# NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Project Report

*Submitted by*

Team ID:PNT2022TMID25119

|  |  |
| --- | --- |
| RANJITH KUMAR M | 210519205040 |
| RAGUL UM | 210519205039 |
| SURIYA A.J | 210519205051 |
| VISHWA R | 210519205059 |

# INTRODUCTION

* 1. Project Overview
  2. Purpose

# LITERATURE SURVEY

* 1. Existing problem
  2. References
  3. Problem Statement Definition

# IDEATION & PROPOSED SOLUTION

* 1. Empathy Map Canvas
  2. Ideation & Brainstorming
  3. Proposed Solution
  4. Problem Solution fit

# REQUIREMENT ANALYSIS

* 1. Functional requirement
  2. Non-Functional requirements

# PROJECT DESIGN

* 1. Data Flow Diagrams
  2. Solution & Technical Architecture
  3. User Stories

# PROJECT PLANNING & SCHEDULING

* 1. Sprint Planning & Estimation
  2. Sprint Delivery Schedule
  3. Reports from JIRA

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

* 1. Feature 1
  2. Feature 2
  3. Database Schema (if Applicable)

# TESTING

* 1. Test Cases
  2. User Acceptance Testing

# RESULTS

* 1. Performance Metrics

# ADVANTAGES & DISADVANTAGES

1. **CONCLUSION**

# FUTURE SCOPE

1. **APPENDIX**

Source Code

GitHub & Project Demo Link

**CHAPTER 1 INTRODUCTION**

## 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. 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.

## Purpose

The purpose of this project to detect the natural disaster and reduce, or avoid, the potential losses from hazards, assure prompt and appropriate assistance to victims of disaster, and achieve rapid and effective recovery.

# CHAPTER 2 LITERATURE SURVEY

**TITLE** : A Deep Learning Approach of Recognizing Natural Disasters on Images.

## PROPOSED WORK

First, this work introduces to the research community a new dataset for the joint classification of natural disaster types and intensity. Moreover, this study primarily aims to explore natural disasters recognition using a convolutional neural network and transfer learning. An open source tool is used for finding and removing the repeated images for analysis. Wildfire, Earthquake, Flood and Volcanic eruption are taken. In particular, this study attempts to build and train a lightweight convolutional neural network that can jointly recognize natural disaster types and intensity. Based on the intensity, it classifies as Severe, Moderate, Insignificant Lastly, this study attempts to measure the model performance using four performance measures; accuracy, precision, recall, and F1-Score.

## TOOLS USED/ALGORITHM

* Image Processing
* Slope NDVI
* Location API
* Cloud Architecture
* Google Earth Engine
* K-Means and Classification Algorithm
* RGB Scale

**TECHNOLOGY :** Artificial Intelligence

**TITLE :** Disaster Intensity-Based Selection of Training Samples for Remote Sensing Building Damage Classification.

## PROPOSED WORK

In this proposed work, two fully automatic procedures for the detection of severely damaged buildings are introduced. The fundamental assumption is that samples that are located in areas with low disaster intensity mainly represent nondamaged buildings. Furthermore, areas with moderate to strong disaster intensities likely contain damaged and nondamaged buildings. Under this assumption, a procedure that is based on the automatic selection of training samples for learning and calibrating the standard support vector machine classifier is utilized. The second procedure is based on the use of two regularization parameters to define the support vectors. These frameworks avoid the collection of labeled building samples via field surveys and/or visual inspection of optical images, which requires a significant amount of time. The performance of the proposed method is evaluated via application to three real cases. The resulted accuracy ranges between 0.85 and 0.89, and thus, it shows that the result can be used for the rapid allocation of affected buildings.

## TOOLS USED/ALGORITHM

* Automatic labelling
* Building damage
* Multi regularization parameters
* Demand Parameter
* Support Vector Machine (SVM)

**TECHNOLOGY:** Machine Learning

**TITLE:** Hurricane Damage Detection using Machine Learning and Deep Learning Techniques

## PROPOSED WORK

In this proposed work, Disaster detection can be done through social media and satellites. Images obtained from satellites are widely used since capturing and processing of these images can be done in a shorter span of time. Satellite images help to recognize damage pattern caused by the disasters. The images from social media are also useful since they provide information on an immediate basis. Since manual methods are error-prone, deep learning and machine learning are used which used for detecting the damage caused by disasters effectively.

## TOOLS USED/ALGORITHM

* + Social-media
  + Satellite imagery
  + Deep learning techniques
  + CNN,VGG-16, ResNet
  + Machine learning techniques
  + Support Vector Machine, Decision trees, random forest.

**TECHNOLOGY:** Machine Learning, Deep Learning

## Existing Problem

Earlier we focus on post disaster relief and rehabilitation measures. Now the focus is shifted. As per sec.2 (e) of DM Act 2005, Disaster Management means a coordination and integrated process of planning, organizing, coordinating, and implementing measures which are necessary or expedient for-

* + 1. Prevention of danger or threat of any disaster
    2. Preparedness to deal with any disaster
    3. Prompt response to any threatening disaster situation or disaster
    4. Assessing the severity or magnitude of effects of any disaster
    5. Evacuation, rescue, and relief
    6. Rehabilitation and reconstruction

## References

1. Mignan, A.; Broccardo, M. Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations. Seism. Res. Lett. 2020, 91, 2330–2342. [CrossRef]
2. Tonini, M.; D’Andrea, M.; Biondi, G.; Degli Esposti, S.; Trucchia, A.; Fiorucci, P. A Machine Learning-Based Approach for Wildfire Susceptibility Mapping. The Case Study of the Liguria Region in Italy. Geosciences 2020, 10,

105. [CrossRef]

1. Islam, A.R.M.T.; Talukdar, S.; Mahato, S.; Kundu, S.; Eibek, K.U.; Pham, Q.B.; Kuriqi, A.; Linh, N.T.T. Flood susceptibility modelling using advanced ensemble machine learning models. Geosci. Front. 2021, 12, 101075. [CrossRef]
2. Schlemper, J.; Caballero, J.; Hajnal, V.; Price, A.N.; Rueckert, D. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans. Med. Imaging 2017, 37, 491–503. [CrossRef] [PubMed]
3. Tang, C.; Zhu, Q.; Wu, W.; Huang, W.; Hong, C.; Niu, X. PLANET: Improved convolutional neural networks with image enhancement for image classification. Math. Probl. Eng. 2020, 2020. [CrossRef]

## Problem Statement Definition

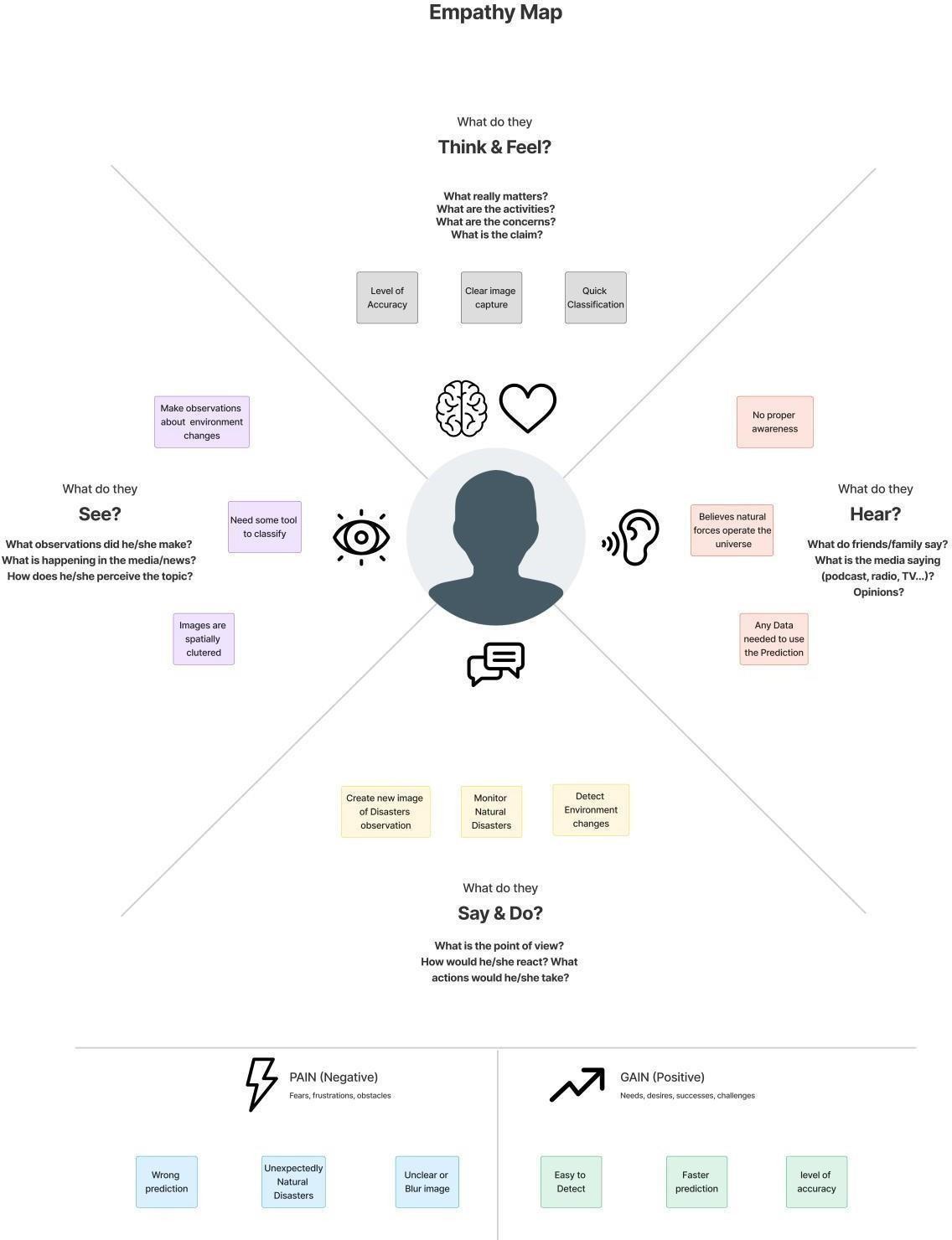
People needs a way to classify and analyze the natural disaster so that they can prevent themselves from losses due to the disaster and millions of lives.

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

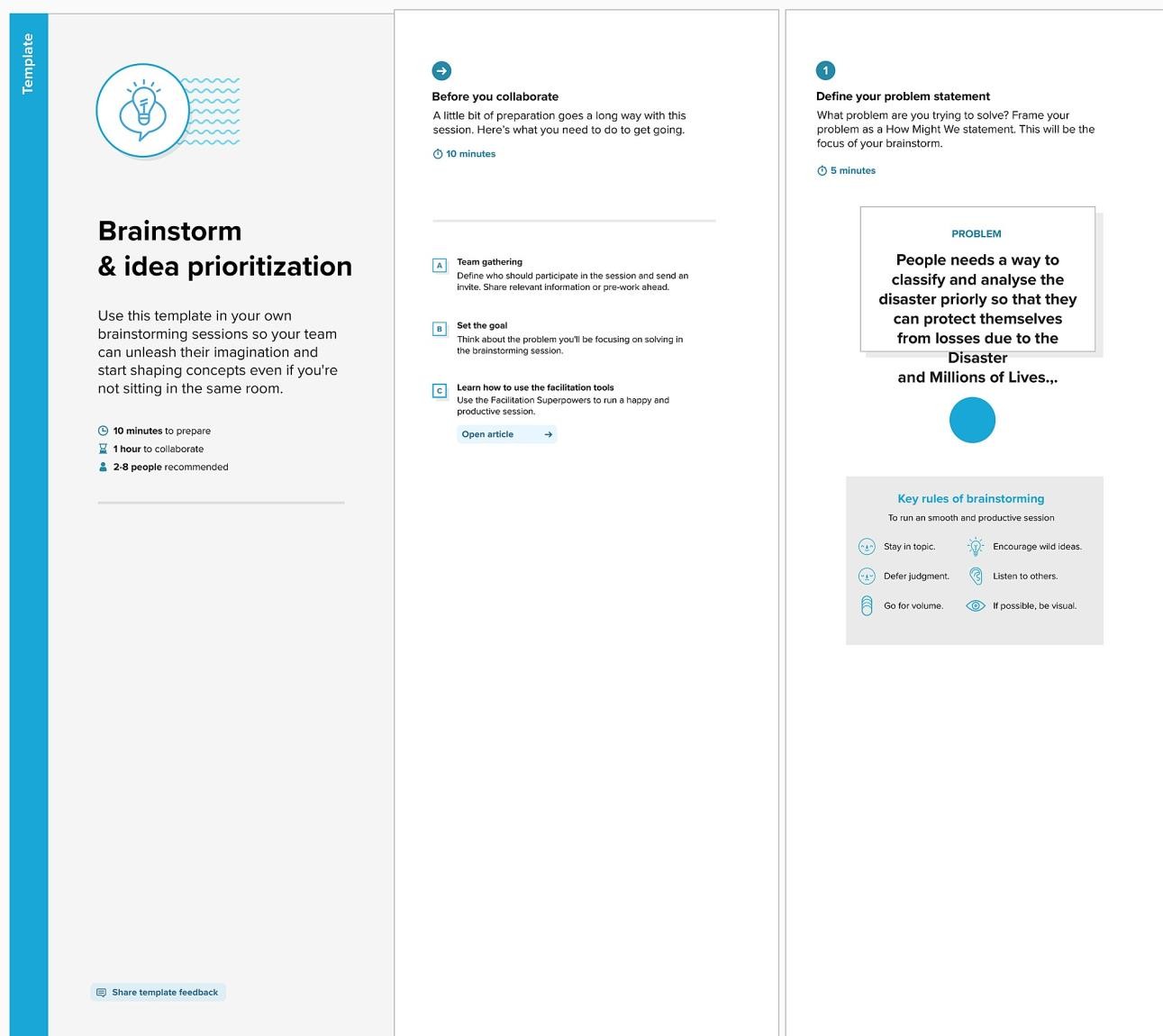
# CHAPTER 3

**IDEATION & PROPOSED SOLUTION**

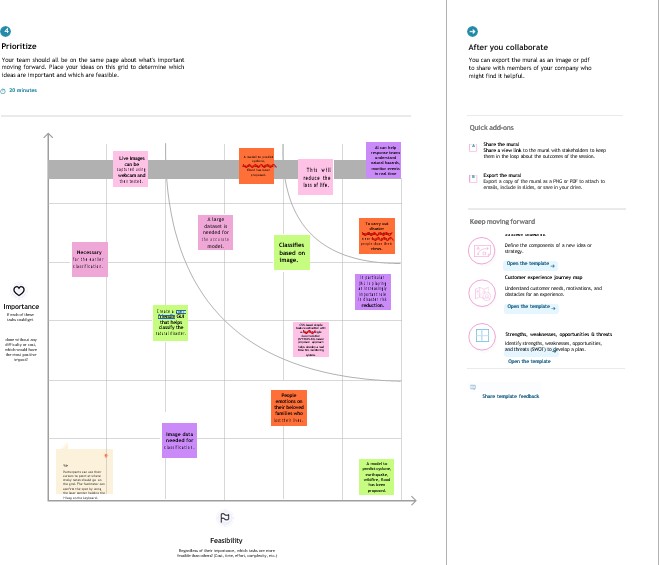
## Empathy Map Canvas



* 1. **Ideation & Brainstorming**



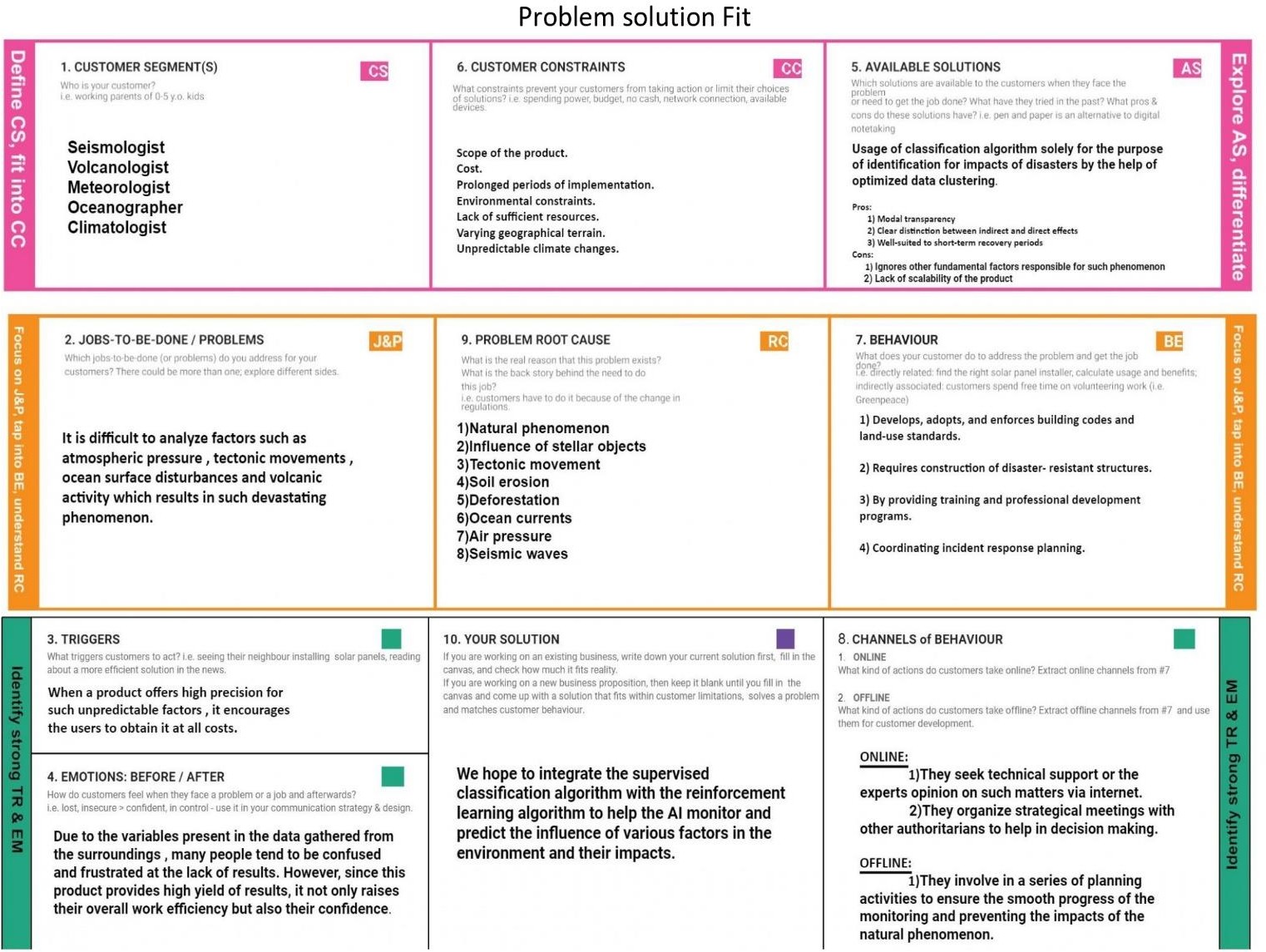
 



## Proposed solution

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to  be solved) | People needs a way to classify and  analyze the Disaster priorly so that they can protect themselves from losses due to the Disaster and Millions of Lives., |
| 2. | Idea/Solution description | This project uses Multi-layered Deep Convolutional Neural Network (pre- trained) model to classify Natural Disaster and calculate the intensity of  the Disaster. |
| 3. | Novelty/Uniqueness | To reduce the issues due to imbalance structure of images, the model uses an integrated webcam to capture the video frame and test data is compared with  pretrained data. |
| 4. | Social impact/Customer  Satisfaction | By the Application, economic damage  caused by Disaster can be reduced. Detection of Natural Disaster will become easier while using videos in Deep CNN instead of images. |
| 5. | Business Model (Revenue  Model) | Multi-layered Deep Convolutional  Neural Network Model. |
| 6. | Scalability of the Solution | Highly expandible, dependable,  reliable, scalable and has robustness. |

* 1. **Problem Solution Fit**



# CHAPTER 4 REQUIREMENT ANALYSIS

## Functional Requirement

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Functional Requirement(Epic)** | **Sub Requirement (Story / Sub-Task)** |
