Natural Disasters Intensity Analysis and Classification using Artificial Intelligence

"NALAIYA THIRAN"

Team ID : PNT2022TMID54341 Team Leader : HARIHARASUDHAN S

M1 Lead : ARAVIND KUMAR S M2 Lead : Arun K S M3 Lead: BHARATH R

1. INTRODUCTION

1.1 Overview

Natural catastrophes not only disrupt the ecology that supports human life, but they also obliterate vital facilities and properties in human society, changing the ecosystem permanently. Natural occurrences like earthquakes, cyclones, floods, and wildfires can bring disaster. To mitigate ecological losses from natural disasters, several deep learning approaches have been used by numerous researchers. However, identification of natural disasters still has difficulties because of the complex and unbalanced image structures. We created a multilayered deep convolutional neural network model that identifies natural disasters and indicates their intensity in order to address this issue.

1.2 Purpose

- To prevent ecological losses from natural disasters as we all know that “Prevention is better than cure”.

2. LITERATURE SURVEY

2.1 Existing Problem

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.

2.2 References

SURVEY 1

Title:

"Neural Network Applications In Earthquake Prediction" Meta-Analytic And Statistical Insights On Their Limitations.Arnaud Mignan And Marco Broccardo neural Network Applications In Earthquake Prediction ; Meta-Analytic And Statistical Insights On Their Limitations Seismological Research Letters (May 2020).

Methods:

Deep learning has solved seemingly intractable problems, boosting the hope to find approximate solutions to problems that now are considered unsolvable. Earthquake prediction, the Grail of Seismology, is, in this context of continuous exciting discoveries, an obvious choice for deep learning exploration. We review the entire literature on artificial neural network (ANN) applications for earthquake prediction (77 articles, 1994-2019 period) and find two emerging trends: an increasing interest in this domain, and a complexification of ANN models over time, towards deep learning. Despite apparent positive results observed in this corpus, we demonstrate that simpler models seem to offer similar predictive powers, if not better ones. Due to the structured, tabulated nature of earthquake catalogs, and the limited number of features so far considered, simpler and more transparent machine learning models seem preferable at the present stage of research. Those baseline models follow first physical principles and are consistent with the known empirical laws of Statistical Seismology, which have minimal abilities to predict large earthquakes.

SURVEY 2

Title:

"Simultaneous Earthquake Detection On Multiple Stations Via A Convolutional Neural Network" Shaobo Yang; Hu; Haijiang Zhang; Guiquan Liu, Seismological Research Letter(2021)

Methods:

As the amount of seismic data has grown rapidly, it is very important to develop a fast and reliable event detection and association algorithm. Generally, event detection is first performed on individual stations followed by event association through linking phase arrivals to a common event generating them. This study considers earthquake detection as the problem of image classification and convolutional neural networks (CNNs), as some of the widely used deep-learning tools in image processing, can be well used to solve this problem. In contrast to existing studies training the network using seismic data from individual stations, in this study, we train a CNN model jointly using records of multiple stations. Because the CNN automatically synthesizes information among multiple stations, the detector can more reliably detect seismic events and is less affected by spurious signals. The CNN is trained using aftershock data of the 2013 Mw 6.6 Lushan earthquake. We have applied it to two very different datasets of Gofar transform fault, East Pacific Rise and Changning shale gas field in southern Sichuan basin, China. The tests show that the trained CNN has strong generalization ability and is flexible with the number of available stations, different instrument types, and different data sampling rates. It can detect many more events than the conventional short-term average/long-term average detector and is more efficient than template-matching methods.

SURVEY 3

Title:

"A Deep Learning Approach of Recognizing Natural Disasters on Images using Convolutional Neural Network and Transfer Learning" International Conference on Artificial Intelligence and its ApplicationsDaryl B. ValdezRey Anthony G. Godmalin December 2021.

Methods:

Natural disasters are uncontrollable phenomena occurring yearly which cause extensive damage to lives, and 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. Methods used include: Deep learning(DL),Convolutional Neural Network(CNN)

SURVEY 4

Title:

“Storm intensity estimation using symbolic aggregate approximation and artificial neural network”, Arthit Buranasing, Akara Prayote, 06 December 2014, IEEE.

Methods:

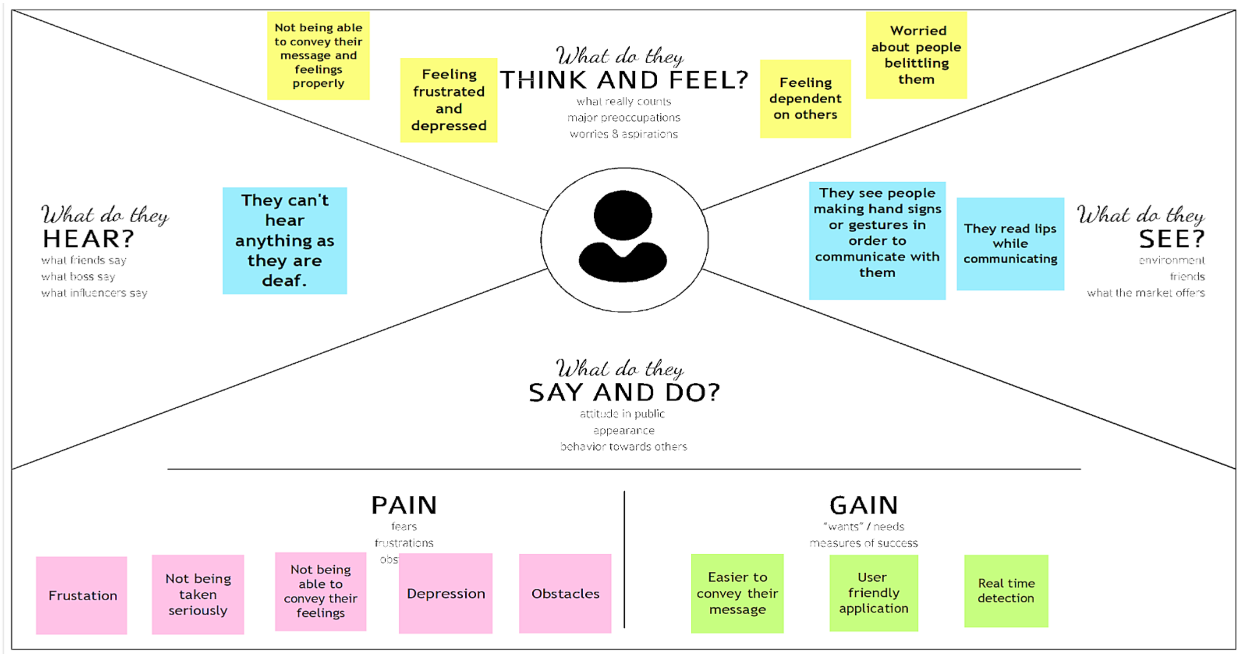
A storm disaster is one of the most destructive natural hazards on earth and the main cause of death or injury to humans as well as damage or loss of valuable goods or properties, such as buildings, communication systems, agricultural land, etc. Storm intensity estimation is also important in evaluating the storm track prediction and risk area that will be affected by the storm. In this paper, proposed the storm intensity estimation model by using only 8 features to categorize major types of storm with symbolic aggregate approximation (SAX) and artificial neural network (ANN). The performance of the model is satisfactory, giving an average F-measure of 0.93 or 93%.

