IBM PROJECT

Natural Disasters Intensity Analysis and Classification Using Artificial intelligence

Team ID	PNT2022TMID53145
Project Name	Project - Natural Disasters Intensity Analysis And Classification Using Artificial intelligence

Team members:

- Abitha P - 195001301
- Chitraju Vishnusree - 195001801
- Meena Muthukumar - 195001063
- Madhuri Mahalingam - 195001057

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. Many machine learning approaches have been used for wildfire predictions since the 1990s. A recent study used a machine learning approach in Italy. As the population is growing rapidly, people need to acquire land to live on, and as a result the ecosystem is disturbed horrifically, which causes global warming and increases the number of natural disasters.

As the technologies are continuously improving, aviation systems have begun adopting smart technologies to develop unmanned aerial vehicles (UAVs) equipped with cameras, which can reach distant areas to identify aftereffects of natural disasters on human life, infrastructure, and transmission lines by capturing images and videos.

The proposed system is explained in the following paper.

1.2 Purpose

Populations in underdeveloped countries cannot afford damages and disasters caused to infrastructures. The aftermath of disasters leaves the humans in miserable situations, and sometimes the devastating effects cannot be detected; additionally, rescue operations cannot take place in most of the places and victims are unable to be identified due to geographical factors of the different areas. Disasters such as forest fires spread rapidly in dense areas, so firefighting is difficult to carry out; in this case, development of the strategy to predict such circumstances is crucial so that such disasters can be prevented beforehand.

As the population is growing rapidly, people need to acquire land to live on, and as a result the ecosystem is disturbed horrifically, which causes global warming and increases the number of natural disasters.

If the intensity of the disaster is known, it will be easier to deal with the aftermath of the disaster and help with the disaster management. The intensity of the damages will be predicted and hence it will be helpful for the government to take necessary measures to get back to normal condition. The number of people affected will be predicted and necessary camps can be created to take care of people.

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

S. No	PAPER	WORK	LINK	TOOLS FOR ALGORITHM	FINDINGS
1	Natural Disasters Intensity Analysis and Classification Based on Multispectral Images Using Multi-Layered Deep Convolutional Neural Network	Deep learning, natural disasters intensity and classification, convolutional neural network	https://www.researchgate.net/publication/350830884_Natural_Disasters_Intensity_Analysis_and_Classification_Based_on_Multispectral_Images_Using_Multi-Layered_Deep_Convolutional_Neural_Network	Convolutional Neural Network	The proposed model works in two blocks: Block-I convolutional neural network (B-I CNN), for detection and occurrence of disasters, and Block-II convolutional neural network (B-II CNN), for classification of natural disaster intensity types with different filters and parameters

2	Recent Efforts in Earthquake Prediction (1990–2007) Ashif Panakkat and Hojjat Adeli	Data collection, Seismicity, Mathematical prediction	https://www.researchgate.net/publication/248880503_Recent_Efforts_in_Earthquake_Prediction_1990-2007		Prediction studies can be broadly grouped based on the basic approach, which vary from purely theoretical geophysics, to genetic mutations and biology, to statistical, mathematical, and computational modeling of earthquake parameter data recorded in historical catalogs of seismic regions. The papers reviewed in this article are classified into two groups: (1) studies based on recording and analyzing earthquake precursors (seismic monitoring); and (2) studies based on historic earthquake data analysis.
3	UAV Image-based Forest Fire Detection Approach Using Convolutional Neural Network	Wildfire detection; fire classification; fire segmentation; vision transformers; UAV; aerial image	https://www.researchgate.net/publication/335865644_UAV_Image-based_Forest_Fire_Detection_Approach_Using_Convolutional_Neural_Network	CNN	EfficientNet-B5 and DenseNet-201 models, is proposed to identify and classify wildfire using aerial images. In addition, two vision transformers (TransUNet and TransFire) and a deep convolutional model

					(EfficientSeg) were employed to segment wildfire regions and determine the precise fire regions.
4	UAVs in Disaster Management: Application of Integrated Aerial Imagery and Convolutional Neural Network for Flood Detection Authors : afiz Suliman Munawar Mohammad Mojtahedi	Convolutional neural network (CNN); Disaster management; aerial imagery; flood detection; unmanned aerial vehicles (UAVs)	https://www.researchgate.net/publication/353015053_UAVs_in_Disaster_Management_Application_of_Integrated_Aerial_Imagery_and_Convolutional_Neural_Network_for_Flood_Detection	CNN	For training phase, 2150 image patches are created by resizing and cropping source images.These patches in training dataset train CNN model to detect and extract the regions where a flood related change has occurred.This model is tested against both pre and post disaster images to validate it .0 highlights the occurrence of a disaster, whereas 1 represents no Disaster.
5	Current efforts for prediction and assessment of natural disasters: Earthquakes, Tsunamis, Volcanic eruptions, tornados, and floods Author: P.AMezquita-Samez, Valtiera-Rodrigu, H.Adeli	Big data prediction	https://www.magiran.com/paper/1779408/?lang=e n		signal and image processing techniques, and statistical analyses used for prediction and assessment of natural disasters

