PROJECT REPORT

1.INTRODUCTION

1.1 PROJECT OVERVIEW

A **natural disaster** is "the negative impact following an actual occurrence of natural hazard in the event that it significantly harms a community". A natural disaster can cause loss of life or damage property, and typically leaves some economic damage in its wake. The severity of the damage depends on the affected population's resilience and on the infrastructure available. Some examples of the natural disasters are cyclone, earthquake, forest fire, floods, tornados etc.

Fortunately, new technologies can help detect and prepare for extreme weather and other hazards, as well as communicate to people and communities effectively about the necessary response. AI can help response teams understand natural hazards, monitor events in real time, and anticipate specific risks in the face of impending or on-going disasters. Using Artificial Intelligence, we are trying to reduce the impact on natural disaster on Human Civilization.

1.2 PURPOSE

The purpose of the project is to make use of Artificial Intelligence to help humans to overcome the impacts of natural disasters and getting the better understanding of the natural disasters. The artificial intelligence algorithm will able to classify what the natural disaster is and predict the intensity. For the problem of classifying the natural disaster we are using convolutional neural network. The entire thing can be deployed into cloud that every single user can use this application.

1. LITERATURE SURVEY

1.1 EXISTING PROBLEM

An-made climate change has increased the frequency and the severity of calamities such as droughts, floods, famines, and hurricanes. This has led to automating disaster preparedness and recovery actions using AI and ML solutions. In the past 50 years, the number of recorded disasters has increased five times, and the resulting economic losses have increased seven times, according to the 2020 State of Climate Services Report by the World Meteorological Organization (WMO). In 2018, around 108 million people required humanitarian aid because of natural disasters. The WMO estimates the number to go up by almost 50 percent by 2030 at a staggering cost of USD 20 billion. Disaster Management committees are adopting technology such as Artificial Intelligence (AI) and Machine Learning (ML) to minimize the damage caused by such disasters by predicting their occurrences with great accuracy and assisting with the relief efforts.

Drone imagery, satellite data, and climate data can train AI and ML models as Disaster preparedness technology. They can study weather patterns to predict the intensity of rain or heat waves. They can also analyse sensor data and satellite imagery to forecast the location of future hurricanes.

1.2 REFERENCE

@article{article,

author = {Muhammad Aamir, Muhammad Aamir and Ali, Tariq and Irfan, Muhammad and Shaf, Ahmad and Azam, Muhammad and Glowacz, Adam and Brumercik, F. and Glowacz, Witold and Alqhtani, Samar M. and Rahman, Saifur},

year = {2021},

month = {04}, pages

= {2648},

title = {Natural Disasters Intensity Analysis and Classification Based on Multispectral Images

Using Multi-Layered Deep Convolutional Neural Network},

volume = {21}, journal = {Sensors}, doi =

{10.3390/s21082648}

}

TY - JOUR

AU - Madichetty, Sreenivasulu

AU - M., Sridevi

PY - 2021

DA - 2021/01/01

TI - A stacked convolutional neural network for detecting the resource tweets during a disaster

JO - Multimedia Tools and Applications

SP - 3927

EP - 3949

VL - 80

IS - 3

AB - Social media platform like Twitter is one of the primary sources for sharing real-time information at the time of events such as disasters, political events, etc. Detecting the resource tweets during a disaster is an essential task because tweets contain different types of information such as infrastructure damage, resources, opinions and sympathies of disaster events, etc. Tweets are posted related to Need and Availability of Resources (NAR) by humanitarian organizations and victims. Hence, reliable methodologies are required for detecting the NAR tweets during a disaster. The existing works don't focus well on NAR tweets detection and also had poor performance. Hence, this paper focus on detection of NAR tweets during a disaster. Existing works often use features and appropriate machine learning algorithms on several Natural Language Processing (NLP) tasks. Recently, there is a wide use of Convolutional Neural Networks (CNN) in text classification problems. However, it requires a large amount of manual labeled data. There is no such large labeled data is available for NAR tweets during a disaster. To overcome this problem, stacking of Convolutional Neural Networks with traditional feature based classifiers is proposed for detecting the NAR tweets. In our approach, we propose several informative features such as aid, need, food, packets, earthquake, etc. are used in the classifier and CNN. The learned features (output of CNN and classifier with informative features) are utilized in another classifier (meta-classifier) for detection of NAR tweets. The classifiers such as SVM, KNN, Decision tree, and Naive Bayes are used in the proposed model. From the experiments, we found that the usage of KNN (base classifier) and SVM (meta classifier) with the combination of CNN in the proposed model outperform the other algorithms. This paper uses 2015 and 2016 Nepal and Italy earthquake datasets for experimentation. The experimental results proved that the proposed model achieves the best accuracy compared to baseline methods.

SN - 1573-7721

UR - https://doi.org/10.1007/s11042-020-09873-8

DO - 10.1007/s11042-020-09873-8

ID - Madichetty2021

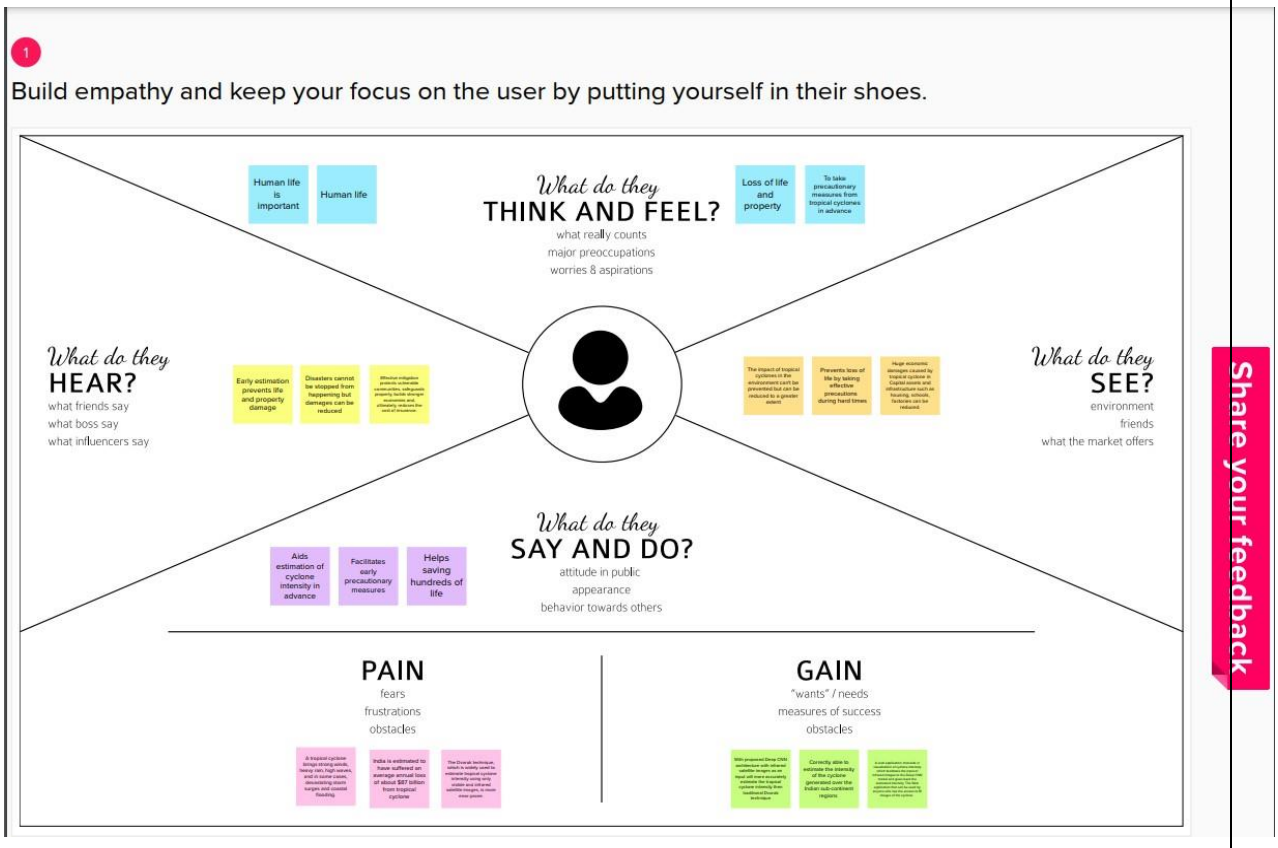
ER -

1.3 PROBLEM STATEMENT DEFINITION

We are building the CNN model, that takes the live video of the natural disaster and classify which class it belongs to. Deploying the entire architecture will enable all the user to access it.

2. IDEATION & PROPOSED SOLUTION

2.1 EMPATHY MAP CANVAS



2.2 IDEATION AND BRAINSTROMING

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

1 / 1

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICA

To detect the natural disasters

5 minutes

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

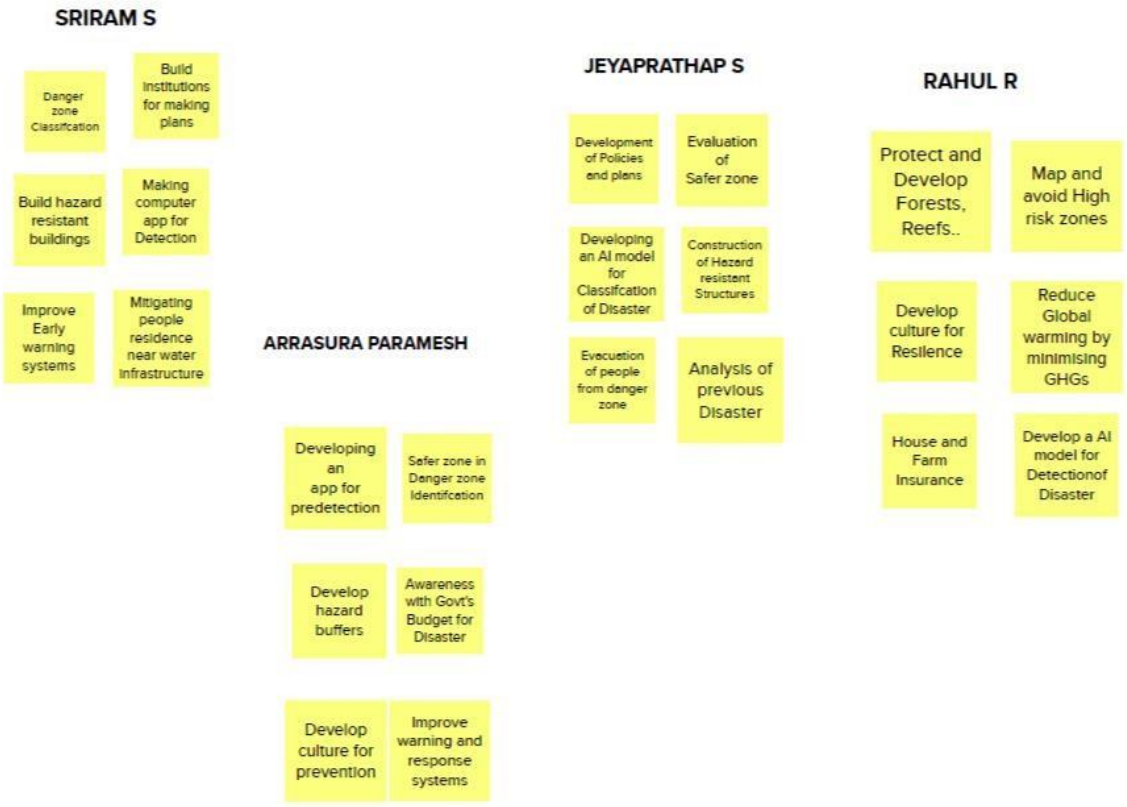
- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

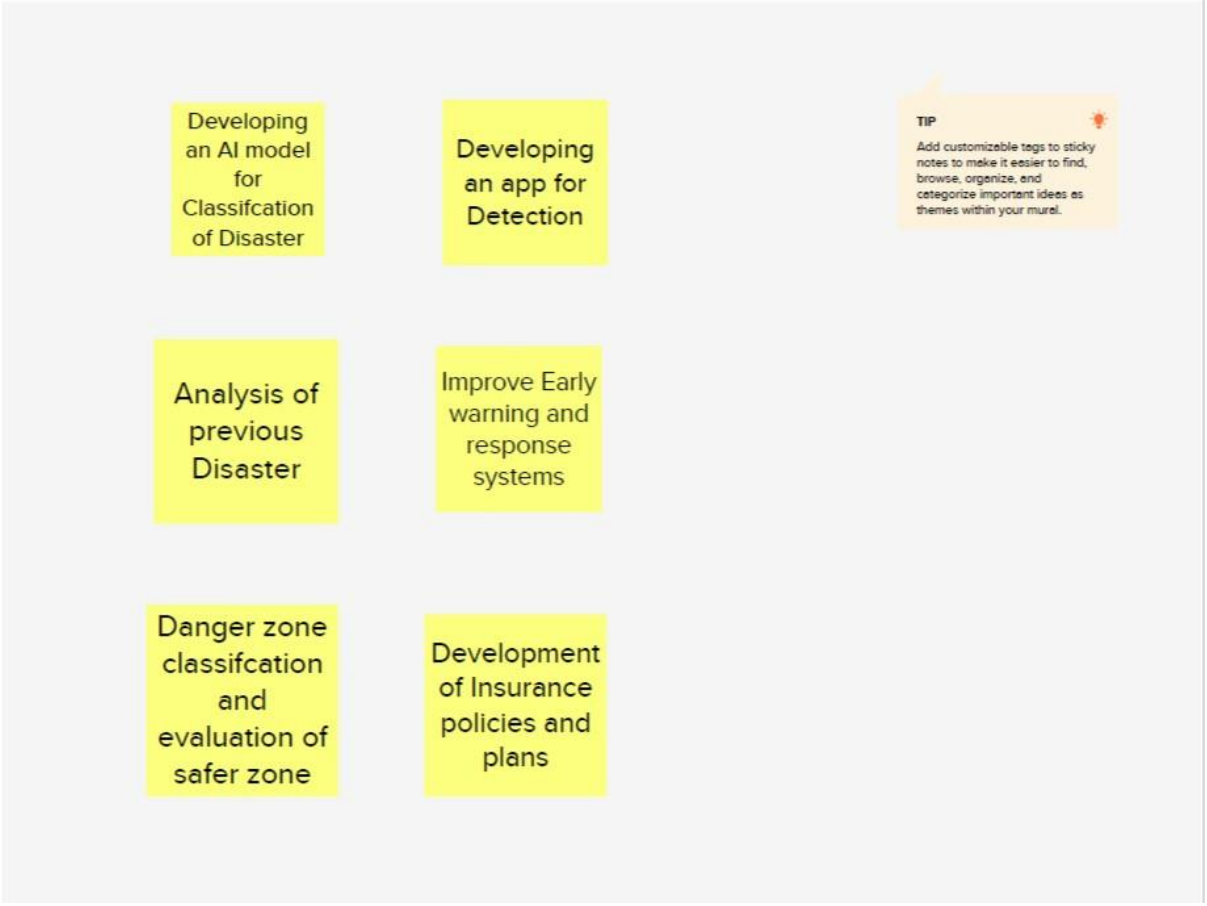


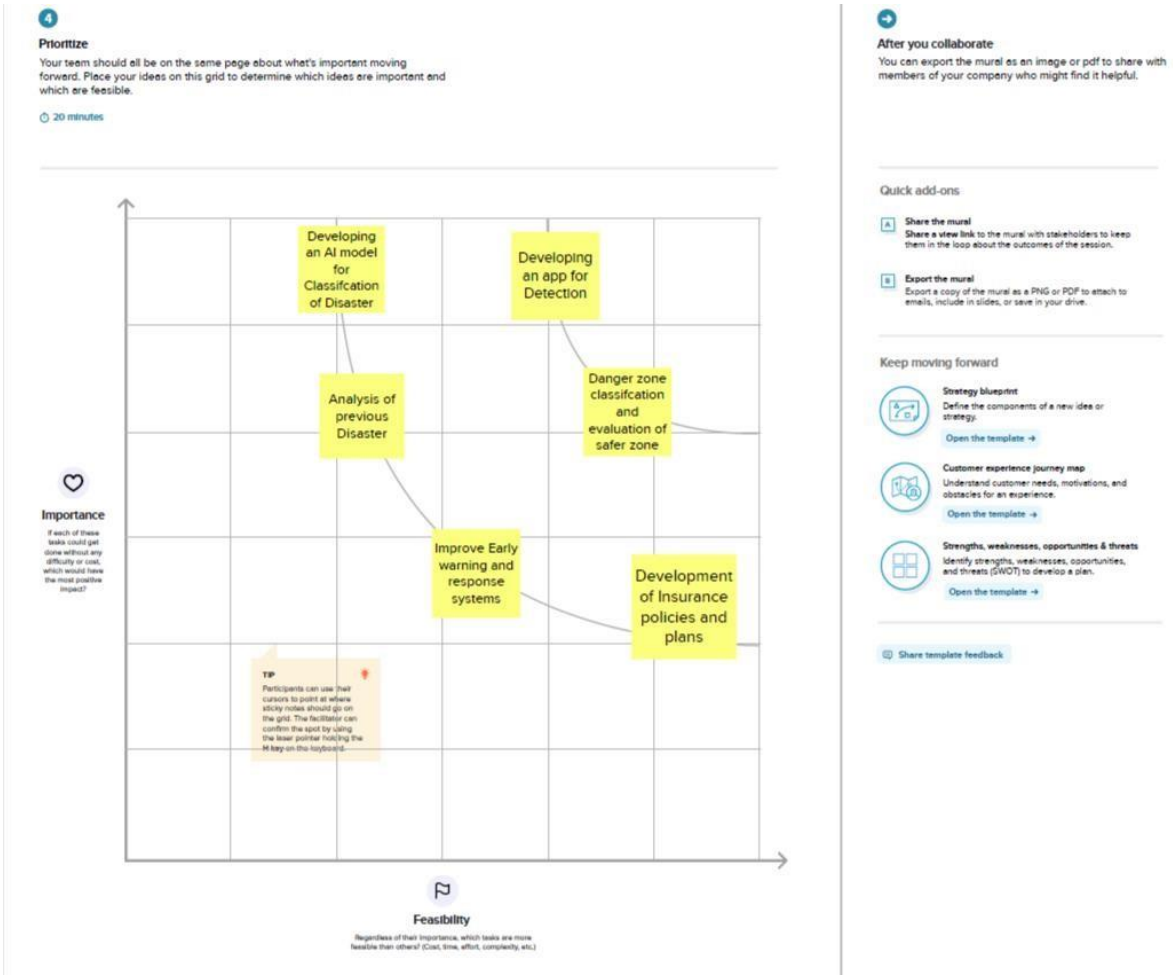
3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes



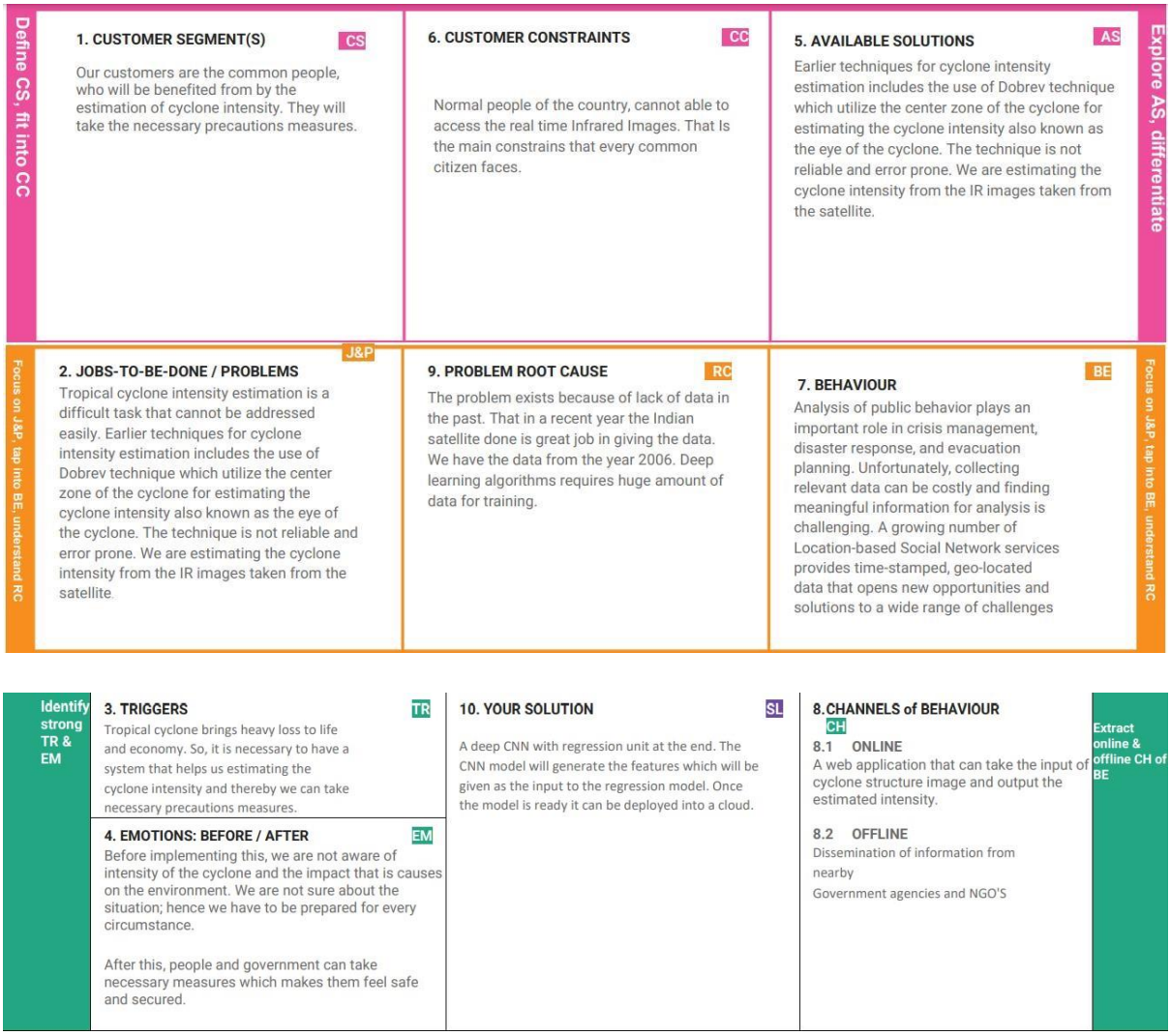


2.3 PROPOSED SOLUTION

S No	Parameter	Description
1	Problem Statement (Problem to be solved)	Tropical Cyclone brings heavy loss to life and property across the nation. The devasting effects of the cyclone cannot be prevented at all, but safety measures can be taken in advance. The solution aims at estimating the intensity of the tropical cyclone from the IR images of cyclone.
2	Idea / Solution description	We are proposing the Deep Learning based CNN architecture

		for estimating the intensity of the cyclone. The UNet based CNN model is used, which takes the IR images of the cyclone taken from the INSAT-3RD satellite generates the features. The features obtained from the CNN model is given into the regression model which estimate the intensity of the cyclone
3	Novelty / Uniqueness	We are proposing our very own architecture, which has single CNN model for feature extraction and a machine learning model for regression.
4	Social Impact / Customer Satisfaction	The model will be hosted in the cloud. In the web application, the user can upload the IR image of the cyclone it gives back the estimated intensity of the cyclone. The IR images can be obtained from mosdac website. The INSAT 3RD satellite will send the IR imagery of Indian subcontinent for every 15 mins
5	Business Model (Revenue Model)	Once the model is deployed, we can use a pay per use or subscription model. Where use can pay for their usage
6	Scalability of the Solution	The proposed architecture is highly scalable, It can correctly predict the intensity of cyclone that can occur over any parts of Indian Subcontinent.

2.4 PROPOSED SOLUTION FIT



3. REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR NO 1	User Registration	Registration through organization
FR NO 2	User Confirmation	Confirmation via Email Confirmation via OTP
FR NO 3	User Accessibility	Confirmation of accessibility for image data to certain organization via email provided
FR NO 4	User Accessibility 2	Accessible to instant weather data.
FR NO 5	Location Accessibility	Gain access to user location via mobile GPS to show the instant cyclone intensity

FR NO 6	Internet Availability	Internet is required to access the web page.

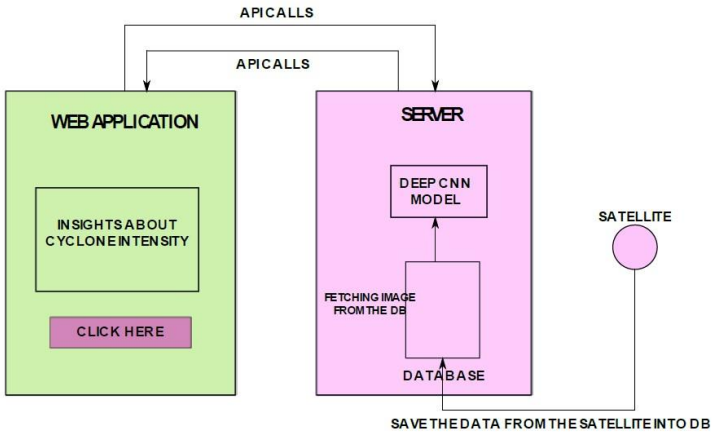
3.2 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements of the proposed solution.

