8px 2px grey;
          overflow: hidden;
          padding-left:20px;
          font-family: 'Josefin Sans';
          font-size: 2vw;
          width: 100%;
          height:8%;
          text-align: center;
        }
        .topnav {
  overflow: hidden;
  background-color: #FCAD98;
}

.topnav-right a {
  float: left;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover {
  background-color: #FCAD98;
  color: black;
```

The above sample source code iis used to design the analysis and classification on the given datasets. The output pedicts the accurate results in identifying the disaster that is occurred using an image as an input.

Output:



The following images shows the output that is obtained at the time of execution. Each image contains different natural disaster's image. Each image is placed before the webcam. The webcam analysis the image and verifies its previously trained data and predicts the accurate output.



The figure shows the output predicted by the code using webcam when different natural disasters images are shown

Demo Link:

#Natural-disaster_intensity_analysis-and-classification_using_AIhttps://youtu.be/x9zD-gwixX4

GitHub Link:

https://github.com/Jyothi-119/Natural-disaster_intensity_analysis-and-classification_using_AI.git