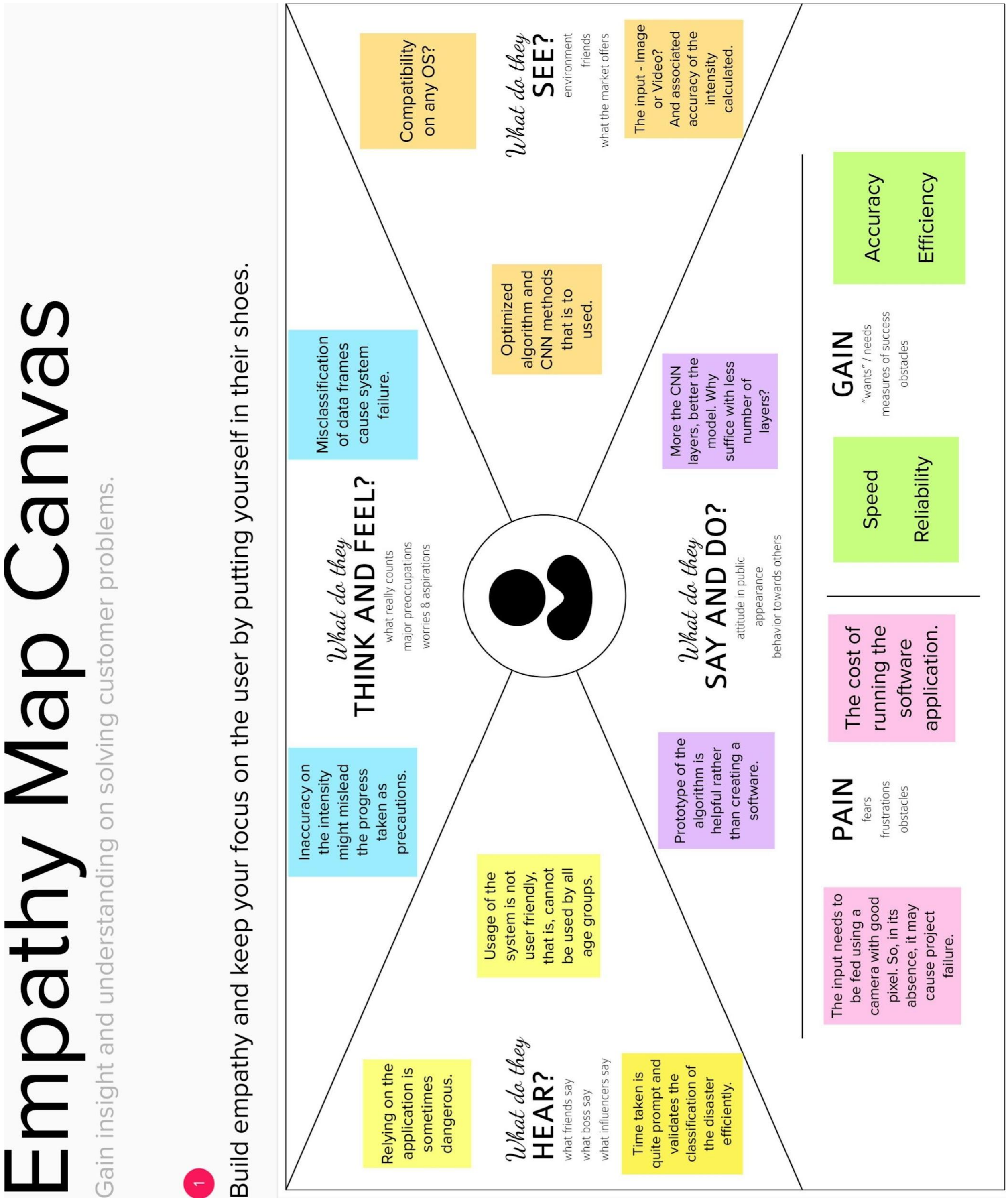
2.3 Problem Statement Definition

Natural disasters are inevitable, and the occurrence of disasters drastically affects the economy, ecosystem and human life. Buildings collapse, ailments spread and sometimes natural disasters such as tsunamis, earthquakes, and forest fires can devastate nations. When earthquakes occur, millions of buildings collapse due to seismological effects. 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.

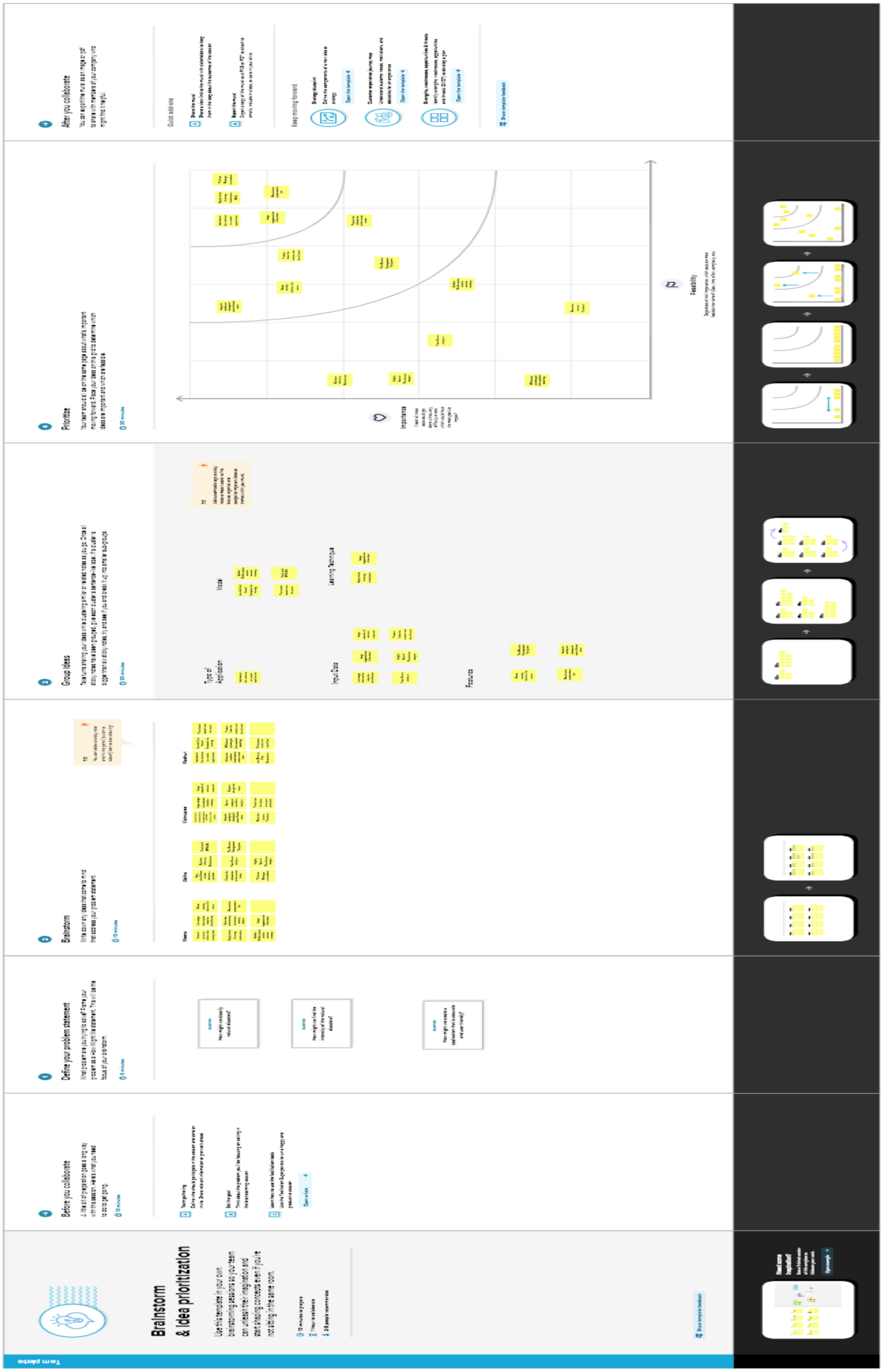
Question	Description
What does the problem affect?	People having normal lives are getting affected by unexpected natural calamities.
What are the boundaries of the problem?	Normal life, geographic, worklife
Why is it important to fix this problem?	Since, the intensity of the disaster is important to be reviewed in order to fix the damages which are caused by the calamity, we need to analyze the intensity of the damages. It should be able to predict the damages occurred in a short time. It helps to get the help faster to help the people who got affected because of the natural calamity. So, it is important to study and predict the intensity of the natural disaster.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



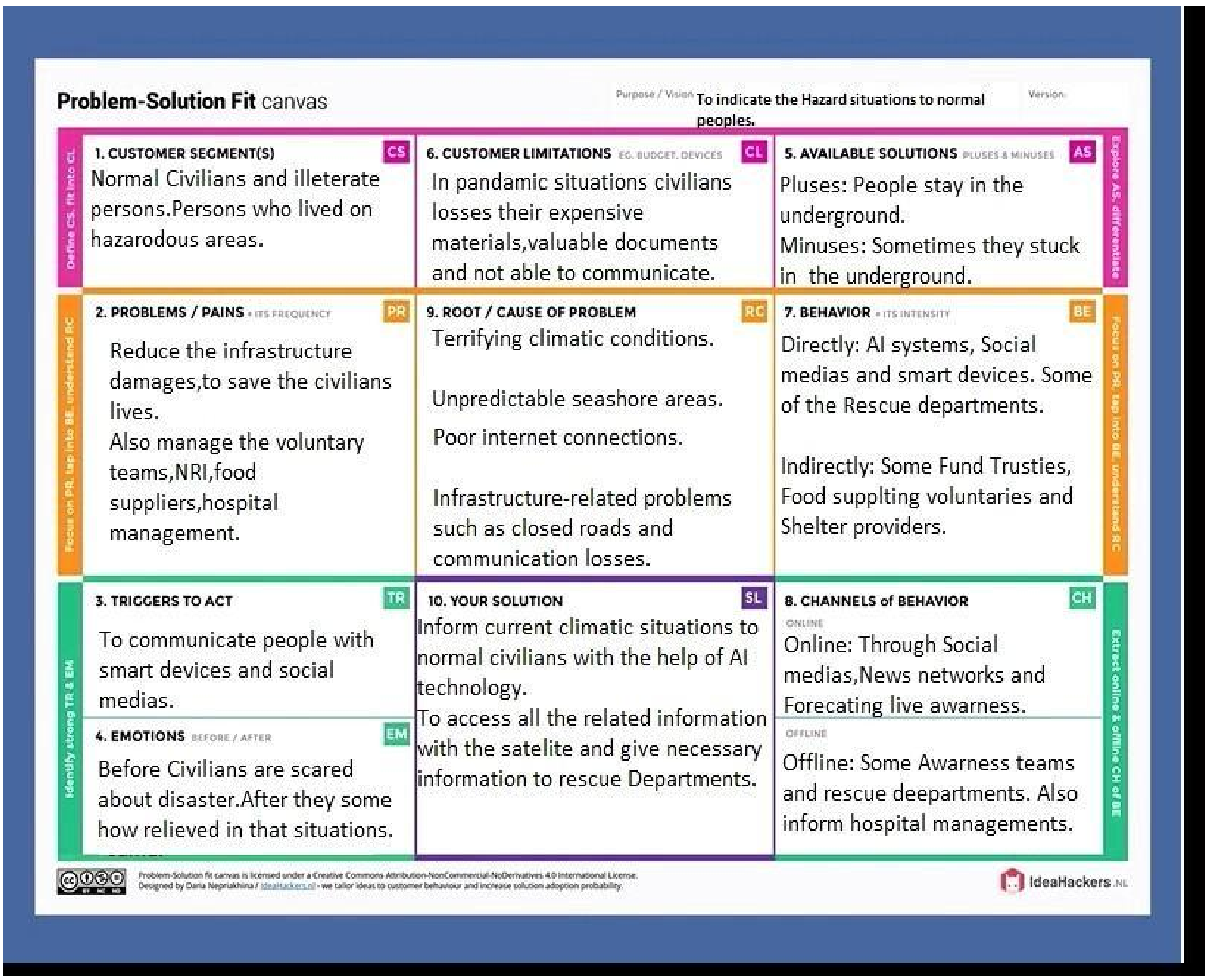
3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement	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 (CNN) that classifies the naturaldisaster 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



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

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 an online application using Gmail.
FR-2	User Confirmation	Confirmation via Email
FR-3	User Preparation	Ensures safety of all people and provision of food.
FR-4	User Evacuation	Safe evacuation ways would be advised.