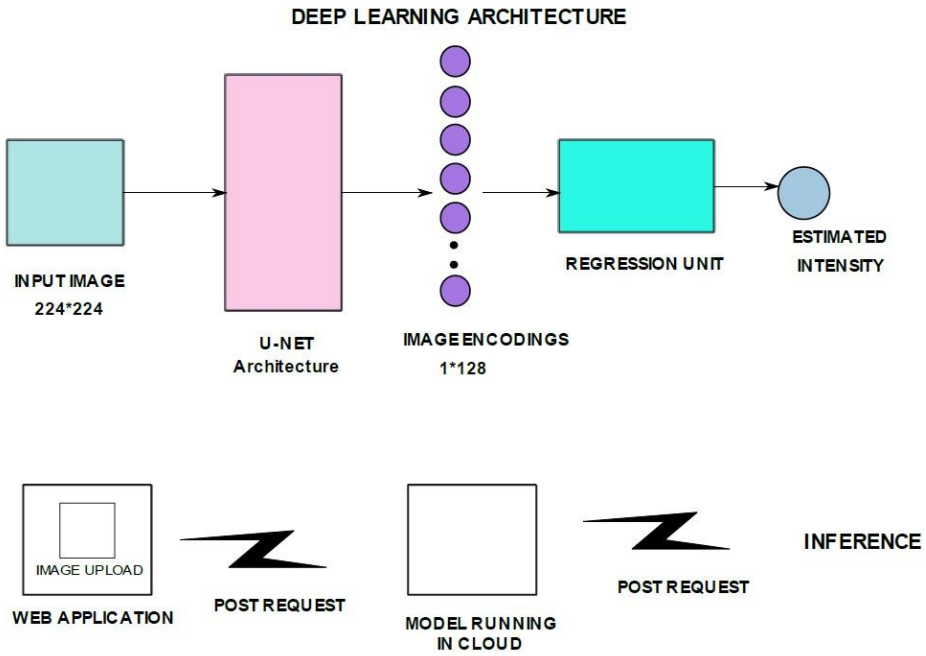
NFR No.	Non-Functional Requirement	Description
FR No. 1	Usability	Interactive UI is developed which facilitate easy navigation on the web page.
FR No. 2	Security	It is highly secured since; the weather IR image data can be accessed by users of certain organization
FR No. 3	Reliability	It is highly reliable since intensity of the cyclone is calculated for every 15 mins.
FR No. 4	Performance	Since the model is trained with range of data it shows increased performance on every new instance
FR No. 5	Availability	It can be made available 24/7.
FR No. 6	Scalability	It is highly scalable since fast API is used to make the API which can handle more request per second.

4. PROJECT DESIGN

4.1 DATA FLOW DIAGRAMS



4.2 SOLUTIONS AND TECHNICAL ARCHITECTURE



4.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	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
Customer	Login	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
Customer	Login	USN-3	As a user, I can register for the application through Gmail	I can assure that	High	Sprint-2
Customer	Login	USN-4	As a user, I can log into the application by entering email & password	Yes, I can assure that	Medium	Sprint-1
Customer	Login	USN-5	As a user I can gain access to weather insights	Yes, about the intensity	High	Sprint-1
Customer	Dashboard	USN-6	As a user under registered organisation, I get access to IR image of cyclone data.	Yes, that is certain	Medium	Sprint-2
Customer	Model	USN-7	Model should be made available so that, it makes estimation regarding the intensity	Yes, that is necessary	High	Sprint-2
Customer	APIs		API should be made available which facilitates the transfer of IR images from the satellite.	Yes, that is necessary	High	Sprint-2
Customer	APIs 2		Update the weather information for every 15 minutes based on the data from the satellite	Yes, that is necessary	High	Sprint-2

5. PROJECT PLANNING & SCHEDULING

5.1 SPRINT PLANNING AND ESTIMATION

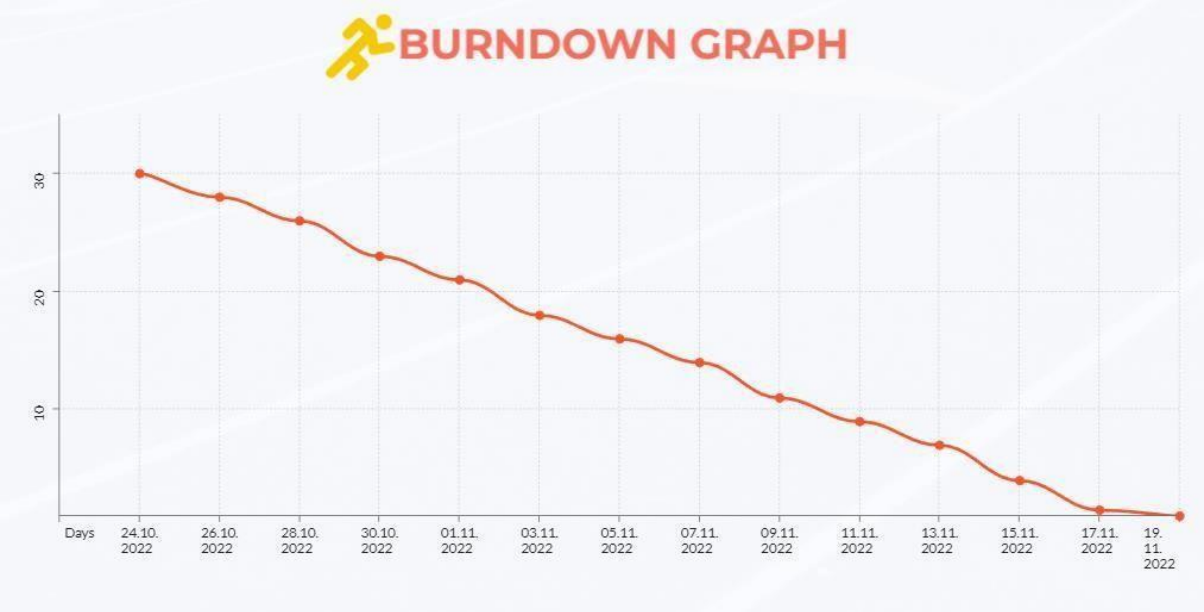
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Design a Web application	USN-1	Design a web application which facilitates the image input.	20	High	Arrasura Paramesh, Rahul R
Sprint-2	Data Collection	USN-2	The data required for building the model has to be collected from mosdac website for training the model.	10	High	Sriram S, Jeyaprathap S
Sprint-2	Image Preprocessing	USN-3	Pre-process the collected data which is downloaded from the website it prevents the unnecessary variance or Bias problem	10	Medium	Rahul R, Arrasura Paramesh
Sprint-3	Model Building	USN-4	Computer Vision Model for tropical cyclone intensity estimation is important so that user can	10	High	Sriram S, Jeyaprathap
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
			send image for prediction			S
Sprint-3	Model Testing	USN-5	Once the model is trained completely, test the model on data that it has not seen before to ensure its performance	10	Medium	Sriram S, Rahul R, Jeyaprathap S
Sprint-4	Building the APIs	USN-6	Building an API's which facilitates the image input to the model running in the server and send back the predicted intensity back to the web page	10	High	Sriram S, Arrasura Paramesh, Rahul R
Sprint-4	Testing	USN-7	Once the web application is built successfully perform series of test on the application to ensure its performance	5	Medium	Arrasura Paramesh, Jeyaprathap

5.2 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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022

Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	04 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	18 Nov 2022

5.3 REPORT FROM JIRA



6. CODING & SOLUTIONING

7.1 FEATURE 1

```
from flask import Flask, Response,
render_template
import cv2 from keras.models import
load_model import io from PIL
import Image import numpy as np
 app = Flask(__name__)
video = cv2.VideoCapture(0)

MODEL_PATH = 'Disaster_Classification_model.h5'
model = load_model(MODEL_PATH) predictions
= str()
```

```
@app.route('/') def
index():
 return render_template('home.html')

@app.route('/intro') def
intro():
 return render_template('introduction.html')
 def
gen(video):
    video = cv2.VideoCapture(0)
while True:                success,
image = video.read()
         image_bytes = cv2.imencode('.jpg',
image)[1].tobytes()          img =
Image.open(io.BytesIO(image_bytes))          img =
img.resize((64,64))          img = np.array(img)          img =
img / 255.0          img = img.reshape(1,64,64,3)
predictions = model.predict(img)          pred =
np.argmax(predictions, axis = 1)          classes = ["Cyclone",
"Earthquake", "Flood",
"wildfire"]
print(classes[pred[0]])
predictions = classes[pred[0]]
         cv2.putText(image, predictions,
(50,50)
,cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 3)
         ret, jpeg = cv2.imencode('.jpg',
image)
         frame =
jpeg.tobytes()
         yield (b'--frame\r\n'                b'Content-Type:
image/jpeg\r\n\r\n' + frame + b'\r\n\r\n'
               )

@app.route('/video_feed') def video_feed():     global
video     return Response(gen(video),
mimetype='multipart/x-mixed-replace; boundary=frame')
 if __name__ ==
'__main__':     app.run()
```

Section	Total Cases	Not Tested	Fail	Pass

Home Page	3	0	0	3
Introduction Page	1	0	0	1
Open with Cam	2	0	0	2

The web application has the features to open web cam. When we click on that, it records the video and display the video and classify what class it belongs to. It can be done by send the video back into server. In the server side the video is split into multiple frames. Each frame is applied image preprocessing as specified while training. Once the prediction is done, it is labelled as what class it is belongs to and send back the frame into the web application.

7. TESTING

7.1 TEST CASES 7.2 USER ACCEPTANCE TESTING

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	4	2	3	18
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	18	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	23	9	10	22	64

Prediction	9	0	0	9

Accuracy	8	0	0	8
Final Report Output	4	0	0	4

8.2 TEST CASE

TESTCASE REPORT TEMPLATE

				Date	03-Nov-22					
				Team ID	PNT2022TMID06841					
				Project Name	Project -NATURAL DISASTERS INT					
				Maximum Marks	4 marks					
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets
HomePage_TC_OO3	UI	Home page	Verify that user is able to see the images of various natural disasters and its explaination.		1.Enter URL and click go		User should able to see the images & explaination in the homepage	Working as expected	Pass	
IntroductionPage_TC_OO1	UI	Introduction Page	Verify the user is able to visualize the Introduction Page.		1.Enter URL and click go 2.Click on Introduction Button		Introduction Page should display .	Working as expected	Pass	
OpenWithCam_TC_OO1	Functional	Live Streaming with the help of cam	Verify user is able to see the web cam.		1.Enter URL and click go 2.Click on open with cam button.		Web Cam is opened.	Working as expected	Pass	
OpenWithCam_TC_OO2	Functional	Live Streaming with the help of cam.	Verify user is able to see the correct prediction of natural disasters.		1.Enter URL and click go 2.Click on open with cam button.		Application should show the correct prediction.(Cyclone,Floods,Earthquake,Wildfire)	Working as expected	Pass	

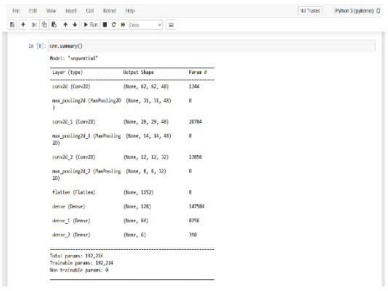
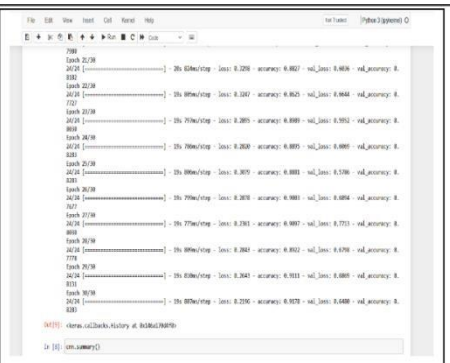
Date	03-Nov-22
Team ID	PNT2022TMID06841
Project Name	Project -NATURAL DISASTERS I
Maximum Marks	4 marks

Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
	Enter URL and click enter.		HomePage should display.	Working as expected	Pass				Sriram S,JeyaPrathap S
	1.Enter URL and click go 2.Verify the buttons are working: a.HomePage(button) b.Introduction(button) c.open with cam (button)		Application should show below UI elements: a.HomePage b.Introduction c.Open with cam	Working as expected	Pass				Arrasura Paramesh,Rahul R
	1.Enter URL and click go		User should able to see the images & explaination in the homepage	Working as expected	Pass				Sriram S,JeyaPrathap S
	1.Enter URL and click go 2.Click on Introduction Button		Introduction Page should display .	Working as expected	Pass				Arrasura Paramesh,Rahul R
	1.Enter URL and click go 2.Click on open with cam button.		Web Cam is opened.	Working as expected	Pass				Sriram S , Arrasura Paramesh
	1.Enter URL and click go 2.Click on open with cam button.		Application should show the correct prediction.(Cyclone,Floods,Earthquake,Wildfire)	Working as expected	Pass				JeyaPrathap S, Rahul R

	Test Scenarios	
1	Verify user is able to see Home Page?	YES
2	Verify user is able to see Introduction Page?	YES
3	Verify user is able to open the web cam?	YES
4	Verify user is able to get the correct prediction?	YES

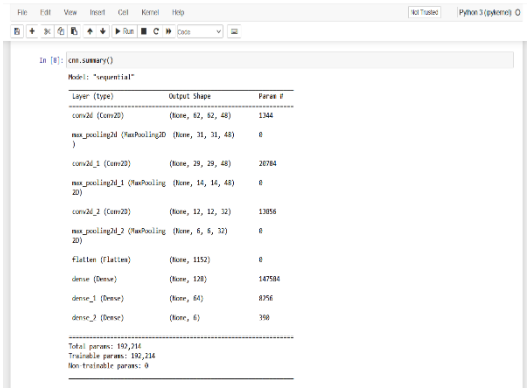
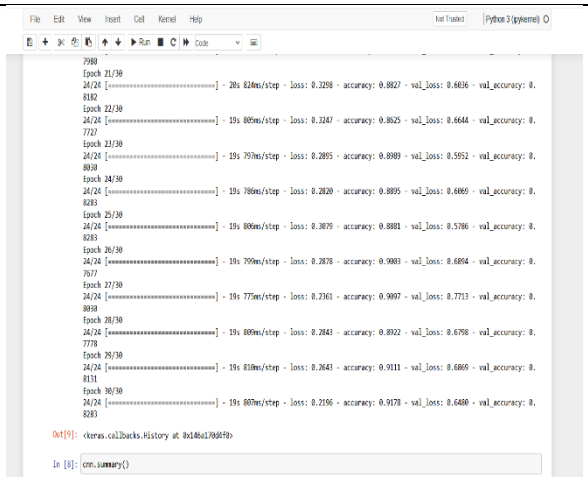
PERFORMANCE TESTING

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**3 convolutional layer architecture** **1 fully connected layer** **Softmax at the end**	
2.	Accuracy	Training Accuracy – 0.9178 Validation Accuracy -0.8283	
3.	Confidence Score (Only Yolo Projects)	NOT REQUIRED	NOT REQUIRED

9 RESULTS

The results can very surprising, the model can correctly able to classify among the four classes – floods, earthquake, wildfire, cyclone. With the training accuracy of 95.3% and test accuracy of 87.5%.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**3 convolutional layer architecture** **1 fully connected layer** **Softmax at the end**	
2.	Accuracy	Training Accuracy – 0.9178 Validation Accuracy -0.8283	

3.	Confidence Score (Only Yolo Projects)	NOT REQUIRED	NOT REQUIRED

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES:

The model performs extremely well on trained data distributions.

DISADVANTAGES:

Only four classes can be correctly predicted. More classes can be added later. But the model does perform poorly on the data that it has not seen in training.

11 CONCLUSION

In this project, we can able to successfully build the CNN model for natural disaster intensity analysis and classification with the test accuracy of 87.5%. The entire application is successfully deployed.

12 FUTURE SCOPE

In near future we are planning to add more classed for classifications like tornado, tsunami, volcanic eruptions etc.

13 APPENDIX

app.py

```
from flask import Flask, Response, render_template
```

```
import cv2 from keras.models import
load_model import io from PIL
import Image import numpy as np
 app = Flask(__name__)
video = cv2.VideoCapture(0)

MODEL_PATH = 'Disaster_Classification_model.h5'
model = load_model(MODEL_PATH) predictions =
str()

@app.route('/') def
index():
 return render_template('home.html')

@app.route('/intro') def
intro():
 return render_template('introduction.html')
 def
gen(video):
    video = cv2.VideoCapture(0)
    while True:
success, image = video.read()
         image_bytes = cv2.imencode('.jpg',
image)[1].tobytes()         img =
Image.open(io.BytesIO(image_bytes))         img =
img.resize((64,64))         img = np.array(img)         img =
img / 255.0         img = img.reshape(1,64,64,3)
predictions = model.predict(img)         pred =
np.argmax(predictions, axis = 1)         classes = ["Cyclone",
"Earthquake", "Flood",
"wildfire"]
print(classes[pred[0]])
predictions = classes[pred[0]]
         cv2.putText(image, predictions,
(50,50)
,cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 3)
         ret, jpeg = cv2.imencode('.jpg',
image)
         frame =
jpeg.tobytes()
         yield (b'--frame\r\n'                b'Content-
Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n'
                )
```

```
@app.route('/video_feed') def video_feed():     global
video     return Response(gen(video),
mimetype='multipart/x-mixed-replace; boundary=frame')
 if __name__ ==
'__main__':
    app.run()
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>AI BASED NATURAL DIASTER ANALYSIS</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initialscale=1">
<link rel="stylesheet" href="styles/styles/css/style.css">
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="icon"type="image/ico" href="images/mus.png">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/boot
strap.min.css"> <link rel="stylesheet"
href="https://bootstrap.bundle.min.js/bootstrap.bundle.js">
<link rel="stylesheet" type="text/css"
href="styles/bootstrap/css/bootstrap.min.css">
</head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery
.min.js"></script>   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootst
rap.min.js"></script>
<script src="styles/styles/js/jquery.min.js"></script>
<script src="styles/styles/js/bootstrap.min.js"></script>
<style>
 .header1{
height:90px;
 background-
color:black; border:2px
solid black;
}
```

```
.rel{
position:absolute;
top:10%; margin-
left:0px;
}



.glow {   font-size: 35px;   color: #fff;   text-
align: center;   animation: glow 1s ease-in-out
infinite alternate;
}

@-webkit-keyframes glow {   from {     text-shadow: 0 0
10px #fff, 0 0 20px #fff, 0 0 30px
#e60073, 0 0 40px #e60073, 0 0 50px #e60073, 0 0 60px #e60073,
0 0 70px #e60073;
  }   to {    text-shadow: 0 0 20px #fff, 0 0 30px #ff4da6,
0 0 40px
#ff4da6, 0 0 50px #ff4da6, 0 0 60px #ff4da6, 0 0 70px #ff4da6,
0 0 80px #ff4da6;
  } } body { background-color:teal; background-repeat:no-
repeat; background-size:cover; height:100%; width:100%; }
.cards {     display: grid;     grid-template-columns:
repeat(auto-fill, minmax(230px,
1fr));
    grid-gap: 20px;
}

.card {
    display: grid;
    grid-template-rows: max-content 200px 1fr;
}

.card img {     object-
fit: cover;     width:
100%;     height: 100%;
}



</style>
```

```
<body>

<div>
      <div class="header1">
<img class="img-circle" style="margin-top:3px"  src=" {{
url_for('static' , filename = 'logo.jpg') }}" alt="logo.jpg"
width="75px" height="75px">
<div class="rel">
<h1 width="100%" style="margin-left:90px; color:white;
margintop:-50px" class="glow"><b class="text-default"
style="fontsize:40px" >AI BASED NATURAL DIASTER ANALYSIS
</b></h1>
</div>
</div>
</div>

<div style="margin-left:800px; margin-top:-40px">
    <button type="button"  > <a href="{{ url_for('index')
}}">  HOME</a></button>
  <button type="button" style="margin-left:10px" ><a href="{{
url_for('intro') }}"> INTRODUCTION</a></button>
  <button type="button" style="margin-left:10px" ><a href= "{{
url_for('video_feed') }}">OPEN WITH CAM</a></button>


  </div>



  <div class="container"
style="margintop:10px;height:600px;width:100%">



  <div class="w3-container">


  <div class="cards" style="margin-top:100px;marginleft:30px">
    <article class="card">
        <header>
            <h2><b>CYCLONE</b></h2>
        </header>
        <img src=" {{ url_for('static' , filename =
'cyclone.jpg') }}" alt="dis">
```

```
        <div class="content">
            <p><i>A cyclone is a general term for a weather
system in which winds rotate inwardly to an area of low
atmospheric pressure.</i></p>
        </div>

    </article>

     <article class="card"style="margin-left:40px">
        <header>
            <h2><b>WILD FIRE</b></h2>
        </header>
        <img src=" {{ url_for('static' , filename =
'wildimg.jfif') }}" alt="dis">
        <div class="content">
            <p><i>A wildfire is an uncontrolled fire that
burns in wildland vegetation, often in rural areas.</i></p>
        </div>
    </article>

    <article class="card"style="margin-left:40px">
        <header>
            <h2><b>EARTH QUAKE</b></h2>
        </header>
        <img src = " {{ url_for('static' , filename =
'earthquake.jfif') }}" alt="dis">
        <div class="content">
            <p><i>An earthquake is the shaking of the surface
of the Earth resulting from a sudden release of energy in the
Earth's lithosphere that creates seismic waves</i></p>
        </div>
    </article>
    <article class="card"style="margin-left:40px">
        <header>
            <h2><b>FLOOD</b></h2>
        </header>
        <img src=" {{ url_for('static' , filename =
'flood.jfif') }}" alt="dis">
        <div class="content">
            <p><i>A flood is an overflow of water that
submerges land that is usually dry.</i> </p>
        </div>
    </article>
```

```
</div>

</div>
</div>


</body>
</html>
```

Introduction.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>AI BASED NATURAL DIASTER ANALYSIS</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initialscale=1">
<link rel="stylesheet" href="styles/styles/css/style.css">
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="icon"type="image/ico" href="images/mus.png">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/boot
strap.min.css"> <link rel="stylesheet"
href="https://bootstrap.bundle.min.js/bootstrap.bundle.js">
<link rel="stylesheet" type="text/css"
href="styles/bootstrap/css/bootstrap.min.css">
</head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery
.min.js"></script>   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootst
rap.min.js"></script>
<script src="styles/styles/js/jquery.min.js"></script>
<script src="styles/styles/js/bootstrap.min.js"></script>
<style>
 .header1{
height:90px;
 background-
color:black; border:2px
solid black;
}
```

```
.rel{
position:absolute;
top:10%; margin-
left:0px;
}



.glow {   font-size:
35px;   color: #fff;
text-align: center;
animation: glow 1s
ease-in-out infinite
alternate;
}

@-webkit-keyframes glow {   from {     text-shadow: 0 0
10px #fff, 0 0 20px #fff, 0 0 30px
#e60073, 0 0 40px #e60073, 0 0 50px #e60073, 0 0 60px #e60073,
0 0 70px #e60073;
  }   to {    text-shadow: 0 0 20px #fff, 0 0 30px #ff4da6,
0 0 40px
#ff4da6, 0 0 50px #ff4da6, 0 0 60px #ff4da6, 0 0 70px #ff4da6,
0 0 80px #ff4da6;
  } } body { background-color:teal; background-repeat:no-
repeat; background-size:cover; height:100%; width:100%; }
.cards {     display: grid;     grid-template-columns:
repeat(auto-fill, minmax(230px,
1fr));
    grid-gap: 20px;
}
 .card {     display: grid;     grid-template-
rows: max-content 200px 1fr;
}

.card img {     object-
fit: cover;     width:
100%;     height: 100%;
}
```

```
</style>
<body>

<div>
      <div class="header1">
<img class="img-circle" style="margin-top:3px"  src=" {{
url_for('static' , filename = 'logo.jpg') }}" alt="logo"
width="75px" height="75px">
<div class="rel">
<h1 width="100%" style="margin-left:90px; color:white;
margintop:-50px" class="glow"><b class="text-default"
style="fontsize:40px" >AI BASED NATURAL DIASTER ANALYSIS
</b></h1>
</div>
</div>
</div>

<div style="margin-left:800px; margin-top:-40px">
 <button type="button"><a href="{{
url_for('index')}}">HOME</a></button>
  <button type="button" style="margin-left:10px" ><a href="{{
url_for('intro') }}">INTRODUCTION</a></button>
  <button type="button" style="margin-left:10px" ><a href="{{
url_for('video_feed') }}">OPEN WITH CAM</a></button>

  </div>



  <div class="container"
style="margintop:10px;height:600px;width:100%">


 <p style="font-size:20px;text-
align:center;margintop:60px"><i>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.</i></p>


</div>
</div>


</body>
</html>
```

Model.ipynb

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator import
numpy as np

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

#loding training data
train_generator = train_datagen.flow_from_directory(
    r"C:\Users\paramesh reddy\Desktop\Nalayathiran_Project\dataset\train_set",
target_size=(64, 64),      batch_size=32,
    class_mode='categorical')

# loading testing data
```

```
test_datagen = ImageDataGenerator(rescale=1./255) test_generator
= train_datagen.flow_from_directory(
      r"C:\Users\paramesh reddy\Desktop\Nalayathiran_Project\dataset\test_set",
target_size=(64, 64),       batch_size=32,       class_mode='categorical')

# initialising sequential model and adding layers to it cnn
= tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=48,    kernel_size=3,          activation='relu',
input_shape=[64, 64, 3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

cnn.add(tf.keras.layers.Conv2D(filters=48, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

cnn.add(tf.keras.layers.Flatten())

cnn.add(tf.keras.layers.Dense(128, activation='relu'))
cnn.add(tf.keras.layers.Dense(64, activation='relu')) cnn.add(tf.keras.layers.Dense(4,
activation='softmax'))

# finally compile and train the cnn
cnn.compile(optimizer="adam",       loss="categorical_crossentropy", metrics=["accuracy"])
cnn.fit(x=train_generator, validation_data=test_generator, epochs=30) cnn.summary()

probability_model = tf.keras.Sequential([cnn,
                              tf.keras.layers.Softmax()])
from PIL import Image

img                              =                              Image.open(r"C:\Users\paramesh
```

```
reddy\Desktop\Nalayathiran_Project\Test_data\download (4).jpg")
img = img.resize((64,64)) img = np.array(img) img = img / 255.0 img
= img.reshape(1,64,64,3)

predictions = probability_model.predict(img) pred
= np.argmax(predictions, axis = 1)
classes = ["Cyclone", "Earthquake", "Flood", "wildfire"] print(classes[pred[0]])
cnn.save("Disaster_Classification_model.h5")
```

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-3830-1658648131

Video Demo Link: https://www.loom.com/share/63b96a9f2d524d42a8e195462d9d8c43