PROJECT REPORT

Natural Disasters Intensity Analysis and Classification using Artificial Intelligence

Submitted by

PNT2022TMID42920

Sabarishan S	-	712819104025
Kavinraj S B	-	712819104717
Arjun Kumar G	-	712819104711
Manoj A	-	712819104018

TABLE OF CONTENTS

CHAPTER 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 lead 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 deep convolutional 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

Disaster management plays an integral role in **keeping communities safe**. It involves coordinating the resources, such as pollution control systems, and responsibilities, such as following best practice policies, needed to prevent, prepare for, respond to, and recover from emergencies.Natural disasters generally constitute an emergency since they require immediate intervention due to their **high impact on human health and safety**; they affect the normal functioning of working infrastructure, interrupting normal day activities and representing a risk for residents and workers in affected areas.

CHAPTER 2
LITERATURE SURVEY

EXISTING PROBLEM

Many researchers have attempted to use different deep learning methods for detection of natural disasters. However, the detection of natural disasters by using deep learning techniques still faces various issues due to noise and serious class imbalance problems. To address these problems, we proposed a multilayered deep convolutional neural network for detection and intensity classification of natural disasters. The proposed method works in two blocks—one for detection of natural disaster occurrence and the second block is used to remove imbalanced class issues. The results were calculated as average statistical values: sensitivity, 97.54%; specificity, 98.22%; accuracy rate, 99.92%; precision, 97.79%; and F1-score, 97.97% for the proposed model. The proposed model achieved the highest accuracy as compared to other state-of-the-art methods due to its multilayered structure. The proposed model performs significantly better for natural disaster detection and classification, but in the future the model can be used for various natural disaster detection processes.

REFERENCES

1. Mignan, A.; Broccardo, M. Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations. Seism. Res. Lett. 2020, 91, 2330–2342. [CrossRef]

2. Tonini, M.; D'Andrea, M.; Biondi, G.; Degli Esposti, S.; Trucchia, A.; Fiorucci, P. A Machine Learning-Based Approach forWildfire Susceptibility Mapping. The Case Study of the Liguria Region in Italy. Geosciences 2020, 10, 105. [CrossRef]

3. Islam, A.R.M.T.; Talukdar, S.; Mahato, S.; Kundu, S.; Eibek, K.U.; Pham, Q.B.; Kuriqi, A.; Linh, N.T.T. Flood susceptibility modelling using advanced ensemble machine learning models. Geosci. Front. 2021, 12, 101075. [CrossRef]

4. Schlemper, J.; Caballero, J.; Hajnal, V.; Price, A.N.; Rueckert, D. A deep

cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans. Med. Imaging 2017, 37, 491–503. [CrossRef] [PubMed]

5. Tang, C.; Zhu, Q.; Wu, W.; Huang, W.; Hong, C.; Niu, X. PLANET: Improved convolutional neural networks with image enhancement for image classification. Math. Probl. Eng. 2020, 2020. [CrossRef]

6. Ashiquzzaman, A.; Oh, S.M.; Lee, D.; Lee, J.; Kim, J. Context-aware deep convolutional neural network application for fire and smoke detection in virtual environment for surveillance video analysis. In Smart Trends in Computing and Communications, Proceedings of the SmartCom 2020, Paris, France, 29–31 December 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 459–467.

7. Li, T.; Zhao, E.; Zhang, J.; Hu, C. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. Electronics 2019, 8, 1131. [CrossRef]

8. Mangalathu, S.; Burton, H.V. Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions. Int. J. Disaster Risk Reduct. 2019, 36, 101111. [CrossRef]

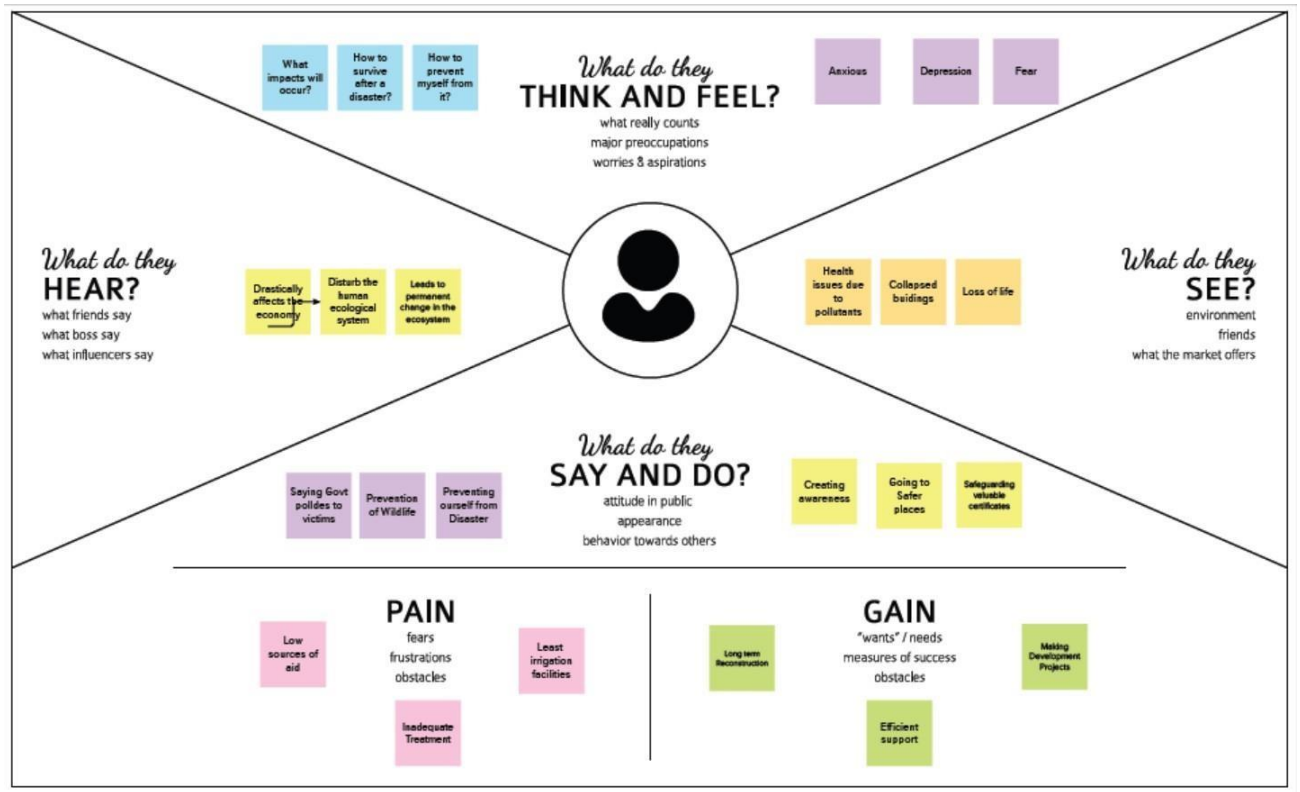
PROBLEM STATEMENT DEFINITION

"IMD (Indian Meteorology department) is responsible to issue warnings for the rainfall and CWC (Central Water Commission) keeps a record of water reservoirs, however there is a lack of collation of data issued from both these departments. This prevents us from determining the impact/seriousness and due to which there are times where adequate forewarnings are not provided. There are several High rainfall areas, low lying areas or flood prone areas. Currently there are limitations that these areas cannot be alerted before the critical situation because of the data unavailability or unavailability of simulation models which can calculate and predict the data. There is a requirement of data on the area likely to be inundated(depth) by release of water from reservoirs. 3D models may help in calculation of such data.a) Adequate forewarning for the area where floods are likely to occur. b) Low lying areas may be alerted about the release of accurate quantity of water from the reservoirs and thus evacuation/shifting of the people can be planned. c) It will help the Response forces to deploy their resources accordingly d) Prediction of release of water based on rainfall in catchment area and dissemination of an information to the affected public through mobile and other mediums."

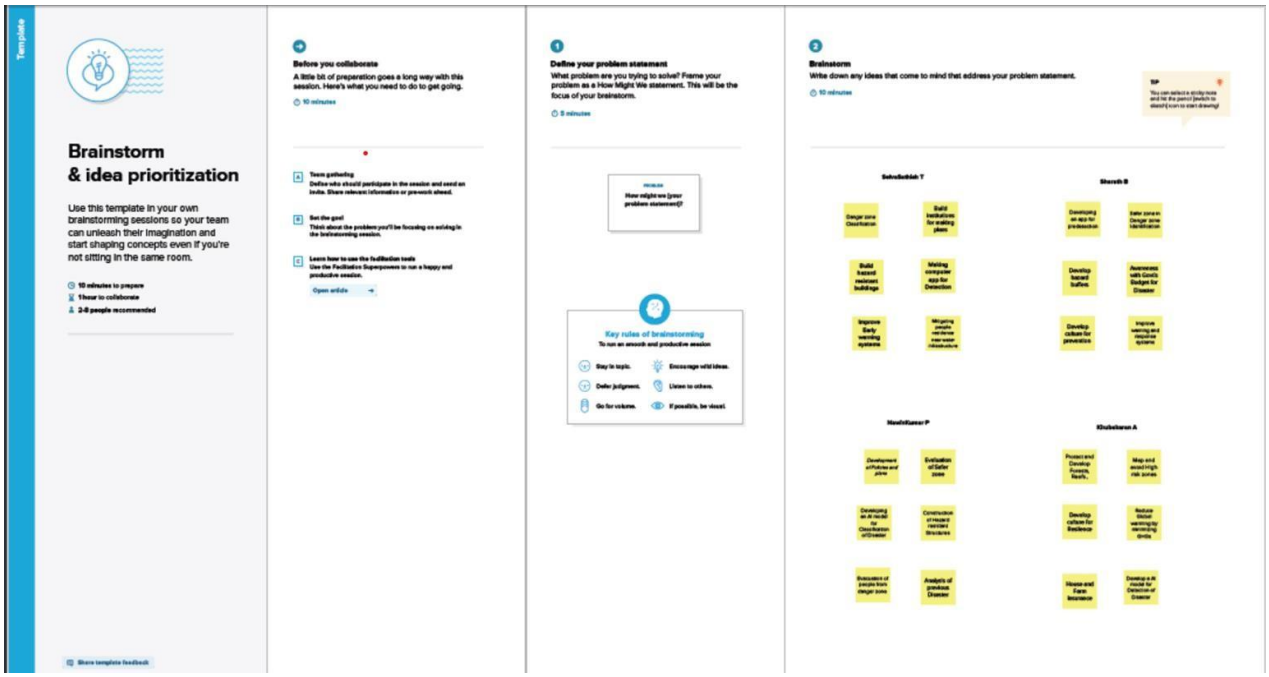
CHAPTER 3

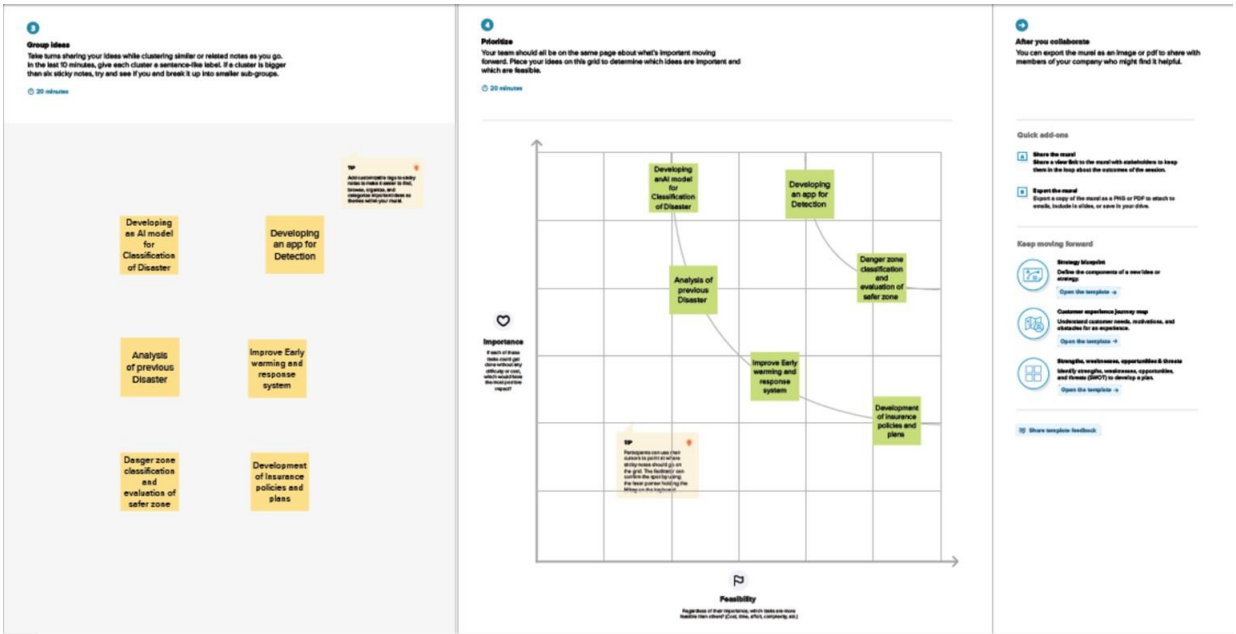
IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS



IDEATION & BRAINSTORMING

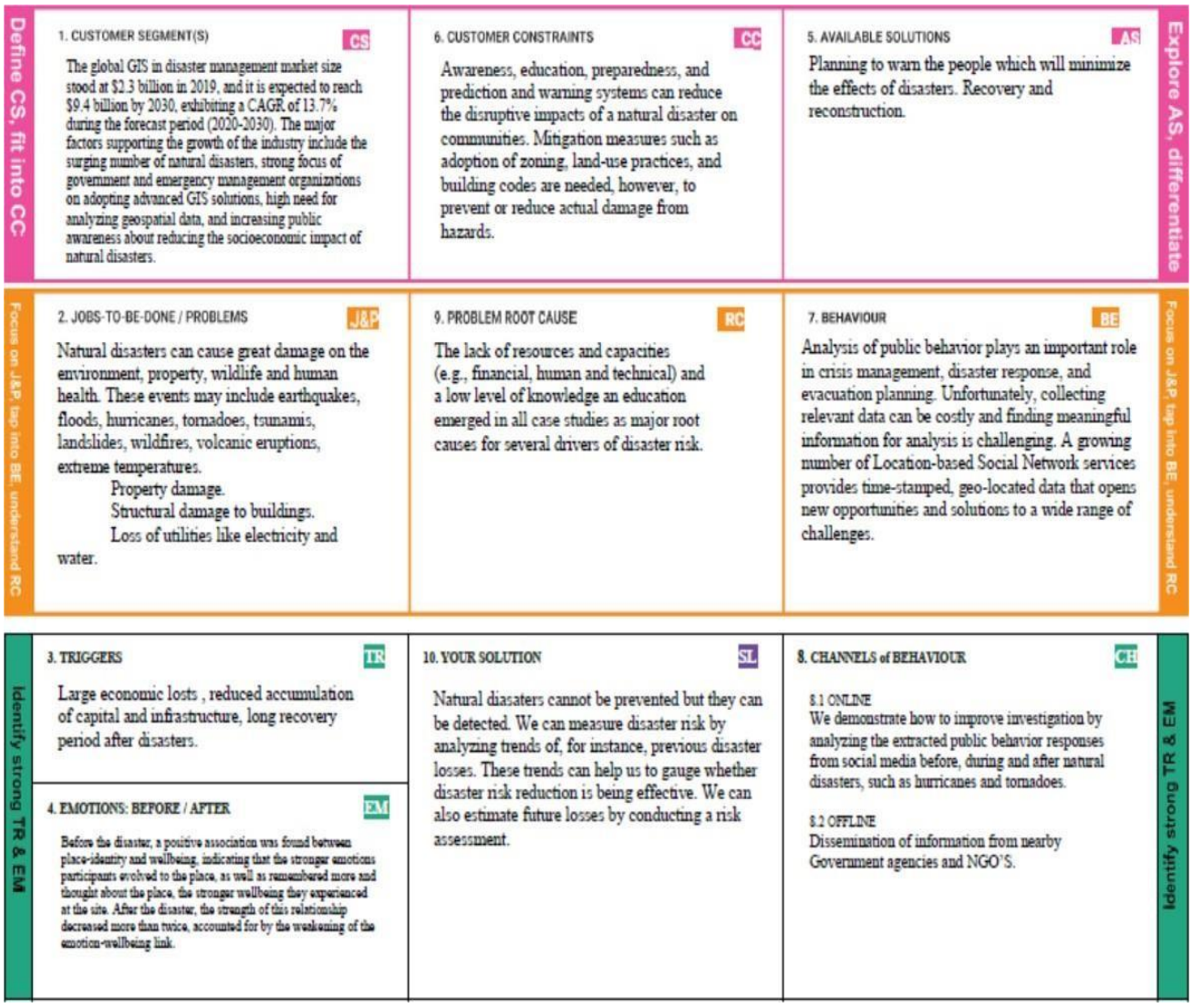




PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To classify the natural disaster and the effect based on the webcam image given as input using Artificial Intelligence.
2.	Idea / Solution description	The classification is done by deep learning techniques such as Convolutional Neural Network (CNN) and Machine Learning Techniques.
3.	Novelty / Uniqueness	It is based on the satellite and multispectral image and the classification using Multilayered Deep Convolutional Neural Networks.
4.	Social Impact / Customer Satisfaction	The people can easily identify the type of natural disaster and its effect on the environment which leads to the earlier identification and reduced damage in the ecosystem.
5.	Business Model (Revenue Model)	We build a system that classifies the natural diasater and its intensity and it is believed that the website is useful for all people and also the website works for a long time effectively.
6.	Scalability of the Solution	The website will be made available for all the people who needs to classify the type of natural disaster. The machine learning and deep learning algorithms that are being used made it easier for the classification and intensity analysis.

PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR - 1	User Registration	1. Registering via Google Accounts 2. Registering via Product's own user managementsystem
FR - 2	User Authentication	1. Verification through OTP 2. Verification through EmailLink
FR - 3	Designation of Region	1. Ease of selection of necessary areasto bemonitored 2. Versatile and Flexible operations on designatedareas
FR - 4	Analysis of Required Phenomenon	1. Simple and easy analysis on thespecificphenomenon to be observed
FR - 5	Accumulation of required Data	1. Fast and Efficient data gathering capabilities regarding past eventanalysis and futureprediction
FR - 6	Organizing Unstructured data	1. Processing of raw and clustered data into clearandrefined data which is useful for analysis and prediction tasks

FR - 7	Algorithm selection	1. The freedom to choose fromseveral classes ofalgorithm to be used in the process 2. Customization of algorithm to suit the needs of aspecific purpose
FR - 8	Prediction and analysis of data	1. Accurate results of the analysis provided by theprocess 2. Advanced visualization techniques to help visualizethe processed data for effective observation
FR - 9	Report generation	1. Restructuring of obtained results intoclearanddetailed reportfor future studies

NON-FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
NFR - 1	**Usability**	It is well suitedfor fields requiring diverseapplication of processes with efficiency , precision and ease.

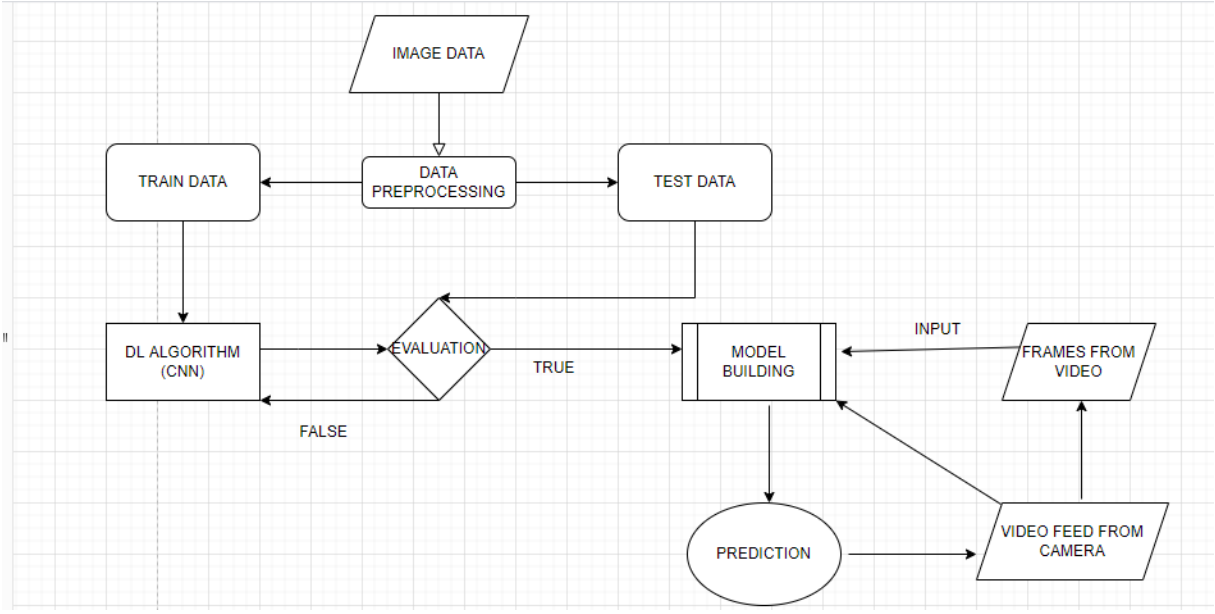
NFR - 2	**Security**	It provides a distinct and secure encryption layer to the system interface for additional security standards.
NFR - 3	**Reliability**	The product is robust and is capable of execution of processes even in the most difficult and unpredictable environments.
NFR - 4	**Performance**	The product boasts a high precision and efficient working capacity which helps in escalating its performance to the highest degree.
NFR - 5	**Availability**	Despite the complexity and degree of difficulty in its operation, the product is equipped with all-round maintenance andreadily available technical services whichprovides the necessary support any individual requires in their duties.
NFR - 6	**Scalability**	The product also possess enoughroom forthe improvement of its specifications to upgrade its capabilities according to the needs of the user and theirorganization

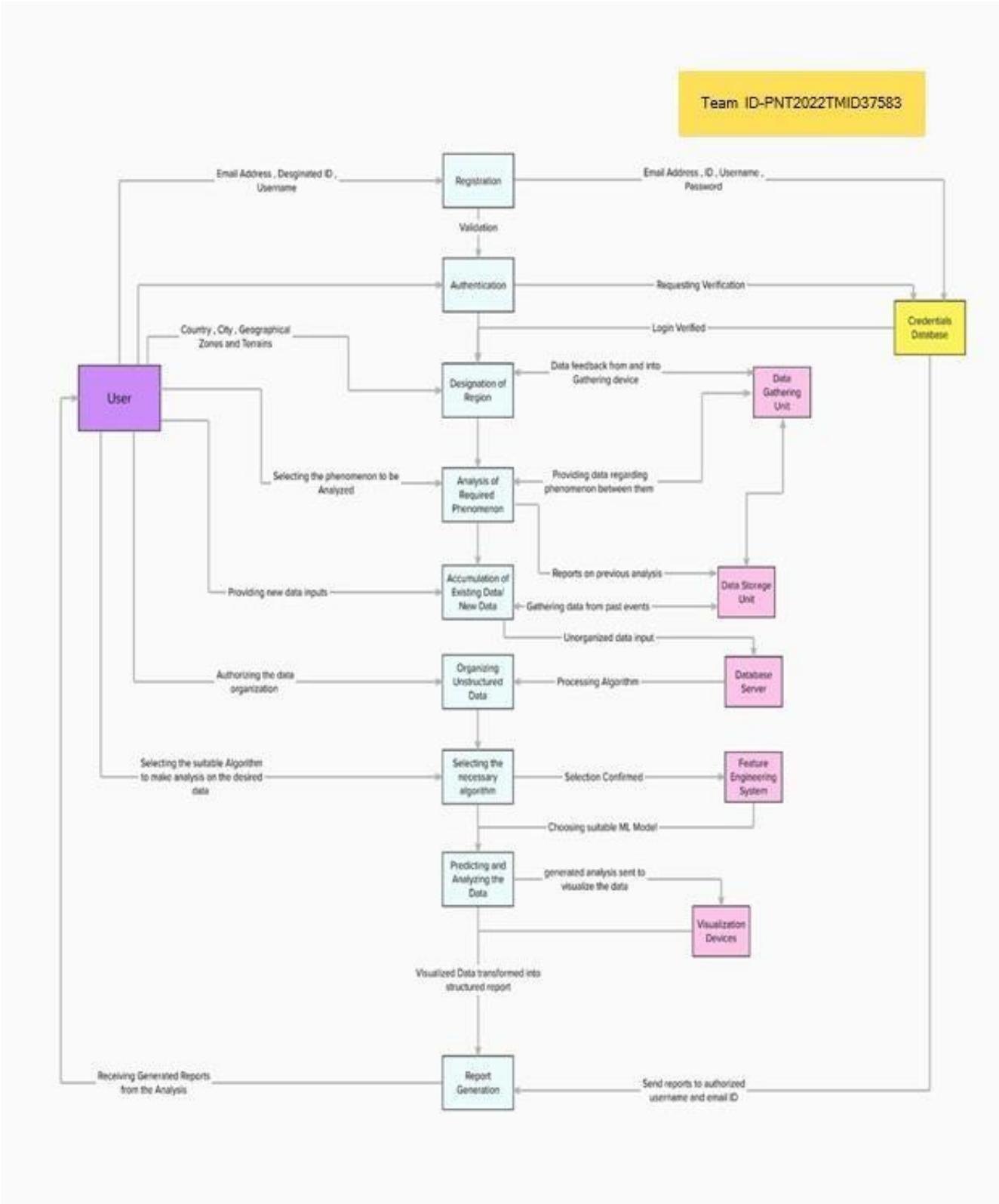
CHAPTER 5

PROJECT DESIGN

DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depictthe right amountof the systemrequirement graphically. It shows how data entersand leaves the system, what changes the information, and where datais stored.




Team ID-PNT2022TMID37583
Email Address , Designated ID , Username
Registration
Email Address , ID , Username , Password
Validation
Authentication
Requesting Verification
Credentials Database
Login Verified
Country , City , Geographical Zones and Terrains
User
Designation of Region
Data feedback from and into Gathering device
Data Gathering Unit
Selecting the phenomenon to be Analyzed
Analysis of Required Phenomenon
Providing data regarding phenomenon between them
Reports on previous analysis
Data Storage Unit
Providing new data inputs
Accumulation of Existing Data/ New Data
Gathering data from past events
Unorganized data input
Authorizing the data organization
Organizing Unstructured Data
Processing Algorithm
Database Server
Selecting the suitable Algorithm to make analysis on the desired data
Selecting the necessary algorithm
Selection Confirmed
Feature Engineering System
Choosing suitable ML Model
Predicting and Analyzing the Data
generated analysis sent to visualize the data
Visualization Devices
Visualized Data transformed into structured report
Receiving Generated Reports from the Analysis
Report Generation
Send reports to authorized username and email ID

USER STORIES

User Type	Functional Requirement (Epic)	User StoryNumber (USN)	User Story / Task	Acceptancecriteria	Priority	Release
End user (Customer)	Registration	USN - 1	As a user, I am able to register with theProduct using my valid emailaddress	I should be able to register with my account credentials	High	Sprint - 1
End User (Customer)	Authentication	USN - 2	As a user, I am able to login into the system with my credentials	It should ensure smooth login capabilities withoutdelay	High	Sprint - 1
End User (Customer)	Designation ofRegion	USN - 3	I can select the regionof interest to be monitored and analyzed	I mustbe able to choose certain specific places without error	High	Sprint - 1
End User (Customer)	Analysis of Required Phenomenon	USN - 4	I am able to monitor certain factors that influence the actions ofthe phenomenon	It should consider and monitormostof the factors involved in the action	High	Sprint - 2
End User (Customer)	Accumulation of required Data	USN - 5	I am able to gather data regarding past events and a detailed report on past analysis	It shouldallow the storage of data ofpast events for certain extent	Medium	Sprint - 2
End User (Customer)	Organizing Unstructured data	USN - 6	I am able to organizeand restructure the raw data into refineddata	It should ensureeasy and efficient processing methods	Low	Sprint - 3

End User (Customer)	Algorithmselection	USN - 7	I am ableto choose the required Algorithmfor a specific analysis	It mustprovide various options for the algorithm to be used	High	Sprint - 2
End User (Customer)	Prediction and analysis of data	USN - 8	I am able to easily predict and visualize thedata	It shouldallow easy to use prediction and visualization techniques	High	Sprint - 3
End User (Customer)	Report generation	USN - 9	I am able to generatea clear and detailed report on the analysis	Report generation must be fast and efficient andshouldnot be complex	Medium	Sprint - 4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement(Epic)	User story Number	User story / Task	Story points	Priority	Team members
Sprint-1	Registration	USN – 1	As a user, Registering into the product usinga valid email address	5	High	Sabarishan S
Sprint-2	Registration	USN – 2	As a user, Registering into the product usinga valid username andpassword	3	Medium	Kavinraj S.B
Sprint-1	Authentication	USN – 3	As a user , I adept to logginginto the system with credentials	4	High	Sabarishan S
Sprint-2	Authentication	USN - 4	As a user , I adept to logging intothe systemwithOTP	2	High	Kavinraj S.B
Sprint-1	Designation ofRegion	USN – 5	selecting the region ofinterest to be monitored and analysed	3	High	Sabarishan S
Sprint-2	Analysis of Required Phenomen On	USN – 6	Regulating certain factors influencing theactions of the phenomenon	3	High	Kavinraj S.B
Sprint-2	Accumulation ofrequired Data	USN – 7	Gathering data and detailed report on pastevent analysis	4	Medium	Kavinraj S.B

Sprint-4	Organizing Unstructureddata	USN – 8	Organizing and reorienting the raw data into a refined data	3	Low	Manoj A
Sprint-2	AlgoriThm selection	USN – 9	Choosing a required algorithm for specificanalysis	2	High	Kavinraj S.B
Sprint-3	Prediction and analysis of data	USN – 10	Predicting and visualizing the dataeffectively	6	High	Arjun kumar G
Sprint-4	Report generation	USN – 11	Generating a clear anddetailed report on productdata analysis	3	High	Manoj A

SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	12	6 days	24 Oct 2022	29 Oct 2022	12	30 Oct 2022
Sprint-2	14	6 days	31 Oct 2022	5 Nov2022	14	6 Nov 2022
Sprint-3	6	6 days	07 Nov2022	12 Nov2022	6	8 Nov 2022
Sprint-4	6	6 days	14 Nov2022	19 Nov2022	6	20 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE

```
from flask import Flask,render_template,request
import cv2
from tensorflow.keras.models import load_model
import numpy as np
from werkzeug.utils import secure_filename

app=Flask(__name__,template_folder="templates")
model=load_model('disaster.h5')
print("Loaded model from disk")

@app.route('/',methods=['GET'])
def home():
    return render_template("index.html")
@app.route('/home',methods=['GET'])
def back():
    return render_template("index.html")
@app.route('/upload',methods=['GET'])
def index():

    cap=cv2.VideoCapture(0)
    H=None
    W= None
```

```
while True:
        (grabbed,frame)= cap.read()

        if not grabbed:
            break
        if W is None or H is None:
            (H,W)= frame.shape[:2]
        output= frame.copy()

        frame=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
        frame=cv2.resize(frame,(64,64))
        x=np.expand_dims(frame,axis=0)

        result = np.argmax(model.predict(x),axis=-1)
        index =['Cyclone','Earthquake','Flood','Wildfire']

        output=str(index[result[0]])

        print(result)
        return render_template("output.html",output=output)
        # cv2.putText(output,"activity:{}".format(result),(10,120),cv2.FONT_HERSHEY_PLAIN,1,(0,255,255),1 )
        # cv2.imshow("output",output)
#         if cv2.waitKey(2) & 0xFF==ord('x'):
#             break
# print("[info] cleaning up....")
# cap.release()
# cv2.destroyAllWindows()
```

```
    # return render_template("output.html",output=result)
if __name__=='__main__':
    app.run(host='0.0.0.0',port=8000,debug=False)
```

CHAPTER 8

TESTING

TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the HomePage	The Home pagemust be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screensizes	The Home pagemust be displayed properly in allsizes	The UI is not displayed properlyin screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if thepage redirects to the resultpage once the input is given	The page shouldredirect to the results page	Working as expected	PASS
BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image	The model shouldrescale the imageand predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the disaster	The model shouldpredict the disaster	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input	The model should predict the disaster in the	The model fails to identify it since the model is not built to	FAIL

				compexfeed	handle such data	
RP_TC_001	UI	Result Page	Verify UI elements in the ResultPage	The Result pagemust be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the resultis displayed properly	The result shouldbe displayed properly	Working as expected	PASS
RP_TC_003	UI	Result Page	Check if the other predictions are displayed properly	The other redictions should be displayed properly	Working as expected	PASS

USER ACCEPTANCE TESTING
DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2

Total	6	1	4	3	14

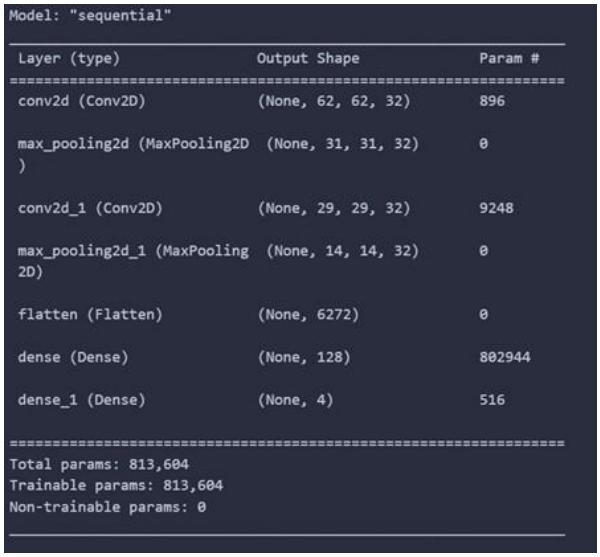
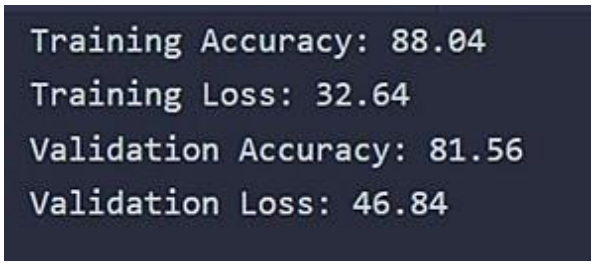
TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy – 88.04% Validation Accuracy -81.56%	

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES:-

1. Humans also need breaks and time offs to balance their work life and personal life. But AI can work endlessly without breaks.
2. With the use of various AI-based techniques, we can also anticipate today's weather and the days ahead.
3. Helpful in getting life back on track..
4. Their Alert nature able to respond effectively and efficiently which defend the society from large scale damages.

DISADVANTAGES:-

1. It involves huge money to be equipped.
2. Problems faced in life basic needs.
3. One application of artificial intelligence is a robot, which is displacing occupations and increasing unemployment .
4. Machines can perform only those tasks which they are designed or programmed to do, anything out of that they tend to crash or give irrelevant outputs which could be a major backdrop.

CHAPTER 11

CONCLUSION

Many researchers have attempted to use different deep learning methods for detection of natural disasters. However, the detection of natural disasters by using deep learning techniques still faces various issues due to noise and serious class imbalance problems.To address these problems, we proposed a multilayered deep convolutional neural network for detection and intensity classification of natural disasters. The proposed method works in two blocks—one for detection of natural disaster occurrence and the second block is used to remove imbalanced class issues. The results were calculated as average statistical values: sensitivity, 97.54%; specificity, 98.22%; accuracy rate, 99.92%; precision, 97.79%; and F1-score, 97.97% for the proposed model. The proposed model achieved the highest accuracy as compared to other state-of-the-art methods due to its multilayered structure. The proposed model performs significantly better for natural disaster detection and classification, but in the future the model can be used for various natural disaster detection processes.

CHAPTER 12

FUTURE SCOPE

AI -smart technology, which has enabled accurate and speedy solutions. If harnessed properly, the technology has the potential of predicting, preventing and providing response faster than ever.

AI data setups are trained to predict seismic data to analyze the patterns of earthquake occurrences, rainfall records and monitor flooding, measure the intensity of hurricanes and read the geological data to understand volcanic eruptions, such systems can reduce the catastrophic impact of natural disasters.

Last year, Google's Pilot project to monitor flood in India with the help of AI, was a successful one – it was a Patna project. They were able to predict floods and the regions that it would be affected due to the natural disaster with an accuracy of over 90%. It was possible owing to the combination of data from government agencies that provide on-ground information – from measuring devices placed on the spot and satellite captured images of flood-prone areas. They ran hundreds of thousands of simulations on its machine learning (ML) models to predict the flow of water.In the future, leveraging AI can help disaster management bodies install drones, sensors and robots to provide accurate information about damaged buildings and landscapes, potential floods, making rescue missions safer and less time-consuming.

There is a need for smart technology to be integrated within our local communities. Immediate response and tech-based solutions can help reduce the extent of damage. However, since AI is based on machine codes, there is a scope of limitations and errors. However, the amalgamation of human, empathy and alertness, could do wonders in the field of crisis management.

APPENDIX

SOURCE CODE
MODEL CREATION

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator
```

```
data_path = './Dataset/'
batch_size = 32
target_size = (64, 64)
```

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   width_shift_range=0.1,
                                   height_shift_range=0.1,
                                   horizontal_flip=True,
                                   validation_split=0.2)

test_datagen = ImageDataGenerator(rescale=1. / 255, validation_split=0.2)
```

```
X_train = train_datagen.flow_from_directory(data_path,
                                            target_size=target_size,
                                            batch_size=batch_size,
                                            color_mode="rgb",
                                            subset="training",
                                            class_mode='categorical')


X_test = test_datagen.flow_from_directory(data_path,
                                          target_size=target_size,
                                          batch_size=batch_size,
                                          color_mode="rgb",
                                          subset="validation",
                                          class_mode='categorical')
```

```
Found 3544 images belonging to 4 classes.
Found 884 images belonging to 4 classes.
```

```
model = Sequential()
```

```
    model.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(32, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())

    model.add(Dense(units=128, activation='relu'))
    model.add(Dense(units=4, activation='softmax'))
```

```
    model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )
```

```
 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 4)                 516

=================================================================
Total params: 813,604
Trainable params: 813,604
Non-trainable params: 0
_________________________________________________________________
```

```
   model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
model.fit(X_train,
          steps_per_epoch=len(X_train),
          epochs=20,
          validation_data=X_test,
          validation_steps=len(X_test))
```

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/20
111/111 [==============================] - 63s 495ms/step - loss: 1.0808 - accuracy: 0.5282 - val_loss: 0.7762 - val_accuracy: 0.7025
Epoch 2/20
111/111 [==============================] - 52s 470ms/step - loss: 0.7012 - accuracy: 0.7322 - val_loss: 0.7050 - val_accuracy: 0.7353
Epoch 3/20
111/111 [==============================] - 50s 448ms/step - loss: 0.6575 - accuracy: 0.7525 - val_loss: 1.1470 - val_accuracy: 0.6210
Epoch 4/20
111/111 [==============================] - 47s 428ms/step - loss: 0.5846 - accuracy: 0.7847 - val_loss: 0.8129 - val_accuracy: 0.6980
Epoch 5/20
111/111 [==============================] - 49s 442ms/step - loss: 0.5523 - accuracy: 0.7999 - val_loss: 0.6012 - val_accuracy: 0.7760
Epoch 6/20
111/111 [==============================] - 52s 470ms/step - loss: 0.5448 - accuracy: 0.7923 - val_loss: 0.7817 - val_accuracy: 0.7048
Epoch 7/20
111/111 [==============================] - 46s 415ms/step - loss: 0.4935 - accuracy: 0.8149 - val_loss: 0.6035 - val_accuracy: 0.7602
Epoch 8/20
111/111 [==============================] - 45s 406ms/step - loss: 0.4554 - accuracy: 0.8361 - val_loss: 0.5008 - val_accuracy: 0.8111
```

```
Epoch 9/20
111/111 [==============================] - 45s 405ms/step - loss: 0.4598 - accuracy: 0.8335 - val_loss: 0.6498 - val_accuracy: 0.7557
Epoch 10/20
111/111 [==============================] - 45s 403ms/step - loss: 0.4260 - accuracy: 0.8420 - val_loss: 0.7311 - val_accuracy: 0.7217
Epoch 11/20
111/111 [==============================] - 45s 406ms/step - loss: 0.4475 - accuracy: 0.8352 - val_loss: 0.4500 - val_accuracy: 0.8224
Epoch 12/20
111/111 [==============================] - 45s 403ms/step - loss: 0.4096 - accuracy: 0.8507 - val_loss: 0.5084 - val_accuracy: 0.7952
Epoch 13/20
...
Epoch 19/20
111/111 [==============================] - 43s 387ms/step - loss: 0.3500 - accuracy: 0.8685 - val_loss: 0.4097 - val_accuracy: 0.8473
Epoch 20/20
111/111 [==============================] - 44s 392ms/step - loss: 0.3264 - accuracy: 0.8804 - val_loss: 0.4684 - val_accuracy: 0.8156

<keras.callbacks.History at 0x19d4c8a69b0>
```

```
model.save("disaster.h5")
```

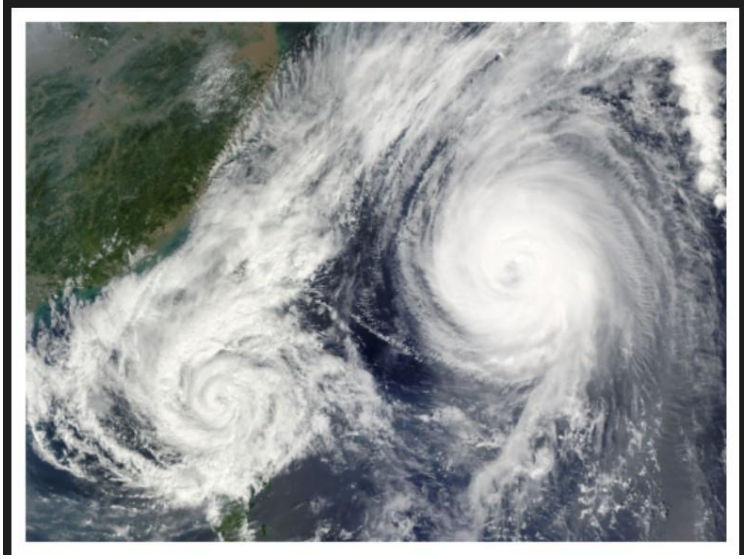
```python
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt

def predict(image_path):
    img = image.load_img(image_path, target_size=target_size)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    labels = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']

    pred = model.predict(x)
    prediction = labels[np.argmax(pred[0])]

    print(f'Disaster: {prediction}')
    plt.imshow(plt.imread(image_path))
    plt.axis('off')
    plt.show()
```

```python
predict('./Sample/1.jpg')
```

```
1/1 [==============================] - 0s 26ms/step
Disaster: Wildfire
```

```
predict('./Sample/2.jpg')
```

```
1/1 [==============================] - 0s 26ms/step
Disaster: Cyclone
```



```
predict('./Sample/3.jpg')
```

```
1/1 [==============================] - 0s 31ms/step
Disaster: Earthquake
```

```
predict('./Sample/4.jpg')
```

```
1/1 [==============================] - 0s 26ms/step
Disaster: Flood
```

FLASK APP.PY

```
from flask import Flask,render_template,request
import cv2
from tensorflow.keras.models import load_model
import numpy as np
from werkzeug.utils import secure_filename

app=Flask(__name__,template_folder="templates")
model=load_model('disaster.h5')
print("Loaded model from disk")

@app.route('/',methods=['GET'])
def home():
    return render_template("index.html")
@app.route('/home',methods=['GET'])
def back():
    return render_template("index.html")
@app.route('/upload',methods=['GET'])
def index():

    cap=cv2.VideoCapture(0)
    H=None
    W= None
    while True:
            (grabbed,frame)= cap.read()

            if not grabbed:
                break
            if W is None or H is None:
                (H,W)= frame.shape[:2]
```

```
            output= frame.copy()

            frame=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
            frame=cv2.resize(frame,(64,64))
            x=np.expand_dims(frame,axis=0)

            result = np.argmax(model.predict(x),axis=-1)
            index =['Cyclone','Earthquake','Flood','Wildfire']

            output=str(index[result[0]])

            print(result)
            return render_template("output.html",output=output)
            # cv2.putText(output,"activity:{}".format(result),(10,120),cv2.FONT_HERSHEY_PLAIN,1,(0,255,255),1 )
            # cv2.imshow("output",output)
    #         if cv2.waitKey(2) & 0xFF==ord('x'):
    #             break
    # print("[info] cleaning up....")
    # cap.release()
    # cv2.destroyAllWindows()

    # return render_template("output.html",output=result)
if __name__=='__main__':
    app.run(host='0.0.0.0',port=8000,debug=False)
```

INDEX . HTML

```
<html>
    <head>

       <meta charset="utf-8" name="viewport" content= "width=device-width, initial-scale=1.0">
      <style>
         *{
          margin: 0;
          padding: 0;
         }
         body{
          width: 100vw;
          height: 100vh;
         }
         .content{
          display: flex;
          flex-direction: column;
          margin-top: 5rem;

         }

         .top{
              height: 3rem;
              background-color: rgb(12, 104, 161);
              color: white;
              display: flex;
              letter-spacing: 2px;
              align-items: center;
              font-weight: 900;
              font-size: 1.2rem;
```

```
            font-family: monospace;
            padding-left: 2.5rem;
        }
        .images{
            display: flex;
            justify-content: center;
            align-items: center;

        }
        img{

            width:200px;
            height:300px ;
            border-radius: 1rem;
            margin-left: 5rem;
        }
        button{
            position: relative;
            background-color: rgb(12, 104, 161);
            color: white;
            padding: 5px 10px;
            border: 1px solid  rgb(12, 104, 161);
            font-weight: 900;
            font-size: 1.2rem;
            font-family: monospace;
            display: flex;
            margin-left: 46%;
            margin-top: 5rem;
        }
```

```
    </style>
</head>
    <body>


        <div class="top">
            <div id="heading">AI Based Natural Disaster Analysis </div>
        </div>
        <div class="content">
        <div class="images">
                <img src="../static/images/cyclones.jpg">
                <img src="../static/images/earthquake.jpg">
                <img src="../static/images/flood.jpg" >
                <img src="../static/images/wildfire.jpg">
        </div>
        <form action="/upload" method="get">
            <button type="submit">Open Web Camera</button>
        </form>
      </div>



    </body>
</html>
```

OUTPUT . HTML

```
<html>
    <head>
        <style>
            *{
                padding: 0;
                margin: 0;
            }
            body{
                height: 100vh;
                width: 100vw;
            }


           .top{
                height: 3rem;
                background-color: rgb(12, 104, 161);
                color: white;
                display: flex;
                letter-spacing: 2px;
                align-items: center;
                font-weight: 900;
                font-size: 1.2rem;
                font-family: monospace;
                padding-left: 2.5rem;
            }
            .top #home{
                position:absolute;
                background-color: rgb(12, 104, 161);
                color: white;
```

```
                right: 40;
                padding: 2px 8px;
                font-size: 1rem;
                font-family: monospace;
                border: 1px solid rgb(255, 255, 255);
                border-radius: 5px;

            }
            .content{

                display: flex;
                justify-content: center;

            }
             #name{
                padding: 3rem;
                text-transform: uppercase;
                font-size:2rem ;
                font-weight: 900;
                text-align: center;
                font-family:'Times New Roman', Times, serif;
            }
           img{

                width:400px;
                height:400px ;
                border-radius: 4rem;
            }
```

```
        </style>
    </head>
    <body>

        <div class="top">
         <div id="heading">AI Based Natural Disaster Analysis </div>
         <form action="/home" method="get">
             <button type="submit" id="home">Home</button>>
         </form>

        </div>

        <div id="name">{{output}}</div>
        <div class="content">
        <img src="../static/images/emergency.png" alt="">

         </div>
    </body>
</html>
```

GITHUB:-

https://github.com/IBM-EPBL/IBM-Project-46161-1660740232

PROJECT DEMO:-

https://drive.google.com/file/d/1Gbw8juGSmZMz25-FSVUyZpJUXT42iNLr/view?usp=drivesdk