| **FR-1** | Request Permission | Access permission from web camera. |
| **FR-2** | Disaster Detection | Based on the webcam image, natural  disaster is classified. |
| **FR-3** | Accuracy | Since the training and testing images are huge,  The accuracy is higher. |
| **FR-4** | Speed | The generation of results from the input  Images are faster. |
| **FR-5** | Resolution | The resolution of the integrated web camera should be high enough to capturethe video  frames. |
| **FR-6** | User Interface | Maximizing the interaction in Web  Designing Service. |

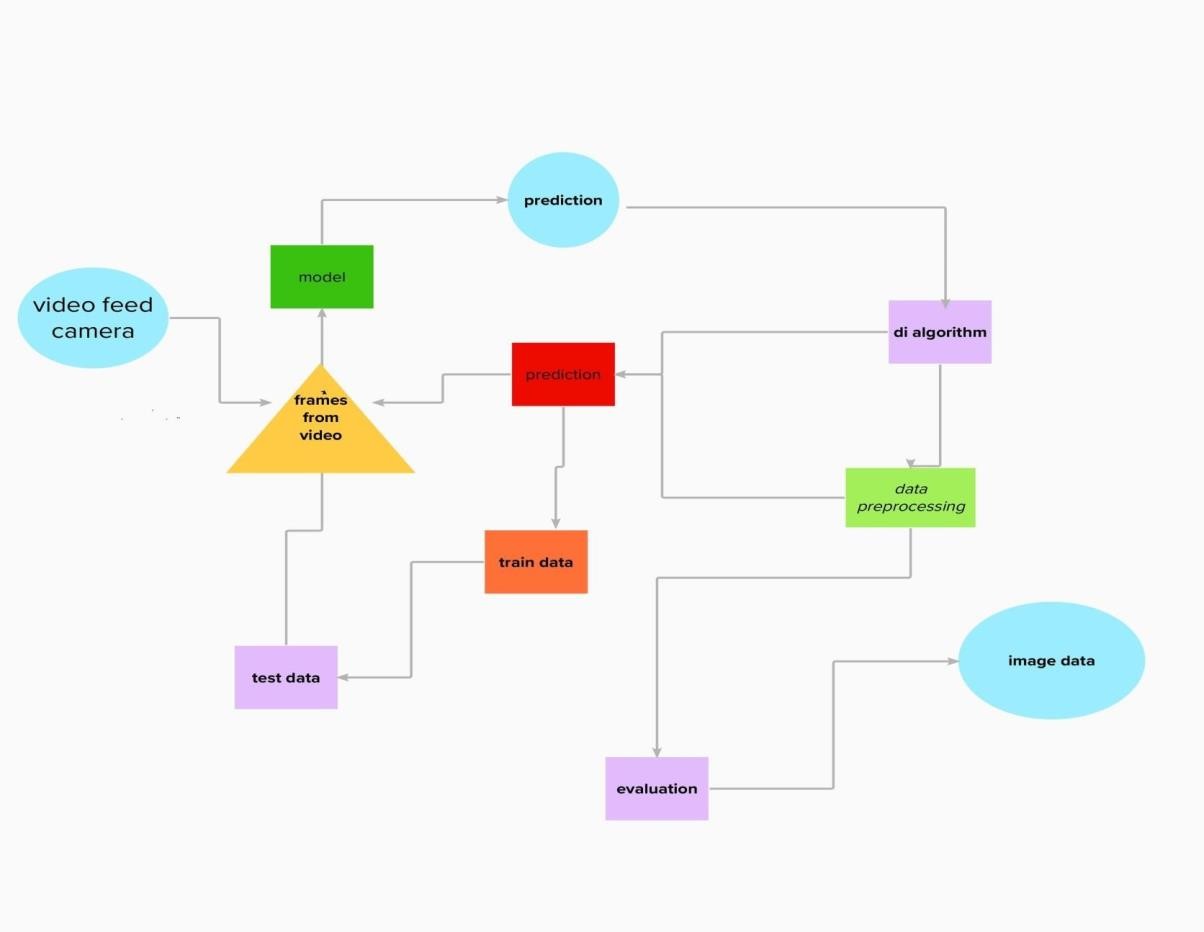
* 1. **Non-Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **NFR. No.** | **Non- Functional**  **Requirement** | **Description** |
| **NFR-1** | Usability | User friendly and classify the disaster easily. |
| **NFR-2** | Security | The model is secure due to the cloud deployment models and also there is no login  issue. |
| **NFR-3** | Reliability | Accurate prediction of the natural disaster  and the website can also be fault tolerant. |
| **NFR-4** | Performance | It is shown that the model gives almost 95  percent accuracy after continuous training. |
| **NFR-5** | Availability | The website will be made available for 24  hours. |
| **NFR-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. |

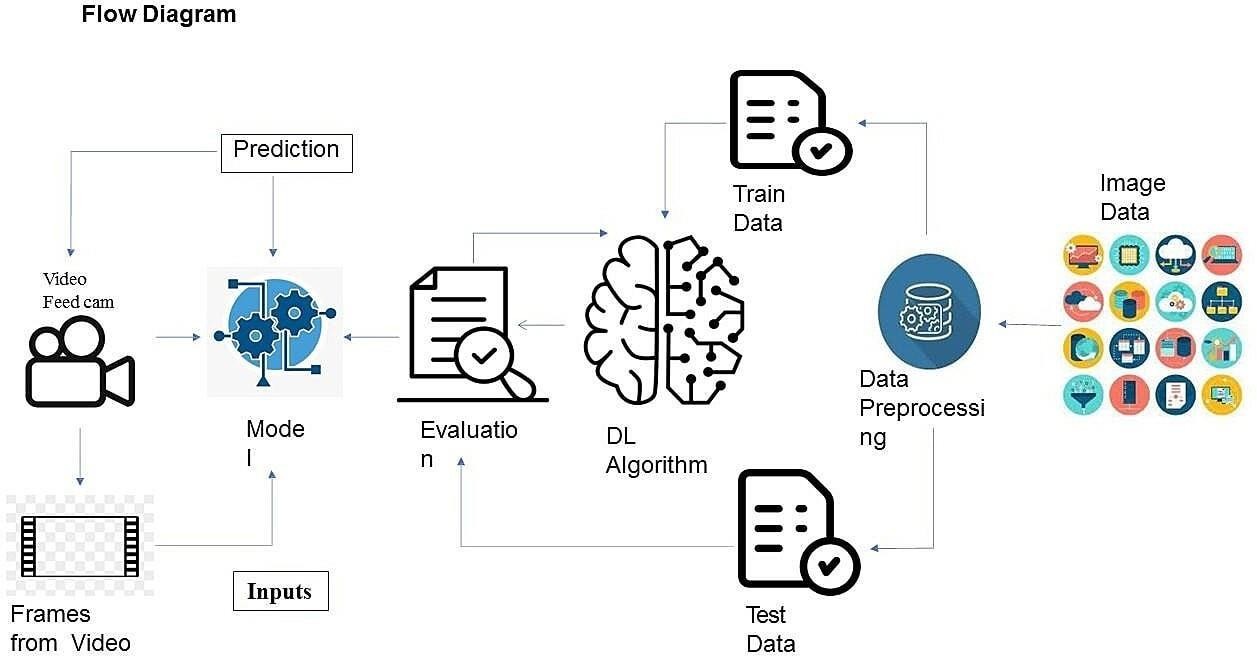
**5.2 Data Flow Diagrams**

# CHAPTER 5 PROJECT DESIGN

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 to be enter and leaves the system, what changes the information, and where data is stored.