2.3 Problem Statement Definition

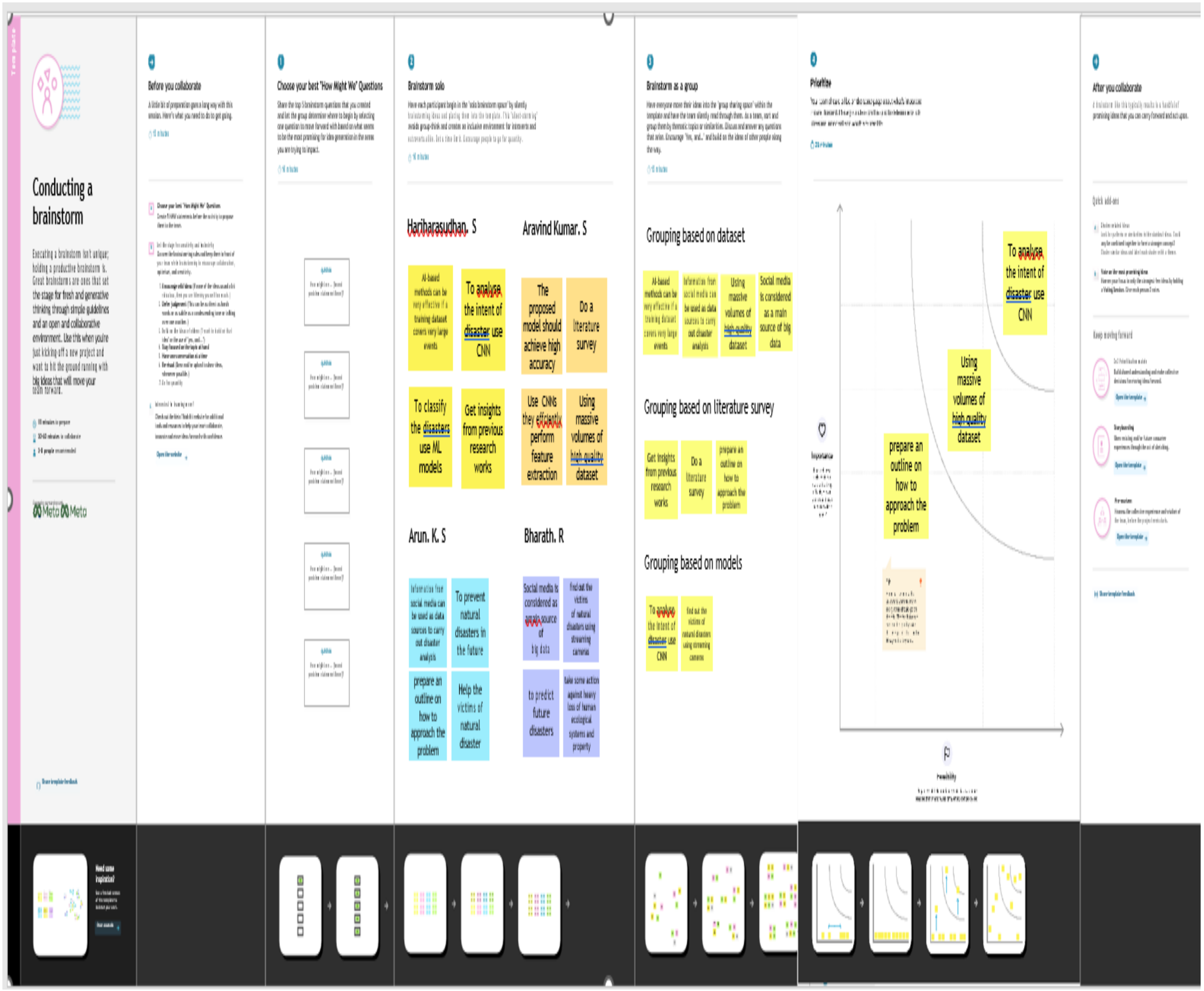
The solution to the problem is Artificial Intelligence, which is being used to implement the proposed system. Artificial intelligence (AI) models have shown remarkable success and superiority to handle huge and nonlinear data owing to their higher accuracy and efficiency, making them perfect tools for disaster monitoring and management. When using AI to detect extreme events such as avalanches or earthquakes, the availability of data can be a limiting factor. AI-based methods can be very effective if a training dataset covers very large events.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