4.2 Non-Functional requirements

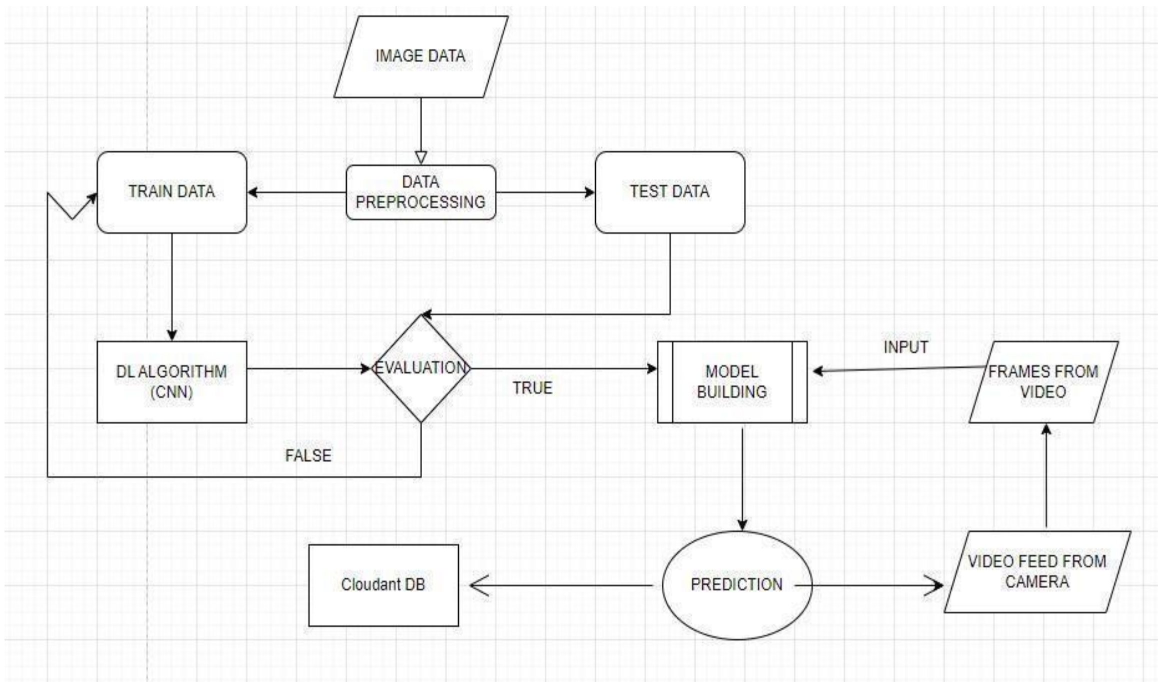
Following are the non-functional requirements of the proposed solution.

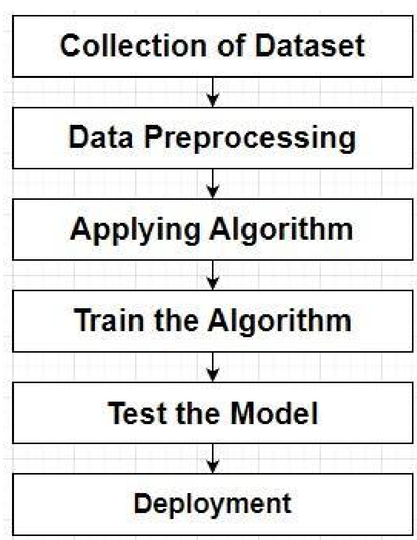
FR No.	Non-FunctionalRequirement	Description
NFR-1	Usability	Prediction of disaster intensity can be done with ease.
NFR-2	Security	The secure pattern shares components with monitor and control for logging and control access for providing audit trails.
NFR-3	Reliability	High reliability since it deals with the lives of people.
NFR-4	Performance	Depends on the throughput of the application and feed of images (dataset).
NFR-5	Availability	It is available 24/7 as far as WIFI exists.
NFR-6	Scalability	Disaster can affect people that can be examined by taking note of the number of fatalities and injuries.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.





Example :DFD Level 0(Industry Standard)

5.2 Solution & Technical Architecture

Solution Architecture:

A network less\issues area we can provide an offline facility.
Our application\website not only provides risk alerts to normal civilians. Also, we provide some of the suggestions to peoples that, what can do in those critical solutions, suggest some safe nearby areas.
We also provide those civilians understandable languages and an easy user interface.
We should provide proper climatic situations alerts.
We also share the hazardous area's locations to the rescue department and also inform the hospital management.
We also include the facilities that the user can also should give the situation in his\her living areas in which the disaster occurred.
We provide another facility where the user can interact with the weather analysts and International Rescue committee.

Reliability:

★ Some of the civilians can live on network coverage less areas. In that situation we can provide some of the offline facilities for what kind of actions they do.
★ Some of the civilians may be illiterate, so we will try to provide an effective user interface.
★ If any civilian can be needed for help, he\she will contact our application\website we will send that request to that nearby rescue department or voluntary team to provide some considerable help.

Usability:

Civilians can protect their lives and valuable documents.
People can know nearby safe areas.
Civilians can know what kind of disaster can occur.
Civilians can also know what kind of actions they do in those critical situations.

Technical Architecture:

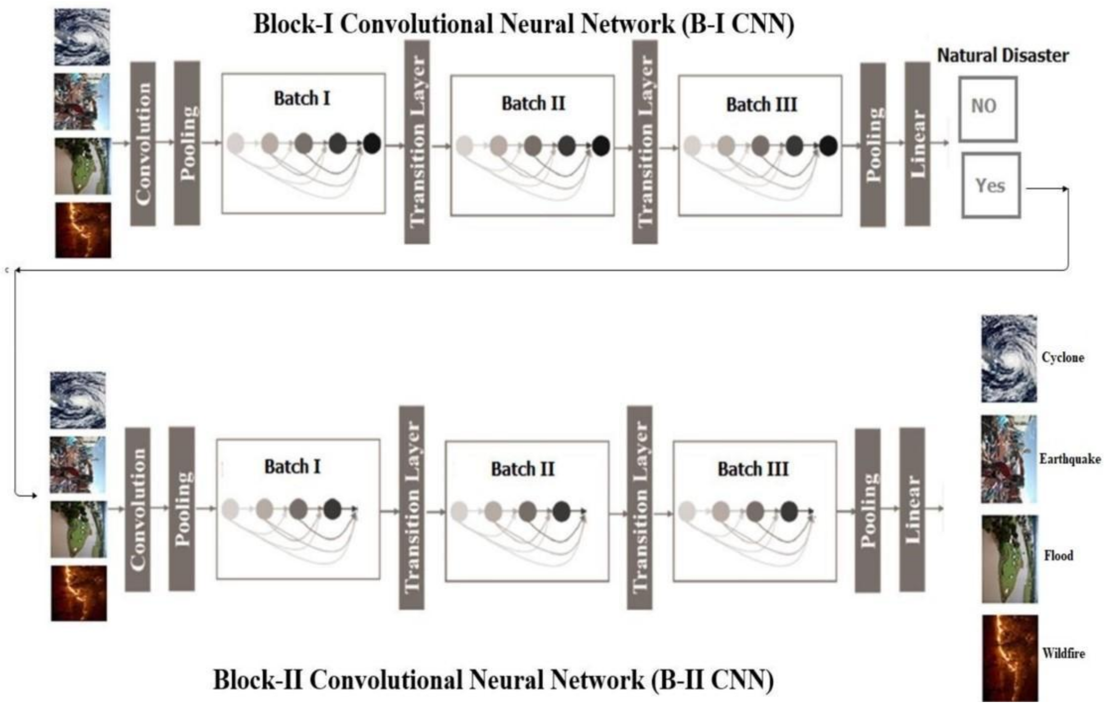


Fig : Technology architecture

Table :Components and Technologies

S.No	Component	Description	Technology
1.	Image processing	Dataset of satellite images of the particular area.	Python.
2.	User Interface	Interaction of the application with the user.	HTML,CSS, JavaScript, ReactJS
3.	Prediction of issue	The issue is predicted by various models.	Python.
4.	Database	Data is stored as image and dataset.	MySQL.
5.	Cloud Database	Database service on IBM Cloud.	IBM DB2
6.	File Storage	CSV files and image formats.	IBM block storage and local file system storage.

7.	External API-1	Google, web browsers and applications	IBM, Weather API
8.	External API-2	To know the current issue in the particular area.	Weather forecast
9.	Machine Learning model.	Model to process the image and data.	Object Recognition Model

Table : Application characteristics

S.No.	Characteristics	Description	Technology
1.	Open-Source framework	Jupyter, spyder, python, colab	Google
2.	Security implementation	The client is the user and server is IBM	SHA-256, Encryptions
3.	Scalable Architecture	Should be able to handle all workloads without consequences.	IBM Cloud
4.	Availability	Availability of application to all the users.	IBM Cloud.
5.	Performance	Application should perform correctly.	IBM Cloud

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web 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 - 4
		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 - 3
	Login	USN - 3	As a user, I can log into the application by entering valid email & password		High	Sprint - 4
Protection Services department	Login	USN - 4	Person can log into the application by entering their name.		High	Sprint - 1
Customer Care service	Register	USN - 5	The user can register for the application through official Gmail account	They can register and access the dashboard	Medium	Sprint - 2

6. PROJECT PLANNING & SCHEDULING
6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the technical papers, research publications etc.	3 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	10 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and Prioritize the top 3 ideas based on the feasibility & importance.	10 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	2 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution Fit document.	29 SEPTEMBER 2022

Solution Architecture	Prepare solution Architecture document.	20 OCTOBER 2022
Solution Requirements	Prepare Solution Requirements (Functional and Non-functional)	17 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow Diagrams and submit for review.	17 OCTOBER 2022
Technology Architecture	Architecture diagram.	15 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & Activity list of the project.	27 OCTOBER 2022
Project Development - Delivery of Sprint- 1, 2, 3 & 4	Develop & submit the developed code by testing it.	18 NOVEMBER 2022

6.2 Sprint Delivery Plan

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collection of Dataset	USN-1	Acquisition of the Natural disasters dataset	5	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-1	Dataset Preprocessing	USN-2	The natural disaster images should be preprocessed using ImageDataGenerator Library and configured using ImageDataGenerator Class.	15	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-2	Building the CNN Model	USN-3	Build a CNN Model for classifying the disasters by using the appropriate layers, and split the preprocessed dataset	4	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-2	Train, Test, and Validate	USN-4	Train the model, validate it using the Metrics and test the model on an anonymous image/video, using the partitioned dataset.	8	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-2	Optimization and Intensity detection	USN-5	Improve on the accuracy and time complexity of the model, and include features for predicting the intensity of classified disaster	8	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-3	User Interface Dashboard and Login	USN-6	As a user, I can register for the application by entering my email, password, and verifying account via mail	10	Medium	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-3	Upload images to the application	USN-7	As a web user, I must capture and upload any images of natural disaster occurrences with better clarity.	10	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-4	Models Outputs through UI and alerts	USN-8	Ensure accurate classification of disaster, and provide the necessary alerts based on intensity to the user.	10	High	Abitha P, Meena Muthukumar, Madhuri Mahalingam,C. Vishnusree
Sprint-4	Login using Third party Service Accounts	USN-9	As a user, I can use feature of OAuth to login using Gmail.	5	Low	Abitha P, Meena

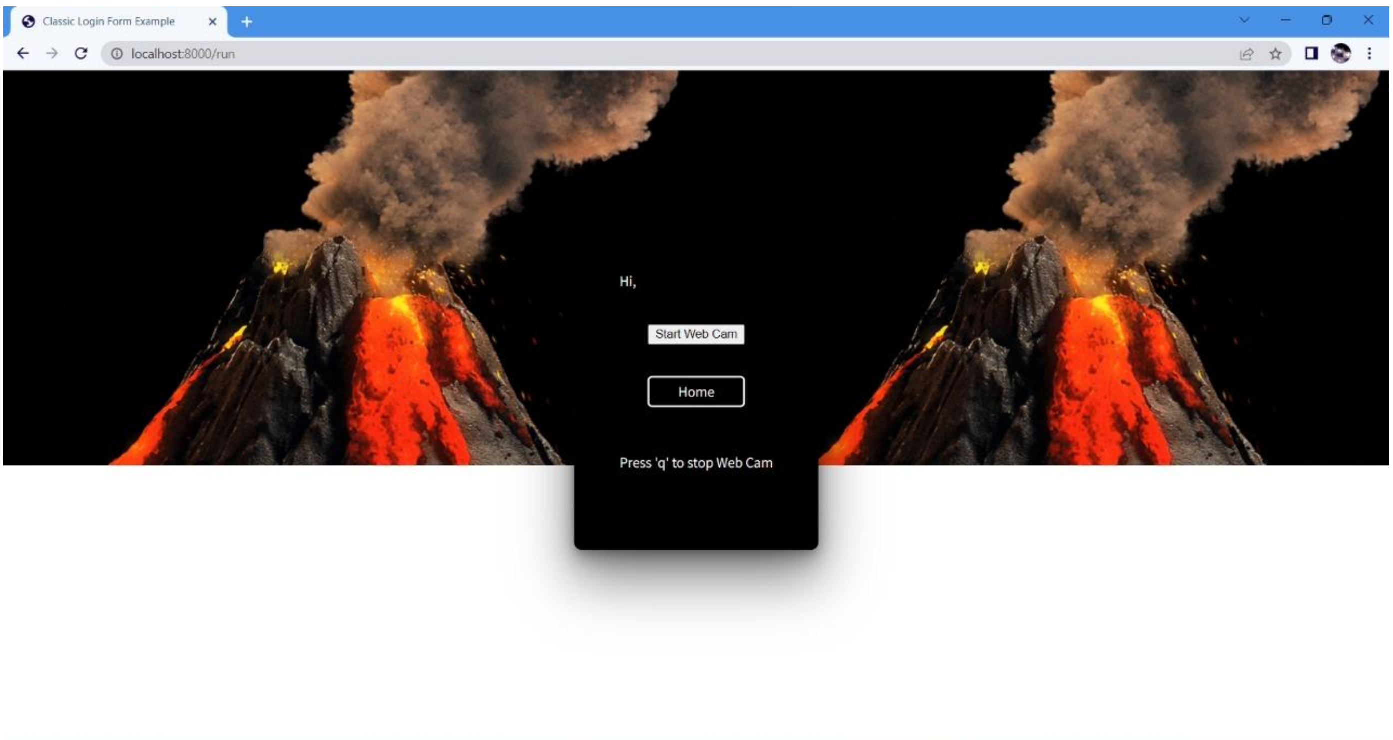
7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

This feature allows third party login of users which enables all the users to access the website.

7.2 Feature 2

Webcam feature is added to capture the live disasters since it will be useful if we can find the intensity in real time.



7.3 Feature 3 : Intensity

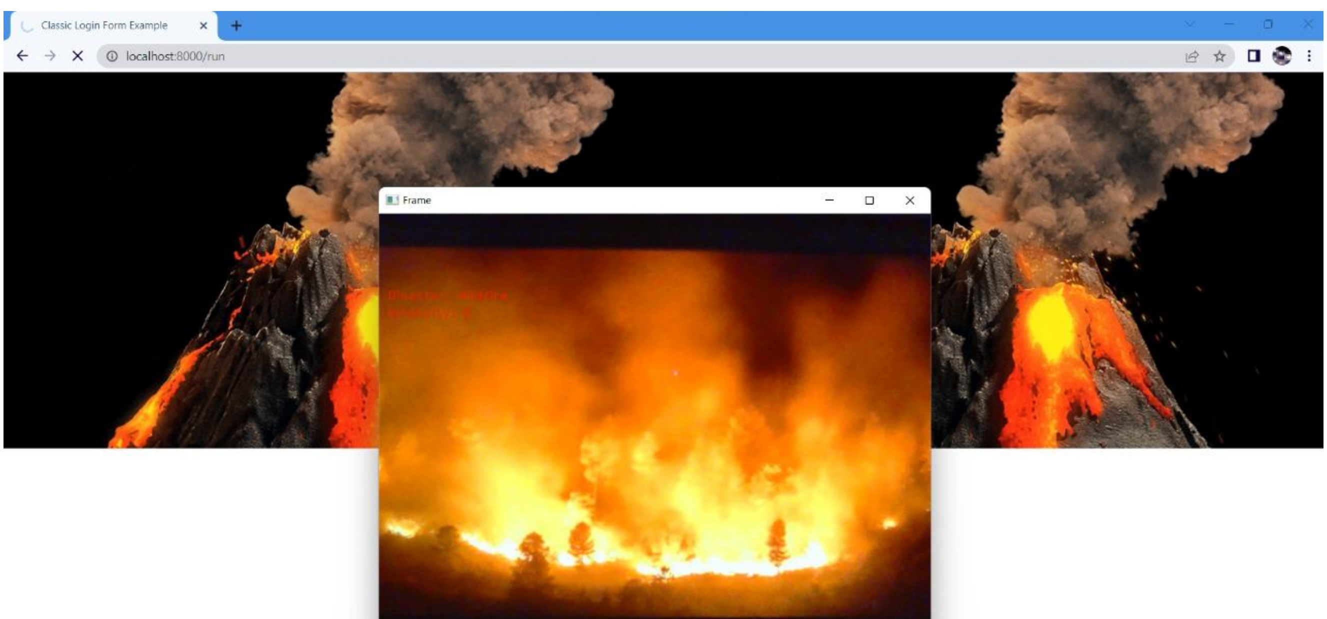
The intensity of the natural disaster is predicted.

