

## **PROJECT DEVELOPMENT**

### **SPRINT-4**

Date	18 November 2022
Team ID	PNT2022TMID37822
Project Name	Natural Disaster Intensity Analysis and Classification using Artificial Intelligence
Team Members	Mohamed Asarudeen Shaik Abdullah Shamir ahmed Sudharsan Ameed Ibrahim

#### **DETECTION AND ANALYSIS OF DATA:**

After Testing and Training the model, data which given in dataset are analysed and visualised effectively to detect the Disaster Type. Using webcam, it can capture image or video stream of Disaster, to detect and analyse the type of Disaster.

```
print(x_train.class_indices)#checking the number of classes
```

```
print(x_test.class_indices)#checking the number of classes
```

```
from collections import Counter as c  
c(x_train.labels)
```

#### **DETECTION AND ANALYSIS OF DATA:**

After Testing and Training the model, data which given in dataset

are analysed and visualised effectively to detect the Disaster Type.

Using webcam, it can capture image or video stream of Disaster, to detect and analyse the type of Disaster.

```
print(x_train.class_indices)#checking the number of classes
```

```
print(x_test.class_indices)#checking the number of classes
```

```
from collections import Counter as c
c(x_train .labels)
```

## Saving the Model

```
In [ ]: # Save the model
classifier.save('disaster.h5')
```

```
In [ ]: model_json = classifier.to_json()
with open("model-bw.json", "w") as json_file:
    json_file.write(model_json)
```

### CREATING app.py:

```
10
11 # import the necessary packages
12 from flask import Flask,render_template,request
13 # Flask-it is our framework which we are going to use to run/serve our application.
14 #request-for accessing file which was uploaded by the user on our application.
15 #import operator
16 import cv2 # opencv library
17 from tensorflow.keras.models import load_model#to load our trained model
18 import numpy as np
19 #import os
20 from werkzeug.utils import secure_filename
21 #from playsound import playsound
22 #from gtts import gTTS
23 ...
24 def playaudio(text):
25     speech=gTTS(text)
26     print(type(speech))
27     speech.save("output1.mp3")
28     playsound("output1.mp3")
29     return
30 ...
31 app = Flask(__name__,template_folder="templates") # initializing a flask app
32 # loading the model
33 model=load_model(r'C:\Users\user\Desktop\IBM\Flask\templates\disaster.h5')
34 print("Loaded model from disk")
35
36
37 app=Flask(__name__,template_folder="templates")
38 @app.route('/', methods=['GET'])
39 def index():
40     return render_template('home.html')
41 @app.route('/home', methods=['GET'])
42 def home():
43     return render_template('home.html')
44 @app.route('/intro', methods=['GET'])
45 def about():
46     return render_template('intro.html')
47 @app.route('/upload', methods=['GET', 'POST'])
```