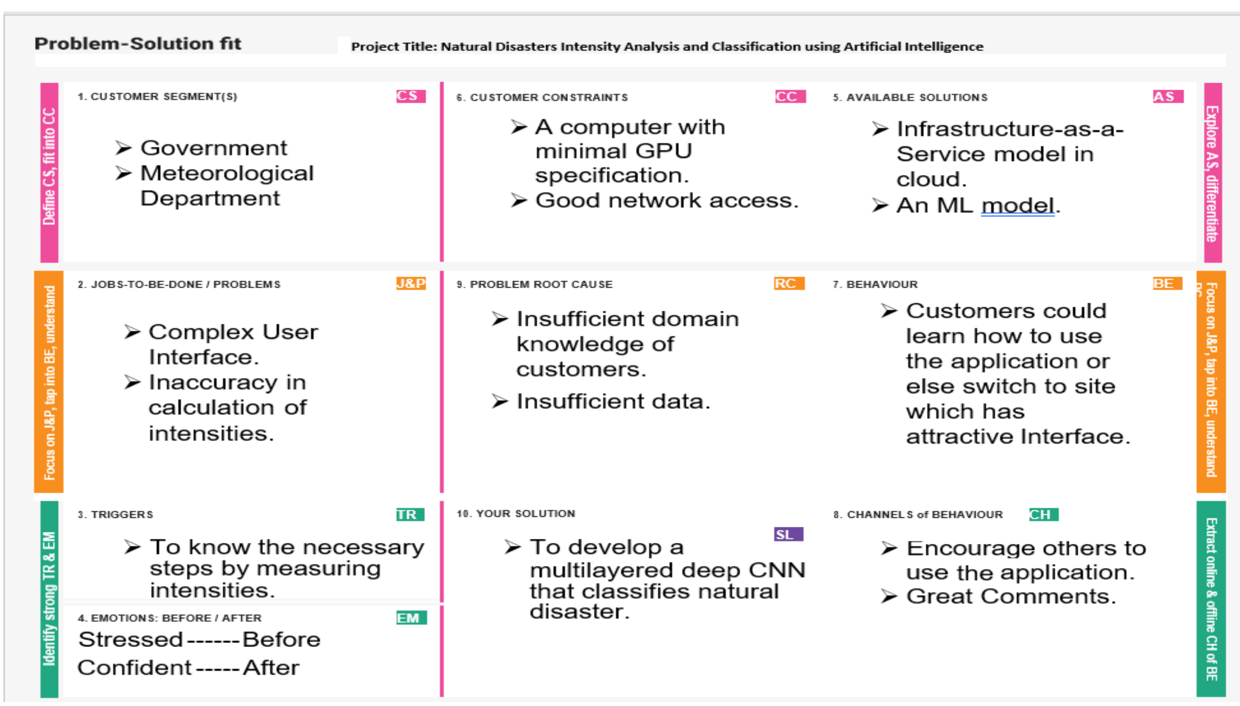
3.2 Ideation and brainstorming


Conducting a brainstorm
Before you collaborate
Choose your best "How Might We" Questions
Brainstorm solo
Hariharasudhan. S
Aravind Kumar. S
AI-based methods can be very effective if a training dataset covers very large events
To analyse the intent of disaster use CNN
The proposed model should achieve high accuracy
Do a literature survey
To classify the disasters use ML models
Get insights from previous research works
Use CNNs they efficiently perform feature extraction
Using massive volumes of high quality dataset
Arun. K. S
Bharath. R
Information from social media can be used as data sources to carry out disaster analysis
To prevent natural disasters in the future
Social media is considered as a main source of big data
Find out the victims of natural disasters using streaming cameras
prepare an outline on how to approach the problem
Help the victims of natural disaster
to predict future disasters
take some action against heavy loss of human ecological systems and property
Brainstorm as a group
Grouping based on dataset
Grouping based on literature survey
Grouping based on models
Prioritize
To analyse the intent of disaster use CNN
Using massive volumes of high quality dataset
prepare an outline on how to approach the problem
Importance
Feasibility
After you collaborate
Quick add-ons
Keep moving forward
Need some inspiration?

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To classify the natural disaster and calculate the intensity of the disaster.
2.	Idea / Solution description	To develop a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster.
3.	Novelty / Uniqueness	We are implementing neural networks to train our model instead using machine learning algorithms which expected to provide with better accuracy.
4.	Social Impact / Customer Satisfaction	With better accuracy in predicting intensities precautions are taken respectively.
5.	Business Model (Revenue Model)	The software is cheap, and the minimum requirements are affordable.
6.	Scalability of the Solution	Better accuracy in measuring the intensities of the natural disaster and in classifying it.

3.4 Problem Solution Fit



Problem-Solution fit

Project Title: Natural Disasters Intensity Analysis and Classification using Artificial Intelligence

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
- Government
- Meteorological Department

6. CUSTOMER CONSTRAINTS CC
- A computer with minimal GPU specification.
- Good network access.

5. AVAILABLE SOLUTIONS AS
- Infrastructure-as-a-Service model in cloud.
- An ML model.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P
- Complex User Interface.
- Inaccuracy in calculation of intensities.

9. PROBLEM ROOT CAUSE RC
- Insufficient domain knowledge of customers.
- Insufficient data.

7. BEHAVIOUR BE
- Customers could learn how to use the application or else switch to site which has attractive Interface.

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

Identify strong TR & EM

3. TRIGGERS TR
- To know the necessary steps by measuring intensities.

4. EMOTIONS: BEFORE / AFTER EM

Stressed------Before

Confident-----After

10. YOUR SOLUTION SL
- To develop a multilayered deep CNN that classifies natural disaster.

8. CHANNELS of BEHAVIOUR CH
- Encourage others to use the application.
- Great Comments.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Reporting/Documenting	Reports should be generated every 24 hours, mention all use cases.
FR-4	Testing	Component, API,UI testing, etc., Tested before nonfunctional testing
FR-5	Authorization levels	Should be a decentralized system.
FR-6	Objective	Describe what the product does

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system needs to be effective and simple for users to use.
NFR-2	**Security**	User information must be secured.
NFR-3	**Reliability**	The produced output should be reliable to the users.
NFR-4	**Performance**	*The system should be able to handle many users without aperformance deterioration.*
NFR-5	**Availability**	The system should be accessible to a user at a given point in time
NFR-6	**Scalability**	*The website pages should load fast with the total number of simultaneous users.*

4.2 Non-Functional Requirement

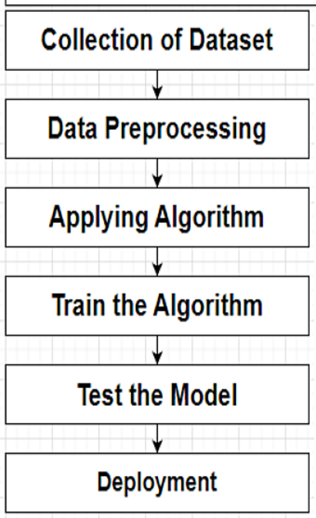
5. PROJECT DESIGN

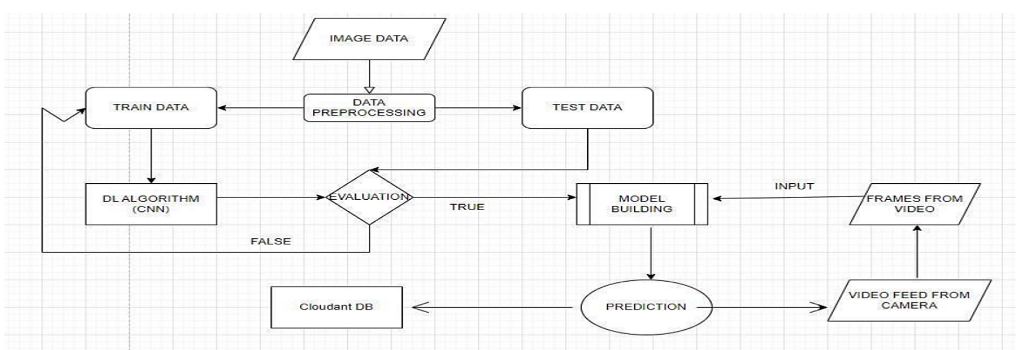
5.1 Data Flow Diagram

Data Flow Diagrams:

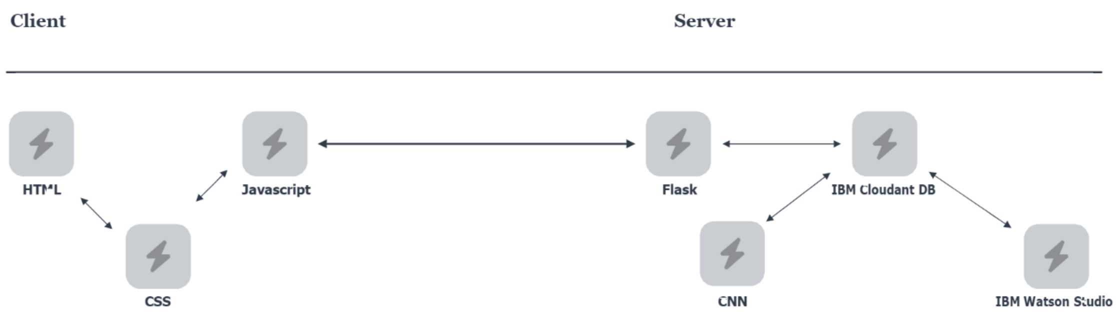
A data flow diagram (DFD) is a visual representation of the information flow through a process or system. DFDs help you better understand process or system operation to discover potential problems, improve efficiency, and develop better processes

Example: DFD Level 0 (Industry Standard)





5.2 Solution & Technical Architecture



S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To classify the natural disaster and calculate the intensity of the disaster.
2.	Idea / Solution description	To develop a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster.
3.	Novelty / Uniqueness	We are implementing neural networks to train our model instead using machine learning algorithms which expected to provide with better accuracy.
4.	Social Impact / Customer Satisfaction	With better accuracy in predicting intensities precautions are taken respectively.
5.	Business Model (Revenue Model)	The software is cheap, and the minimum requirements are affordable.
6.	Scalability of the Solution	Better accuracy in measuring the intensities of the natural disaster and in classifying it.

5.3 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 register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login & access my account with my registered credentials	High	Sprint-1
	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	I can upload the images, I can viewthe result, I can edit my profile and I can view my history	High	Sprint-1
Customer (Web user)	Login	USN-7	As a user, I can log into the web application and access the dashboard	I can login with the same registered credentials and access my account through web application	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer Care Executive	Help Desk	USN-8	As a user, I can get the guidance from the customer care	I can get help from the customer care for carrying out my tasks	High	Sprint-2
Administrator	Management	USN-9	As an administrator, I can collect new datasets and keep the model trained	I can collect and train the model with new dataset frequently	High	Sprint-2
		USN-10	As an administrator, I can update other features of the application	I can update and tune the features of application if needed	Medium	Sprint-1
		USN-11	As an administrator, I can maintain the information about the user	I can maintain information like user type and other such information	Medium	Sprint-1
		USN-12	As an administrator, I can maintain third-party services	I can support and maintain any third-party services	Low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

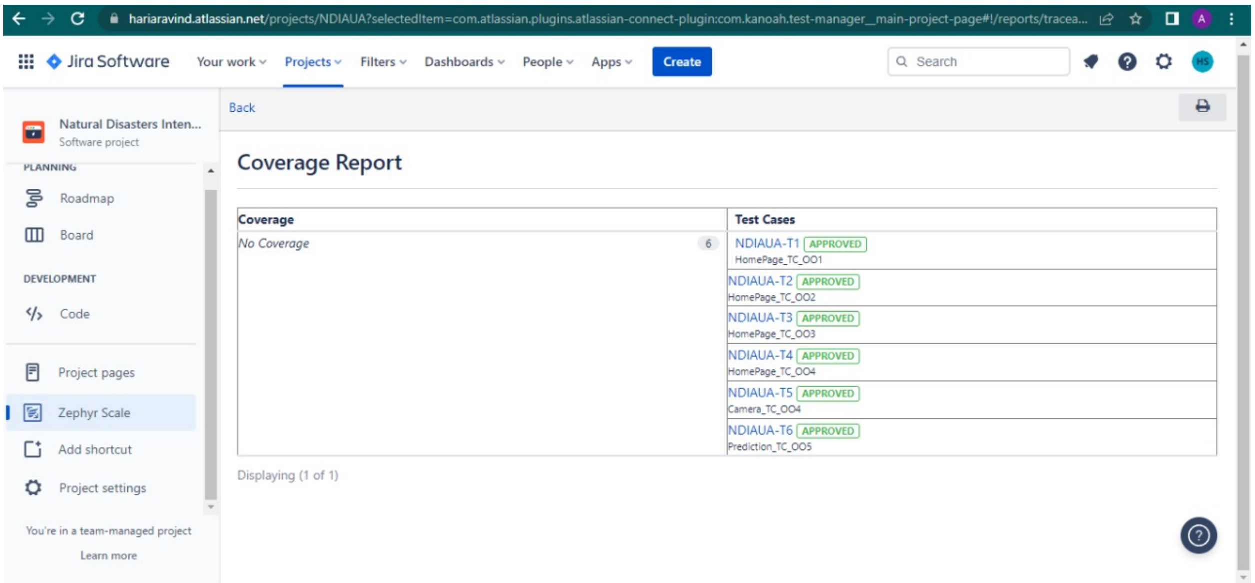
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Hariharasudhan, Aravindkumar
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Bharath, Arun
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Aravindkumar, Bharath
Sprint-2		USN-4	As a user, I can register for the application through Gmail	2	Medium	Hariharasudhan, Arun
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Bharath, Hariharasudhan
Sprint-1	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	2	High	Aravindkumar, Arun
Sprint-1	login	USN−7	As a user, I can log into the web application and access the dashboard	2	High	Bharath, Arun
Sprint-4	Helpdesk	USN-8	As a user, I can get the guidance from the customer care	1	High	Aravindkumar, Hariharasudhan, Arun
Sprint-3	Management	USN-9	As an administrator, I can collect new datasets and keep the model trained	2	High	Hariharasudhan

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3		USN-10	As an administrator, I can update other features of the application	2	Medium	Aravindkumar, Hariharasudhan
Sprint-3		USN-11	As an administrator, I can maintain the information about the user	2	medium	Bharath, Arun
Sprint-4		USN-12	As an administrator, I can maintain third-party services	1	Low	Aravindkumar

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	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	4	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	19 Nov 2022

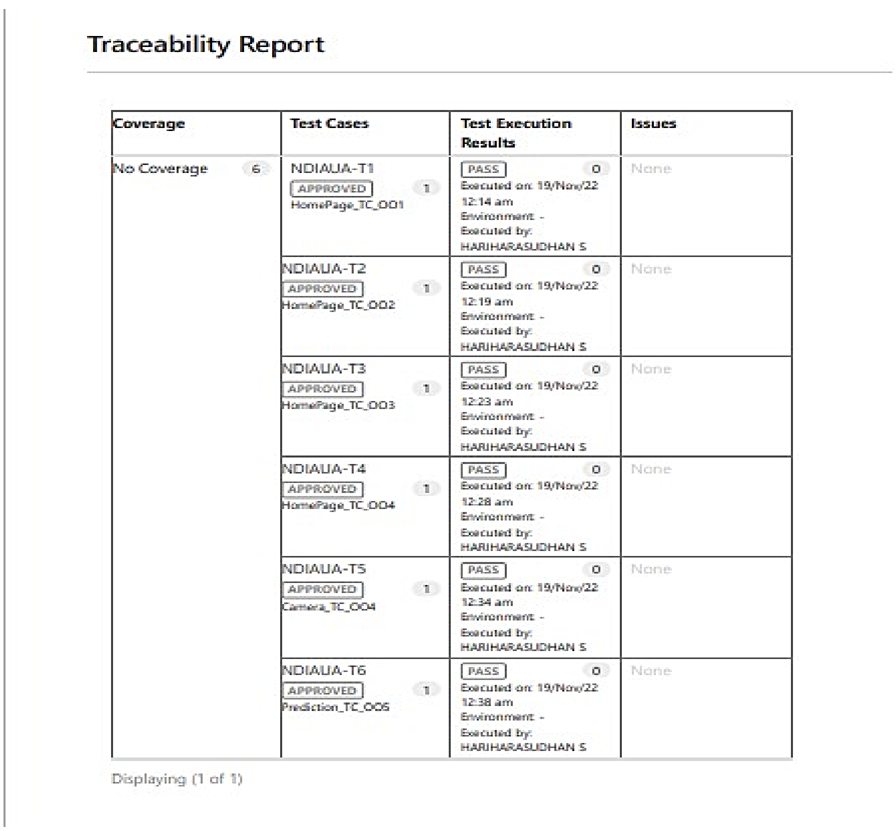
6.3 Reports from JIRA



Coverage Report

Coverage		Test Cases
No Coverage	6	NDIAUA-T1 APPROVED HomePage_TC_001
		NDIAUA-T2 APPROVED HomePage_TC_002
		NDIAUA-T3 APPROVED HomePage_TC_003
		NDIAUA-T4 APPROVED HomePage_TC_004
		NDIAUA-T5 APPROVED Camera_TC_004
		NDIAUA-T6 APPROVED Prediction_TC_005

Displaying (1 of 1)



Traceability Report

Coverage		Test Cases		Test Execution Results		Issues
No Coverage	6	NDIAUA-T1 APPROVED HomePage_TC_001	1	PASS Executed on: 19/Nov/22 12:14 am Environment: - Executed by: HARIHARASUDHAN S	0	None
		NDIAUA-T2 APPROVED HomePage_TC_002	1	PASS Executed on: 19/Nov/22 12:19 am Environment: - Executed by: HARIHARASUDHAN S	0	None
		NDIAUA-T3 APPROVED HomePage_TC_003	1	PASS Executed on: 19/Nov/22 12:23 am Environment: - Executed by: HARIHARASUDHAN S	0	None
		NDIAUA-T4 APPROVED HomePage_TC_004	1	PASS Executed on: 19/Nov/22 12:28 am Environment: - Executed by: HARIHARASUDHAN S	0	None
		NDIAUA-T5 APPROVED Camera_TC_004	1	PASS Executed on: 19/Nov/22 12:34 am Environment: - Executed by: HARIHARASUDHAN S	0	None
		NDIAUA-T6 APPROVED Prediction_TC_005	1	PASS Executed on: 19/Nov/22 12:38 am Environment: - Executed by: HARIHARASUDHAN S	0	None

Displaying (1 of 1)

Traceability Tree

Traceability	Summary
No Coverage	
└ Covered by Test Case NDIAUA-T1	HomePage_TC_OO1
└-- Executed on 19/Nov/22 12:14 am	PASS Executed by HARIHARASUDHAN S
└ Covered by Test Case NDIAUA-T2	HomePage_TC_OO2
└-- Executed on 19/Nov/22 12:19 am	PASS Executed by HARIHARASUDHAN S
└ Covered by Test Case NDIAUA-T3	HomePage_TC_OO3
└-- Executed on 19/Nov/22 12:23 am	PASS Executed by HARIHARASUDHAN S
└ Covered by Test Case NDIAUA-T4	HomePage_TC_OO4
└-- Executed on 19/Nov/22 12:28 am	PASS Executed by HARIHARASUDHAN S
└ Covered by Test Case NDIAUA-T5	Camera_TC_OO4
└-- Executed on 19/Nov/22 12:34 am	PASS Executed by HARIHARASUDHAN S
└ Covered by Test Case NDIAUA-T6	Prediction_TC_OO5
└-- Executed on 19/Nov/22 12:38 am	PASS Executed by HARIHARASUDHAN S

Displaying (1 of 1)

7. CODING & SOLUTIONING

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

app = Flask(__name__,template_folder="templates")
model=load_model("Flask/analysis.h5")
#print(model)

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

@app.route('/home.html',methods=['GET'])
def home():
```

```
  return render_template('home.html')

@app.route('/intro.html',methods=['GET'])
def about():
  return render_template('intro.html')

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

@app.route('/uploader.html',methods=['GET','POST'])
def predict():
  if request.method == "POST":
    f = request.files['filename']
    f.save("Flask/videos/save.mp4")
  cap=cv2.VideoCapture("Flask/videos/save.mp4")
  while(True):
    _,frame = cap.read()
    frame=cv2.flip(frame,1)
    while(True):
      (grabbed,frame) = cap.read()
      if not grabbed:
        break
      output = frame.copy()
      frame = cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
      frame = cv2.resize(frame,(64,64))
      x=np.expand_dims(frame,axis=0)
      result = np.argmax(model.predict(x),axis=1)
      index=['Cyclone','Earthquake','Flood','Wildfire']
      result = str(index[result[0]])
      #print(result)
      cv2.putText(output,"activity:
{}".format(result),(10,120),cv2.FONT_HERSHEY_PLAIN,1,(0,25,255),1)
      cv2.imshow("Output",output)
    if cv2.waitKey(0) & 0xFF==ord('q'):
      break
  print("[INFO]cleaning up...")
  cap.release()
  cv2.destroyAllWindows()
  return render_template("upload.html")
```

if __name__ == '__main__':
  app.run(host='0.0.0.0',port=8000,debug=True)

TRAIN AND TEST THE MODEL

```
from google.colab import drive
drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly remount,
call drive.mount("/content/drive", force_remount=True).
```

In [84]:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,shear
_range=0.2)
test_datagen = ImageDataGenerator(rescale=1./255)
```

In [85]:

```
x_train=train_datagen.flow_from_directory("/content/drive/MyDrive/dataset/t
rain_set",target_size=(64,64),class_mode='categorical',batch_size=5,color_
mode='rgb')
x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/dataset/te
st_set",target_size=(64,64),class_mode='categorical',batch_size=5,color_mo
de='rgb')
Found 742 images belonging to 4 classes.
Found 198 images belonging to 4 classes.
```

In [86]:

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

In [87]:

```
model=Sequential()
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=128,activation='relu'))
model.add(Dense(units=4,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['a
ccuracy'])
```

In [88]:

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 128)               802944

 dense_3 (Dense)             (None, 4)                 516

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

In [89]:

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

```
Epoch 1/20
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
UserWarning: `Model.fit_generator` is deprecated and will be removed in a
future version. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
149/149 [==============================] - 44s 293ms/step - loss: 1.1635 -
accuracy: 0.4798 - val_loss: 0.9364 - val_accuracy: 0.6566
Epoch 2/20
149/149 [==============================] - 41s 273ms/step - loss: 0.8416 -
accuracy: 0.6429 - val_loss: 0.8283 - val_accuracy: 0.6717
Epoch 3/20
149/149 [==============================] - 42s 284ms/step - loss: 0.6678 -
accuracy: 0.7655 - val_loss: 0.7795 - val_accuracy: 0.7323
Epoch 4/20
149/149 [==============================] - 41s 273ms/step - loss: 0.6775 -
```

```
accuracy: 0.7493 - val_loss: 0.6493 - val_accuracy: 0.7626
Epoch 5/20
149/149 [==============================] - 41s 273ms/step - loss: 0.5995 -
accuracy: 0.7749 - val_loss: 0.6781 - val_accuracy: 0.7879
Epoch 6/20
149/149 [==============================] - 41s 273ms/step - loss: 0.5397 -
accuracy: 0.7817 - val_loss: 0.8131 - val_accuracy: 0.7172
Epoch 7/20
149/149 [==============================] - 42s 285ms/step - loss: 0.4696 -
accuracy: 0.8275 - val_loss: 0.6780 - val_accuracy: 0.7879
Epoch 8/20
149/149 [==============================] - 41s 272ms/step - loss: 0.4959 -
accuracy: 0.8194 - val_loss: 0.8018 - val_accuracy: 0.7576
Epoch 9/20
149/149 [==============================] - 41s 273ms/step - loss: 0.3969 -
accuracy: 0.8544 - val_loss: 0.6865 - val_accuracy: 0.7828
Epoch 10/20
149/149 [==============================] - 41s 273ms/step - loss: 0.3885 -
accuracy: 0.8652 - val_loss: 0.8218 - val_accuracy: 0.7677
Epoch 11/20
149/149 [==============================] - 42s 280ms/step - loss: 0.3552 -
accuracy: 0.8652 - val_loss: 1.0350 - val_accuracy: 0.7374
Epoch 12/20
149/149 [==============================] - 41s 273ms/step - loss: 0.3266 -
accuracy: 0.8801 - val_loss: 0.7144 - val_accuracy: 0.7778
Epoch 13/20
149/149 [==============================] - 40s 268ms/step - loss: 0.2738 -
accuracy: 0.8949 - val_loss: 0.6965 - val_accuracy: 0.7879
Epoch 14/20
149/149 [==============================] - 40s 271ms/step - loss: 0.2957 -
accuracy: 0.8827 - val_loss: 0.7882 - val_accuracy: 0.7677
Epoch 15/20
149/149 [==============================] - 42s 283ms/step - loss: 0.2576 -
accuracy: 0.9084 - val_loss: 1.0848 - val_accuracy: 0.7525
Epoch 16/20
149/149 [==============================] - 41s 271ms/step - loss: 0.2901 -
accuracy: 0.8976 - val_loss: 0.8777 - val_accuracy: 0.7828
Epoch 17/20
149/149 [==============================] - 41s 271ms/step - loss: 0.2853 -
accuracy: 0.9097 - val_loss: 1.1820 - val_accuracy: 0.7273
Epoch 18/20
149/149 [==============================] - 42s 276ms/step - loss: 0.2219 -
accuracy: 0.9272 - val_loss: 0.8731 - val_accuracy: 0.7929
```

```
Epoch 19/20
149/149 [==============================] - 41s 275ms/step - loss: 0.1894 -
accuracy: 0.9353 - val_loss: 1.0621 - val_accuracy: 0.7374
Epoch 20/20
149/149 [==============================] - 40s 269ms/step - loss: 0.2297 -
accuracy: 0.9151 - val_loss: 1.1963 - val_accuracy: 0.7374
```

Out[89]:

In [90]:

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

In [91]:

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

In [92]:

```
x_train.class_indices
```

Out[92]:

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

In [93]:

```
img =
image.load_img(r"/content/drive/MyDrive/dataset/test_set/Earthquake/1347.j
pg",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 82ms/step
Earthquake
```

In [94]:

```
img =
image.load_img(r"/content/drive/MyDrive/dataset/test_set/Cyclone/918.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
```

8. TESTING

8.1 Test Cases

TEST CASE ANALYSIS

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

8.2 User Acceptance Testing

PURPOSE OF THE DOCUMENT

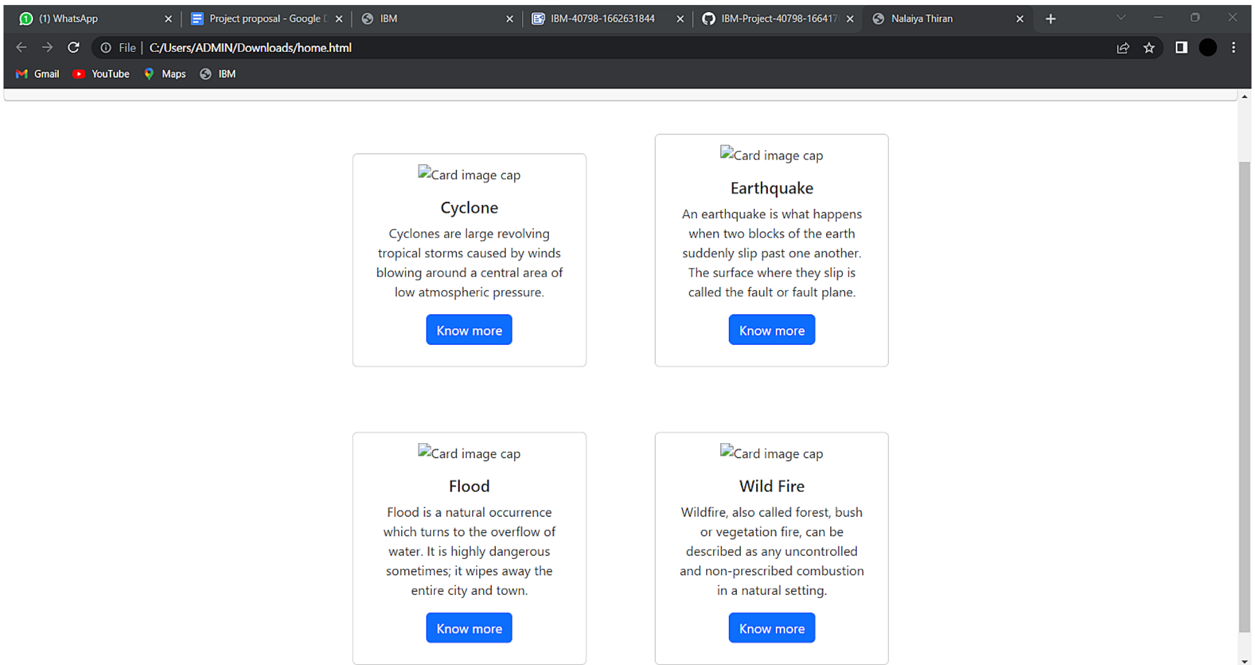
The purpose of this document is to briefly explain the test coverage and open issues of the Natural Disaster Intensity Analysis and Classification project at the time of the release to UserAcceptance Testing (UAT).

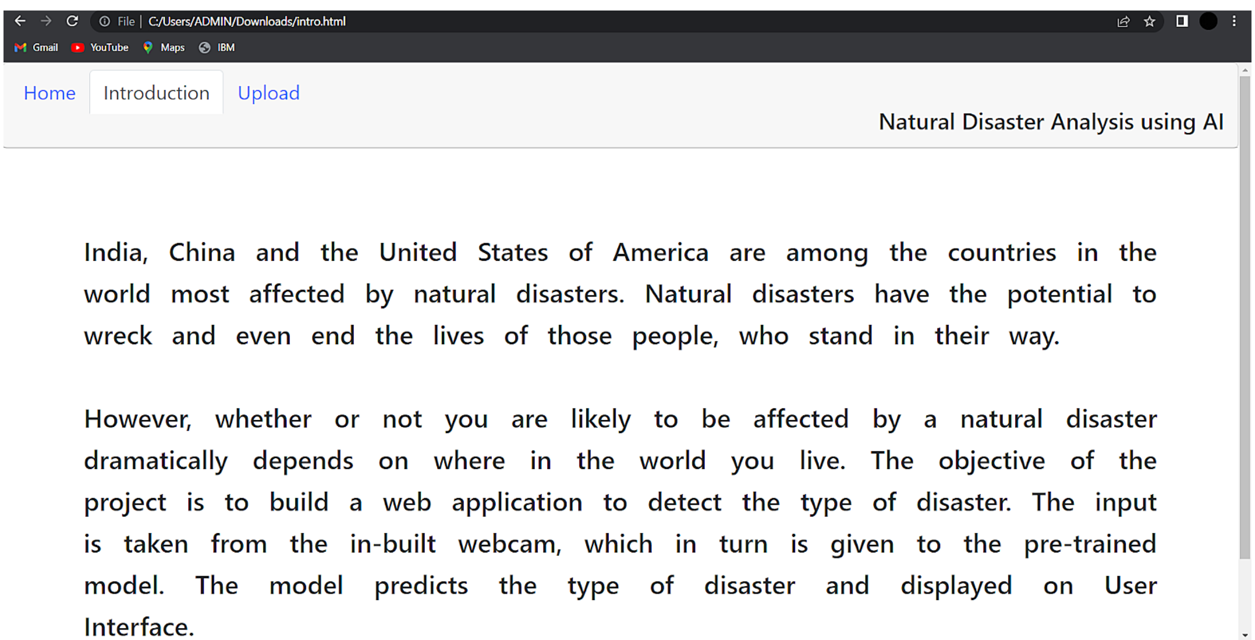
DEFECT ANALYSIS

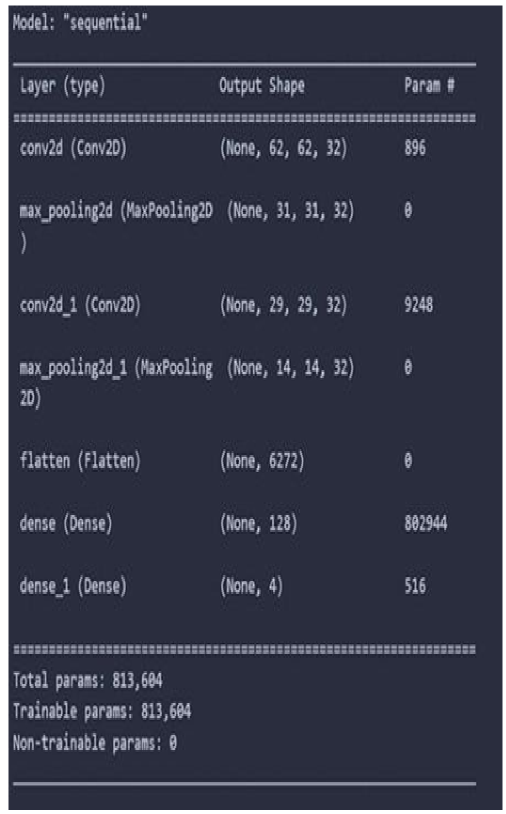
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

9. RESULTS

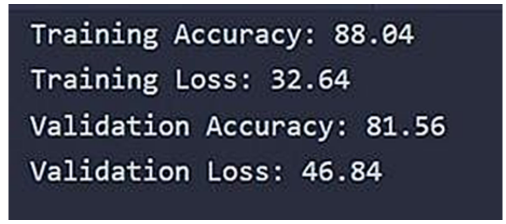
9.1 Performance Metrics





S.No.	Parameter	Values	Screenshot
1.	**Model Summary**	-	

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

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

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

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

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 4)                 516

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

2.	**Accura cy**	**Training Accuracy – 88.04%** **Validation Accuracy - 81.56%**	

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. In order to balance their personal and professional lives, people require breaks and vacation time. However, AI can operate continuously without rest.

2. With the aid of various AI-based methods, we can also predict the weather for the present day and the coming days.

3. Beneficial in regaining control of one's life.

4. Their alert temperament allows them to react quickly and effectively, protecting society from significant harm.

DISADVANTAGES

1. Getting outfitted costs a lot of money.

2. Issues with basic necessities.

3. Robots are one use of artificial intelligence that are replacing jobs and raising unemployment.

4. Machines can only do jobs for which they are created or programmed; if they are asked to complete anything else, they frequently fail or produce useless results, which can create serious problems.

11. CONCLUSION

Numerous researchers have tried to detect natural disasters using various deep learning techniques. Deep learning algorithms for natural disaster detection still have a number of concerns with noise and severe class imbalances. We suggested a multilayered deep convolutional neural network for natural disaster identification and intensity classification to overcome these issues. The suggested method consists of two blocks: the first block is used to identify natural disasters, and the second block is used to address concerns with unequal class representation. As average statistical values, the following findings were derived for the suggested model: sensitivity, 97.54%; specificity, 98.22%; accuracy rate, 99.92%; precision, 97.79%; and F1-score, 97.97%. Due to its multilayered nature, the proposed model outperformed other cutting-edge techniques in terms of accuracy.

12. FUTURE SCOPE

Google's pilot effort in Patna, India, to use artificial intelligence to monitor floods, was a success last year. With an accuracy of over 90%, they were able to foresee floods and the areas that would be impacted by the natural calamity. It was made feasible by a mix of information from government organizations that supply on-the-ground data, including measurements taken with on-the-ground measuring devices and satellite photographs of flood-prone locations.

To forecast the flow of water, they performed hundreds of thousands of simulations using its machine learning (ML) models.

By using AI, disaster management organizations can deploy robots, sensors, and drones in the future to offer precise information on damaged structures and landscapes, impending floods, and safer rescue missions.

Smart technology must be included into our neighborhood communities. The degree of the harm can be decreased with an immediate response and technological remedies. However, there are some restrictions and mistakes with AI because it is based on machine codes. However, combining human empathy with vigilance could be extremely beneficial in the realm of crisis management.

13. APPENDIX

GIT REPO LINK: https://github.com/IBM-EPBL/IBM-Project-40798-1664170001