PROJECT REPORT

NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Submitted by

TEAM ID:PNT2022TMID48626

MANGALA DHARSINI R - 920819104018
BAVITHRA C - 920819104006
SOUNDHARYA V - 920819104040
SUJITHA P - 920819104044

BACHELOR OF ENGINEERING

COMPUTER SCIENCE ENGINEERING

NPR COLLEGE OF ENGINEERING AND TECHNOLOGY

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 lead to permanent changein the ecosystem. Disaster can be caused by naturally occurring events such as earthquake, cyclone, flood, and wildfire. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems**.**

Purpose

The main of the aim of the project to develop a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster of naturalThe model uses an integrated webcam to capture the video frame and the video frame iscompared with the Pre-trained model and the type of disaster is identified and showcased on the OpenCV window.

2.LITERATURE SURVEY

Existing problem

Natural disasters are uncontrollable phenomena occurring yearly which cause extensive damage to lives, property and cause permanent damage to the environment. However by, using Deep Learning, real-time recognition of these disasters can help the victims and emergency response agencies during the onset of these destructive events. At present, there are still gaps in the literature regarding real-time natural disaster recognition.Flood management, which involves flood prediction, detection, mapping, evacuation, and relief activities, can be improved via the adoption of state-of the-art tools and technology. Thus, future efforts need to focus on combiningdisaster management knowledge, image processing techniques and machine learning tools to ensure effective and holistic disaster management across all phases.

References

1. Machine learning technique with numerical weather prediction
   Authors: M.Meadows, M.Wilson
2. Artificial neural network, genetic algorithm and wavelet transfer technique
   Authors : R.R. Sahay, A. Srivastava

Problem statement definition

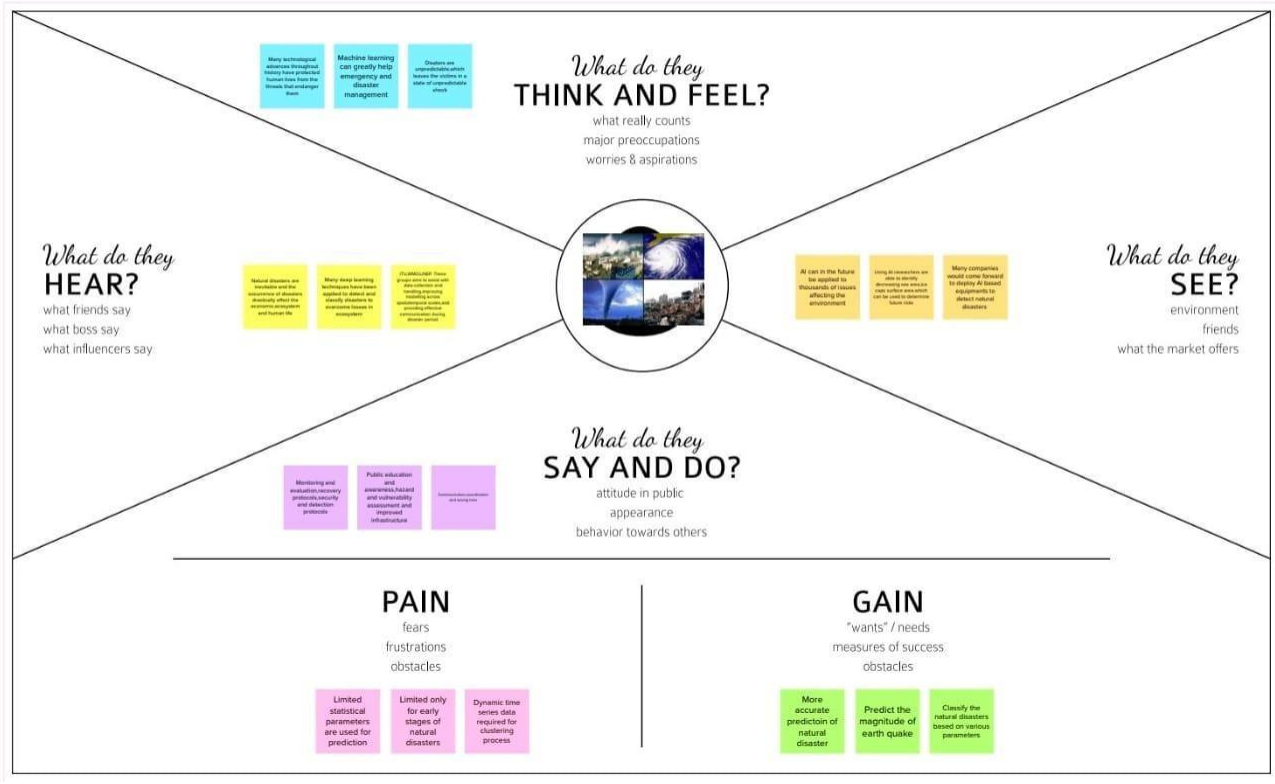
Natural disasters not only disturb the human ecological system but also destroythe properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem.

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 thecomplex and imbalanced structures of images.

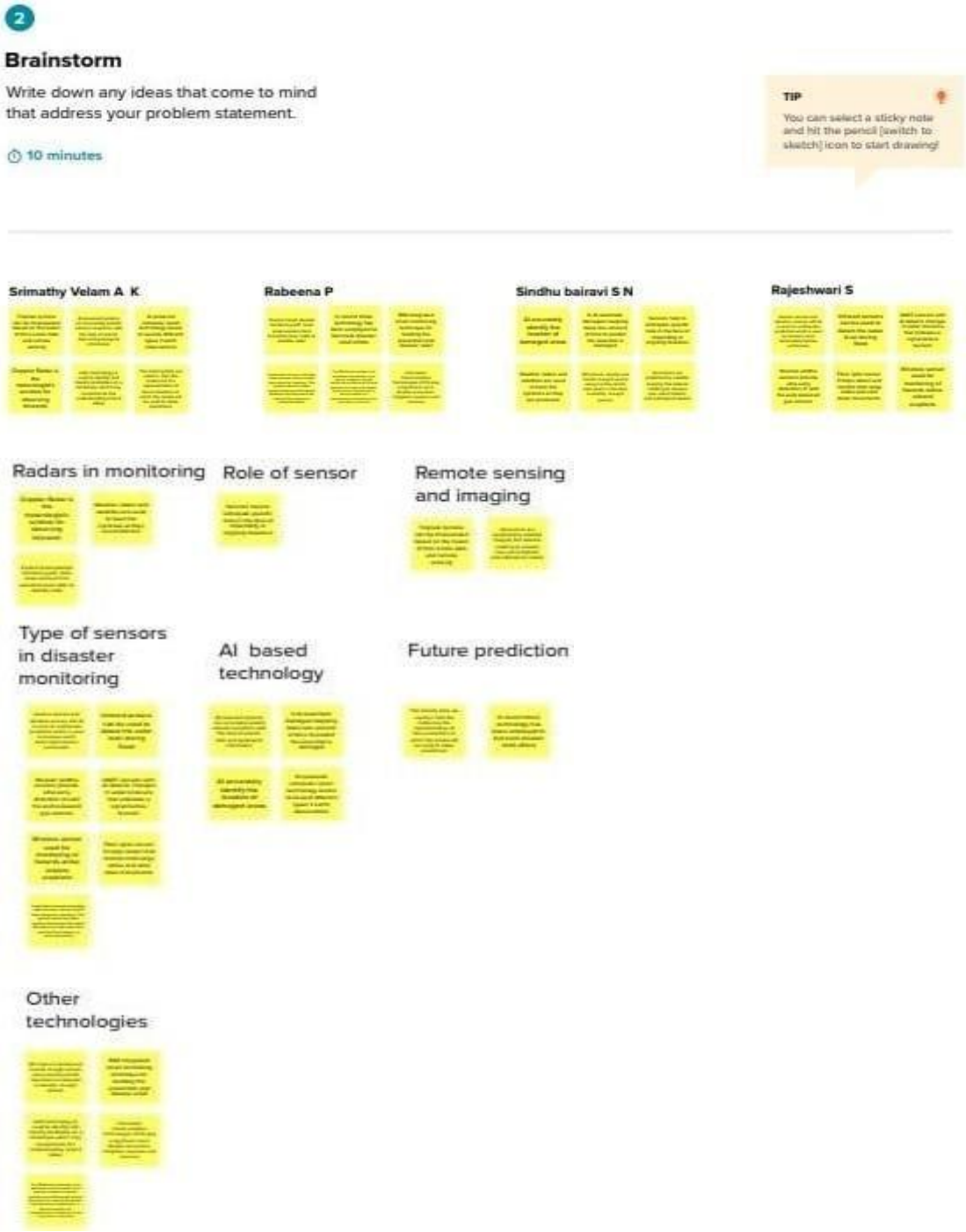
3. IDEATION AND PROPOSED SOLUTION

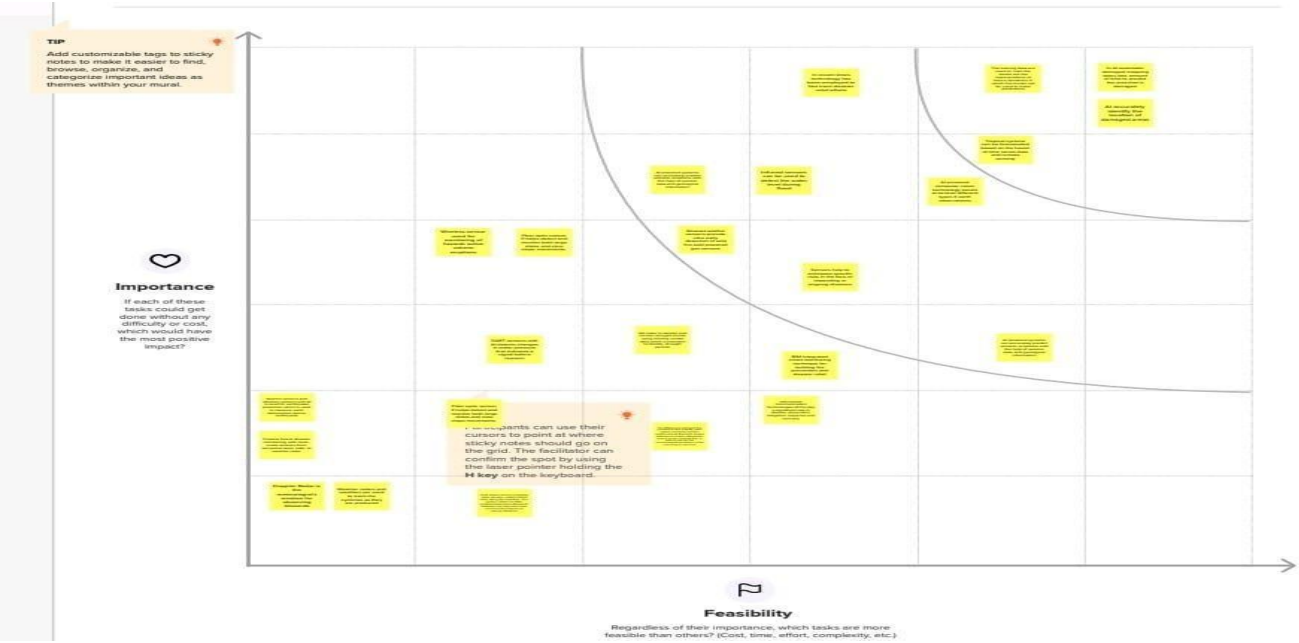
Empathy map canvas

Build empathy and keep your focus on the user by putting yourself in their shoes.



Ideation and Brainstorming



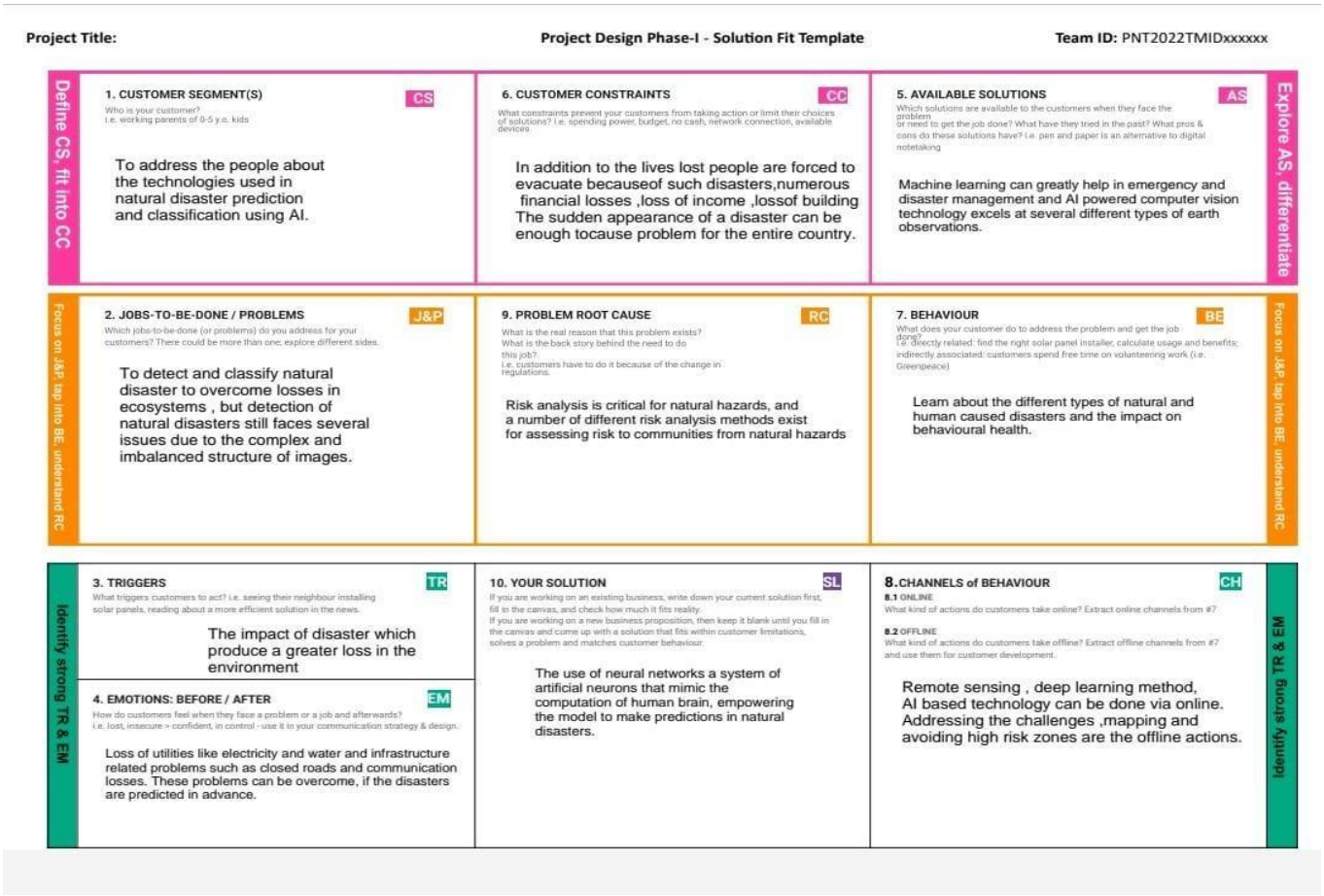


Proposed Solution

S.NO	Parameter	Description
1.	**Problem statement(problem to be solved)**	To detect and classify natural disaster to overcome losses in ecosystems , but detection of natural disasters still faces several issues due to the complex and imbalanced structure of images
2.	**Ideas/solution description**	Using machine learning techniques we can measure the intensity types of the disaster occurred with different filters and parameters.
3.	**Novelty/Uniqueness**	Using machine learning techniques we can measure the intensity types of the disaster occurred with different filters and parameters**.**

4.	**Social impact/customer satisfaction**	By issuing accurate forecasts in advance we can save people and environment from the effects of disasters in future**.**
5.	**Business Model (Revenue model)**	By issuing accurate forecasts in advance we can save people and environment from the effects of disasters in future**.**
6.	**Scalability of the solution**	Studies analyzing the intensity of natural disasters have gained a significant attention in the current decade .Thus a smart and cost effective detection network can be proposed to save the future from any type of disasters

Problem Solution Fit



4.REQUIREMENT ANALYSIS

Functional requirement

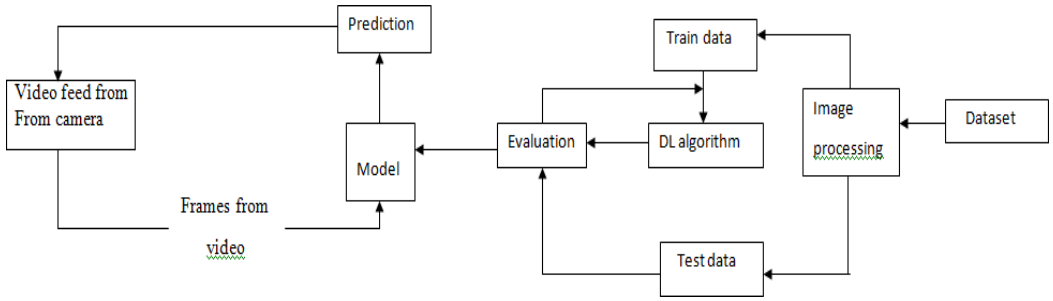
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Authentication	Provides a right or agreed results or effects based on the data set
FR-2	Transaction processing	Big blue
FR-3	External interface	User interface that involves some aspect of artificial intelligence
FR-4	Business rules	supports data based decisions and should not be a autonomous system
FR-5	Reporting	AI methods require large data sets to learn relationships between variables
FR-6	Compliance to laws or regulations	Robust and reliable Safe and secure

Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly and easy to classify the disaster
NFR-2	**Security**	There is no need for user login so that there is no security issues
NFR-3	**Reliability**	Highly reliable because it can process the data without any faults or errors
NFR-4	**Performance**	It gives a greater accuracy of about 90% to 95% and the result is obtained within a short period of time
NFR-5	**Availability**	It can be accessed at any time and the result is also perfect
NFR-6	**Scalability**	The website can be able to run on any web browser

5. PROJECT DESIGN

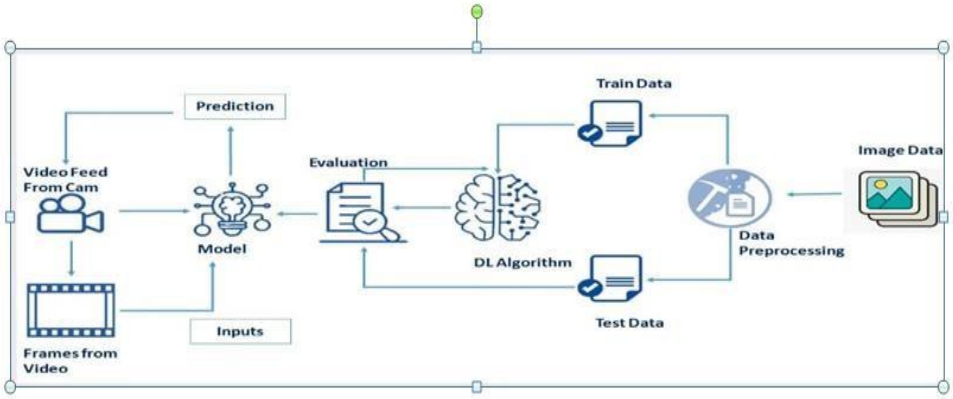
Data flow diagram



User stories

USER TYPE	FUNCTIONAL REQUIREMENT (EPIC)	USER STORY NUMBER	USER STORY/TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
Customer (web user)	Dashboard	USN 1	As a user, I can view the home page where the different types of natural disasters are defined	I can access the dashboard	High	Sprint 4
		USN 2	As a user,I can view the introduction page	I can access the dashboard	Medium	Sprint 4
		USN 3	As a user ,I can open my web camera to stream live		High	Sprint 4
		USN 4	As a user ,I can view the live stream		High	Sprint 4
Administration	Monitoring the website	USN 1	As a user,I can check whether the website is working smoothly		High	Sprint 4
	Monitoring the accuracy	USN 2	As a user,I csn check the accuracy of the result		High	Sprint 2

Technology architecture



Components and technologies

S.No	Component	Description	Technology
1.	User Interface	User can analyse the intensity using the website	HTML, CSS, JavaScript
2.	Application Logic-1	Deployment	Python
3.	Application Logic-2	Training and building deep learning model	IBM Watson Studio
4.	Database	Data collection of various disaster	MySQL
5.	Cloud Database	Database Service on Cloud	IBM Cloudant
6.	File Storage	To store the dataset	IBM Block Storage
7.	Data generation system	To generate and process the data required for the application	IBM Weather API
8.	Database management system	An organized collection of data stored in database to access and manage the data easily	MySQL
9.	Machine Learning Model	To classify the natural disaster	Object Recognition Model
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration:Python flask Cloud Server Configuration : cloud foundry	Local, Cloud Foundry.

Applications and characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open source frame works are free for public use and provide the foundation for building a sofyware application	Keras,tensor flow
2.	Security Implementations	Keeps the model secure and make sure that only those who have permission can use it	Encryption and decryption Example:sending an OTP
3.	Scalable Architecture	A truely excellent sftware needs a continuous process of improvements and updates.Regularly update the app with new features to enrich it	Waterfall approach Spiral approach
4.	Availability	It includes both online and offline work.Good internet connection is needed for online work to explore the software.Offline works includes the saved data to explore later	Caching ,backend server
5.	Performance	User can use mobile application and web application to interact with the model	App development and web development

6.PROJECT PLANNING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email	2	High	Rabeena p Rajeshwari s
Sprint-2	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	Medium	Srimathy Velam A K Sindhu Bairavi S N

Sprint-1	Authentication	USN-3	As a user, I can register login to the system with required credintials	2	High	Rabeena P Rajeshwari S
Sprint-2	Analysis of required data	USN-4	Regulating certain factors influencing the actions of the phenomenon	2	High	Srimathy Velam A K Sindhu Bairavi S N
Sprint-2	Accumulation of data	USN-5	Gathering the data required for the analysis	1	High	Rabeena P
Sprint -3	Organizing the unstructured data	USN-6	An organized collection of data stored in database to access and manage the data easily	2	Low	Rajeshwari S
Sprint-3	Selection of algorithm	USN-7	Choosig a required algorithm for the analysis	3	High	Sindhu Bairavi S N
Sprint-4	Report generation	USN-8	Generating a clear detailed report based on the analysis	3	High	Srimathy Velam A K Sindhu Bairavi S N Rabeena P

7. CODING AND SOLUTIONING

Model building

In [33]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [34]:
```
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,shear_range=0.2)
```

In [36]:
```
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [37]:
```
x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/Disasters/dataset/train_set",target_size=(64,64),
                                          batch_size=5,color_mode='rgb',class_mode='categorical')
```

```
Found 742 images belonging to 4 classes.
```

In [38]:
```
x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/Disasters/dataset/test_set",target_size=(64,64),
                                        batch_size=5,color_mode='rgb',class_mode="categorical")
```

```
Found 198 images belonging to 4 classes.
```

In [39]:
```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Conv2D,MaxPooling2D,Flatten
```

In [42]:
```
model=Sequential()
model.add(Conv2D(32,(3,3),activation="relu",input_shape=(64,64,3)))
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.compile(loss="categorical_crossentropy",metrics=["accuracy"],optimizer='adam')
```

In [43]:
```
model.summary()
```

In [43]:
```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896

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

conv2d_2 (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
_________________________________________________________________
```

In [49]:
```
model.fit_generator(generator=x_train,epochs=20,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

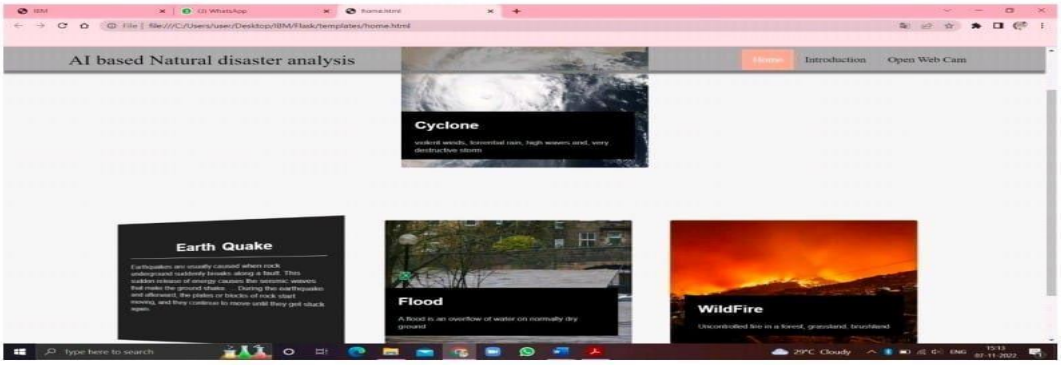
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versio
n. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/20
149/149 [==============================] - 40s 270ms/step - loss: 0.6694 - accuracy: 0.7507 - val_loss: 0.8159 - val_accuracy: 0.6970
Epoch 2/20
149/149 [==============================] - 40s 271ms/step - loss: 0.5971 - accuracy: 0.7736 - val_loss: 0.6273 - val_accuracy: 0.7727
Epoch 3/20
```

```
In [50]:
model.save('disaster.h5')
model_json=model.to_json()
with open("model-bw.json","w") as json_file:
  json_file.write(model_json)

In [51]:
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('disaster.h5')

In [52]:
x_train.class_indices

Out[52]: {'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}

In [53]:
img=image.load_img(r"/content/drive/MyDrive/Disasters/dataset/test_set/Earthquake/1329.jpg",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])

1/1 [==============================] - 0s 133ms/step
Earthquake

In [54]:
img=image.load_img(r"/content/drive/MyDrive/Disasters/dataset/test_set/Cyclone/900.jpg",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])

1/1 [==============================] - 0s 23ms/step
Cyclone
```

Creation of HTML pages

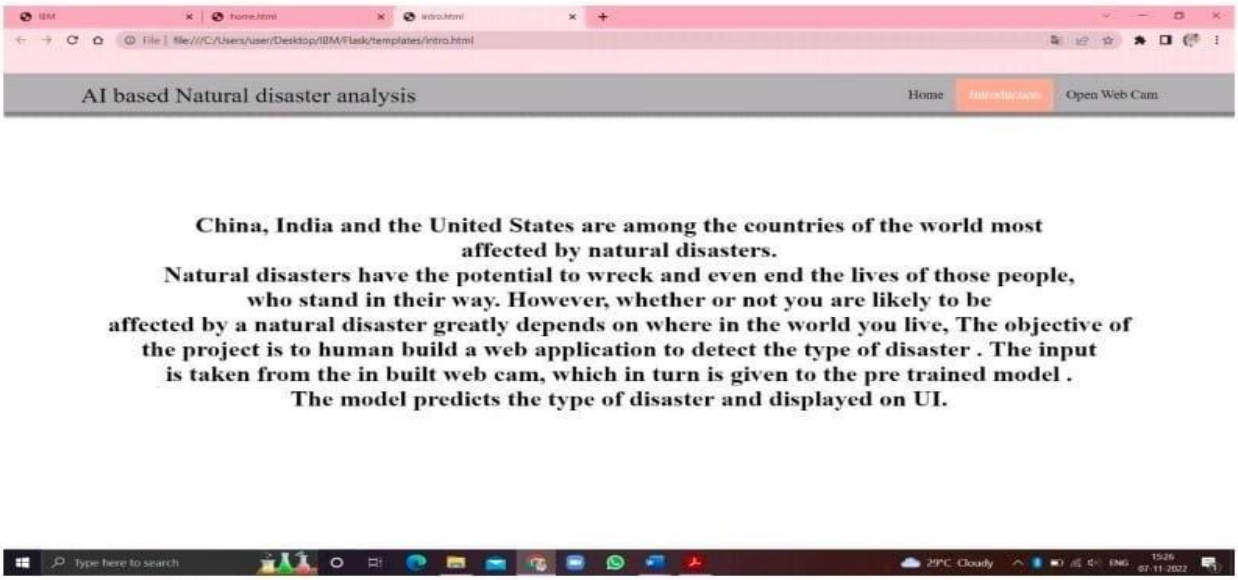
CREATION OF HOME PAGE:

Using HTML and CSS, the Home page is created. From the Home page the User can be able to know the basics of the frequently occurring Disasters. The home.html page is given below:



CREATION OF INTRO PAGE:

Using HTML and CSS, the intro page is created. From the intro page the user can be able to know about the project's introduction or abstract. The intro.html page is given below:



OPENING WEB CAM:

Using HTML and CSS, the upload.html page is created. Through this page User can be able to open the web cam to know about current disaster. The upload.html page is given below:

Building the python code

```
# 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(r'C:\Users\user\Desktop\IBM\Flask\templates\disaster.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'])
```

8. TESTING

Test cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the home page when click on the Local host ID		1 .Click on the local host ID 2.Verify Home page displayed or not	https://127.0.0.1:5000	Home page should display	Working as expected	Pass				
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Home page		1.Click on the Local host ID. 3.Verify Home page with below UI elements: a.Home b.Intro page c.Open Web Cam	https://127.0.0.1:5000	Application should show below UI elements: a.Home b.Intro page c.Open web cam	Working as expected	pass				
HomePage_TC_OO3	UI	Home	Verify user is able to see the some definition of natural disaster in Home.		1.Click on the local host ID 2.Click on Home 3.Verify Home with below UI elements: a.Cyclone with definition b.Earth quake with definition c.Wilde Fire with definition d.Flood with definition	https://127.0.0.1:5000	Application should show below UI elements: a.Cyclone with definition b.Earth quake with definition c.Wilde Fire with definition d.Flood with definition	Working as expected	Pass				
HomePage_TC_OO4	UI	Intro Page	Verify user is able to see introduction in Intro page		1.Click on the local host ID 2.Click on Intro page 3.Verify intro page with some intoduction	https://127.0.0.1:5000	Application should show Some Introduction about natural disaster	Working as expected	pass				
HomePage_TC_OO4	UI	Open web cam	Verify user is able to see UI elements in open web cam		1 .Click on the local host ID 2.Click on the Open web cam 3.Verify open web cam with below elements: a.Upload b.Predict	https://127.0.0.1:5000	Application should show Upload button and predict button	Working as expected	Pass				
HomePage_TC_OO5	UI	Upload	Verify user is able to upload an image		1 .Click on the local host ID 2.Click on the Open web cam 3.click on the Upload button 4.verify user to see images to upload in upload button 5.click on any image shows in upload button	https://127.0.0.1:5000	Application should upload an image	Working as expected	pass				
Home age_TC_006	UI	Predict			1.Click on the local host ID 2.Click on the Open web cam 3.click on the Upload button 4.Click on the image to upload 5.Click on the predict button 6.Verify user able to see output image	https://127.0.0.1:5000	Application should show output ima	working as expected	Fail	Output image not shows			

User acceptance testing

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	6	3	2	1	12
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	12	2	4	5	23
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	3	2	1	6
Totals	21	11	13	9	54

9. RESULTS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	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 (MaxPooling2D) — (None, 14, 14, 32) — 0 flatten (Flatten) — (None, 6272) — 0 dense (Dense) — (None, 128) — 802944 dense_1 (Dense) — (None, 4) — 516
2.	Accuracy	Training Accuracy - Validation Accuracy -	loss: 0.5239 - accuracy: 0.7857 - val_loss: 0.7225 - val_accuracy: 0.7576 loss: 0.4353 - accuracy: 0.8383 - val_loss: 0.7533 - val_accuracy: 0.7323 loss: 0.3964 - accuracy: 0.8544 - val_loss: 1.0309 - val_accuracy: 0.6364 loss: 0.3662 - accuracy: 0.8767 - val_loss: 0.5903 - val_accuracy: 0.7273 loss: 0.4363 - accuracy: 0.8342 - val_loss: 0.5633 - val_accuracy: 0.7475 loss: 0.3292 - accuracy: 0.8814 - val_loss: 0.5497 - val_accuracy: 0.7577

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

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

DISADVANTAGES:-

- It involves huge money to be equipped.
- Problems faced in life basic needs.
- 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**.**

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 twoblocks—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 ascompared 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.

12. FUTURE SCOPE

AI -smart technology, which has enabled accurate and speedy solutions. If harnessedproperly, 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 intensityhurricanes and read the geological data to understand volcanic eruptions, such systems canreduce 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 ableto predict flood and the regions that it would be affected due to the natural disaster with anaccuracy of over 90%. It was possible owing to the combination of data from governmentagencies that provide on-ground information – from measuring devices placed on the spot andsatellite 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 canhelp 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.

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-40254-1660626733

PROJECT DEMO LINK:

https://drive.google.com/file/d/18OxPP2IYHfTHlSLdxhPKlo5QjgawRaNS/view?usp=drivesdk