8. TESTING

8.1 Test Cases

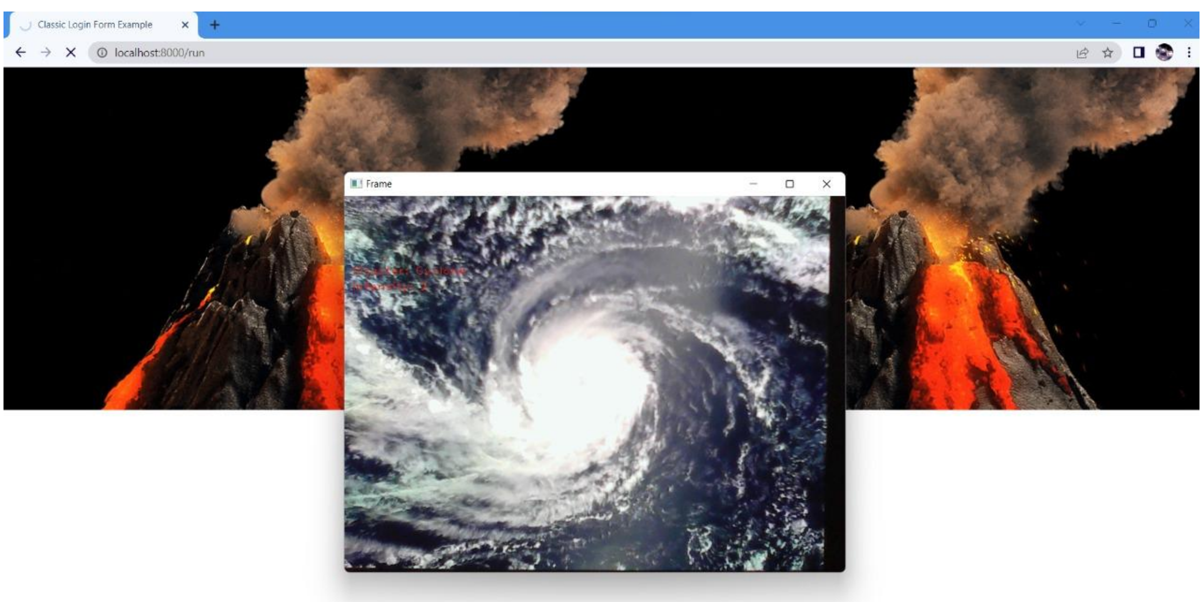
Test case 1:

Output:



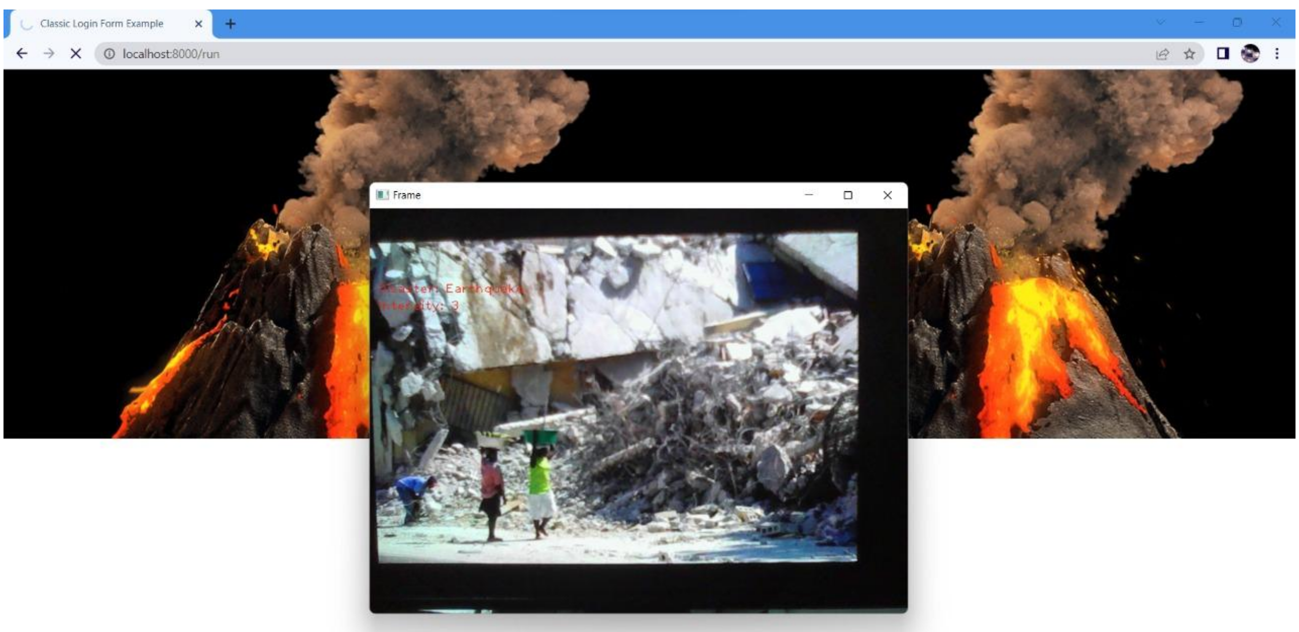
Test case 2:

Output:



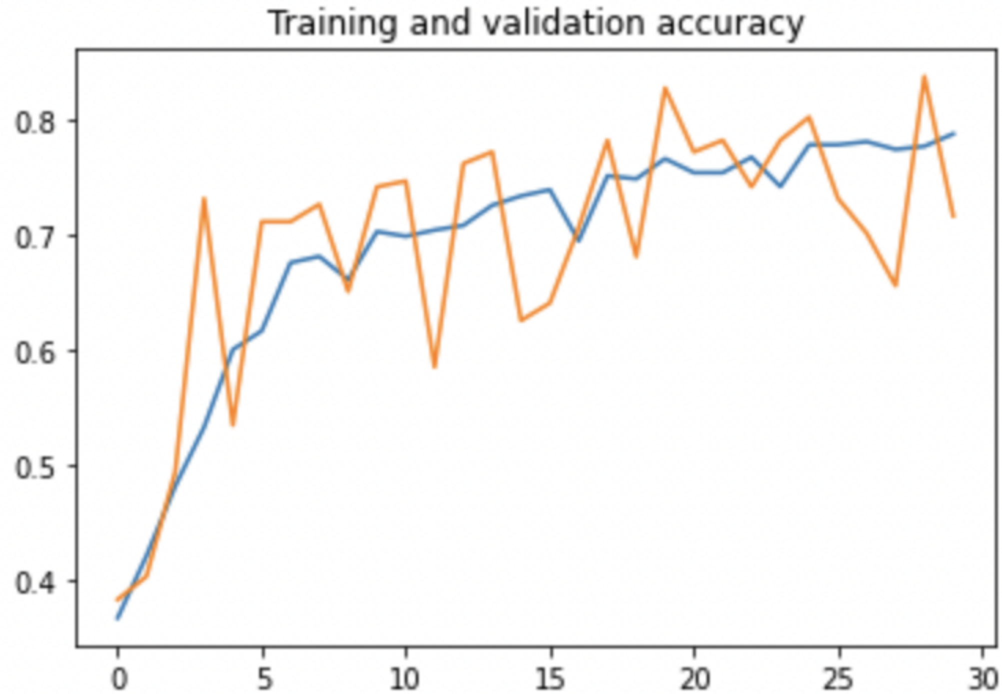
Test case 3:

Output:



9. RESULTS

9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

Disadvantages:

- **Webcam:** High pixel quality webcam should be used to predict the intensity of the disaster with high efficiency, else it can lead to misprediction.
- **Internet facility:** Since a webcam is required to capture the images the internet facility available in that location should be strong else the accuracy and the intensity cannot be calculated efficiently.

Advantage:

- **Frame by frame capture:** Since the data is captured frame by frame,quality and accuracy of the prediction is enhanced. The increase in fire in the region can be captured by using frame by frame capture.
- **Intensity prediction:** Since the intensity is predicted, the disaster management can take place in a much faster way and the measures to be taken to control the situation can be decided much faster.

11. CONCLUSION

Many solutions have been proposed to predict the natural disaster intensity analysis using Deep Learning techniques but still a lot of problems are faced due to noise and imbalance problems. Hence deep convolutional neural networks with multiple layers have been developed to improve accuracy and reduce the errors.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 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. Our approach helps to narrow down searching casualties in a disaster area in which the results presented here may promote further enhancement in detecting natural disasters more efficiently.

12. FUTURE SCOPE

In future, a separate application can be created for this and people can be notified in case of a natural disaster in their locality and also this helps predict the intensity of the disaster by which people can get to know the effect of that disaster in their daily life and can make preventive measures to get to a safer place before the disaster goes out of control. The accuracy and intensity of the disaster classification can be further improved and hence reducing the false alarm rates. A separate model can be created to detect the number of casualties in the affected area which will be helpful for rescue operations. The application can be connected to the government disaster management department making it easier to manage faster management measures.

13. APPENDIX

Source Code:

App.py

```
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("disaster.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('/run.html',methods=['GET'])
def upload():
  return render_template('run.html')

@app.route('/uploader.html',methods=['GET','POST'])
def predict():
      _,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("run.html")

if __name__ == '__main__':
  from waitress import serve
  serve(app, host='0.0.0.0',port=8000)
```

ML model.py:

```
from google.colab import drive
drive.mount('/content/drive')

#Import ImageDatageneratorLibrary

from keras.preprocessing.image import ImageDataGenerator
import numpy as np #numerical analysis
from sklearn.model_selection import train_test_split
from keras.models import Sequential, load_model
from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard
from keras.callbacks import ReduceLROnPlateau
#Dense layer is regularly deeply connected neural network layer
#MaxPooling2D to downsample the image
from keras.layers import Conv2D, Dropout, Dense, Flatten, MaxPooling2D,
SeparableConv2D, Activation, BatchNormalization
import matplotlib.pyplot as plt
import tensorflow as tf
from keras.preprocessing import image
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array

#configure ImageDatGenerator Class
#Define the parameters/arguments for ImageDataGenerator class
train_datagen=ImageDataGenerator(rescale=1./255,
                                shear_range=0.2,
                                rotation_range=180,
                                zoom_range=0.2,
                                width_shift_range=0.2,
                                height_shift_range=0.2,
                                  horizontal_flip=True)
```

```
test_datagen=ImageDataGenerator(rescale=1./255)

#Applying ImageDataGenerator functionality to train data

x_train
train_datagen.flow_from_directory(r"/content/drive/MyDrive/dataset/train_set",
                                                target_size = (64,64),
                                                batch_size=5,
                                                color_mode='rgb',
                                                class_mode='categorical')
Applying ImageDataGenerator functionality to test data

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

#initializing the model
model=Sequential()

#add convolutional, maxpooling and flatten layers
model.add(Conv2D(32,(3,3),activation="relu",input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(3,3),activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

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

model.summary()

#configure the learning process
model.compile(loss='categorical_crossentropy',
             optimizer='adam',
             metrics=['accuracy'])
x_test.class_indices

#Training the model
history = model.fit(x_train,epochs=20,
                   steps_per_epoch=len(x_train),
```

```
                    validation_data=x_test,validation_steps=len(x_test))

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

#load the saved model
model = load_model("disaster.h5")

#load image
img=load_img(r"/content/drive/MyDrive/dataset/test_set/Earthquake/1321.jpg",ta
rget_size=(64,64))
x=img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']

#predict class
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])

#load image
img=load_img(r"/content/drive/MyDrive/dataset/test_set/Cyclone/921.jpg",target_
size=(64,64))
x=img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']

#predict class
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])

# Retrieve a list of accuracy results on training and validation data
# sets for each training epoch
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

# Retrieve a list of list results on training and validation data
# sets for each training epoch
loss = history.history['loss']
val_loss = history.history['val_loss']
```

```
# Get number of epochs
epochs = range(len(acc))

# Plot training and validation accuracy per epoch
plt.plot(epochs, acc)
plt.plot(epochs, val_acc)
plt.title('Training and validation accuracy')

plt.figure()

# Plot training and validation loss per epoch
plt.plot(epochs, loss)
plt.plot(epochs, val_loss)
plt.title('Training and validation loss')
```

GitHub & Project Demo Link:

Github: https://github.com/IBM-EPBL/IBM-Project-27558-1660059840

Project Demo: