Natural Disasters Intensity Analysis and Classification using Artificial Intelligence

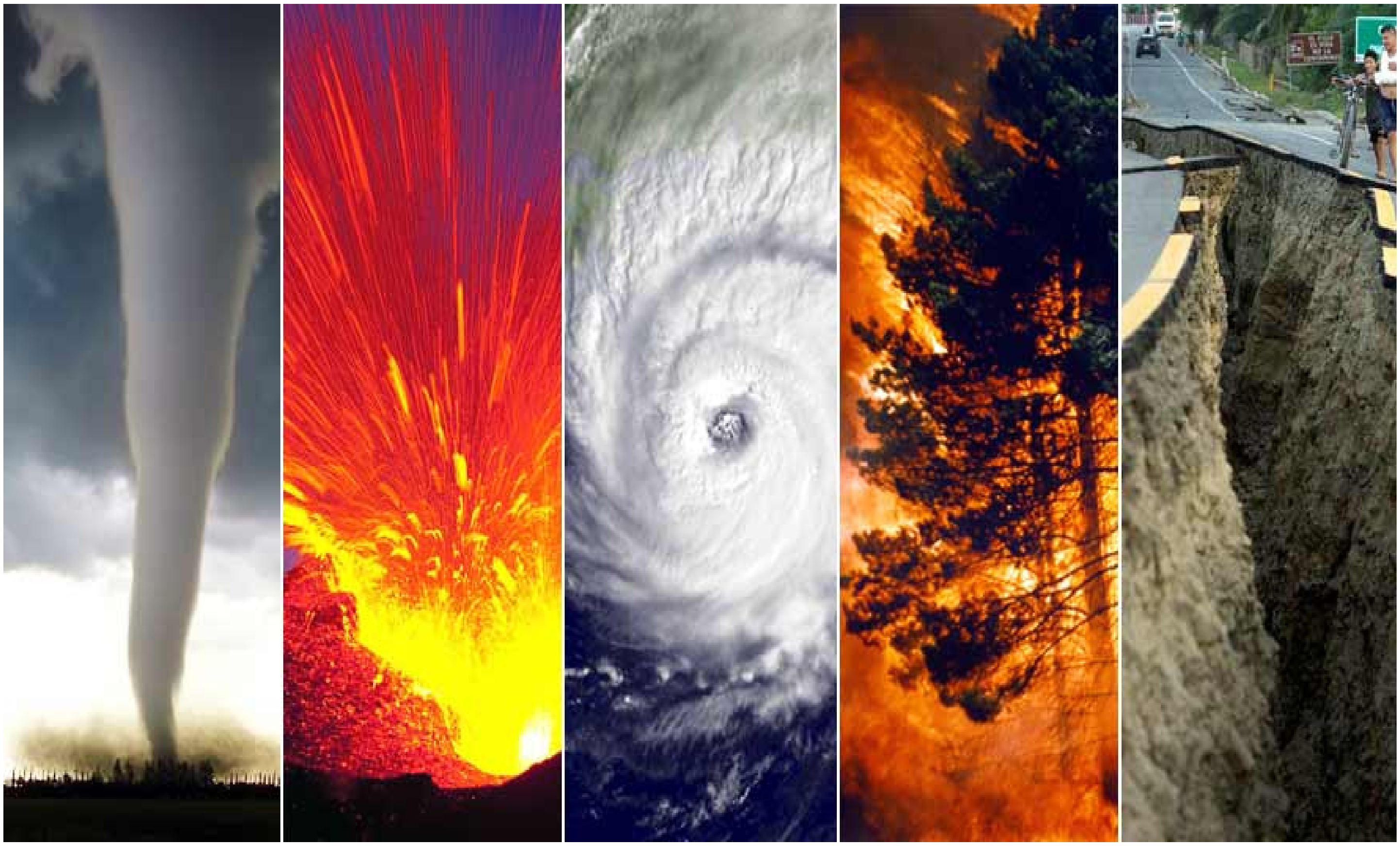


TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, 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 window.

1.2 PURPOSE

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 machine learning approaches have been used for wildfire predictions since the 1990s. A recent study used a machine learning approach in Italy. Floods are the most devastating natural disaster, damaging properties, human lives and infrastructures. To map flood susceptibility, an assembled machine learning technique based on random forest (RF), random subspace (RS) and support vector machine (SVM) was used. As the population is escalating, 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. Populations in underdeveloped countries cannot afford damages disasters cause 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 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 after effects of natural disasters on human life, infrastructure, and transmission lines by capturing images and videos. Data acquired from these UAVs helps to identify the facial expressions of victims, the intensity of their situation and their needs in a post disaster scenario. It helps to take actions and carry out necessary operations to tackle devastating scenarios. Raw images obtained from camera-equipped UAVs are processed and neural network-based feature extraction techniques are applied to analyse the intensity.

A deep learning method for the reconstruction of two-dimensional cardiac magnetic resonance images was proposed to enhance the image data acquisition process. Cascade deep convolutional neural networks use a 10-fold method to reconstruct the feature map for the MR images. In this way, feature extraction sequence becomes swift and it takes less than 5 to 10 s to extract the feature matrix.

Neural networks provide multilevel network architectures, where

Convolutional Neural Networks (CNNs) are the most frequently implemented architecture as the direct input of multidimensional vector images, speech recognition, and image processing can be carried out with low complexity. CNNs efficiently perform feature extraction by de-noising the images and removing interference and achieve highly accurate results.

The proposed multi-layered deep convolutional neural network method works in two blocks of convolutional neural networks. The first block, known as Block-I Convolutional Neural Network (B-I CNN), detects the occurrence of a natural disaster and the second one, known as Block-II Convolutional Neural Network (B-II CNN), defines the intensity of the natural disaster. Additionally, the first block consists of three mini convolutional blocks with four layers each and includes an image input and fully connected layers. On the other hand, the second block also consists of three mini convolutional blocks with two layers each, including an image input layer and fully connected layer.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Natural catastrophes are unavoidable, and their occurrence has a significant impact on the economy, ecological, and quality of life for people. Buildings crumble, diseases spread, and occasionally natural calamities like tsunamis, earthquakes, and forest fires may wreak havoc on entire countries. Due to seismological impacts, millions of buildings fall during earthquakes.

2.2 REFERENCES

Study on Risk assessment model of urban drought in Hilly Area of Central Sichuan Basin (IEEE-2009). In this paper it is used for mathematical model for the drought risk assessment and then use this model to calculate the intensity of drought risk of Nanchong city in Hilly Area of Central Sichuan Basin from different perspective.

Land Surface Temperature retrieval using HJ1B/IRS data and analysis of its effect (IEEE-2009). In this paper, the generalized single channel algorithm is utilized to achieve the LST from HJ-1B/IRS.

Urban Damage Detection Using Decorrelation of SAR Interferometric Data (IEEE-2002). It indicates a fact that the building damage causes the interferometric decorrelation. It can be detected using interferometric decorrelation of ERS and JERS-1 SAR data.

Quantifying change after natural disasters to estimate infrastructure damage with mobile phone data (IEEE-2018). It indicates that how mobility patterns are changing, in the post disaster It indicates that how mobility patterns are changing, in the post disaster timeframe, is crucial in order to settle rescue people from affected area.

Spatio–Temporal Analysis for Understanding the Traffic Demand After the 2016 Kumamoto Earthquake Using Mobile Usage Data (IEEE-2018). It mainly focuses on the effect of natural disasters on the population density transition.

Degree of network damage: A measurement for intensity of network damage (IEEE-2004). To define degree of network damage (DND), a measurement used to classify the effect of a destructive event on network infrastructures, human, and traffic flows.

Natural Disasters Intensity Analysis and Classification Based on Multispectral Images Using Multi-Layered Deep Convolutional Neural Network (2021) by Muhammad Aamir , Tariq Ali, Muhammad Irfan , Ahmad Shaf , Muhammad Zeeshan Azam , Adam Glowacz, Frantisek Brumercik

Earthquake risk assessment in NE India using deep learning and geospatial analysis by Ratiranjan Jena A, Biswajeet Pradhan, Sambit Prasanajit Naik d, Abdullah M. Alamri (2021)

2.3 PROBLEM STATEMENT DEFINITION

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

People needs a way to classify and analyse the natural disaster so that they can prevent themselves from loses due to the disaster and millions of lives.

- Tsunami is a dangerous natural disaster causes health risks like contaminated water which should be analysed and studies to prevent unnecessary deaths, drownings, traumatic injuries are also primary concern.

- People and animals are facing so many issues like loss of life, property, resources and deterioration of the air quality due to wildfire. So, we need to analyse and detect natural disaster and protect them from such disasters.

- Floods are the most frequent type of natural disaster which is caused due to the heavy rainfall, rapid snow-melt., should be classified and intensity is analysed so that the loss of human life, crop destruction, property are prevented for that zone people.

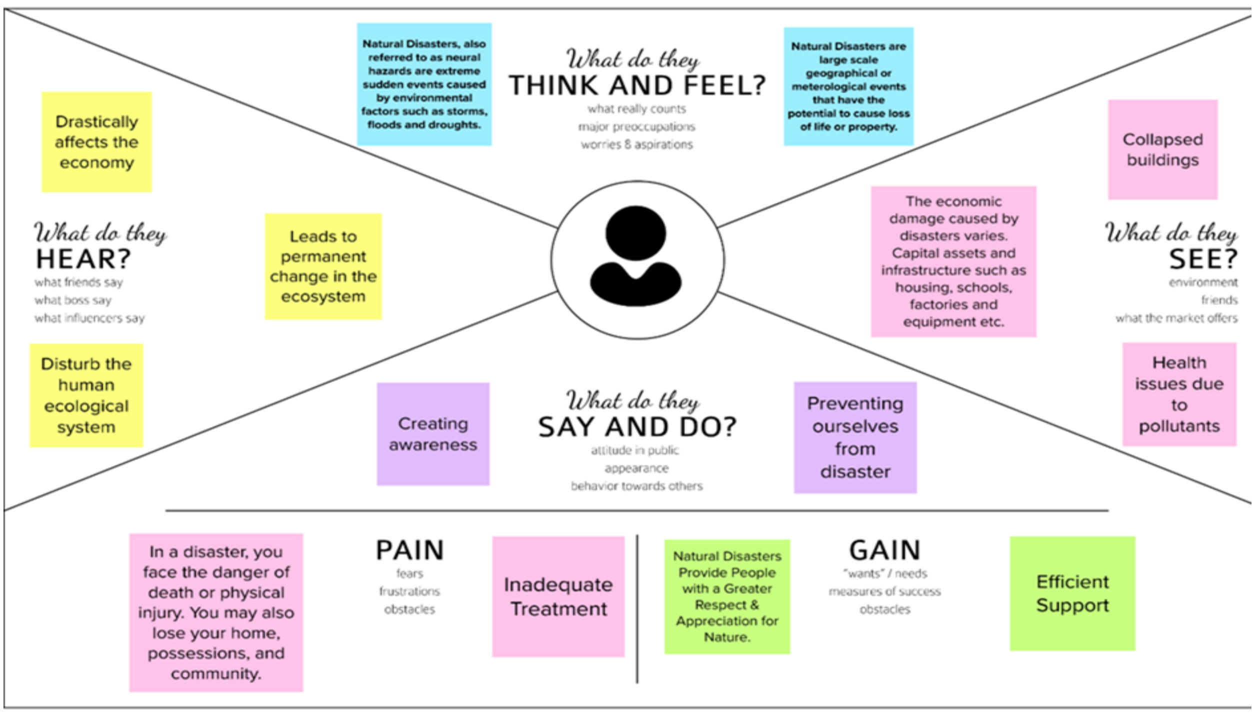
- Earthquake causes seismic waves, it travels through the earth crust, due to which the buildings are destroyed, it needs a way to analyse the intensity of the disaster so that the people in that zone can prevent themselves from major problems.

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Natural Disaster Intensity Analysis and Classification using Artificial Intelligence

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.

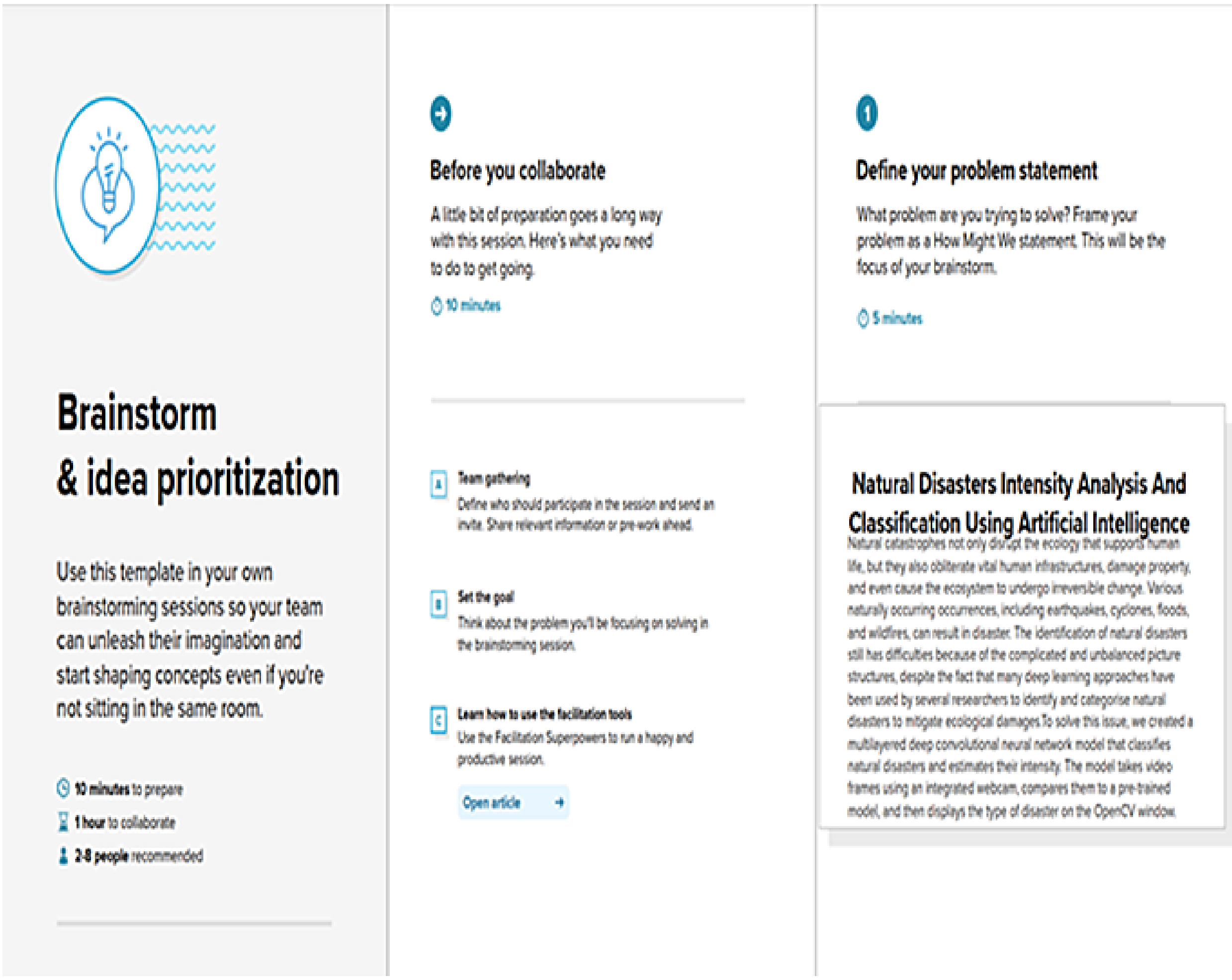


3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

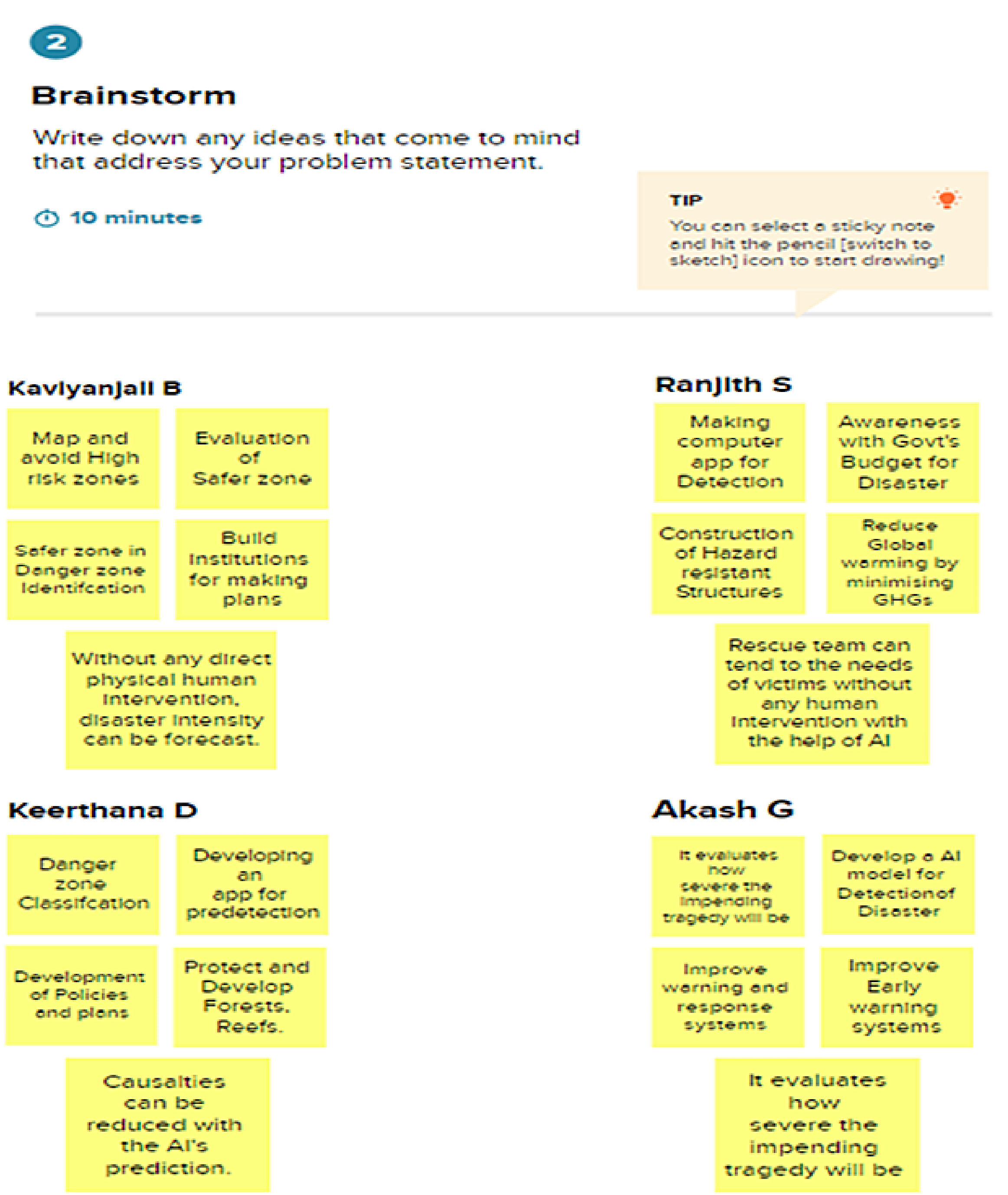
Step-1: Team Gathering, Collaboration and Select the Problem

Statement



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
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
Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.
Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.
Open article
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
Natural Disasters Intensity Analysis And Classification Using Artificial Intelligence
Natural catastrophes not only disrupt the ecology that supports human life, but they also obliterate vital human infrastructures, damage property, and even cause the ecosystem to undergo irreversible change. Various naturally occurring occurrences, including earthquakes, cyclones, floods, and wildfires, can result in disaster. The identification of natural disasters still has difficulties because of the complicated and unbalanced picture structures, despite the fact that many deep learning approaches have been used by several researchers to identify and categorise natural disasters to mitigate ecological damages.To solve this issue, we created a multilayered deep convolutional neural network model that classifies natural disasters and estimates their intensity. The model takes video frames using an integrated webcam, compares them to a pre-trained model, and then displays the type of disaster on the OpenCV window.

Step-2: Brainstorm, Idea Listing and Grouping



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

TIP

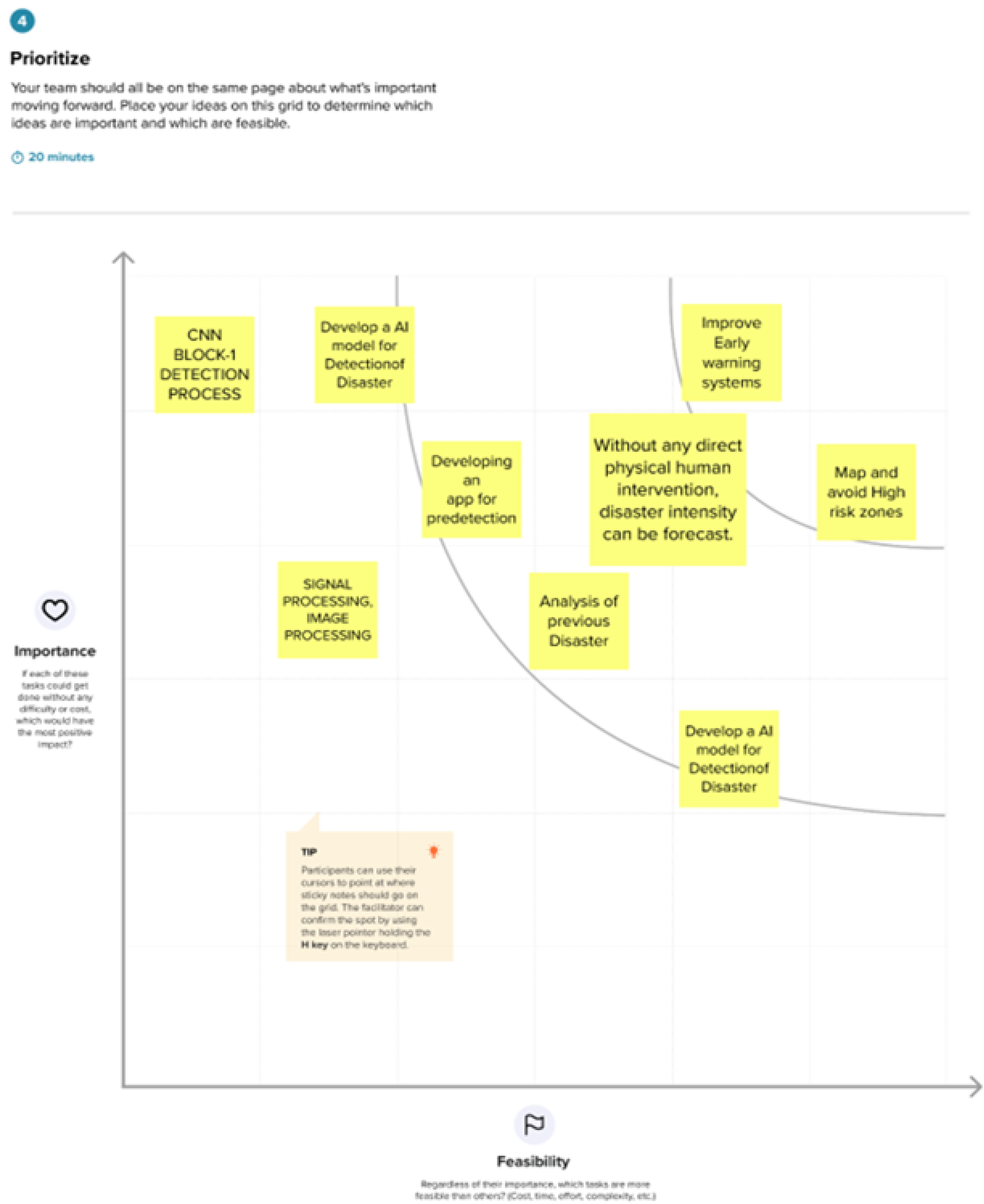
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Develop a AI model for Detectionof Disaster	Construction of Hazard resistant Structures	Analysis of previous Disaster
Improve Early warning systems	Awareness with Govt's Budget for Disaster	Developing an app for predetection
Map and avoid High risk zones	Danger zone Classifcation	Development of Policies and plans

Without any direct physical human intervention, disaster intensity can be forecast.

Rescue team can tend to the needs of victims without any human intervention with the help of AI

Step-3: Idea Prioritization



3.3 PROPOSED SOLUTION

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	The main purpose of this model is to detect and classify the type of disaster with a high accuracy rate.
2	Idea / Solution description	Emergency measures, Investments in risk reduction, Information sharing on newest research findings, Reforestation, Stable buildings, Education, Technology, Governance.
3	Novelty / Uniqueness	A natural disaster is "the negative impact following an actual occurrence of natural hazard if it significantly harms a community".
4	Social Impact / Customer Satisfaction	Increased mental health issues, alcohol misuse, domestic violence, chronic disease and short-term unemployment have resulted from extreme weather events such as bushfire s, severe

		storms, cyclones, floods and earthquakes.
5	Business Model (Revenue Model)	Be Aware of the Natural Disasters that Could Affect Your Business, Create a Disaster Response Plan, Implement Communication Plans, Backup Documents and Data, Protect the Power, Plan to Recover, Review Your Commercial Insurance Coverage.
6	Scalability of the Solution	Scalability: Implementing disaster recovery measures involves identifying new and scalable solutions, such as the cloud.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behaviour patterns and recognize what would work and why

Purpose:

❑ Solve complex problems in a way that fits the state of your customers.
❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behaviour.
❑ Sharpen your communication and marketing strategy with the right triggers and messaging.
❑ Increase touch-points with your company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems.
❑ Understand the existing situation in order to improve it for your target group.

1. Customer Segment(S) :

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

2. Jobs-To-Be-Done / Problems:

Natural disasters can cause great damage on the environment, property, wildlife and human health. These events may include . earthquakes, floods, hurricanes, tornadoes, tsunamis, landslides, wildfires, volcanic eruptions,extreme temperatures. Property damage. Structural damage to buildings. Loss of utilities like electricity and water.

3. Triggers:

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

4. Emotions : Before / After:

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

5. Available Solutions:

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

6. Customer Constraints:

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

7. Behaviour:

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

8. Channels Of Behaviour:

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

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

9. Problem Root Cause:

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

10. Your Solution:

Natural disasters cannot be prevented but they can be detected. We can measure disaster risk by analysing trends of, for instance, previous disaster losses. These trends can help us to gauge whether disaster risk reduction is being effective. We can also estimate future losses by conducting a risk assessment.

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

1. Camera Permission: The user is asked for the camera request permission when they enrolled into the website.

2. Disaster Model: the user needs to have the disaster model that is trained with several datasets, which is used to predict the disaster captured from the webcam image.

3. UI(user interface): UI model needs to be more interactive for the user and it also should be ease to use. making a interactive ui is as important as the making the model accuracy.

4.Resolution: The resolution of the integrated web camera should be high enough to capture the video frames.

5. Accuracy: Making The model's accuracy is the most important factor in the AI model project

Software Requirements:

- Anaconda Navigator: Python version 3.7
  - Jupyter Notebook
  - Spyder

  Packages:
    - Numpy
    - Pandas
    - Scikit-learn
    - Opencv-pyhton
    - CNN
    - Tensorflow 2.3.0
    - Keras 2.4.0
    - Flask

Hardware Requirements :

i. Processor: Minimum 1 GHz; Recommended 2GHz or more.
ii. Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
iii. Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
iv. Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.

4.2 NON-FUNCTIONAL REQUIREMENT

1. Usability: Use friendly and classify the disaster easily.

2. Security: The model is secure due to the cloud deployment models and also there is no login issue.

3. Reliability: Accurate prediction of the natural disaster and the website can also be fault tolerant.

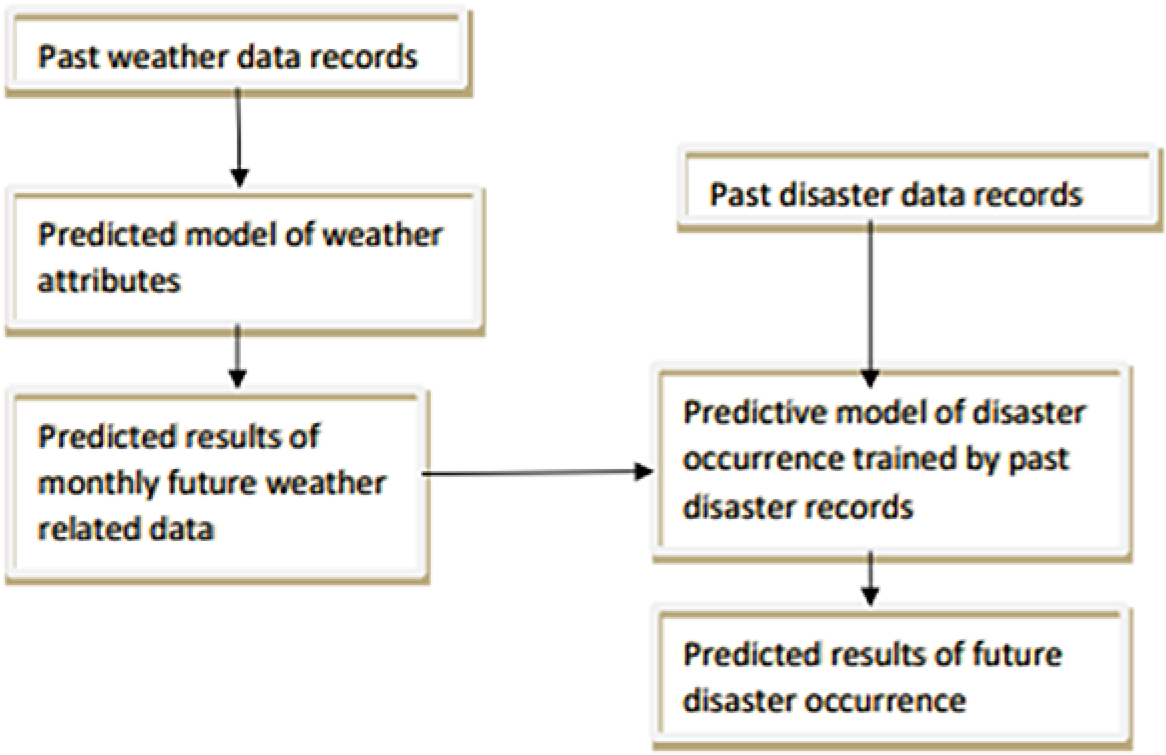
4. Performance: It is shown that the model gives almost 90 percent accuracy after continuous training.

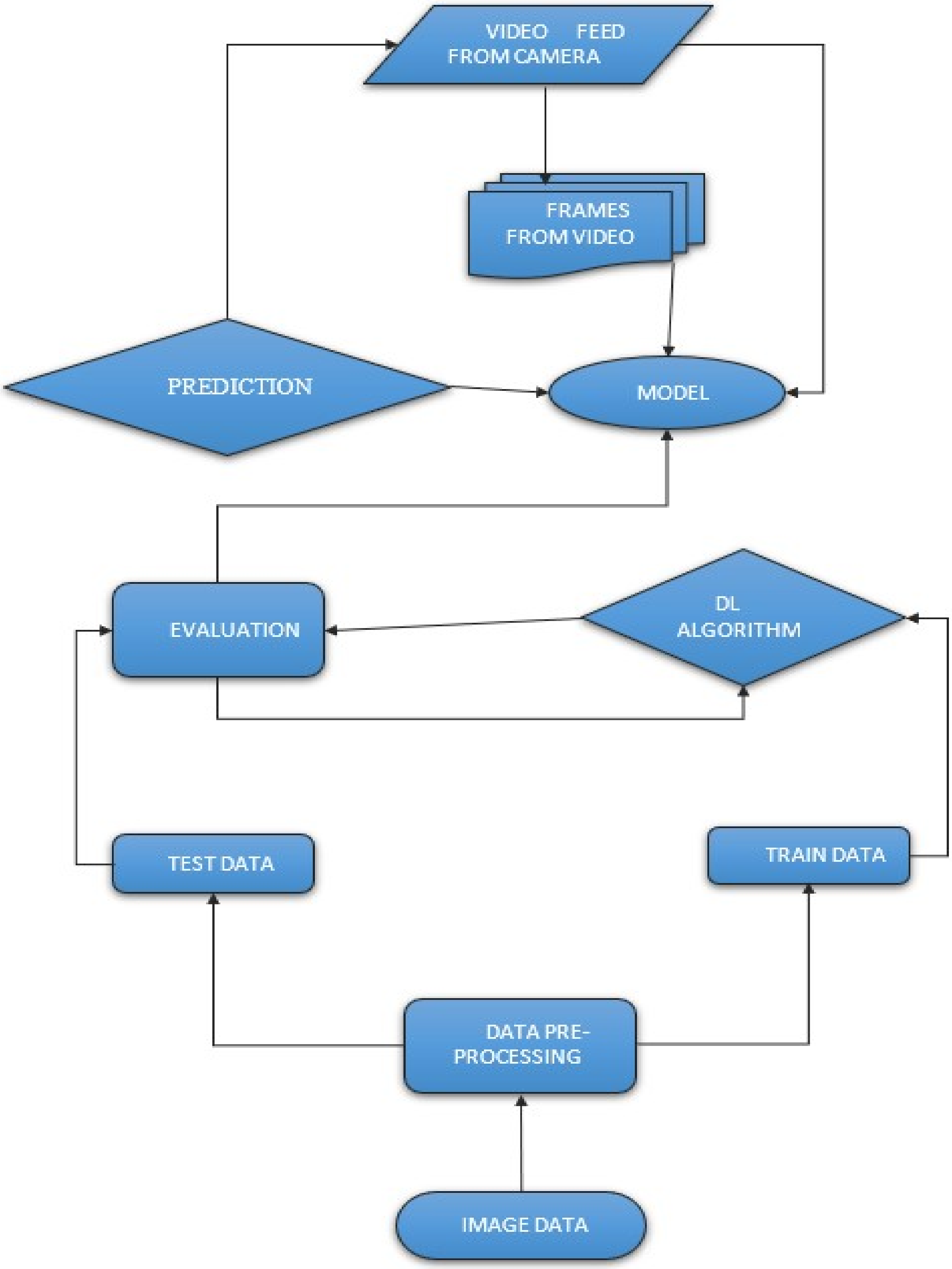
5. Availability: The website will be made available for 24 hours.

6. Scalability: The website can run on web browsers like Google chrome, Microsoft edge and also it can be extended to the NDRF and customers.

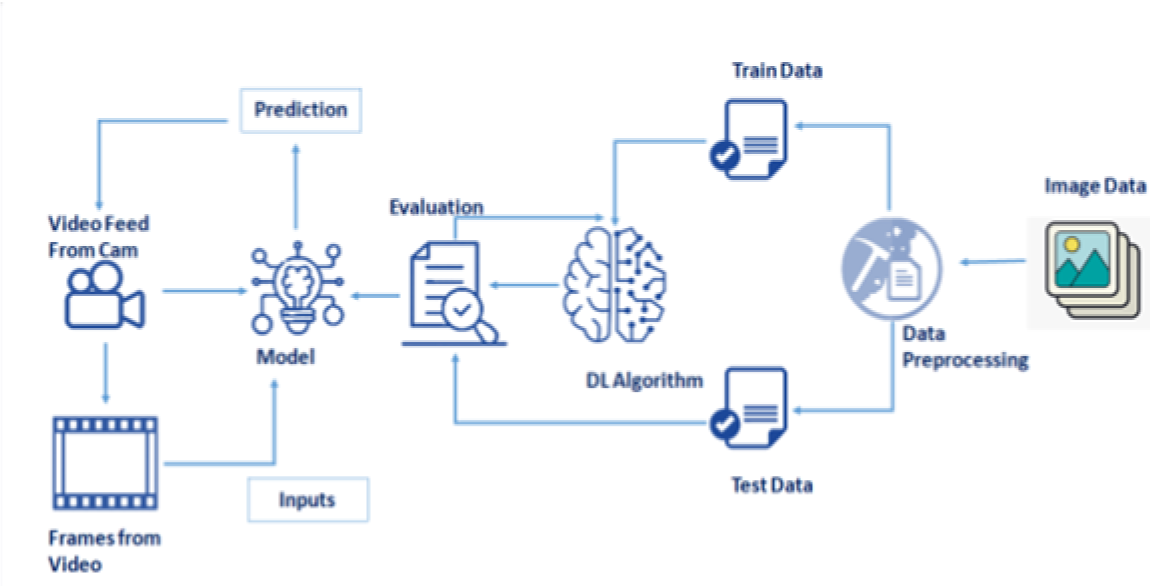
CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS




VIDEO FEED
FROM CAMERA
FRAMES
FROM VIDEO
PREDICTION
MODEL
EVALUATION
DL
ALGORITHM
TEST DATA
TRAIN DATA
DATA PRE-
PROCESSING
IMAGE DATA

5.2 SOLUTION & TECHNICAL ARCHITECTURE



Component	Description	Technology
User Interface	User interacts with application for the prediction of Any Natural disaster which will happen in future minutes.	HTML, CSS, JavaScript, Django, Python.
Data Preprocessing	Algorithms can't make sense of raw data. We have to select, transform, combine, and otherwise prepare our data so the algorithm can find useful patterns.	Image processing, pattern extraction, etc.
Model Training kit	It learns patterns from the data. Then they use	Multiclass Classification Model,

	these patterns to perform particular tasks.	Regression Model, etc
Prediction unit	This function is used to predict outcomes from the new trained data to perform new tasks and solve new problems.	Decision trees, Regression, Neural networks.
Evaluation system	It monitors that how Algorithm performs on data as well as during training.	Chi-Square, Confusion Matrix, etc.
Interactive services	To interact with our model and give it problems to solve. Usually this takes the form of an API, a user interface, or a command-line interface.	Application programming interface, etc.
Data collection unit	Data is only useful if it's accessible, so it needs to be stored ideally in a consistent structure and conveniently in one place.	IBM Cloud, SQL Server
Data generation system	Every machine learning application lives off data. That data has to come from somewhere. Usually, it's generated	Synthetic data generation.

	by one of your core business functions	

5.3 USER STORIES

1.Installation:

As a user, I have to install the webcam where the disasters are occurring.

2.Power Connection:

User has to ensure that the power supply for all the devices.

3.Safety:

User has to ensure that the device has been installed in the secure & safest place which covers the maximum area.

4.Battery Backup:

User has to ensure battery backup for the devices to prevent it from the power loss.

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Title	Milestone Number	Activity	Date
Project Objectives	M-1	To know the fundamental concepts and techniques of the fundamental artificial neural network and convolutional neural network.	25-09-2022
Project Flow	M-2	Data collection, data preprocessing and model building are the activities to complete the project.	25-09-2022
Project Structure	M-3	The project structure that must be adhered to in order to construct Conversation Engine is as follows.	26-09-2022

Prerequisites	M-4	Prerequisites are the needs required for the implementation of the Various phases of the project. They help to render the relevant Technical information regarding the project.	27-09-2022
Prior Knowledge	M-5	We need to have prior knowledge in Supervised and unsupervised learning, Artificial Neural Network, Convolutional Neural Network, Regression classification and clustering and flask	27-09-2022
Collection Of Dataset	M-6	We are collecting data for building our project. We will be Creating	28-09-2022

		two folders one for training and the other for testing. Images present in the training folder will be used for building the model and the testing images will be used for validating our model	
Image processing	M-7	Image pre-processing includes import imagedatagenerator library, configure imagedatagenerator class and applying imagedatagenerator functionalities to the train set and test set	01-11-2022
Model Building	M-8	Model building includes importing the necessary	03-11-2022

		libraries, initialising the model, adding CNN and dense layers, configure the learning process and train the model.	
Application Building	M-9	Building a web application that is integrated into the model that built. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI. This section has the following tasks ● Building HTML Pages ● Building server-side	05-11-2022

		script	
Ideation Phase Literature Survey	M-10	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	12-10-2022
Ideation Phase Empathy Map	M-11	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	28-09-2022
Ideation Phase Ideation	M-12	Preparation of Literature Survey on the selected Project and Information Gathering, empathy map and ideation	29-09-2022
Project Design Phase-1 Proposed Solution	M-13	Prepare the proposed solution document, which includes the	15-10-2022

		novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	
Project Design Phase-1 Problem Solution Fit	M-14	Prepared Customer journey, functional requirements, Data flow diagram and Technology Architecture.	15-10-2022
Project Design Phase-1 Solution Architecture	M-15	In this activity we have created solution architecture document and submitted.	15-10-2022
Project Design Phase-2 Customer Journey	M-16	Prepare the user journey mapsto understand the user interactions &experiences withthe application (entry to exit).	15-10-2022
Project Design	M-17	Prepare the	15-10-2022

Phase-2 Functional Requirement		functional requirement document	
Project Design Phase-2 Data Flow Diagram	M-18	Draw the data flow diagrams and submit for review.	15-10-2022
Project Design Phase-2 Technology Architecture	M-19	Prepare the technology architecture diagram.	15-10-2022
Project Planning Phase Milestone & Activity List	M-20	Prepare the milestones & activity list of the project.	09-11-2022
Project Planning Phase Sprint Delivery Plan	M-21	Prepare the sprint delivery plan.	09-11-2022
Project	M-22	Develop & submit the	In progress

Development Phase Sprint-1 Sprint-2 Sprint-3 Sprint-4		developedcode by testing it.	

6.2 SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from different images of cyclone, earthquake, wildfire and flood.	12	Medium	Kaviyanjali.B Keerthana.D Akash.G
Sprint-	Data Pre-	USN-2	As a	8	High	Kaviyanjali

1	processing		user, I can load the dataset, scaling and split data into train and test.			.B Keerthana.D
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of images of cyclone, earthquake, wildfire and flood..	5	High	Kaviyanjali .B Keerthana.D Ranjith S
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	4	Medium	Kaviyanjali .B Ranjith S
Sprint-	Compiling	USN-5	With both the	4	Medi	Ranjith S

2	the model		training data defined and model defined, it's time to configu re the learning process.		um	Akash.G
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	3	High	Keerth ana D Aka sh G
Sprint-2	Save the model	USN-7	As a user, the model is saved &integrat edwith an android applicati on or web applicati on in order to predict somethin g.	4	Low	Kaviyanjali .B Keerthana.D Akash.G Ranjith S
Sprint-	Buil		As a user,		Medi	Keerthana

3	di ng UI Ap plic ati on	USN-8	we access the camera to capture a live video, grab the video frames from the video by looping over the frames and convert the captured frame image from BGR to RGB.	10	um	D Ranjith S
Sprint-4	Train the model-onIBM	USN-10	As a user, I train the model and integrate them on IBM.	9	High	Kaviyanjali .B Keerthana.D Ranjith S
Sprint-4	Cloud Deployment	USN-11	As a user, I can access the web applicati onand make the use of the product from anywhere.	11	High	Kaviyanjali .B Keerthana.D Akash.G Ranjith S

Project Tracker, Velocity & Burn down Chart:

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	5 Days	24 Oct 2022	29 Oct 2022	20	10 Nov 2022
Sprint-2	20	5 Days	31 Oct 2022	05 Nov 2022	20	10 Nov 2022
Sprint-3	20	5 Days	07 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	5 Days	14 Nov 2022	19 Nov 2022	20	12 Nov 2022

Velocity:

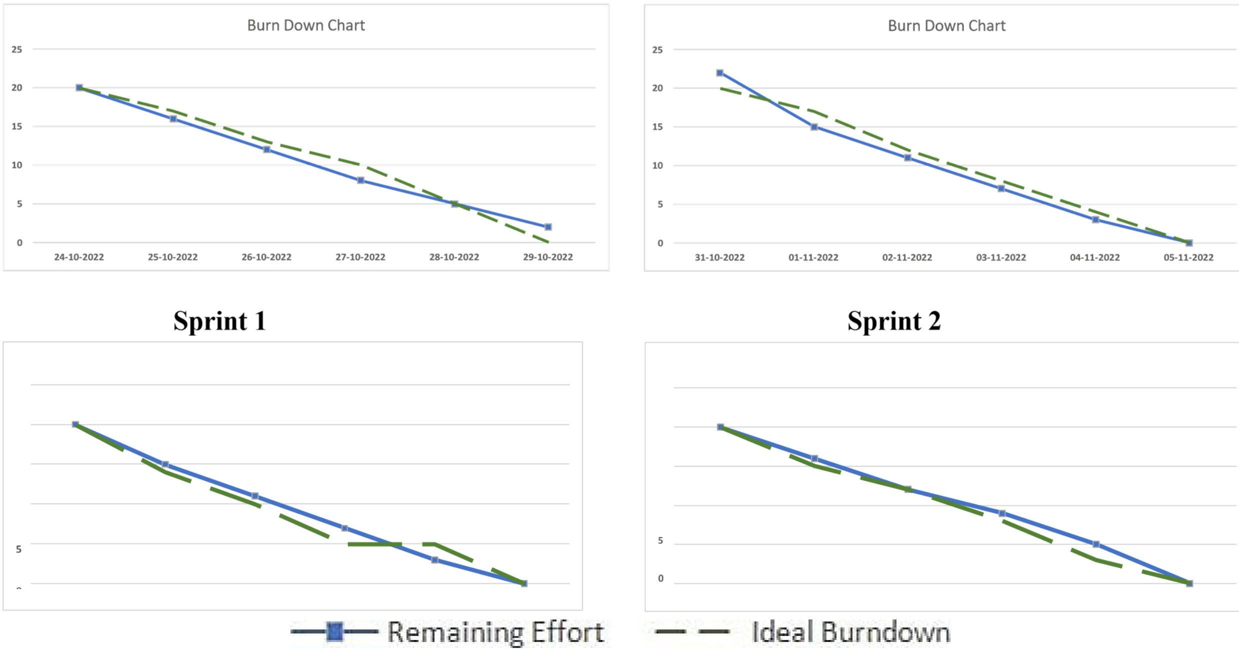
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

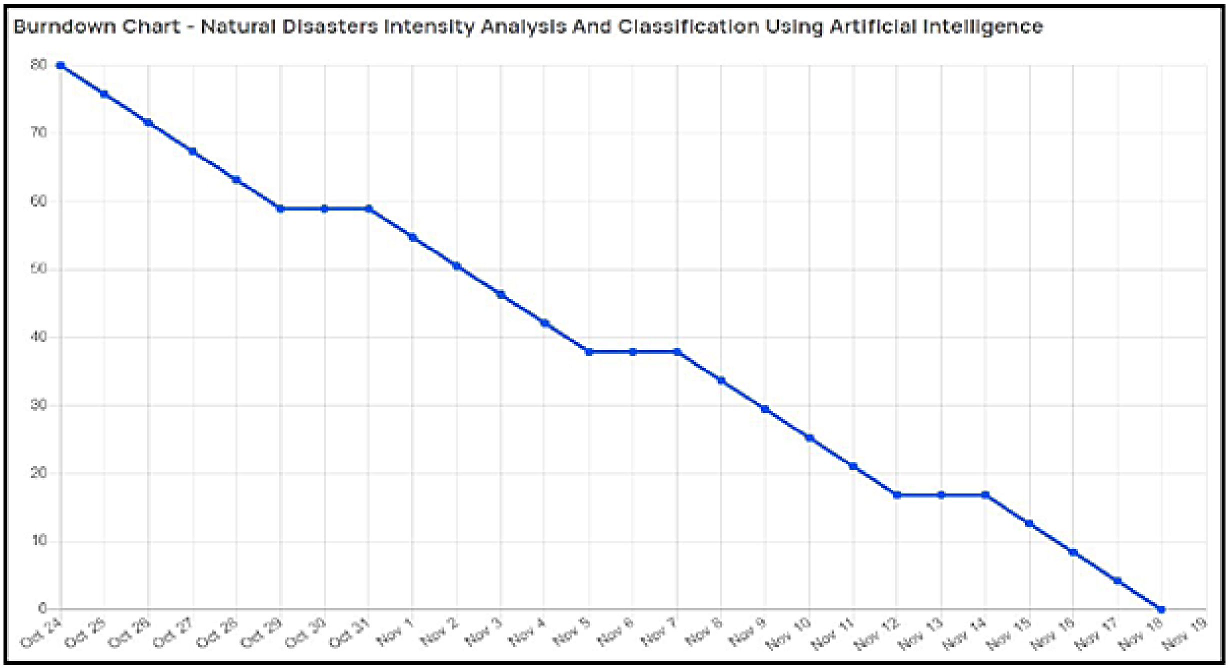
Average Velocity (AV) = 20/4 = 5

Burn down Chart:

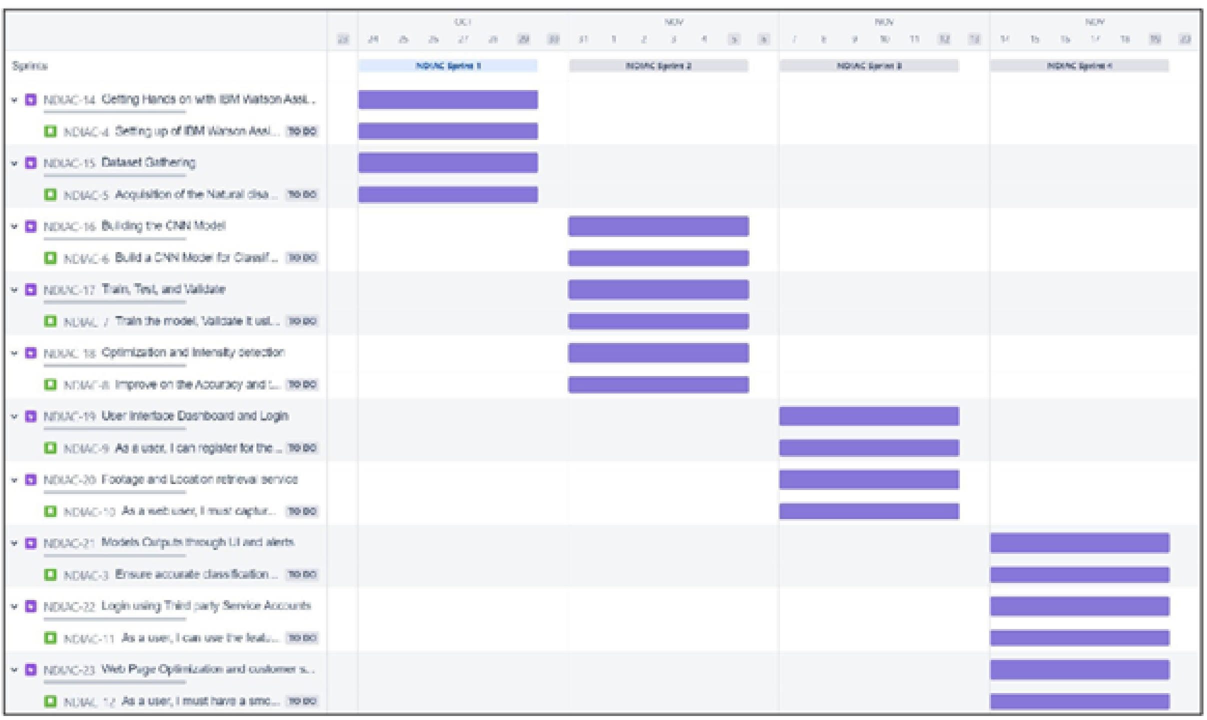
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 REPORTS FROM JIRA



A Roadmap for the Sprint Delivery using Scrum




T
NOV
DEC
Sprints
Sprint 1
Sprint 2
Sprint 3
Sprint 4
NDIA-1 Registration
NDIA-11 I can register for the applicatio... TO DO
NDIA-12 I will receive confirmation emai... TO DO
NDIA-3 login
NDIA-14 , I can log into the application... TO DO
NDIA-25 Designation of Region
NDIA-26 As a user, I can collect the dat... TO DO
NDIA-27 Algorithm selection
NDIA-33 As a user, I can choose the re... TO DO
NDIA-28 Training and Testing
NDIA-34 As a user, I can train and test t... TO DO
NDIA-29 Prediction and analysis of data
NDIA-35 As a user, I can predict and vis... TO DO
NDIA-30 Model building
NDIA-36 As a user, I can build with the... TO DO
NDIA-31 Report generation
NDIA-37 As a user, I can generate detai... TO DO
NDIA-32 Model deployment
NDIA-38 As an administrator, I can mai... TO DO

CHAPTER 7
CODING & SOLUTIONING

7.1 Image Preprocessing:

i. Import the ImageDataGenerator Library

```
from keras.preprocessing.image import ImageDataGenerator
```

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.

ii. Configure ImageDataGenerator Class:

Image Data Agumentation

```
#setting parameter for Image Data agumentation to the traing data
train_datagen = ImageDataGenerator (rescale=1./255, shear_range=0.2,zoom_range=0.2, horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

Python

ImageDataGenerator class is instantiated and the configuration for the types of data augmentation

There are five main types of data augmentation techniques for image data; specifically:

Image shifts via the width_shift_range and height_shift_range arguments

The image flips via the horizontal_flip and vertical_flip arguments.

Image rotations via the rotation_range argument

Image brightness via the brightness_range argument.

iii. Apply ImageDataGenerator Functionality To Trainset And Testset:

Loading our data and performing data augmentation

```python
#performing data agumentation to train data
x_train = train_datagen.flow_from_directory('C:/Users/pesid/Downloads/IBM-Project/project/dataset/train_se
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory('C:/Users/pesid/Downloads/IBM-Project/project/dataset/test_set',
```

Python

```
Found 742 images belonging to 4 classes.
Found 198 images belonging to 4 classes.
```

This function will return batches of images from the subdirectories Cyclone, Earthquake, Flood, Wildfire together with labels 0 to 3{Cyclone: 0, Earthquake: 1, Flood: 2, Wildfire: 3, }

Arguments:

- **directory:** Directory where the data is located. If labels are "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.
- **batch_size:** Size of the batches of data. Default: 32.
- **target_size:** Size to resize images after they are read from disk.
- **class_mode:**

  - **'int'**: means that the labels are encoded as integers (e.g. for sparse_categorical_crossentropy loss).

  - **'categorical'** means that the labels are encoded as a categorical vector (e.g. for categorical_crossentropy loss).

  - **'binary'** means that the labels (there can be only 2) are encoded as float32 scalars with values 0 or 1 (e.g. for binary_crossentropy).

  - **None** (no labels).

7.2 Model Building:

i. Importing Necessary Libraries:

Importing Neccessary Libraries

```python
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of Layers:
from tensorflow.keras import layers #A Layer consists of a tensor-in tensor-out computation function
#Dense Layer is the regular deeply connected neural network Layer
from tensorflow.keras.layers import Dense, Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D, MaxPooling2D #Convolutional Layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

Python

ii. Initializing the Model:

Keras has 2 ways to define a neural network:

1. Sequential
2. Function API

The Sequential class is used to define a linear initializations of network layers which then, collectively, constitute a model. In our examplenbelow, we will use the Sequential constructor to create a model, which will then have layers added to it using the add() method.

```python
model=Sequential()
```

iii. Adding CNN Layers:

```python
# First convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution I
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening the Layers
classifier.add(Flatten())
```

As the input image contains three channels, we are specifying the input shape as (64,64,3).

We are adding a convolution layer with activation function as “relu” and with a small filter size (3,3) and the number of filters (32) followed by a max-pooling layer.

Max pool layer is used to down sample the input.(Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter)

Flatten layer flattens the input. Does not affect the batch size.

iv. Adding Dense Layers:

A dense layer is a deeply connected neural network layer. It is the most common and frequently used layer.

```
# Adding a fully connected Layer
classifier.add(Dense (units=128, activation='relu'))
classifier.add(Dense (units=4, activation='softmax')) # softmax for more than 2
```

The number of neurons in the Dense layer is same as the number of classes in the training set. The neurons in the last Dense layer, use softmax activation to convert their outputs into respective probabilities.

Understanding the model is a crucial phase to use it for training properly and prediction purposes. Keras provides a simple method, summary to get the full information about the model and its layers.

```
classifier.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 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
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
_________________________________________________________________
```

v. Configure The Learning Process:

The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find errors or deviations in the learning process. Keras requires loss function during the model compilation process.

Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer.

Metrics are used to evaluate the performance of your model. It is similar to the loss function, but not used in the training process.

Compiling the model

```
# Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

vi. Train The Model:

The model is trained for 20 epochs and after every epoch, the current model state is saved if the model has the least loss encountered till that time. We can see that the training loss decreases in almost every epoch till 20 epochs and probably there is further scope to improve the mode.

Fit generator functions used to train a deep learning neural network

Arguments:

Steps per epoch: it specifies the total number of steps taken from the generator as soon as one epoch is finished and the next epoch has started. We can calculate the value of steps per epoch as the total number of samples in your dataset divided by the batch size.

Epochs: an integer and number of epochs we want to train our model for.

Validation data can be either:

- an inputs and targets list
- a generator
- an inputs, targets, and sample weights list which can be used to evaluate the loss and metrics for any model after any epoch has ended.

Validation steps: only if the validation data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

Fitting the model

```python
classifier.fit_generator( generator=x_train, steps_per_epoch = len(x_train), epochs=20, validation_data=x_
```

Python

```
Please use Model.fit, which supports generators.
Epoch 1/20
149/149 [==============================] - 23s 153ms/step - loss: 1.1720 - accuracy: 0.4933 - val_loss: 0.8377 - val_accuracy: 0.6667
Epoch 2/20
149/149 [==============================] - 21s 139ms/step - loss: 0.8336 - accuracy: 0.6550 - val_loss: 1.1909 - val_accuracy: 0.4697
Epoch 3/20
149/149 [==============================] - 21s 143ms/step - loss: 0.7105 - accuracy: 0.7399 - val_loss: 0.8390 - val_accuracy: 0.6717
Epoch 4/20
149/149 [==============================] - 21s 141ms/step - loss: 0.5757 - accuracy: 0.7736 - val_loss: 0.9805 - val_accuracy: 0.6263
Epoch 5/20
149/149 [==============================] - 22s 144ms/step - loss: 0.5806 - accuracy: 0.7817 - val_loss: 0.7162 - val_accuracy: 0.6768
Epoch 6/20
149/149 [==============================] - 26s 175ms/step - loss: 0.5214 - accuracy: 0.8032 - val_loss: 0.5987 - val_accuracy: 0.8081
Epoch 7/20
149/149 [==============================] - 21s 140ms/step - loss: 0.4666 - accuracy: 0.8450 - val_loss: 0.5968 - val_accuracy: 0.8283
Epoch 8/20
149/149 [==============================] - 21s 140ms/step - loss: 0.4618 - accuracy: 0.8235 - val_loss: 0.9052 - val_accuracy: 0.7323
Epoch 9/20
149/149 [==============================] - 21s 141ms/step - loss: 0.4026 - accuracy: 0.8450 - val_loss: 0.6366 - val_accuracy: 0.8030
Epoch 10/20
149/149 [==============================] - 21s 139ms/step - loss: 0.3561 - accuracy: 0.8679 - val_loss: 0.8216 - val_accuracy: 0.7727
Epoch 11/20
149/149 [==============================] - 21s 142ms/step - loss: 0.4345 - accuracy: 0.8410 - val_loss: 0.6938 - val_accuracy: 0.7879
...
Epoch 19/20
149/149 [==============================] - 21s 142ms/step - loss: 0.2128 - accuracy: 0.9205 - val_loss: 0.7216 - val_accuracy: 0.8333
Epoch 20/20
149/149 [==============================] - 21s 142ms/step - loss: 0.1734 - accuracy: 0.9434 - val_loss: 0.8815 - val_accuracy: 0.7980
```

CHAPTER 8
TESTING

8.1 Test Cases

- ◎ Verify user can click the button on the results tab
- ◎ Verify user can see that the camera is accessible and open when the button is clicked.
- ◎ Verify user can capture the image from live stream
- ◎ Verify user can see the predicted results in the window

8.2 User Acceptance Testing:

i. Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	0	0	1
Duplicate	1	3	3	1	8
External	2	3	0	0	5
Fixed	2	4	4	2	12
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	6	10	7	4	27

ii. Test Case Analysis

Section	TotalCases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	3	0	0	3
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	1	0	0	1
Final ReportOutput	4	0	0	4
Version Control	2	0	0	2

CHAPTER 9
PERFORMANCE METRICS

9.1 Performance Testing:

i. Model Summary

Total Params: 813,604

Trainable Params:813,604

Non-Trainable Params:0

```
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 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 128)               802944
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 516
=================================================================
```

ii. Accuracy

Training Accuracy: 94.3%

Validation Accuracy: 83.33%

```
Please use Model.fit, which supports generators.
Epoch 1/20
149/149 [==============================] - 23s 153ms/step - loss: 1.1720 - accuracy: 0.4933 - val_loss: 0.8377 - val_accuracy: 0.6667
Epoch 2/20
149/149 [==============================] - 21s 139ms/step - loss: 0.8336 - accuracy: 0.6550 - val_loss: 1.1909 - val_accuracy: 0.4697
Epoch 3/20
149/149 [==============================] - 21s 143ms/step - loss: 0.7105 - accuracy: 0.7399 - val_loss: 0.8390 - val_accuracy: 0.6717
Epoch 4/20
149/149 [==============================] - 21s 141ms/step - loss: 0.5757 - accuracy: 0.7736 - val_loss: 0.9805 - val_accuracy: 0.6263
Epoch 5/20
149/149 [==============================] - 22s 144ms/step - loss: 0.5806 - accuracy: 0.7817 - val_loss: 0.7162 - val_accuracy: 0.6768
Epoch 6/20
149/149 [==============================] - 26s 175ms/step - loss: 0.5214 - accuracy: 0.8032 - val_loss: 0.5987 - val_accuracy: 0.8081
Epoch 7/20
149/149 [==============================] - 21s 140ms/step - loss: 0.4666 - accuracy: 0.8450 - val_loss: 0.5968 - val_accuracy: 0.8283
Epoch 8/20
149/149 [==============================] - 21s 140ms/step - loss: 0.4618 - accuracy: 0.8235 - val_loss: 0.9052 - val_accuracy: 0.7323
Epoch 9/20
149/149 [==============================] - 21s 141ms/step - loss: 0.4026 - accuracy: 0.8450 - val_loss: 0.6366 - val_accuracy: 0.8030
Epoch 10/20
149/149 [==============================] - 21s 139ms/step - loss: 0.3561 - accuracy: 0.8679 - val_loss: 0.8216 - val_accuracy: 0.7727
Epoch 11/20
149/149 [==============================] - 21s 142ms/step - loss: 0.4345 - accuracy: 0.8410 - val_loss: 0.6938 - val_accuracy: 0.7879
...
Epoch 19/20
149/149 [==============================] - 21s 142ms/step - loss: 0.2128 - accuracy: 0.9205 - val_loss: 0.7216 - val_accuracy: 0.8333
Epoch 20/20
149/149 [==============================] - 21s 142ms/step - loss: 0.1734 - accuracy: 0.9434 - val_loss: 0.8815 - val_accuracy: 0.7980
```

Locust Test Report

During: 11/17/2022, 4:34:34 PM - 11/17/2022, 4:39:21 PM

Target Host: http://192.168.0.113:7000

Script: locustfile.py

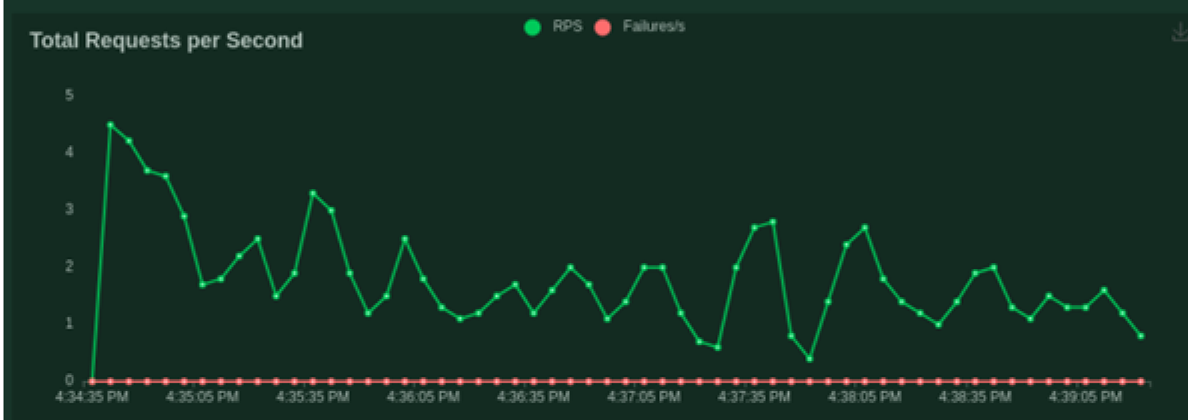
Request Statistics

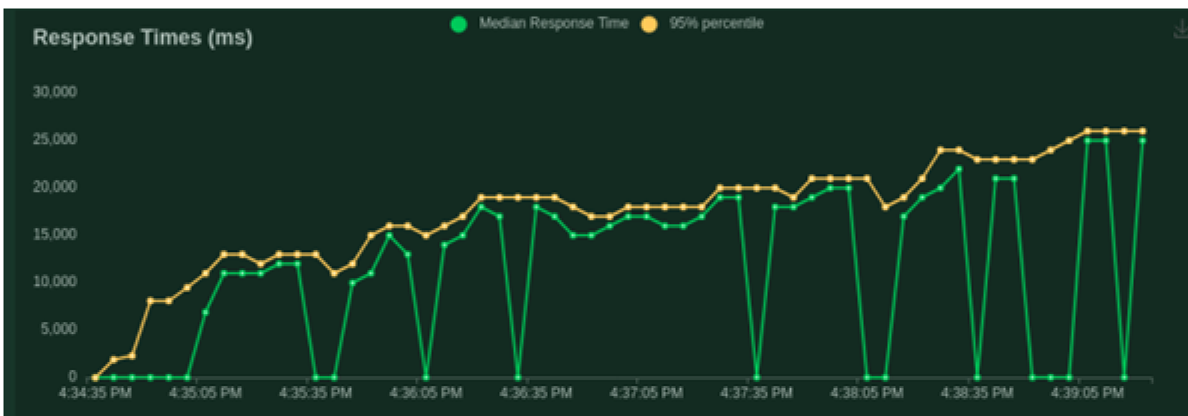
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	20	0	20	14	29	21141	0.1	0.0
GET	/home	236	0	12	8	25	21141	0.8	0.0
GET	/upload	244	0	16482	1416	26317	21141	0.8	0.0
	Aggregated	**500**	**0**	**8050**	**8**	**26317**	**21141**	**1.7**	**0.0**

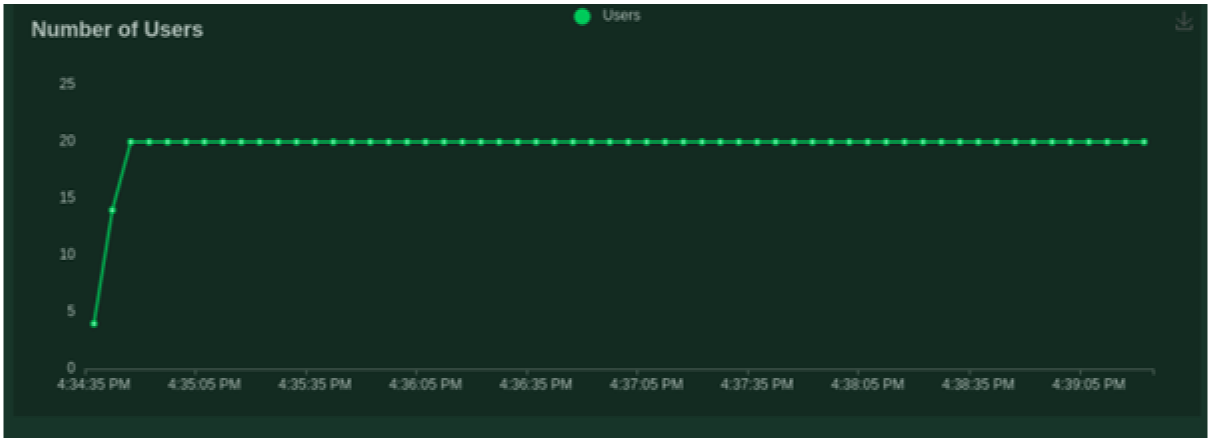
Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	20	20	23	25	26	29	29	29
GET	/home	12	12	13	14	16	19	24	26
GET	/upload	17000	18000	19000	21000	23000	25000	26000	26000
	Aggregated	**25**	**11000**	**16000**	**18000**	**21000**	**23000**	**26000**	**26000**

Charts







Final ratio

Ratio per User class

- 100.0% QuickstartUser
  - 50.0% home
  - 50.0% upload

Total ratio

- 100.0% QuickstartUser
  - 50.0% home
  - 50.0% upload

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- This model can be easily used to classify the disaster and it can also analyse the intensity of the classified disaster.
- We used CNNs to perform feature extraction by denoising the images and removing interference and achieve highly accurate results.
- We used Deep Learning algorithms to classify the natural disaster automatically.

DISADVANTAGES:

- In the built model, it can only take the live capture as an input to classify the disaster.
- We cannot give any image or video as an input to classify or to analyse the intensity.

CHAPTER 11
CONCLUSION

Natural disasters cannot be prevented — but they can be detected.

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 window.

CHAPTER 12
FUTURE SCOPE

As of now we have just built the web application which apparently takes the live capture as an input and then it classifies them and analyse the intensity of that disaster. This can also be done in Mobile applications like android, ios in near future.

We will try to make this model to accept many forms of inputs such as image and video to classify what type of disaster is.

CHAPTER 13
APPENDIX

Source Code:

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SB1 PROJECT</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.mi
n.css">
  <link href="https://fonts.googleapis.com/css?family=Montserrat"
rel="stylesheet" type="text/css">
  <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet"
type="text/css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></s
cript>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.j
s"></script>

  <style>
  body {
    font: 400 15px Lato, sans-serif;
    line-height: 1.8;
    color: #818181;
```

```
}
h2 {
  font-size: 24px;
  text-transform: uppercase;
  color: #303030;
  font-weight: 600;
  margin-bottom: 30px;
}
h4 {
  font-size: 19px;
  line-height: 1.375em;
  color: #303030;
  font-weight: 400;
  margin-bottom: 30px;
}
.jumbotron {
  background-color:black;
  color: #fff;

  font-family: Montserrat, sans-serif;
}
.container-fluid {
  padding: 60px 50px;
}
.bg-grey {
  background-color: #f6f6f6;
}
.logo-small {
  color: #f4511e;
  font-size: 50px;
}
.logo {
```

```css
  color: #f4511e;
  font-size: 200px;
}
.thumbnail {
  padding: 0 0 15px 0;
  border: none;
  border-radius: 0;
}
.thumbnail img {
  width: 100%;
  height: 100%;
  margin-bottom: 10px;
}
.carousel-control.right, .carousel-control.left {
  background-image: none;
  color: #f4511e;
}
.carousel-indicators li {
  border-color: #f4511e;
}
.carousel-indicators li.active {
  background-color: #f4511e;
}
.item h4 {
  font-size: 19px;
  line-height: 1.375em;
  font-weight: 400;
  font-style: italic;
  margin: 70px 0;
}
.item span {
  font-style: normal;
```

```
}
.panel {
  border: 1px solid #f4511e;
  border-radius:0 !important;
  transition: box-shadow 0.5s;
}
.panel:hover {
  box-shadow: 5px 0px 40px rgba(0,0,0, .2);
}
.panel-footer .btn:hover {
  border: 1px solid #f4511e;
  background-color: #fff !important;
  color: #f4511e;
}
.panel-heading {
  color: #fff !important;
  background-color: #f4511e !important;
  padding: 25px;
  border-bottom: 1px solid transparent;
  border-top-left-radius: 0px;
  border-top-right-radius: 0px;
  border-bottom-left-radius: 0px;
  border-bottom-right-radius: 0px;
}
.panel-footer {
  background-color: white !important;
}
.panel-footer h3 {
  font-size: 32px;
}
.panel-footer h4 {
  color: #aaa;
```

```css
  font-size: 14px;
}
.panel-footer .btn {
  margin: 15px 0;
  background-color: #f4511e;
  color: #fff;
}
.navbar {
  margin-bottom: 0;
  background-color:black;
  z-index: 9999;
  border: 0;
  font-size: 12px !important;
  line-height: 1.42857143 !important;
  letter-spacing: 4px;
  border-radius: 0;
  font-family: Montserrat, sans-serif;
}
.navbar li a, .navbar .navbar-brand {
  color: #fff !important;
}
.navbar-nav li a:hover, .navbar-nav li.active a {
  color: #f4511e !important;
  background-color: #fff !important;
}
.navbar-default .navbar-toggle {
  border-color: transparent;
  color: #fff !important;
}
footer .glyphicon {
  font-size: 20px;
  margin-bottom: 20px;
```

```
  color:linear-gradient(to right, red , yellow);
}
.slideanim {visibility:hidden;}
.slide {
  animation-name: slide;
  -webkit-animation-name: slide;
  animation-duration: 1s;
  -webkit-animation-duration: 1s;
  visibility: visible;
}
@keyframes slide {
  0% {
    opacity: 0;
    transform: translateY(70%);
  }
  100% {
    opacity: 1;
    transform: translateY(0%);
  }
}
@-webkit-keyframes slide {
  0% {
    opacity: 0;
    -webkit-transform: translateY(70%);
  }
  100% {
    opacity: 1;
    -webkit-transform: translateY(0%);
  }
}
@media screen and (max-width: 768px) {
  .col-sm-4 {
```

```
    text-align: center;
    margin: 25px 0;
  }
  .btn-lg {
    width: 100%;
    margin-bottom: 35px;
  }
}
@media screen and (max-width: 480px) {
  .logo {
    font-size: 150px;
  }
}

.container {
      padding: 16px;
      max-width: max-content;
  }

  .container {
      max-width: 1376px;
      margin: auto;
      padding: 2rem 1.5rem;
  }

  .cards {
      display: flex;
      flex-wrap: wrap;
      align-items: center;
      justify-content: center;
     }
     .card {
```

```
  cursor: pointer;
  background-color: transparent;
  height: 300px;
  perspective: 1000px;
  margin: 1rem;
  align-items: center;
  justify-content: center;
}
.card h3 {
  border-bottom: 1px #fff solid;
  padding-bottom: 10px;
  margin-bottom: 10px;
  text-align: center;
  font-size: 1.6rem;
  word-spacing: 3px;
}
.card p{
  opacity: 0.75;
  font-size: 0.8rem;
  line-height: 1.4;
}
.card img {
  width: 360px;
  height: 300px;
  object-fit: cover;
  border-radius: 3px;
}
.card-inner {
  position: relative;
  width: 360px;
  height: 100%;
  transition: transform 0.9s;
```

```css
  transform-style: preserve-3d;
}
.card:hover .card-inner {
  transform: rotateY(180deg);
}
.card-front,
.card-back {
  position: absolute;
  width: 360px;
  height: 100%;
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
}
.card-back {
  background-color: #222;
  color: #fff;
  padding: 1.5rem;
  transform: rotateY(180deg);
}
.text-block {
  position: absolute;
  bottom: 20px;
  right: 20px;
  background-color: black;
  color: white;
  padding-left: 20px;
  padding-right: 20px;
}
```

```
.features-section img {
    display: none;
  }
  .testimonials-section {
    background: var(--primary-colour);
    color: white;
  }
  .testimonials-section li {
    background: #0059ff;
    text-align: center;
    width: 80%;
    border-radius: 1em;
  }
  .testimonials-section li img {
    width: 6em;
    height: 6em;
    border: 3px solid #ffffff;
    border-radius: 50%;
    margin-top: -2.5em;
  }

  ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
  }

ul.features-list {
    margin: 0;
    padding-left: .1em;
  }
  ul.features-list li {
```

```
  font-size: 1.1em;
  margin-bottom: 1em;
  margin-left: 2em;
  position: relative;
}
ul.features-list li:before {
  content: '';
  left: -2em;
  position: absolute;
  width: 20px;
  height: 20px;
  background-image: url("#");
  background-size: contain;
  margin-right: .5em;
}
.features-section img {
  display: none;
}



</style>
```

```
</head>
<body id="myPage" data-spy="scroll" data-target=".navbar" data-
offset="60">

<nav class="navbar navbar-default navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#myNavbar">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#myPage">NATURAL DISASTERS</a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav navbar-right">
        <li><a href="#about">ABOUT</a></li>
        <li><a href="#services">SERVICES</a></li>
       <li><a href="#Results">RESULTS</a></li>
      </ul>
    </div>
  </div>
</nav>

<div class="jumbotron text-center" >
  <br><br><br>
  <h1>AI Based Natural Disaster Analysis</h1>
</div>

<!-- Container (About Section) -->
<div id="about" class="container-fluid" style="align-items: center; align-
```

content: center;">
<div class="row">
<div class="text-center" style="align-items: center; align-content: center; padding-left: 25px;padding-right: 25px;">
<center><h1 style="color: black; font-weight: bold;">Natural Disaster</h1></center>

<h3 style="color: black;">A natural disaster is a major adverse event resulting from natural processes of the Earth; examples include firestorms, duststorms, floods, hurricanes, tornadoes, volcanic eruptions, earthquakes, tsunamis, storms, and other geologic processes. A natural disaster can cause loss of life or damage property and typically leaves some economic damage in its wake, the severity of which depends on the affected population's resilience and on the infrastructure available.</h3>
<h3 style="color: black;">The number of deaths from natural disasters can be highly variable from year-to-year; some years pass with very few deaths before a large disaster event claims many lives.

If we look at the average over the past decade, approximately 60,000 people globally died from natural disasters each year. This represents 0.1% of global deaths.

In the visualizations shown here we see the annual variability in the number and share of deaths from natural disasters in recent decades.

What we see is that in many years, the number of deaths can be very low – often less than 10,000, and accounting for as low as 0.01% of total deaths. But we also see the devastating impact of shock events: the 1983-85 famine and drought in Ethiopia; the 2004 Indian Ocean earthquake and tsunami; Cyclone Nargis which struck Myanmar in 2008; and the 2010 Port-au-Prince earthquake in Haiti. All of these events pushed global disasters deaths over 200,000 – more than 0.4% of deaths in these years.</h3>

```
    </div>
  </div>
</div>

<div class="container-fluid">
    <div class="container">

        <div class="cards">

            <div class="card">
                <div class="card-inner">
                    <div class="card-front">
                        <img src="https://images.unsplash.com/photo-
1454789476662-
53eb23ba5907?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8f
Hw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=689&q=80"
                            alt="">
                        <div class="text-block">
                            <h1>Cyclone</h1>
                            <h3>violent winds, torrential rain, high waves and, very
destructive storm</h3>
                        </div>

                    </div>
                    <div class="card-back">
                        <h3>Cyclone</h3>
                        <h3>The effects of tropical cyclones include heavy rain,
strong wind, large storm surges near
                            landfall, and tornadoes. The destruction from a tropical
cyclone, such as a hurricane or
                            tropical storm, depends mainly on its intensity, its size, and
its location.</h3>
```

```
                </div>
            </div>
        </div>



        <div class="container">

            <div class="cards">

                <div class="card">
                    <div class="card-inner">
                        <div class="card-front">
                            <img src="https://images.unsplash.com/photo-
1603869311144-
66b03d340b32?ixid=MXwxMjA3fDB8MHxzZWFyY2h8M3x8ZWFydGhxdWF
rZXxlbnwwfHwwfA%3D%3D&ixlib=rb-
1.2.1&auto=format&fit=crop&w=500&q=60"
                                alt="">
                            <div class="text-block">
                                <h1>Earth Quake</h1>
                                <h3>Sudden release of stored energy in the Earth's
crust that creates seismic waves.
                                </h3>
                            </div>

                        </div>
                        <div class="card-back">
                            <h3>Earth Quake</h3>
                            <h3>Earthquakes are usually caused when rock
underground suddenly breaks along a fault.
                                This sudden release of energy causes the seismic
waves that make the ground shake.
```

```
                  ... During the earthquake and afterward, the plates or
blocks of rock start moving,
                  and they continue to move until they get stuck
again.</h3>
              </div>
            </div>
          </div>



          <div class="container">

            <div class="cards">

              <div class="card">
                <div class="card-inner">
                  <div class="card-front">
                    <img src="https://images.unsplash.com/photo-
1547683905-
f686c993aae5?ixid=MXwxMjA3fDB8MHxzZWFyY2h8MXx8Zmxvb2R8ZW5
8MHx8MHw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"
                      alt="">
                    <div class="text-block">
                      <h1>Flood</h1>
                      <h3>A flood is an overflow of water on normally
dry ground</h3>
                    </div>

                  </div>
                  <div class="card-back">
                    <h3>Flood</h3>
                    <h3>During heavy rain, the storm drains can become
```

```
overwhelmed or plugged by
                              debris and flood the roads and buildings nearby.
Low spots, such as
                              underpasses, underground parking garages,
basements, and low water crossings
                              can become death traps. Areas near rivers are at
risk from floods.</h3>
                        </div>
                    </div>
                </div>



                <div class="container">

                    <div class="cards">

                        <div class="card">
                            <div class="card-inner">
                                <div class="card-front">
                                    <img src="https://images.unsplash.com/photo-
1473260079709-
83c808703435?ixid=MXwxMjA3fDB8MHxzZWFyY2h8NHx8d2lsZGZpcmV8
ZW58MHx8MHw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"
                                        alt="">
                                    <div class="text-block">
                                        <h1>WildFire</h1>
                                        <h3>Uncontrolled fire in a forest, grassland,
brushland</h3>
                                    </div>

                                </div>
```

```
                                <div class="card-back">
                                  <h3>Wildfire</h3>
                                  <h3>Wildfires can be caused by an
accumulation of dead matter (leaves,
                                    twigs, and trees) that can create enough heat
in some instances to
                                    spontaneously combust and ignite the
surrounding area. Lightning
                                    strikes the earth over 100,000 times a day. 10
to 20% of these
                                    lightning strikes can cause fire.</h3>
                                </div>
                              </div>
                            </div>



</div>






<!-- Container (Services Section) -->
<div id="services" class="container-fluid text-center">
  <h2>SERVICES</h2>
  <h4>What we offer</h4>
  <br>
  <div style="align-items: center;"
    <div class="row slideanim">
        <span class="glyphicon glyphicon-camera logo-small"></span>
        <h4>SURVEILLANCE-CAM</h4>
```

```
        <p>If you have any camera near the natural disaster areas, then it will
predict the disaster automatically.</p>
    </div>
  </div>



  <h2>Stats of Natural Disasters</h2>
  <div id="myCarousel" class="carousel slide text-center" data-
ride="carousel">
    <!-- Indicators -->
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
    </ol>

    <!-- Wrapper for slides -->
    <div class="carousel-inner" role="listbox">
      <div class="item active">
        <h4>"Natural disasters kill on average 60,000 people per year and are
responsible for 0.1% of global deaths."<br><span>What we see is that in
many years, the number of deaths can be very low – often less than 10,000,
and accounting for as low as 0.01% of total deaths.</span></h4>
      </div>
      <div class="item">
        <h4>"On average, as many as 27 million people become displaced
from their homes every decade because of natural
disasters."<br><span>Floods and storms are the largest culprits, as they
account for over eighty percent of all weather induced
displacements.</span></h4>
      </div>
      <div class="item">
```

```
        <h4>"The report is based on EM-DAT data from the period between 1994 and 2013, which includes 6,873 natural disasters worldwide."<br><span>It claimed 1.35 million lives or almost 68,000 lives on average each year, and affected 218 million people on average per annum during this 20-year period</span></h4>
      </div>
    </div>

    <!-- Left and right controls -->
    <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
      <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
      <span class="sr-only">Previous</span>
    </a>
    <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
      <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
      <span class="sr-only">Next</span>
    </a>
  </div>
</div>

<!-- Container (Pricing Section) -->
<div id="Results" class="container-fluid">
  <div class="text-center">
    <h2>Results</h2>
    <h4>Choose the desired way that works for you</h4>
  </div>
  <div class="row slideanim">
    <!--
```

```
    <div class="col-sm-4 col-xs-12">
      <div class="panel panel-default text-center">
        <div class="panel-heading">
          <h1>Photo input</h1>
        </div>
        <div class="panel-body">
          <strong><p>Upload   the</p>
          <p><strong>image</strong>   of</p>
          <p><strong>natural</strong>   disaster</p></strong>
        </div>
        <div class="panel-footer">
          <h3><span class="glyphicon glyphicon-picture logo-
small"></span></h3>
          <a href='/image'>
          <button class="btn btn-lg">Open</button></a>
        </div>
      </div>
    </div>       -->
    <div class="col-sm-4 col-xs-12" style="width: 500px;">
      <div class="panel panel-default text-center">
        <div class="panel-heading">
          <h1>Live Stream</h1>
        </div>
        <div class="panel-body">
          <strong><p>Live   Stream</p></strong>
                    <br>
                    <br>
                    <br>
        </div>
        <div class="panel-footer">
          <h3><span class="glyphicon glyphicon-camera logo-
small"></span></h3>
```

```
        <a href='/upload'>
        <button class="btn btn-lg">Open</button></a>
      </div>
    </div>
  </div>
 </div>
</div>
```

```
<script>
$(document).ready(function(){
 // Add smooth scrolling to all links in navbar + footer link
 $(".navbar a, footer a[href='#myPage']").on('click', function(event) {
  // Make sure this.hash has a value before overriding default behavior
  if (this.hash !== "") {
   // Prevent default anchor click behavior
   event.preventDefault();

   // Store hash
   var hash = this.hash;

   // Using jQuery's animate() method to add smooth page scroll
   // The optional number (900) specifies the number of milliseconds it
takes to scroll to the specified area
   $('html, body').animate({
    scrollTop: $(hash).offset().top
   }, 900, function(){
    // Add hash (#) to URL when done scrolling (default click behavior)
    window.location.hash = hash;
   });
```

```
    } // End if
  });
  $(window).scroll(function() {
    $(".slideanim").each(function(){
      var pos = $(this).offset().top;

      var winTop = $(window).scrollTop();
        if (pos < winTop + 600) {
          $(this).addClass("slide");
        }
    });
  });
})
</script>

</body>
</html>
```

app.py

```
# import the necessary packages
from flask import Flask, render_template, request, redirect, url_for
# Flask-It is our framework which we are going to use to run/serve our application.
# request-for accessing file which was uploaded by the user on our application.
#import operator
import cv2  # opencv library
from tensorflow.keras.models import load_model  # to load our trained model
import numpy as np
#import os
from werkzeug.utils import secure_filename
```

```
#from playsound import playsound
#from gtts import gTTS

app = Flask(__name__, template_folder="templates")  # initializing a flask
app
# Loading the model
model = load_model("disaster.h5")
print("Loaded model from disk")


# app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
   return render_template('home.html')


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


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


@app.route('/upload', methods=['GET', 'POST'])
def predict():

   # Get a reference to webcam #0 (the default one)

   print("[INFO] starting video stream...")
```

```
vs = cv2.VideoCapture(0)
#writer = None
(W, H) = (None, None)

# loop over frames from the video file stream
while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break

        # if the frame dimensions are empty, grab them
    if W is None or H is None:
        (H, W) = frame.shape[:2]

        # clone the output frame, then convert it from BGR to RGB
        # ordering and resize the frame to a fixed 64x64
    output = frame.copy()
    # print("apple")
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame = cv2.resize(frame, (64, 64))
    #frame = frame.astype("float32")
    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)
    # result=result.tolist()
```

```
        cv2.putText(output, "activity: {}".format(result), (10, 120),
cv2.FONT_HERSHEY_PLAIN,
                    1, (0, 255, 255), 1)
        #playaudio("Emergency it is a disaster")
        cv2.imshow("Output", output)
        key = cv2.waitKey(1) & 0xFF

        # if the `q` key was pressed, break from the loop
        if key == ord("q"):
            break

    # release the file pointers
    print("[INFO] cleaning up...")
    vs.release()
    cv2.destroyAllWindows()
    return render_template("upload.html")

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

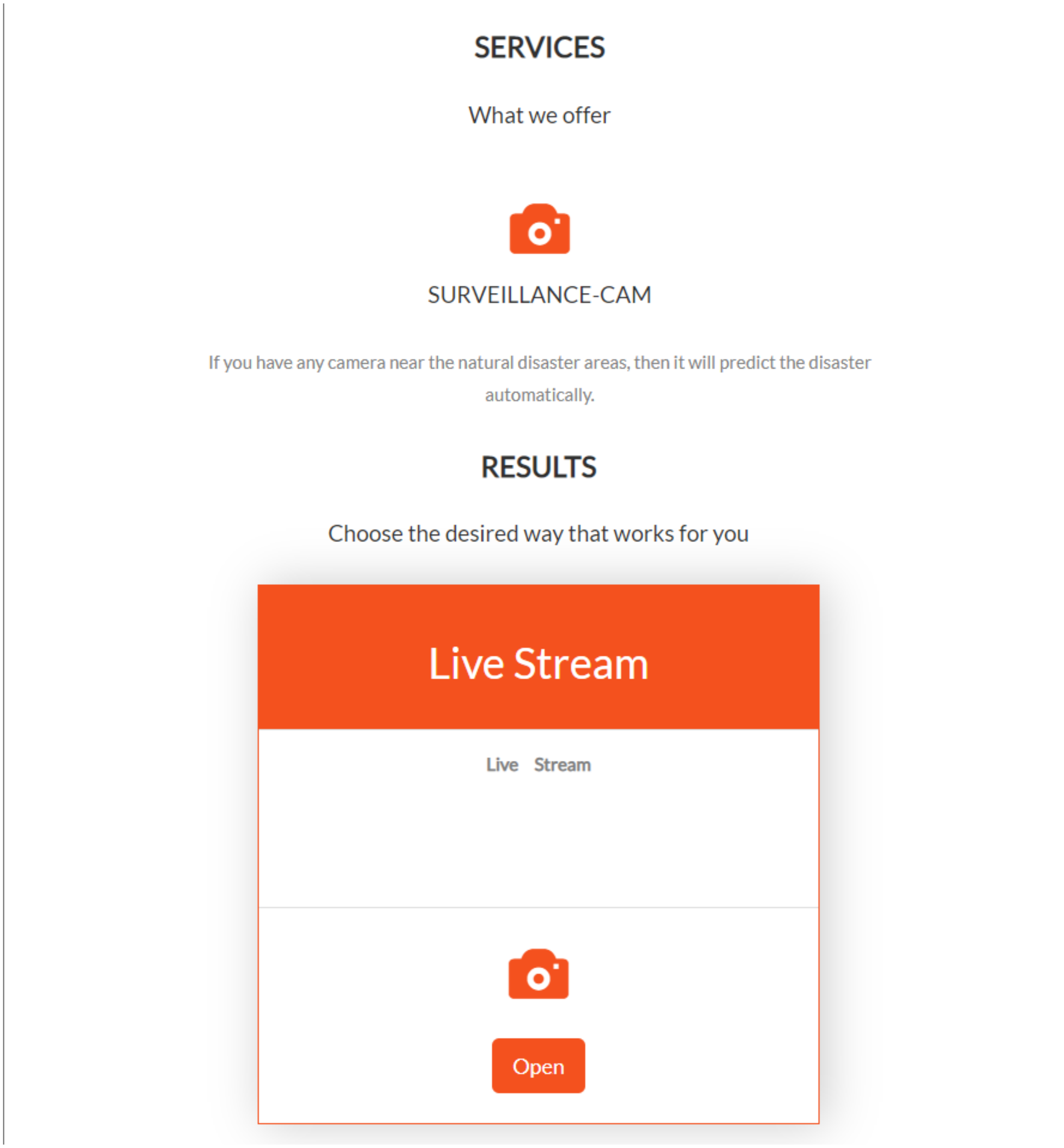
OUTPUT:

AI Based Natural Disaster Analysis

Natural Disaster

A natural disaster is a major adverse event resulting from natural processes of the Earth; examples include firestorms, duststorms, floods, hurricanes, tornadoes, volcanic eruptions, earthquakes, tsunamis, storms, and other geologic processes. A natural disaster can cause loss of life or damage property and typically leaves some economic damage in its wake, the severity of which depends on the affected population's resilience and on the infrastructure available

The number of deaths from natural disasters can be highly variable from year-to-year; some years pass with very few deaths before a large disaster event claims many lives. If we look at the average over the past decade, approximately 60,000 people globally died from natural disasters each year. This represents 0.1% of global deaths. In the visualizations shown here we see the annual variability in the number and share of deaths from natural disasters in recent decades. What we see is that in many years, the number of deaths can be very low – often less than 10,000, and accounting for as low as 0.01% of total deaths. But we also see the devastating impact of shock events: the 1983-85 famine and drought in Ethiopia; the 2004 Indian Ocean earthquake and tsunami; Cyclone Nargis which struck Myanmar in 2008; and the 2010 Port-au-Prince earthquake in Haiti. All of these events pushed global disasters deaths over 200,000 – more than 0.4% of deaths in these years.



GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-32257-1660208909

DEMO LINK: NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE