NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Team ID: LTVIP2023TMID08016

Submitted by

G. Priyanka (20NM1A0454)
G. Bhanu Sravani (20NM1A0463)
E. Pradeepthi Sai Sri (20NM1A0451)
A. Krishnaveni (20NM1A0403)

BACHELOR OF ENGINEERING

IN

Electronics and Communication Engineering

VIGNANS INSTITUTE OF ENGINEERING FOR WOMEN

TABLE OF CONTENTS

1.INTRODUCTION

PROJECT OVERVIEW

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even to permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters 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 neural network model that classifies the natural disaster and tells the intensity of disaster of natural. The model uses an integrated webcam to capture the video frame and the video frame is compared with the Pre-trained model and the type of disaster is identified and showcased on the OpenCV window.

PURPOSE:

The purpose of natural disaster intensity analysis and classification using AI is to build a deep learning model that can classify and tell the intensity of a natural disaster based on images. This can help to overcome losses in ecosystems, human lives, and properties by providing timely and accurate information for disaster management and response. Basically the Min objective of natural disaster management reduce the damage. However, there are several on Objectives that are integrated with it. Identifying the hazard and its cause. Reducing vulnerability and potential loses of hazard. Assessing ,reviewing and controlling the risk. Reducing the damage, death, sufferings and destruction of any natural and human induced disaster Giving protection to victims. Increasing the strength among people to survive among disasters .Building up capacity in every sector like-individual, social, economic, environmental, national and international. Ensuring the availability of local emergency equipment and transportation. Promote the culture of disaster risk prevention and mitigation at all levels.

PROBLEM STATEMENT :

Disaster can be caused by natural occurring events such as earthquakes, cyclones , floods and wildfires. Many deep-learning techniques have been applied by various researches to detect natural disasters to overcome losses in ecosystem ,but detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem we process a multi-layered deep convolutional neutral network. A civilian who is aware about natural disasters and takes the prediction methods to save nature. Due to natural disasters ,there are droughts ,economic crisis ,capital destruction etc. Natural disasters are increasing because of population growth , urbanization (a lot of people in small places),alteration of the natural environment(man-made islands).

PROPOSED SOLUTION:

The main purpose of this model is to detect and classify the type of disaster with a high accuracy rate .To prevent natural disasters in the future ,said model can be used to predict future disasters and take some action against heavy loss of human ecological systems and property. We propose a multi-layered deep convolution neural network .The proposed model works in two blocks: Block-1 convolutional neural network (B-1 CNN), for detection and occurrence of disasters. block-2 convolutional neural network (B-1 CNN), for classification of natural disasters intensity type with different filters and parameters. Building collapse , ailments spread and sometimes natural disasters such as tsunamis can devastate nations. The proposed multi-layered deep convolutional neural network was simulated on the computer system with core i7 ,central processing**.**

2. REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS

The following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**LOGIN**	Login by giving a mobile number, gmail or google account and their loca on.
FR-2	**ALERT**	The alert message is given to all the users when the cyclone hits.
FR-3	**MONITORING**	Continuous monitoring of cyclones and climate changes.
FR-4	**REPORTS**	Keeping the records of the previous cyclone and refer news from meteorologist for live updates.
FR-5	**END USERS**	The information is sent to the farmers using the database.
FR-6	**END GOAL**	Inform farmers about the cyclone and its intensity.

2 .2 NON-FUNCTIONAL REQUIREMENTS

The following are the non-functional requirements of the proposed solution :

FR No.	Non-Functional Requirement	Description

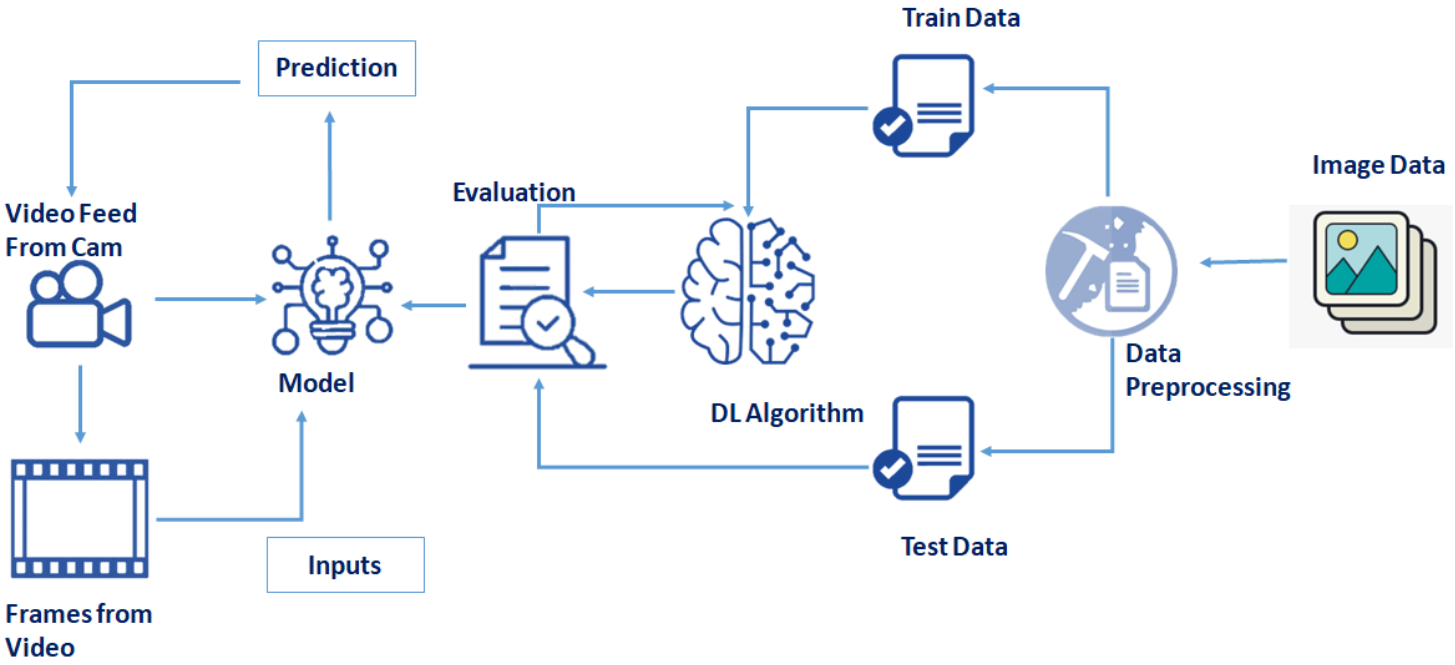
NFR-1	**USABILITY**	While using this system, people turn on their current loca on. They receive alert messages as no fica on. The local officials can also inform and guide their nearby people and farmers by an alert message.
NFR-2	**SECURITY**	It does not share any personal informa on to strangers. Their informa on is to be encrypted and
NFR-3	**RELIABILITY**	As the details collected from satellite image and meteorologist and updated details in this system, so it is trustworthy.
NFR-4	**PERFORMANCE**	It runs in minimum storage space. It will run efficiently when 1000 users login the same time.

NFR-5	**AVAILABILITY**	It should be available in all Android phones and laptops.
NFR-6	**SCALABILITY**	As the product we created is user friendly and it will be very useful for farmers and agriculture.

3. PROJECT DESIGN

3.1 TECHNICAL ARCHITECTURE

The technical architecture also provides information about the outputs and inputs of each entity and the process itself.



3.2 PROJECT FLOW

- The user interacts with the UI (User Interface) to open the integrated webcam.
- The video frames are captured and analyzed by the model which is integrated with flask application.
- Once model analyses the video frames, the prediction is showcased on the UI and OpenCV window

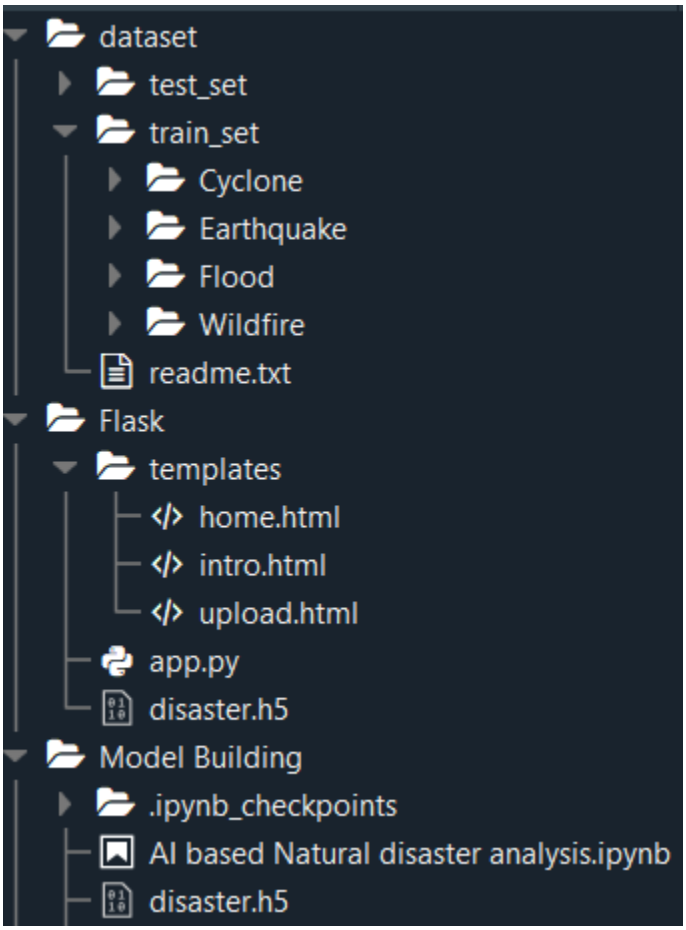
To accomplish this, we have to complete all the activities and tasks listed below

- Data Collection.
  - Collect the dataset or Create the dataset
- Data Preprocessing.
  - Import the ImageDataGenerator library
  - Configure ImageDataGenerator class
  - ApplyImageDataGenerator functionality to Trainset and Testset
- Model Building
  - Import the model building Libraries

3.3 PROJECT STRUCTURE

- Dataset folder contains the training and testing images for training our model.
- We are building a Flask Application that needs HTML pages stored in the templates folder and a python script app.py for server side scripting
- we need the model which is saved and the saved model in this content is a disaster.h5

- templates folder contains home. html, intro.html, upload.html pages.



4. PROJECT PLANNING AND SCHEDULING

4.1 PREREQUISITES

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook,

QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder

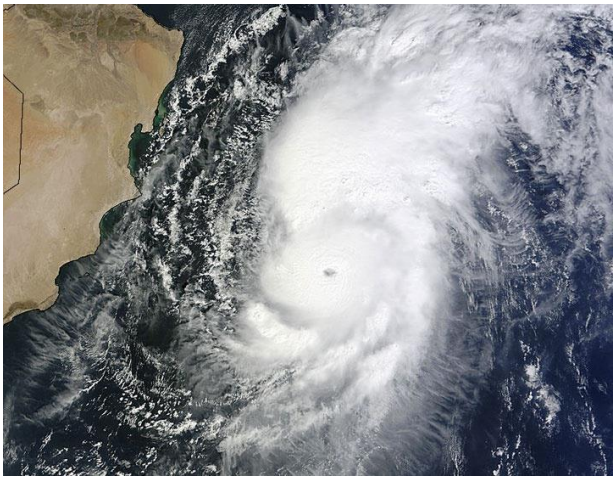
To build Machine learning models you must require the following packages

- **Numpy**:
  - It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations

- **Scikit-learn:** forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy

- **OpenCV**
  - OpenCV is a library of programming functions mainly aimed at real-time computer vision. Here, OpenCV is used to capture frames by accessing the webcam in real-time.
  - Open anaconda prompt and type command
    "pip install opencv-contrib-python"
- **Flask:**
  Web framework used for building Web applications
- **Python packages:**
  - open anaconda prompt as administrator
  - Type "pip install numpy" and click enter.
  - Type "pip install pandas" and click enter.
  - Type "pip install scikit-learn" and click enter.
  - Type "pip install opencv-contrib-python" and click enter.
  - Type "pip install tensorflow==2.3.0" and click enter.
  - Type "pip install keras==2.4.0" and click enter.
  - Type "pip install Flask" and click enter.

5. IMAGES OF NATURAL DISASTERS



CYCLONE

- Tropical cyclones are one of the biggest threats to life and property even in the formative stages of their development. They include a number of different hazards that can individually cause significant impacts on life and property, such as storm surge, flooding, extreme winds, tornadoes and lighting. Combined, these hazards interact with one another and substantially increase the potential for loss of life and material damage.

EARTHQUAKE

- Earthquakes occur most often along geologic faults, narrow zones where rock masses move in relation to one another. The major fault lines of the world are located at the fringes of the huge tectonic plates that make up Earth's crust.

FLOODS

- Floods are a type of natural disaster that can cause heavy destruction to life and property. It is a condition when rainwater accumulates at a place, flooding populated areas. They can also lead to the loss of numerous lives. At times, it can be highly dangerous and can wipe off an entire village or city.

WILDFIRE

Wildfires occur when vegetated areas are set alight and are particularly common during hot and dry periods. They can occur in forests, grasslands, brush and deserts, and with sufficient wind can rapidly spread. Unchecked, such fires can cause devastation to forests and other areas of vegetation.

6. CODING AND SOLUTIONING

6.1 FEATURE 1:

HTML

Home page:

```
<!DOCTYPE html>

<html>
                    <head>
                            <title>Home page</title>
                    </head>
                    <body>
                            <a href = "Intro page.html">Intro page </a>
                            <a href = "Upload page.html">Upload page </a>

                            <h1>AI BASED NATURAL DISASTER ANALYSIS</h1>
                            <img src = "D:\Disasters\CYCLONE.jpg">
                            <img src = "D:\Disasters\EARTHQUAKE.jpg">
                            <img src = "D:\Disasters\FLOOD.jpg">
                            <img src = "D:\Disasters\FOREST FIRE.jpg">
                              </body>
</html>
```

Intro page:

```
<!DOCTYPE html>

<html>
                <head>
                        <title>Intro page</title>
                </head>
                <body>
                        <a href = "Home page.html">Home page </a>
                        <a href = "Upload page.html">Upload page </a>

                        <h1>AI BASED NATURAL DISASTER ANALYSIS</h1>
                         <p>A Disaster is a serious problem occuring over a short or long period of time that causes widespread human, material, economic or environm
                          </body>
</html>
```

Upload page:

```
<!DOCTYPE html>

<html>
                <head>
                        <title>Upload page</title>
                </head>
                <body>
                        <a href = "Intro page.html">Intro page </a>
                        <a href = "Home page.html">Home page </a>

                        <h1>AI BASED NATURAL DISASTER ANALYSIS</h1>
                        <img src = "D:\Emergency\Emergency alert.jpeg">
                  </body>
</html>
```

6.2 FEATURE 2:

PYTHON

```
from flask import Flask,render_template,request
import cv2
from tensorflow.keras.models import load_model#to load our trained model
import numpy as np
from werkzeug.utils import secure_filename
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('model.h5')
print("Loaded model from disk")
@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')

def predict():

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

        print("[INFO] starting video stream...")
        vs = cv2.VideoCapture(0)
        #writer = None
        (W, H) = (None, None)

# loop over frames from the video file stream
        while True:
            # read the next frame from the file
            (grabbed, frame) = vs.read()

            # if the frame was not grabbed, then we have reached the end
            # of the stream
            if not grabbed:
                break
            if W is None or H is None:
                (H, W) = frame.shape[:2]

            # clone the output frame, then convert it from BGR to RGB
            # ordering and resize the frame to a fixed 64x64
            output = frame.copy()
            #print("apple")
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            frame = cv2.resize(frame, (64, 64))
            #frame = frame.astype("float32")
            x=np.expand_dims(frame, axis=0)
            result = np.argmax(model.predict(x), axis=-1)
            index=['Cyclone','Earthquake','Flood','Wildfire']
            result=str(index[result[0]])
            #print(result)
            #result=result.tolist()

            cv2.putText(output, "activity: {}".format(result), (10, 120
cv2.FONT_HERSHEY_PLAIN,
                        1, (0,255,255), 1)
            #playaudio("Emergency it is a disaster")
            cv2.imshow("Output", output)
            key = cv2.waitKey(1) & 0xFF

                # if the `q` key was pressed, break from the loop
```

```
            if key == ord("q"):
                break

        # release the file pointers
        print("[INFO] cleaning up...")
        vs.release()
        cv2.destroyAllWindows()
        return render_template("upload.html")

if __name__ == '_main_':
      app.run(host='0.0.0.0', port=8000,debug=False)
```

7. CONCLUSION

Artificial intelligence has the potential to enhance the detection and classificationof natural disasters, as well as the resilience and relief efforts
of affected communities. By using deep learning techniques, AI can analyze complex and imbalanced images of disasters and provide accurate and timely information. However, AI also faces challenges such as data quality, ethical issues, and human-AI collaboration. Therefore, it is essential to develop robust and reliable AI systems that can complement human expertise and judgment in disaster management.

AI can help predict the occurrence and impact of natural disasters by using historical data, satellite imagery, and weather models. This can enable early warning systems and preparedness plans for vulnerable areas. AI can also assist in the recovery and reconstruction of disaster-affected regions by providing insights into the needs and priorities of the survivors, as well as the best allocation of resources and funds.

AI can also support the learning and improvement of disaster management practices by analyzing the lessons learned from past disasters and identifying the gaps and opportunities for future interventions.

8. FUTURE SCOPE

To develop more advanced and efficient deep learning models that can handle the complexity and diversity of natural disaster images, and provide accurate and reliable results.

To integrate multiple sources and types of data, such as text, audio, video, and sensor data, to enhance the analysis and classification of natural disasters and their impacts.

To explore the ethical and social implications of using AI for natural disaster management, such as the privacy, security, and accountability of the data and the algorithms, and the potential biases and risks of the AI outputs. To evaluate the performance and impact of AI for natural disaster management, and compare it with other methods and tools, such as human experts, traditional models, and manual processes. To foster the collaboration and communication among different stakeholders, such as researchers, practitioners, policymakers, and communities, to share the best practices and challenges of using AI for natural disaster management, and to co-create solutions that meet the needs and expectations of the users.

To promote the awareness and education of the public and the decision-makers on the benefits and limitations of AI for natural disaster management, and to encourage the participation and feedback of the affected people and groups.