PROJECT REPORT

NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Submitted by

K. DURGA VIJAYA LAKSHMI - 950819106015
T. SOWNDARYA - 950819106090
P.SRI DEVI - 950819106091
T. SUDARMATHI - 950819106093

In partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING In
ELECTRONICS AND COMMUNICATION ENGINEERING,

GOVERNMENT COLLEGE OF ENGINEERING,
TIRUNELVELI - 627 007.
ANNA UNIVERSITY : CHENNAI - 600 025.

NOVEMBER 2022.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 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.

1.2 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 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.

2.LITERATURE SURVEY

2.1 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-ofthe-art tools and technology. Thus, future efforts need to focus on combining disaster management knowledge, image processing techniques and machine learning tools to ensure effective and holistic disaster management across all phases.

2.2 References

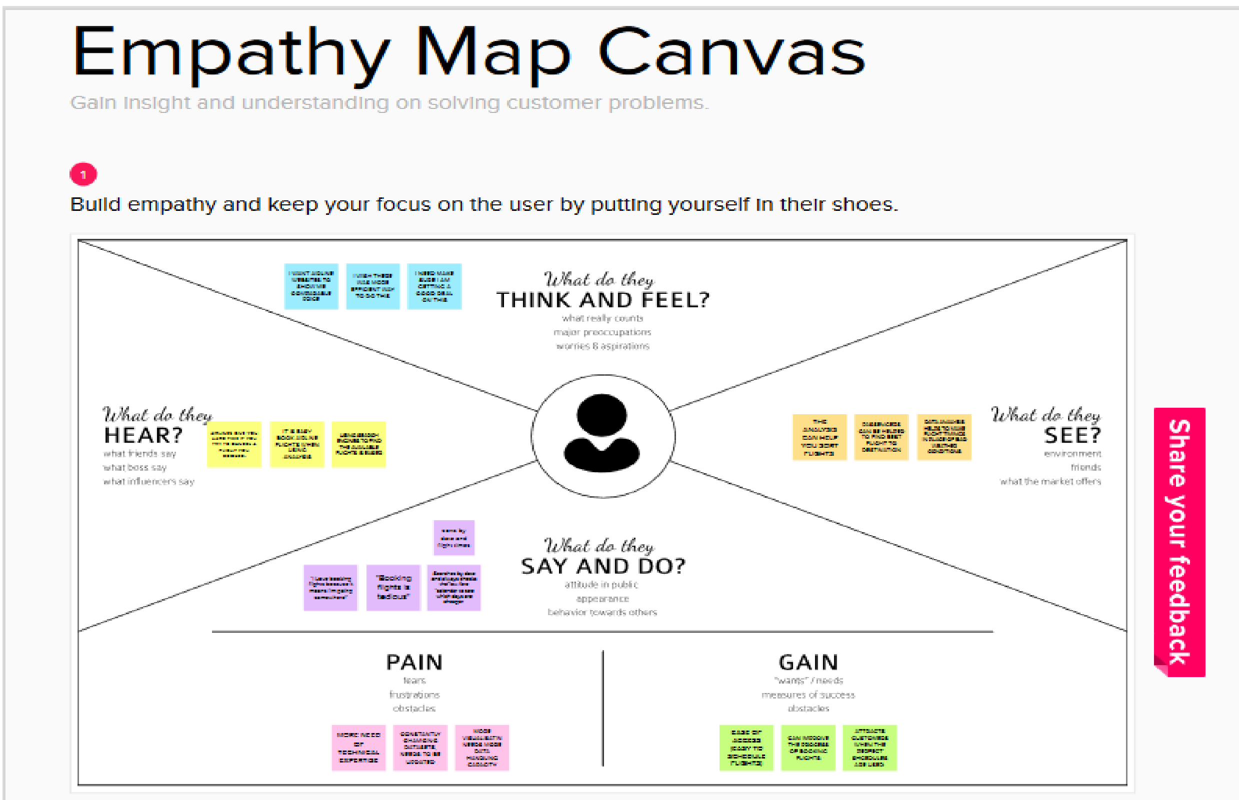
Deep Learning Based Forest Fire Classification And Detection In Satellite Images.
Author: **R.Shanmuga priya, K.Kani**

2.3 Problem statement definition

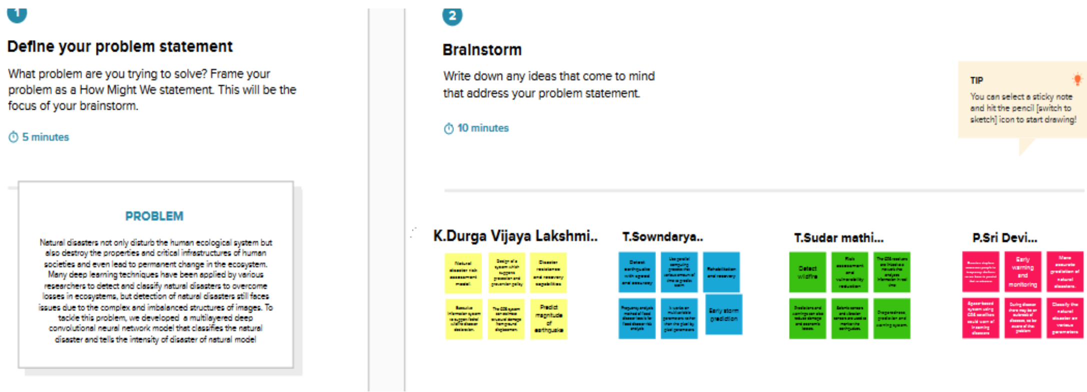
Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem.Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images.

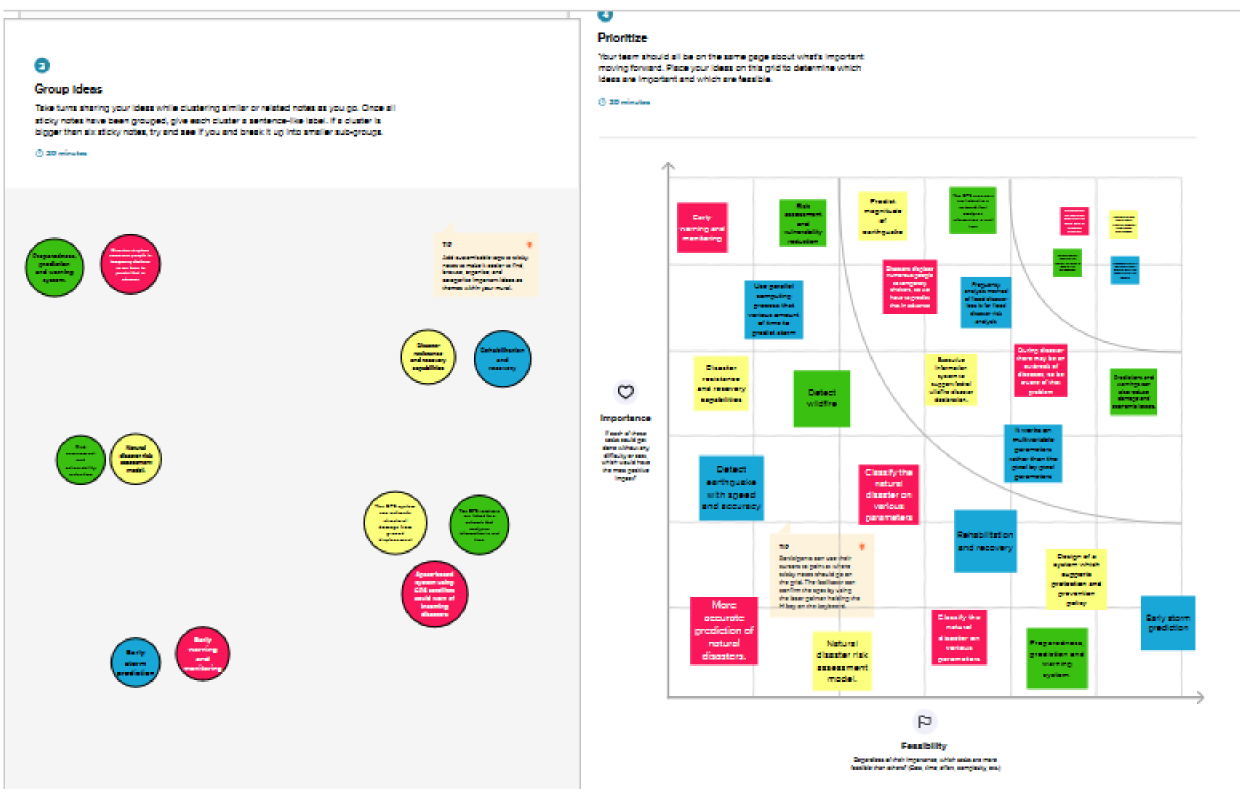
3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy map canvas



3.2 Ideation and Brainstorming





3.3 Proposed Solution

S.NO	Parameter	Description
1.	**Problem statement(problem to be solved)**	To tackle the problem of detecting natural disasters ,we developed a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of natural disaster.
2.	**Ideas/solution description**	By predicting to occurrence of natural disaster, we can save thousands of lives and take appropriate measures to reduce property damage.
3.	**Novelty/Uniqueness**	It finds the magnitude of impact , length of fore warming and duration of impact.
4.	**Social impact/customer satisfaction**	The most vulnerable are citizens and children .it can save lives of people can minimize the loss of infrastructure, finance.
5.	**Business Model (Revenue model)**	The government and private companies make use of this to get revenue in future.
6.	**Scalability of the solution**	Disaster damages are measured involves examining the number of fatalities, of injuries, of people affected.

3.4 Problem Solution Fit

Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

The global GIS in disaster management market size stood at $2.3 billion in 2019, and it is expected to reach $9.4 billion by 2030, exhibiting a CAGR of 13.7% during the forecast period (2020 – 2030). The major factors supporting the growth of the industry include the surging number of natural disasters, strong focus of government and emergency management organizations on adopting advanced GIS solutions, high need for analyzing geospatial data, and increasing public awareness about reducing the socioeconomic impact of natural disasters.

6. CUSTOMER CONSTRAINTS — CC

Awareness, education, preparedness, and prediction and warning systems can reduce the disruptive impacts of a natural disaster on communities. Mitigation measures such as adoption of Zoning, land-use practices, and building codes are needed, however, to prevent or reduce actual damage from haZards.

5. AVAILABLE SOLUTIONS — AS

Planning to warn the people which will minimize the effects of disasters .Recovery and reconstruction.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

Natural disasters can cause great damage on the environment, property, wildlife and human health. These events may include earthquakes, floods, hurricanes, tornadoes, tsunamis, landslides, wildfires, volcanic eruptions,extreme temperatures.

Property damage.

Structural damage to buildings.

Loss of utilities like electricity and water.

9. PROBLEM ROOT CAUSE — RC

The lack of resources and capacties (e.g., financial, human and technical) and a low level of knowledge an education emerged in all case studies as major root causes for several drivers of disaster risk.

7. BEHAVIOUR — BE

Analysis of public behavior plays an important role in crisis management, disaster response, and evacuation planning. Unfortunately, collecting relevant data can be costly and finding meaningful information for analysis is challenging. A growing number of Location-based Social Network services provides time-stamped, geo-located data that opens new opportunities and solutions to a wide range of challenges.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

Large economic losts, reduced accumulation of capital and infrastructure, long recovery period after disasters.

4. EMOTIONS: BEFORE / AFTER — EM

Before the disaster, a positive association was found between place-identity and wellbeing, indicating that the stronger emotions participants evolved to the place, as well as remembered more and thought about the place, the stronger wellbeing they experienced at the site. After the disaster, the strength of this relationship decreased more than twice, accounted for by the weakening of the emotion-wellbeing link

10. YOUR SOLUTION — SL

Natural disasters cannot be prevented but they can be detected.

We can measure disaster risk by analysing trends of, for instance, previous disaster losses. These trends can help us to gauge whether disaster risk reduction is being effective. We can also estimate future losses by conducting a risk assessment.

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE

We demonstrate how to improve investigation by analyzing the extracted public behavior responses from social media before, during and after natural disasters, such as hurricanes and tornadoes.

8.2 OFFLINE

Dissemination of information from nearby Government agencies and NGO'S.

Extract online & offline CH of BE

AMALTAMA

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

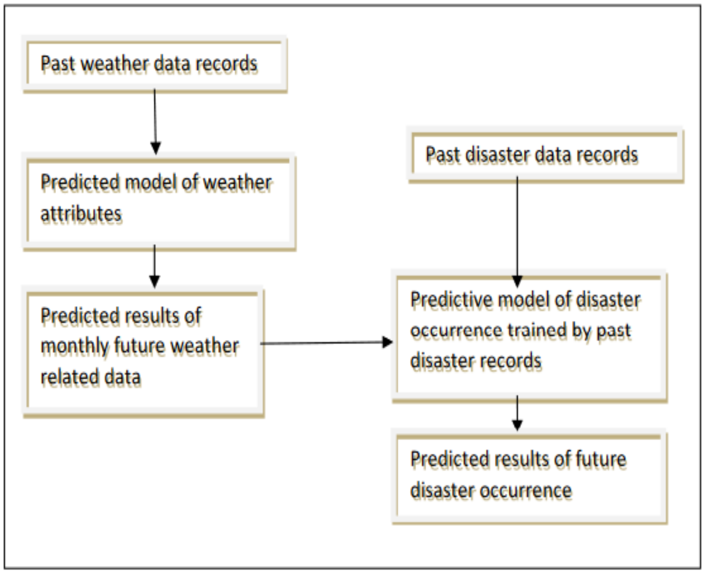
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Register through mobile application Call the given emergency number
FR-2	User Confirmation	Confirmation via Call back Confirmation via Text
FR-3	User Preparation	Ensure safety of all people Supply of canned food
FR-4	User evacuation	Waiting for evacuation team Take refugee in nearest safe location

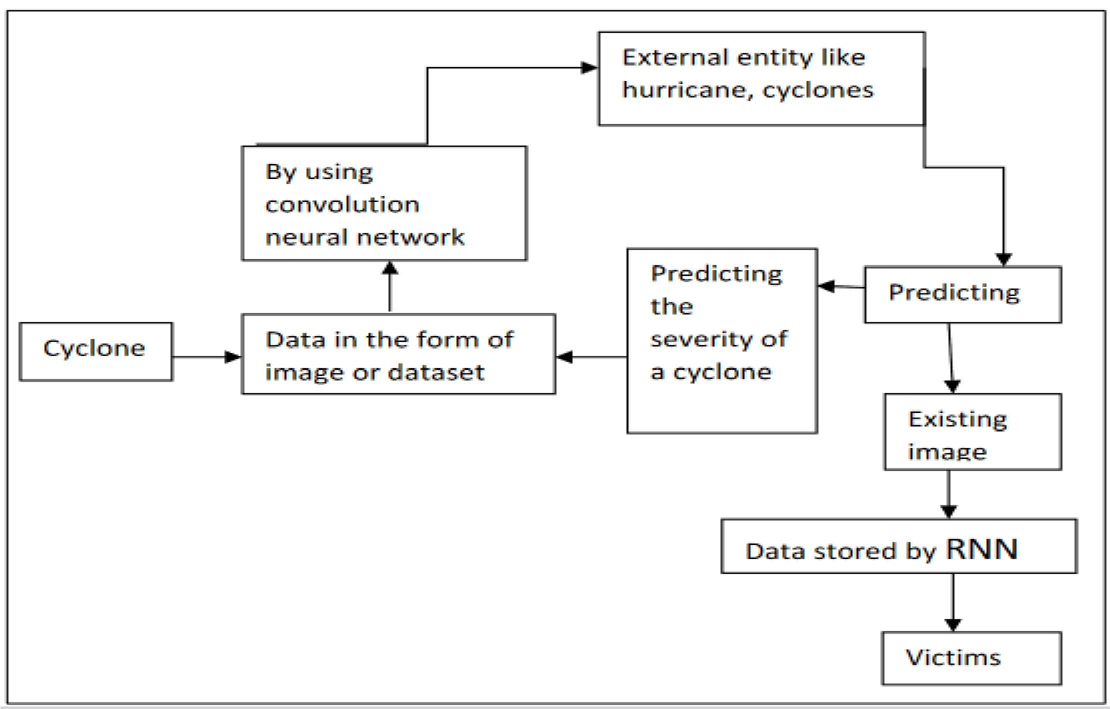
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is easy and quick method to predict the disasters.
NFR-2	**Security**	The secure pattern shares components with monitor and control for logging and control access and for providing audit trails.
NFR-3	**Reliability**	It should be highly reliable.
NFR-4	**Performance**	It deals with the measure of the system's response time.
NFR-5	**Availability**	It can be available at the any time and we can access during any disasters.
NFR-6	**Scalability**	Disaster damages are measured involves examining the number of fatalities, of injuries, of people affected.

5.PROJECT DESIGN

5.1 Data flow diagram





5.2 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can login with my password	Medium	Sprint-1
	Login			I can see	High	Sprint-1

		USN-5	As a user, I can log into the application by entering email & password	the dashboard now		
	Dashboard	USN-6	As a user, I can update Disaster incidents.	I can update now.	Medium	Sprint-1
Customer (Web user)		USN-7	As a user, I can view Map Data.	I can see Map Data.	Medium	Sprint-1
Customer Care Executive	Authentication	USN-8	As a Community Leader, I can log into the application using my password	I can access my account.	High	Sprint-1
		USN-9	As a Community Leader, I can apply for membership.	I can apply membership.	High	Sprint-1
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
		USN-10	As a Community Leader, I can verify Disaster.	Disaster verification	High	Sprint-1
System Administrator	Membership Approval	USN-11	As a administrator, I can approve the Membership	I can approve membership.	High	Sprint-1

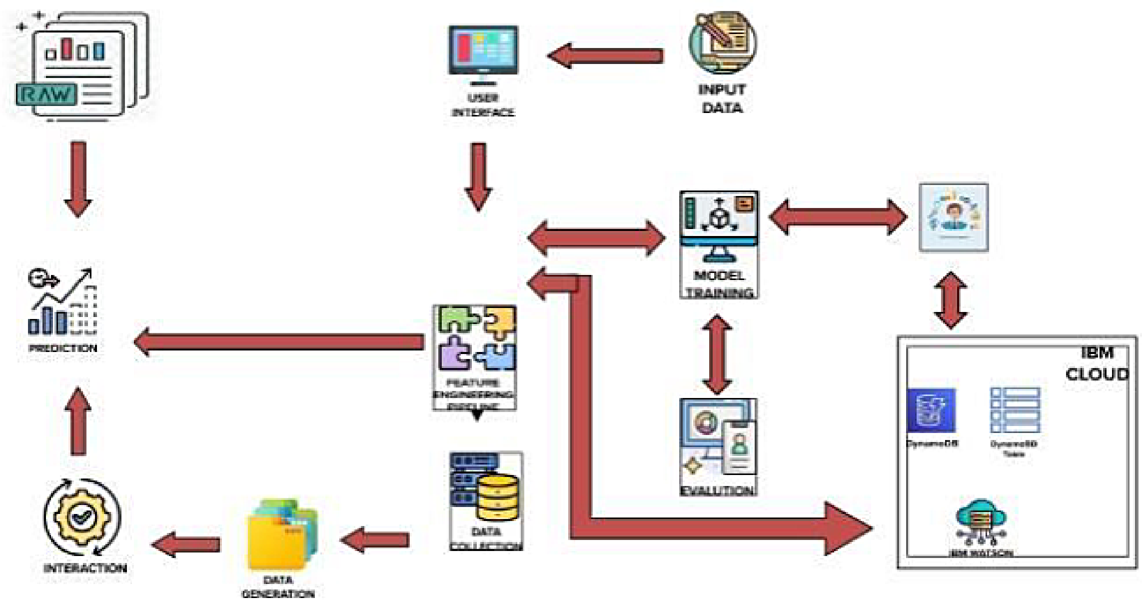
			application.			
	Update Disaster information	USN-12	As a administrator, I can update information about Disaster.	I can update disaster informatio n.	High	Sprint-1
	Disaster verification	USN-13	As a administrator, I can verify disaster.	I can verify Disaster	High	Sprint-1
Community Leader and System Administrat or	Disaster Queries	USN-14	Both are can able to ask disaster queries.	We can ask Queries about disaster.	Low	Sprint-2

5.3 Solution And Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
3. Define features, development phases, and solution requirements.

Provide specifications according to which the solution is defined, managed, and delivered.



6 PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation

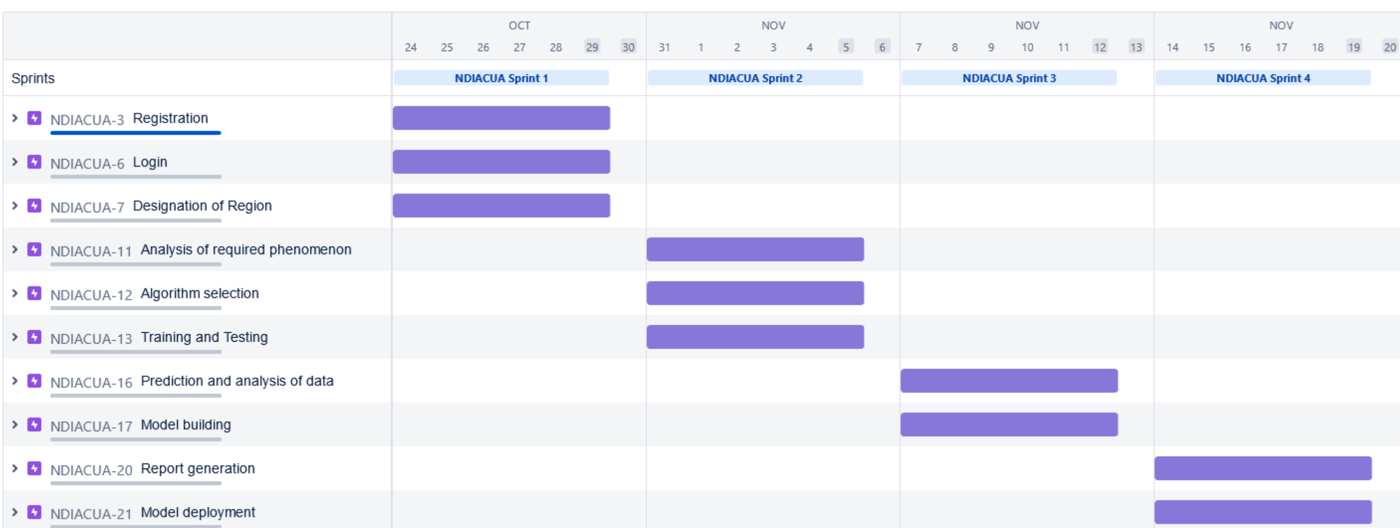
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming that.	2	Low	Haja Mydeen
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application.	3	High	Harini
Sprint-1	Login	USN-3	As a user, I adapt to logging into the system with	2	Low	Vanish

			credentials.			
Sprint-1	Designation of Region	USN-4	As a user, I can collect the dataset and select the region of interest to be monitored and analysed.	5	Medium	Veeramakali
Sprint-2	Analysis of required phenomenon	USN-5	As a user, I can regulate certain factors influencing the action and report on past event analysis.	4	High	Vanish
Sprint-2	Algorithm selection	USN-6	As a user, I can choose the required algorithm for specific analysis.	4	Medium	Harini
Sprint-2	Training and Testing	USN-7	As a user, I can train and test the model using the algorithm.	4	High	Veeramakali
Sprint-3	Prediction and analysis of data	USN-8	As a user, I can predict and visualise the data effectively.	4	High	Haja Mydeen
Sprint-3	Model building	USN-9	As a user, I can build with the web application.	8	High	Vanish
Sprint-4	Report generation	USN-10	As a user, I can generate detailed report on product data analysis.	4	High	Harini
Sprint-4	Model deployment	USN-11	As an administrator, I can maintain third-party services.	**8**	High	Veeramakali

6.2 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	29 Oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	12	6 Days	07 Nov 2022	12 Nov 2022	12	12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

6.3 Reports from JIRA



7 CODING AND SOLUTIONING

7.1 Feature 1

Train and Test(1).ipynb ×

C: > Users > USER > Downloads > Train and Test(1).ipynb > ls

+ Code + Markdown | Run All Clear Outputs of All Cells | Outline ···

```
[6]
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[7]
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,shear_range=0.2)
```

```
[8]
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
[14]
x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/Disaster/dataset/train_set",target_size=(64,64),
                                          batch_size=5,color_mode='rgb',class_mode='categorical')
```

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

```
[15]
x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/Disaster/dataset/test_set",target_size=(64,64),
                                        batch_size=5,color_mode='rgb',class_mode="categorical")
```

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

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

```
model=Sequential()
model.add(Conv2D(32,(3,3),activation="relu",input_shape=(64,64,3)))
model.add(MaxPooling2D(poo activation: Any
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')
```

```
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
```

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

```
model.fit_generator(generator=x_train,epochs=20,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

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

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('disaster.h5')
```

```
x_train.class_indices
```

```
{'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}
```

```
img=image.load_img(r"/content/drive/MyDrive/Disaster/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 121ms/step
Earthquake
```

```
img=image.load_img(r"/content/drive/MyDrive/Disaster/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 20ms/step
Cyclone
```

7.2 Feature 2

application.py* ×

```python

from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('homepage.html', title='Disaster Classifier | Home', active_page='home')

@app.route('/intro')
def intro():
    return render_template('intro.html', title='Disaster Classifier | About', active_page='intro')

@app.route('/launch')
def launch():
    return render_template('launch.html', title='Disaster Classifier | Launch', active_page='launch')

if __name__ == '__main__':
    app.run(debug=True)
```

launch.html × layout.html × layout1.html × intro.html ×

```
"""

{%extends "layout.html"%}

{%block head%}
  <link rel="stylesheet" href="{{url_for('static', filename='intro_style.css')}}">
{%endblock head%}

{%block body%}
<div>
    <div class="block animated fadeIn text">
        <div>
            <p>
                China, India and United States tend to be the most affected by Natural Disasters. These disasters can wreck havoc and
                even end the lives of those who stand in their way. However, The Geographical location where people live mostly decides the extent
                that its residents get affected by Disasters.
            </p>
            <br>
            <p>
                This Web App is built with the objective of Detecting and Alerting the Public about the Type of Disaster.
                Upload an Image or a Video Feed captured or obtained, and this data is fed to a Trained CNN Model.
                The Deep Learning Model predicts the type of Natural Disaster among Cyclone, Earthquake, Flood, Wildfire, and
                alerts it to the user.
            </p>

        </div>
    </div>
    <div class="animated fadeIn text">
    <img src="static\Blank diagram.png"/>
    </div>
    <div class="block animated fadeIn text">
    </div>
</div>
{%endblock body%}
```

8 TESTING

8.1 Test Case

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	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 infodaction	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 bellow 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				
ome 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 shown			

8.2 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

9.1 Performance Metrics

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Model Summary</td><td>-</td><td>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 2D)   (None, 14, 14, 32)   0

flatten (Flatten)   (None, 6272)   0

dense (Dense)   (None, 128)   802944

dense_1 (Dense)   (None, 4)   516

===</td></tr>
</table>

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: 0.7538 - val_accuracy: 0.7323 loss: 0.3964 - accuracy: 0.8544 - val_loss: 1.0309 - val_accuracy: 0.6364 loss: 0.3662 - accuracy: 0.8787 - val_loss: 0.6908 - val_accuracy: 0.7273 loss: 0.4363 - accuracy: 0.8342 - val_loss: 0.6638 - val_accuracy: 0.7475 loss: 0.3292 - accuracy: 0.8814 - val_loss: 0.6497 - val_accuracy: 0.7577

10. ADVANTAGES AND 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 weatherand the days ahead.

3. Helpful in getting life back on track..

4. Their Alert nature able to respond effectively and efficiently which defend the societyfrom 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 andincreasing 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.

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.

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 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 thefuture, 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

Train and Test(1).ipynb

C: > Users > USER > Downloads > Train and Test(1).ipynb > ls

```
[6]
from tensorflow.keras.preprocessing.image import ImageDataGenerator

[7]
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,shear_range=0.2)

[8]
test_datagen=ImageDataGenerator(rescale=1./255)

[14]
x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/Disaster/dataset/train_set",target_size=(64,64),
                                          batch_size=5,color_mode='rgb',class_mode='categorical')

Found 742 images belonging to 4 classes.

[15]
x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/Disaster/dataset/test_set",target_size=(64,64),
                                        batch_size=5,color_mode='rgb',class_mode="categorical")

Found 198 images belonging to 4 classes.
```

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

```python
model=Sequential()
model.add(Conv2D(32,(3,3),activation="relu",input_shape=(64,64,3)))
model.add(MaxPooling2D(poo activation: Any
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')
```

```python
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
```

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

```python
model.fit_generator(generator=x_train,epochs=20,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

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

```python
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('disaster.h5')
```

```python
x_train.class_indices
```

```
{'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}
```

```python
img=image.load_img(r"/content/drive/MyDrive/Disaster/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 121ms/step
Earthquake
```

```python
img=image.load_img(r"/content/drive/MyDrive/Disaster/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 20ms/step
Cyclone
```

Flask app.py

application.py*

```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('homepage.html', title='Disaster Classifier | Home', active_page='home')

@app.route('/intro')
def intro():
    return render_template('intro.html', title='Disaster Classifier | About', active_page='intro')

@app.route('/launch')
def launch():
    return render_template('launch.html', title='Disaster Classifier | Launch', active_page='launch')

if __name__ == '__main__':
    app.run(debug=True)
```

HTML Code

launch.html | layout.html | layout1.html | intro.html

```html
"""

{%extends "layout.html"%}

{%block head%}
  <link rel="stylesheet" href="{{url_for('static', filename='intro_style.css')}}">
{%endblock head%}

{%block body%}
<div>
    <div class="block animated fadeIn text">
        <div>
            <p>
                China, India and United States tend to be the most affected by Natural Disasters. These disasters can wreck havoc and
                even end the lives of those who stand in their way. However, The Geographical location where people live mostly decides the extent
                that its residents get affected by Disasters.
            </p>
            <br>
            <p>
                This Web App is built with the objective of Detecting and Alerting the Public about the Type of Disaster.
                Upload an Image or a Video Feed captured or obtained, and this data is fed to a Trained CNN Model.
                The Deep Learning Model predicts the type of Natural Disaster among Cyclone, Earthquake, Flood, Wildfire, and
                alerts it to the user.
            </p>

        </div>
    </div>
    <div class="animated fadeIn text">
    <img src="static\Blank diagram.png"/>
    </div>
    <div class="block animated fadeIn text">
    </div>
</div>
{%endblock body%}
```

GITHUB :

https://github.com/IBM-EPBL/IBM-Project-34152-1660232021

PROJECT DEMO :

https://drive.google.com/file/d/1hTAEWRoRxu8eeLW8kzOyoSEE604HWa6B/view?usp=share_link