## Flow Diagram

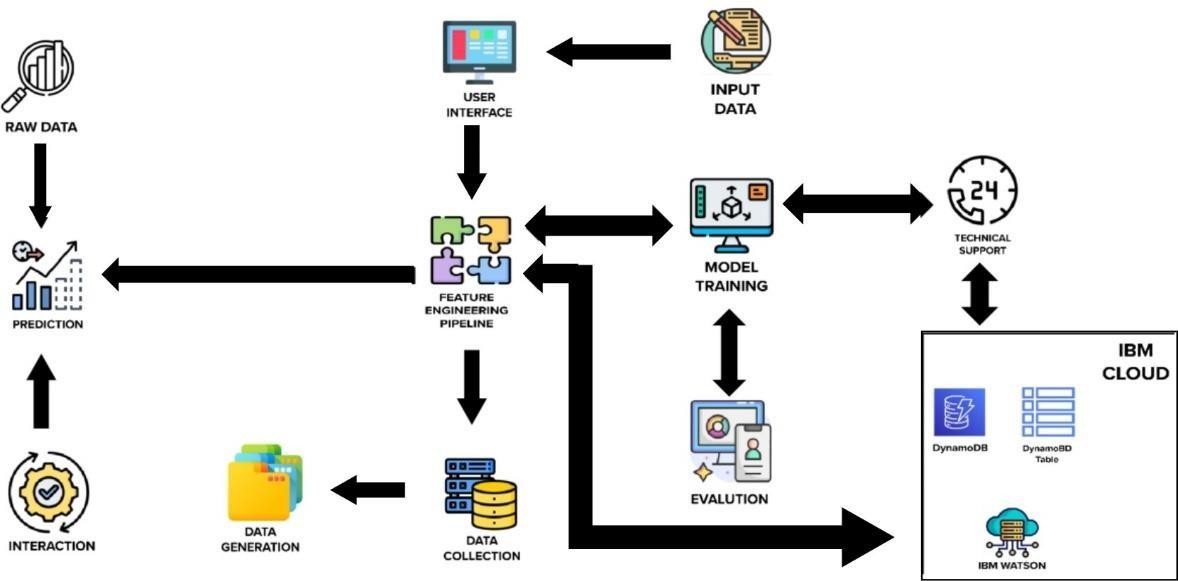


* 1. **Solution & Technical Architecture Solution Architecture**

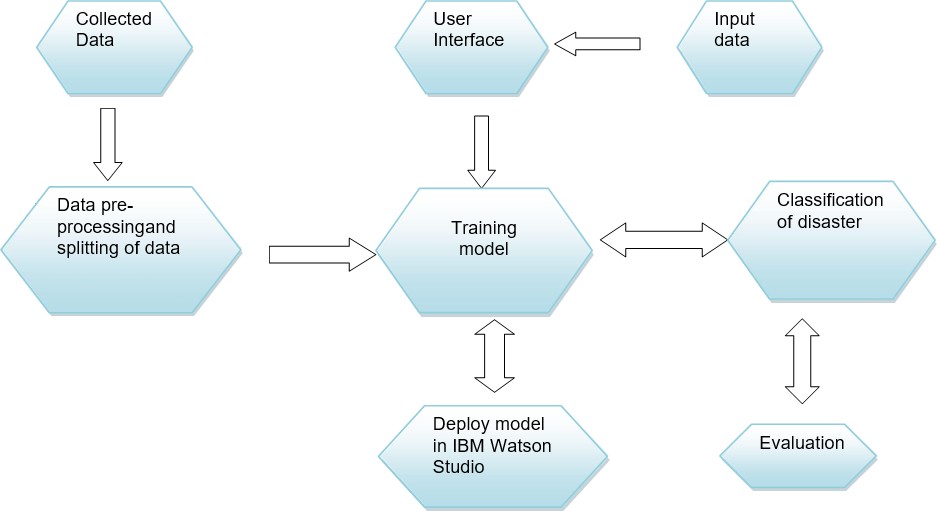
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

* + - Find the best tech solution to solve existing business problems.
    - Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
    - Define features, development phases, and solution requirements.
    - Provide specifications according to which the solution is defined, managed, and delivered.

## Solution Architecture Diagram



**Technical Architecture**



## Components &Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | User interacts with application for the detection  of any Natural disaster’s intensity and classify which happened just before. | HTML, CSS,  JavaScript, Django, Python. |
| 3. | Disaster | This function is used to detect, | Decision |
| Detection | Outcomes from | trees, Regression, |
|  | the new trained data to perform new | Convolutional |
|  | tasks and solve new problems. | Neural networks |
| 4. | Evaluation system | It monitors that how Algorithm performs on data as  well as during training. | Chi-Square, Confusion  Matrix, etc. |
| 5. | Input data | 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. |

|  |  |  |  |
| --- | --- | --- | --- |
| 6. | 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, SQLServer. |
| 7. | Database  management system | An organized collection of data stored in database,so that it can be easily accessed and managed. | MySQL,  DynamoDB etc. |

**Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | An open source framework is a template for software development that is designed by a social network of software developers. These frameworks are free for public use and provide the foundation for  building a software application. | Keras, Tensorflow. |
| 2. | Authentication | This keeps our models secure and makes sure only those who have  permission can use them. | Encryption and Decryption  (OTP). |

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | Application interface | User uses mobile application and web application to interact with model | Web Develop ment (HTML, CSS) |
| 4. | Availability (both Online and Offline work) | Its include both online and offline work. As good internet connection is need for online work to explore the software perfectly. Offline work  includes the saved data to explore for  later time. | Caching, backend server. |
| 5. | Regular Updates | The truly excellent software product needs a continuous process of improvements and updates. Maintain your server and make sure that your content is always up-to- date. Regularly update an app and enrich it with new features. | Waterfall Approach, Incremental Approach, Spiral Approach |
| 6. | Personalization | Software has features like flexible fonts, backgrounds, settings, color themes, etc. which make a software  interface looks good and functional. | * CSS |

* 1. **User Stories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Requirement**  **(Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** |
| Collection of dataset | USN-1 | As a user, I can collect the dataset for monitoring and analyzing. | Enough data collected for training Model. |
| Home Page | USN-2 | As a user, I want to know to about the basics of frequently occurring  Disasters. | I can get the idea about the Application. |
| Intro page | USN-3 | As a user, I want to about the introduction of Disaster in particular  areas. | I can get idea about the disaster and where it occurs. |
| Open webcam | USN-4 | As a user, I adapt with the webcam to analyze and classify the Disaster from video capturing | I can capture a video or image of particular disaster to analyze and classify. |
| Analysis of required phenomenon | USN-5 | As a user, I can regulate certain factors influencing the action and report on past event analysis. | Model should be easy to use & working fine from the web app. |

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm selection | USN-6 | As a user, I can choose the required algorithm for  specific analysis. | Selection must give the better accuracy and better  output. |
| Training and Testing | USN-7 | As a user, I can train and test the model using the  algorithm. | Training the model to classify and analyze the  intensity |
| Detection and analysis of data | USN-8 | As a user, I can detect and visualize the data  effectively. | I can capture a video or image of particular disaster  to analyze and detect. |
| Model building | USN-9 | As a user I can build with the web application | Model should be predicting occurrence of the disaster and intensity level of  disaster. |
| Integrate the web app with the AI Model | USN-10 | As a user, I can use Flask app to use model easily through web app. | Model should be easy to use and working fine from the web app. |
| Model deployment | USN-11 | As an administrator, I can deploy the AI model in IBM Cloud. | Model’s prediction should be available for users to make decision. |

# CHAPTER 6

**PROJECT PLANNING & SCHEDULING**

## Sprint planning & Estimation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement**  **(Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** |
| Sprint-1 | Collection of  Dataset | USN-1 | As a user, I can collect the dataset  for monitoring and analysing. | 5 |
| Sprint-1 | Home page | USN-2 | As a user, I want to know to about the basics of frequently occurring  Disasters. | 5 |
| Sprint-1 | Intro page | USN-3 | As a user, I want to about the introduction of Disaster in  particular areas. | 5 |
| Sprint-1 | Open webcam | USN-4 | As a user, I adapt with the webcam to analyze and classify the Disaster from video capturing. | 5 |
| Sprint-2 | Analysis of required  phenomenon | USN-5 | As a user, I can regulate certain factors influencing the action and  report on past event analysis. | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint-2 | Algorithm selection | USN-6 | As a user, I can choose the required Algorithm for specific  analysis. | 5 |
| Sprint-2 | Training and  Testing | USN-7 | As a user, I can train and test the  model using the algorithm. | 10 |
| Sprint-3 | Detection and analysis of  data | USN-8 | As a user, I can detect and visualize the data effectively. | 10 |
| Sprint-3 | Model  building | USN-9 | As a user, I can build  with the web application. | 10 |
| Sprint-4 | Integrate the web app with the AI  model | USN-11 | As a user, I can use Flask app to use model easily through web app. | 10 |
| Sprint-4 | Model  deployment | USN-12 | As an administrator, I can deploy  the AI model in IBM Cloud. | 10 |

* 1. **Sprint Delivery schedule**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 20 | 19 Nov 2022 |

## Reports from Jira

**Velocity:**

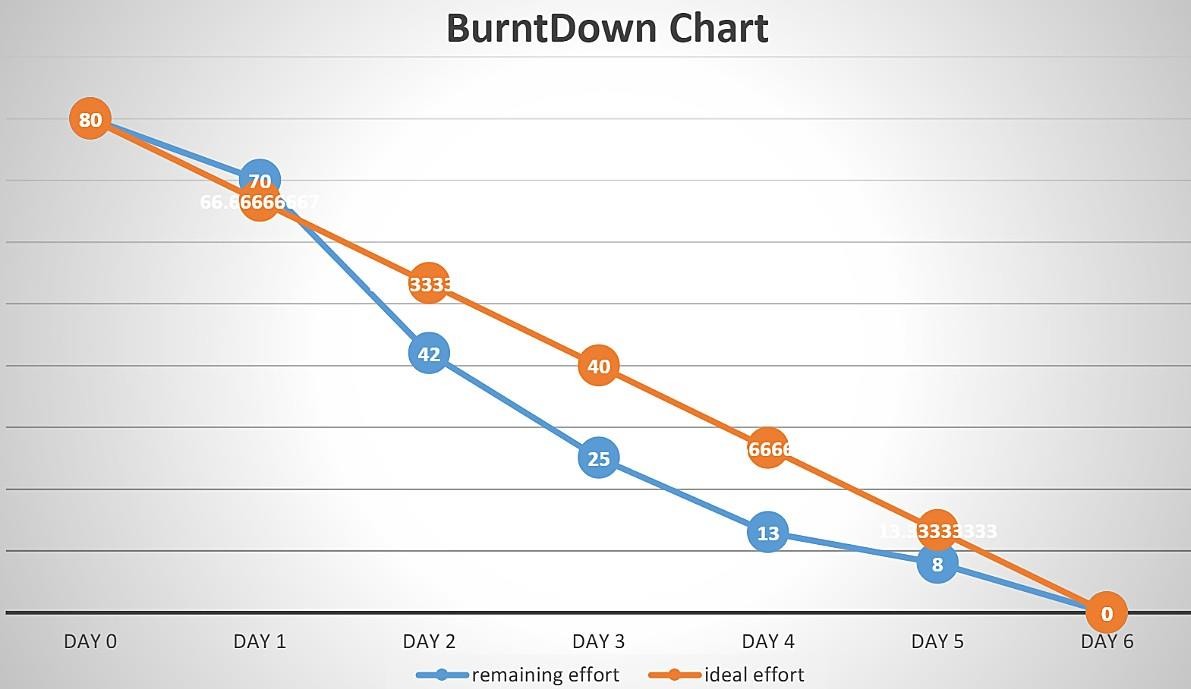
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

## Average velocity = Sprint duration / velocity

**=20/6**

## =3

**Burndown Chart:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile [software development m](https://www.visual-paradigm.com/scrum/what-is-agile-software-development/)ethodologies such as [Scrum.](https://www.visual-paradigm.com/scrum/scrum-in-3-minutes/) However, burn down charts can be applied to any project containing measurable progress over time.

# CHAPTER 7

## CODING & SOLUTIONING

* 1. **Feature 1**

The project focuses on the analysis of intensity of Disaster for giving precautionary measures for the people living in the Danger zone.

It focuses on classifying the type of Disaster which oftenly occurs in that particular zone.

## Feature 2

The accuracy of the project is improved more better than the previously submitted models.

The accuracy is improved by training and testing more images in the dataset.

# CHAPTER 8 TESTING

## Test cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Component** | **Test Scenario** | **Expected Result** | **Actual Result** | **Status** |
| TC\_001 | Home Page | Verify user is able to see the Home  page | Home page should display | Working as  expected | Pass |
| TC\_002 | Home Page | Verify the UI elements in Home page | Application should show below UI elements:  Home page button  Intro page button | Working as expected | Pass |
| Open webcam button |
| TC\_003 | Home Page | Verify user is able to see the cards  about Disaster | Application should show the cards about  Disaster. | Working as  expected | Pass |
| TC\_004 | Home Page | Verify user is able to navigate to the required page | Application should navigate to the Intro page | Working as  expected | Pass |
| TC\_005 | Intro Page | Verify user is able  to see the Intro page | Intro page should  display | Working  as expected | Pass |
| TC\_006 | Intro Page | Verify the UI Elements in Intropage | Application should show below UI elements:  Home page  Intro page | Working as expected | Pass |
| Open webcam button |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_007 | Intro Page | Verify the user is able to see the introduction of the  Disaster | Application should show the sentences about the Disaster | Working as expected | Pass |
| TC\_008 | Intro Page | Verify user is able to navigate  to the required  page | Application should navigate to the Open webcam page | Working as expected | Pass |
| TC\_009 | Webcam page | Verify user is able to see the webcam  page | Webcam page is displayed | Working as  expected | Pass |
| TC\_010 | Webcam  page | Verify the  Emergency pull button is visible while the webcam  is not connected | Application should  show below UI elements:  a. Emergency pull  button | Working  as expected | Pass |
| TC\_011 | Webcam page | Verify user is able to see the  output  window | Application should detect the type of Disaster from the real time video | Working as expected | Pass |

* 1. **User Acceptance Testing**

It is to briefly explain the test coverage and open issues of the natural disasters intensity analysis and classification using artificial intelligence project at the time of the release to User Acceptance Testing (UAT).

## Defect Analysis:

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 1 | 0 | 0 | 2 | 3 |
| Duplicate | 1 | 0 | 0 | 0 | 1 |
| External | 0 | 0 | 0 | 0 | 0 |
| Fixed | 1 | 0 | 0 | 2 | 3 |
| Not Reproduce | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 0 | 1 | 1 |
| Won't Fix | 0 | 0 | 0 | 0 | 0 |
| Totals | 3 | 0 | 0 | 5 | 8 |

## Test Case Analysis:

This report shows the number of test cases that have passed, failed, and untested.

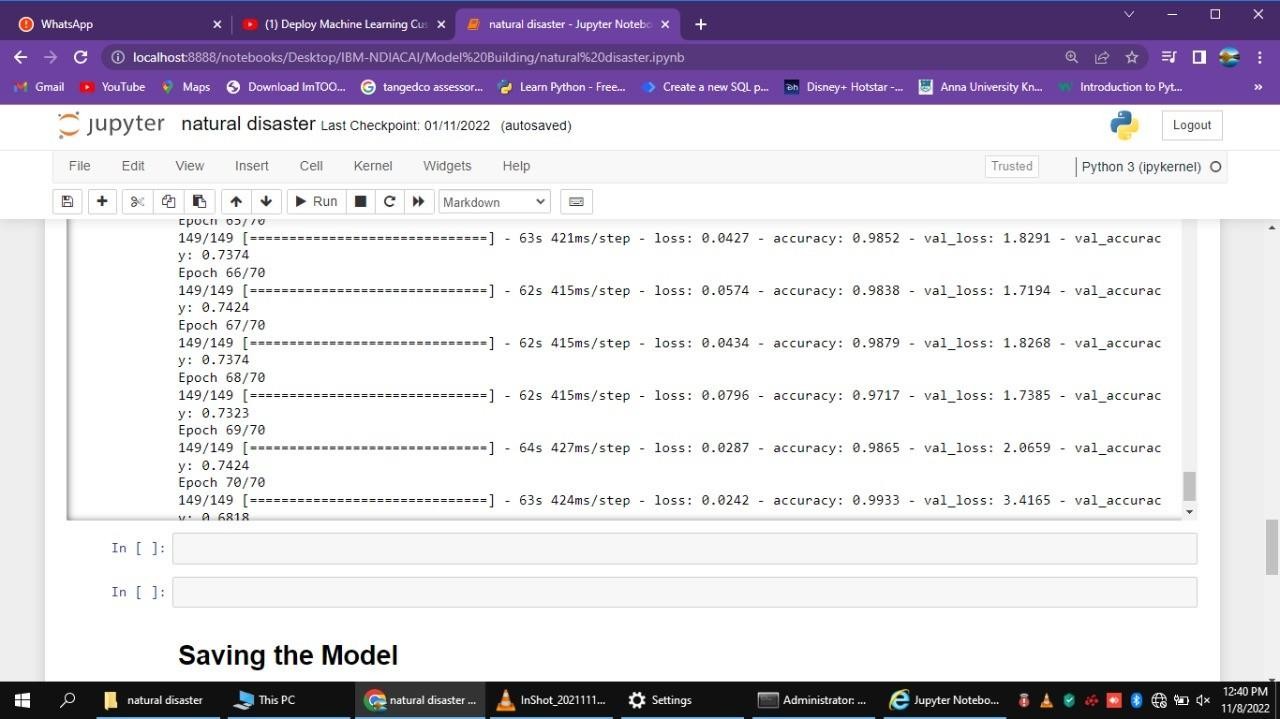
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Test Cases** | **Not Tested** | **Fail** | **Pass** |
| Home Page | 4 | 0 | 0 | 4 |
| Intro Page | 4 | 0 | 0 | 4 |
| Open Webcam | 3 | 0 | 0 | 3 |

# CHAPTER 9 RESULTS

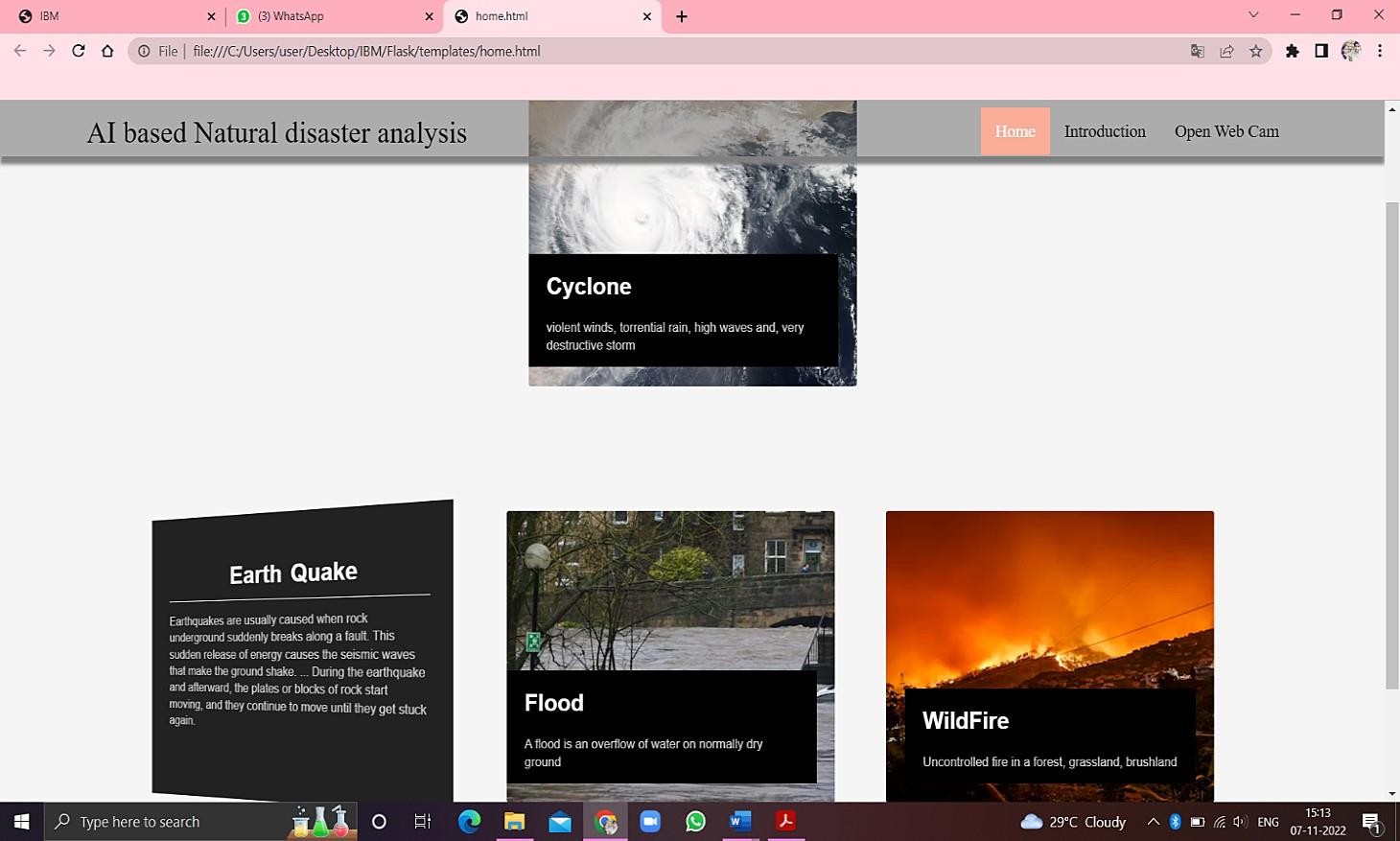
## Performance Metrics

The nature disaster intensity analysis and classification with test data and train data has been executed successfully. The model has been trained over 1000+ images and the model have an accuracy of nearly 99% and the model has been tested with the data which is separate from the trained data and has predicted the data well.

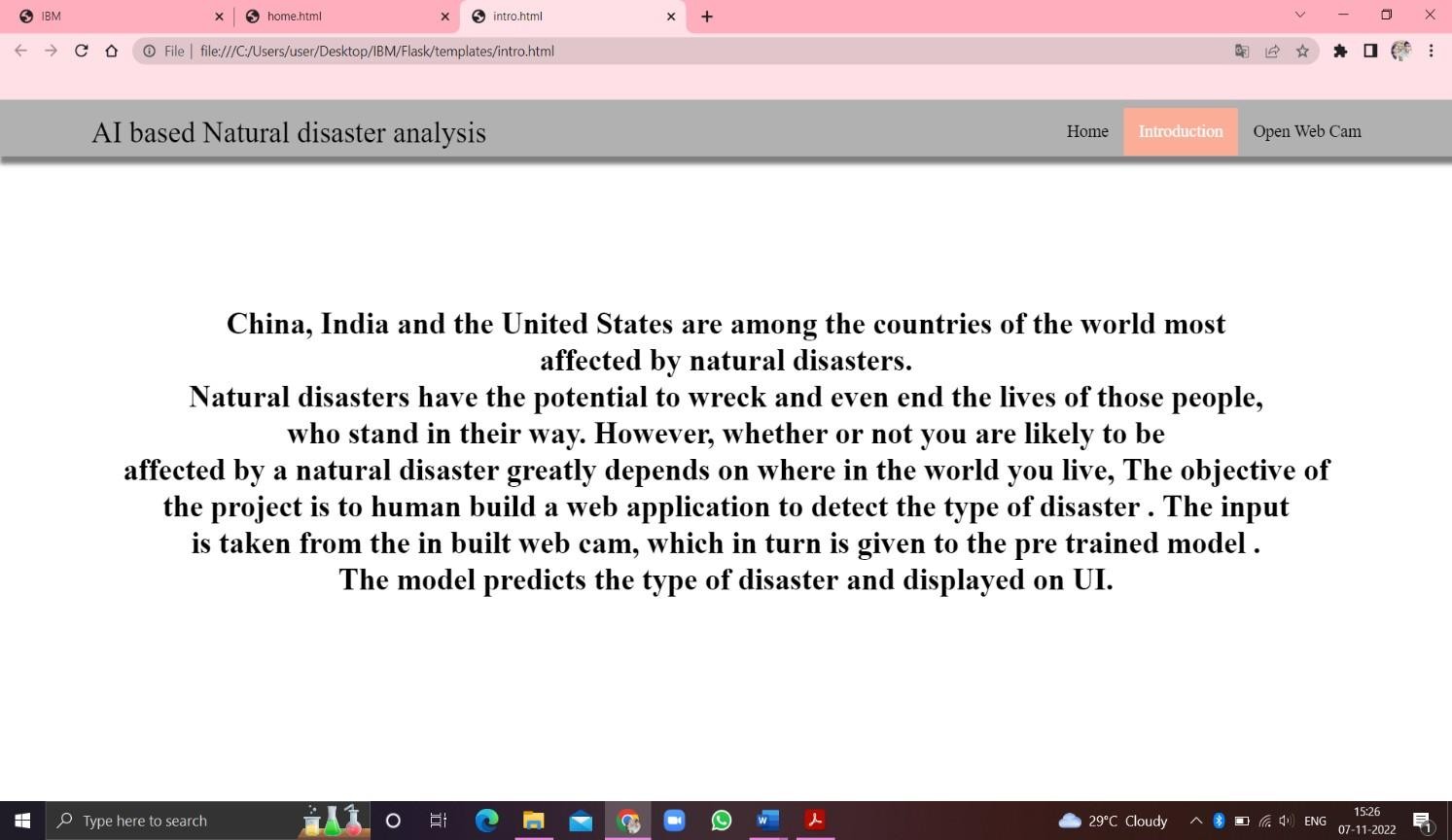
**Output of application**



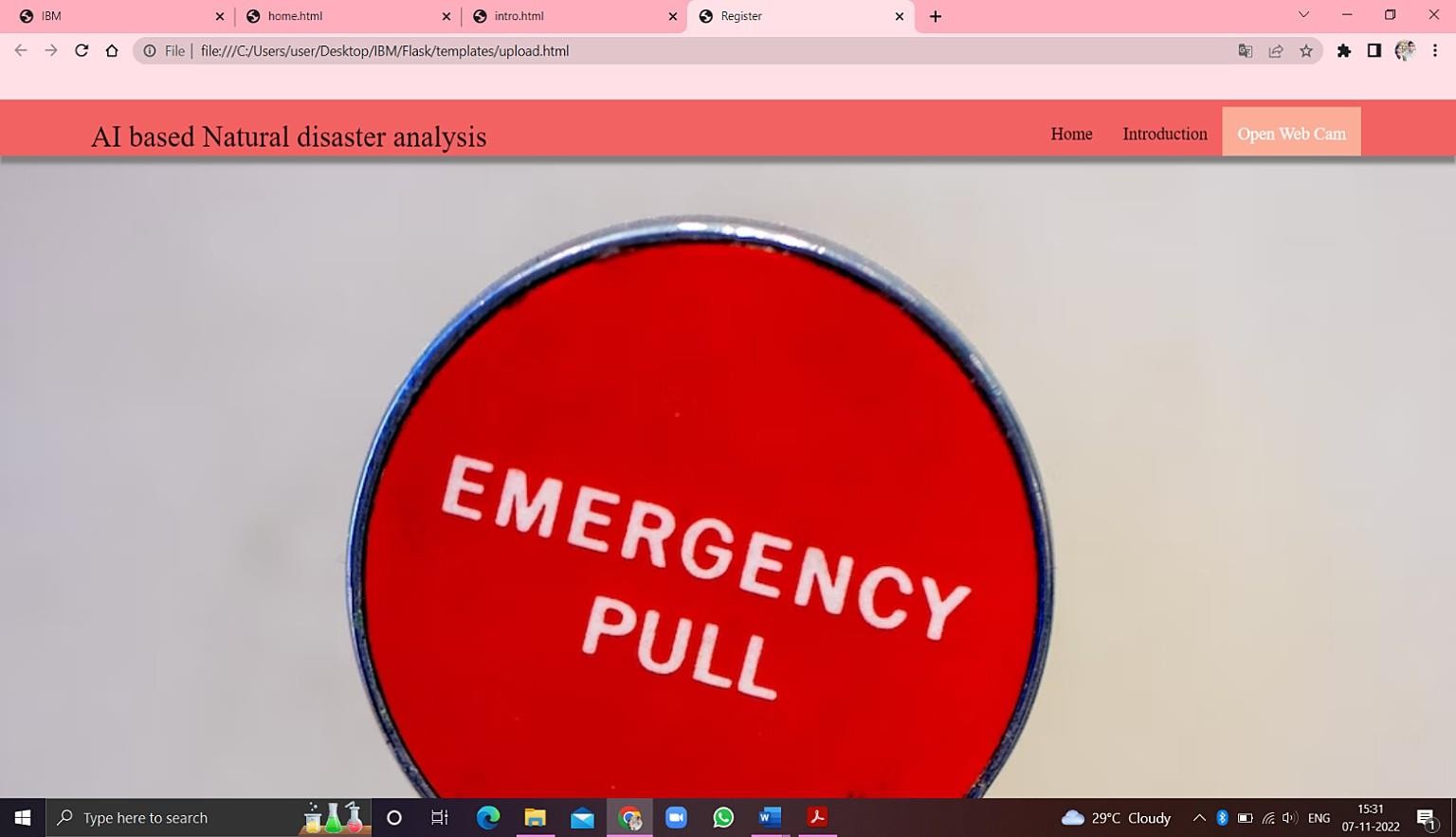
# HOME PAGE



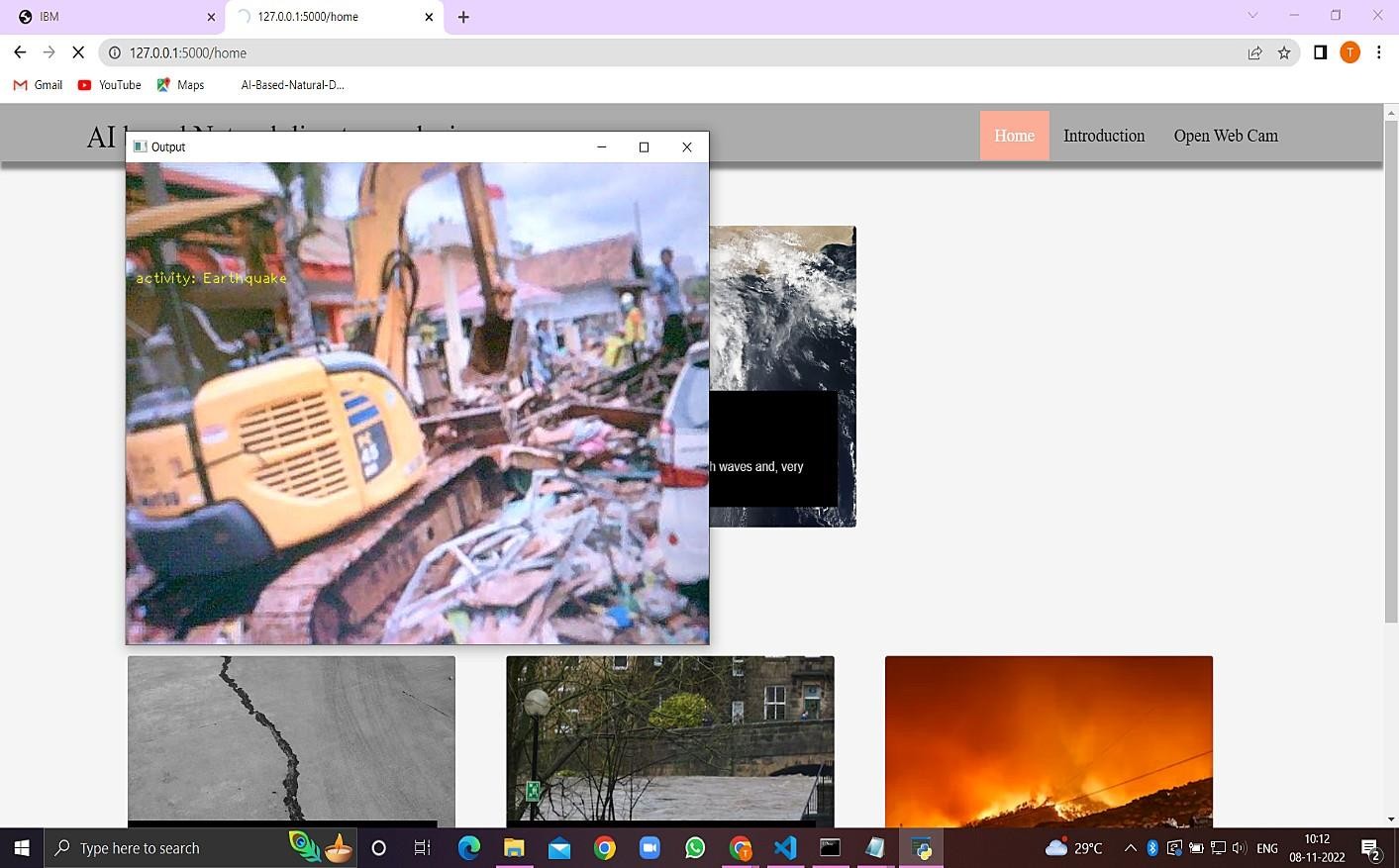
## INTRODUCTION PAGE

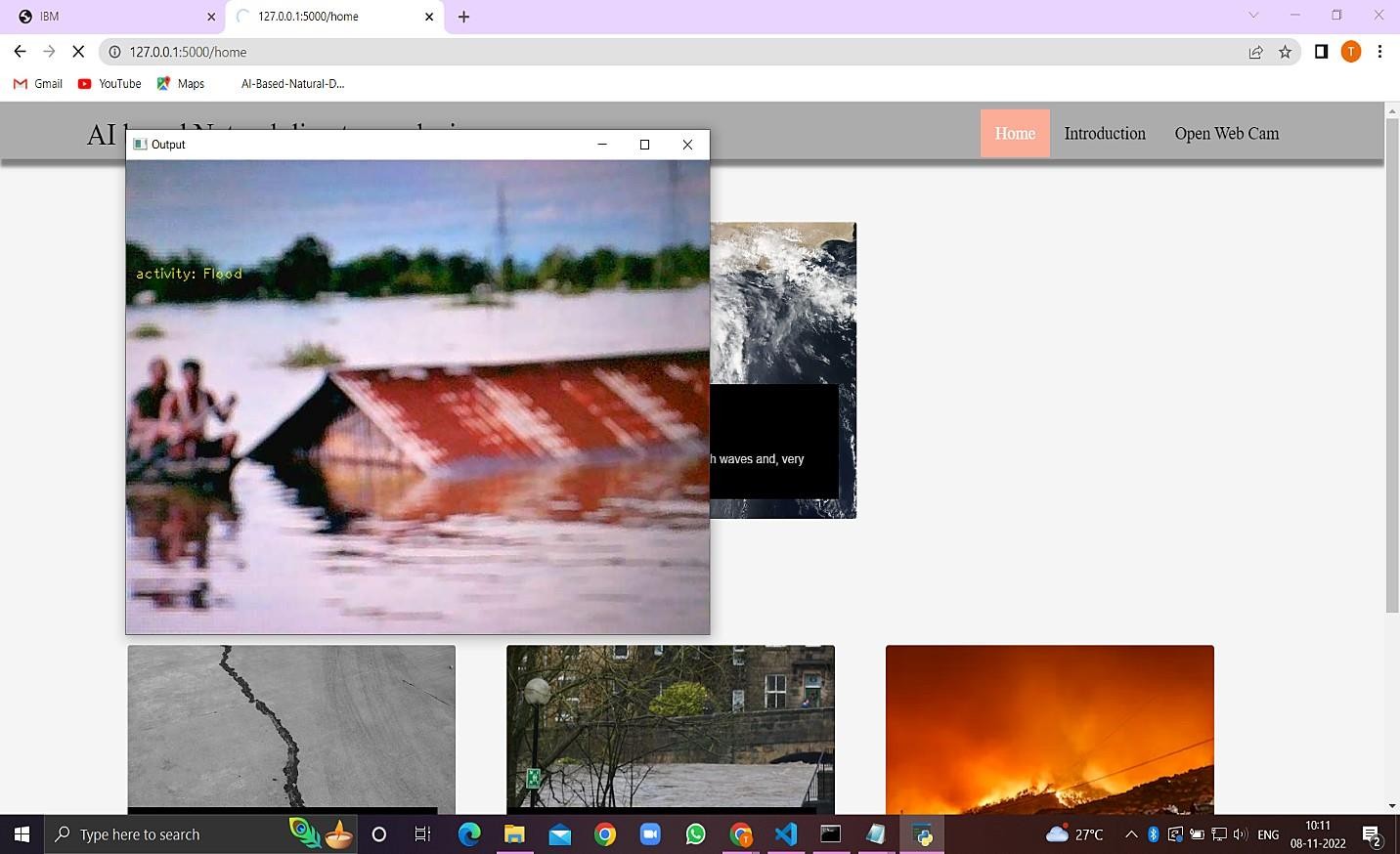


**WEB CAM**



**DETECTION OF NATURE DISASTER**





# CHAPTER 10 ADVANTAGES & DISADVANTAGES

## ADVANTAGES

* + 1. The proposed model will be used as a real time natural disaster detection model and provide some upcoming predictions for future disasters.
    2. The model is to detect and classify the type of disaster and The model have a high accuracy rate ( 99.33).
    3. The model was used to prevent natural disasters in the future and model can be used to predict future disasters and take some action against heavy loss of human ecological systems and property.
    4. The proposed system helps to reduce the impact of hazards occur during natural disaster. This provides an efficient way to warn and educate people about disaster prone areas.
    5. It will help us be prepared in times of disaster

## DISADVANTAGES

1. The resultant model unable to validate the model performance under uncontrolled conditions.
2. The model cannot be used for various natural disaster

# CHAPTER 11 CONCLUSION

It focused how image from given dataset (trained dataset) in field and past data set used predict the pattern of different nature disaster using CNN model. In the system had applied different type of CNN compared the accuracy. The natural disaster in Indonesia frequently happened, due to the geographical position of the country. Thus, natural disasters mostly occurred as an impact of the natural condition. However, the weather and climate condition has also influenced and triggered the disasters.

# CHAPTER 12 FUTURE SCOPE

In the future, the research will be continued to obtain the data from all over the country, not only west java province, and with the use of more complete analysis, so that the government or related institution could make a better anticipation work as a mitigation effort.

# CHAPTER 13 APPENDIX

## Inserting necessary libraries

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 tensorflow. version

tensorflow.keras. version

## Image Data Augumentation

**#setting parameter for Image Data agumentation to the training 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)

## Loading our data and performing Data Augumentation #performing data agumentation to train data

x\_train=train\_datagen.flow\_from\_directory(r'C:\Users\vasanth\Desktop\IBM Project\dataset\train\_set',target\_size=(64, 64),batch\_size=5,

color\_mode='rgb',class\_mode='categorical')

## #performing data agumentation to test data

x\_test=test\_datagen.flow\_from\_directory(r'C:\Users\vasanth\Desktop\IBM Project\dataset\test\_set',target\_size=(64, 64),batch\_size=5,

color\_mode='rgb',class\_mode='categorical') print(x\_train.class\_indices**)#checking the number of classes**

print(x\_test.class\_indices)#**checking the number of classes**

from collections import Counter as c c(x\_train .labels)

## Creating the Model

**# Initializing the CNN**

classifier = Sequential()

## # First convolution layer and poolingo

classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 3), activation='relu'))

classifier.add(MaxPooling2D(pool\_size=(2, 2)))

classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 3), activation='relu'))

## # 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 layer

classifier.add(MaxPooling2D(pool\_size=(2, 2)))

classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 3), activation='relu'))

## # Flattening the layers

classifier.add(Flatten())

## # Adding a fully connected layer

classifier.add(Dense(units=128, activation='relu')) classifier.add(Dense(units=4, activation='softmax')) **# softmax for more than 2**

classifier.summary() #**summary of our model # Compiling the Model**

## # Compiling the CNN

**# categorical\_crossentropy for more than 2**

classifier.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

## # Fitting the Model

classifier.fit\_generator( generator=x\_train,steps\_per\_epoch = len(x\_train),

epochs=10, validation\_data=x\_test,validation\_steps = len(x\_test**))# No of images in test set**

**# Saving the Model** classifier.save('disaster.h5') model\_json = classifier.to\_json()

with open("model-bw.json", "w") as json\_file: json\_file.write(model\_json)

## # Predicting Results

from tensorflow.keras.models import load\_model from keras.preprocessing import image

model = load\_model("disaster.h5") **#loading the model for testing**

img=image.load\_img(r"C:\Users\vasanth\Desktop\IBMProject\dataset\test\_set\Cyc lone\921.jpg",grayscale=False,target\_size= (64,64)) **#loading of the image\n**

x = image.img\_to\_array(img**)#image to array\n",**

x = np.expand\_dims(x,axis = 0)**#changing the shape\n",**

pred = model.predict\_classes(x**)#predicting the classes\n",**

pred index=['Cyclone','Earthquake','Flood','Wildfire'] result=str(index[pred[0]])

result

**Source Code:**

home.html:

<!DOCTYPE html>

<html lang=‖en‖>

<title>Home – Natural Disasters Database</title>

<meta charset=‖UTF-8‖>

<meta name=‖viewport‖ content=‖width=device-width, initial-scale=1‖>

<link rel=‖stylesheet‖ href=https:/[/www.w3schools.com/w3css/4/w3.css>](http://www.w3schools.com/w3css/4/w3.css)

<link rel=‖stylesheet‖ href=https://fonts.googleapis.com/css?family=Lato>

<link rel=‖stylesheet‖ href=https://fonts.googleapis.com/css?family=Montserrat>

<link rel=‖stylesheet‖ href=https://cdnjs.cloudflare.com/ajax/libs/font- awesome/4.7.0/css/font-awesome.min.css>

<style>

Body,h1,h2,h3,h4,h5,h6 {font-family: ―Lato‖, sans-serif}

.w3-bar,h1,button {font-family: ―Montserrat‖, sans-serif}

.fa-anchor,.fa-coffee {font-size:200px}

</style>

<body>

<!—Navbar



<div class=‖w3-top‖>

<div class=‖w3-bar w3-black w3-card w3-left-align w3-large‖>

<a class=‖w3-bar-item w3-button w3-hide-medium w3-hide-large w3-right w3- padding-largew3-hover-white w3-large w3-red‖ href=‖javascript:void(0);‖ onclick=‖myFunction()‖ title=‖Toggle Navigation Menu‖><i class=‖fa fa- bars‖></i></a>

<a href=‖{% url ‗home‘ %}‖ class=‖w3-bar-item w3-button w3-hide-small w3- padding-largew3-hover- white‖>Home</a>

<a class=‖w3-bar-item w3-button w3-padding-large w3-white‖>Earthquake</a>

<a href=‖{%url ‗tsunami‘%}‖ class=‖w3-bar-item w3-button w3-hide-small w3- padding-largew3-hover-white‖>Tsunami</a>

<a href=‖{%url ‗tornado‘%}‖ class=‖w3-bar-item w3-button w3-hide-small w3- padding-largew3-hover-white‖>Tornado</a>

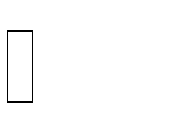
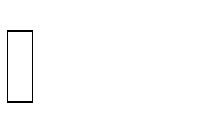
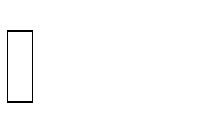
<a href=‖{%url ‗volcano‘%}‖

class=‖w3-bar-item w3-button w3-hide-small w3- padding-largew3- hover-

white‖>Volcanic Activity</a>

</div>

<!—Navbar on small screens



<div id=‖navDemo‖ class=‖w3-bar-block w3-white w3-hide w3-hide-large w3- hide-mediumw3-large‖>

<a href=‖#‖ class=‖w3-bar-item w3-button w3-padding-large‖>Earthquake</a>

<a href=‖#‖ class=‖w3-bar-item w3-button w3-padding-large‖>Tsunami</a>

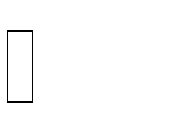
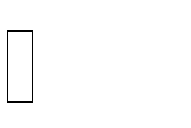
<a href=‖#‖ class=‖w3-bar-item w3-button w3-padding-large‖>Tornado</a>

<a href=‖#‖ class=‖w3-bar-item w3-button w3-padding-large‖>Volcanic Activity</a>

</div>

</div>

<!—Header



<header class=‖w3-container w3-grey w3-center‖ style=‖padding:128px 16px‖>

<h1 class=‖w3-margin w3-jumbo‖>Earthquakes</h1>

<p class=‖w3-xlarge‖>Natural Disasters Database</p>

</header>

<div class=‖w3-container‖>

<h2>Earthquakes</h2>

<table class=‖w3-table-all‖>

<tr>

<th>Earthquake\_id</th>

<th>Intensity</th>

<th>Date</th>

<th>Country</th>

<th>Place</th>

<th>Latitude</th>

<th>Longitude</th>

{% for quake in all\_quakes %}

<tr>

<td>{{quake.earthquake\_id}}</td>

<td>{{quake.intensity}}</td>

<td>{{quake.date}}</td>

<td>{{quake.country}}</td>

<td>{{quake.place}}</td>

<td>{{quake.latitude}}</td>

<td>{{quake.longitude}}</td>

</tr>

{% endfor %}

</table>

</div>

<div class=‖w3-container‖>

<h2>Damage caused by the quakes</h2>

<table class=‖w3-table-all‖>

<tr>

<th>Earthquake\_id</th>

<th>Amount (in million)</th>

<th>Deaths (in thousands)</th>

<th>House\_destroyed (in thousands)</th>

</tr>

{% for d in damage %}

<tr>

<td>{{d.earthquake\_id}}</td>

<td>{{d.amount}}</td>

<td>{{d.deaths}}</td>

<td>{{d.house\_destroyed}}</td>

</tr>

{% endfor %}

</table>

</div>

<div class=‖w3-container w3-black w3-center w3-opacity w3-padding-50‖>

<h1 class=‖w3-margin w3-xlarge‖>Thanks for visiting the website</h1>

</div>

<!—Footer



<footer class=‖w3-container w3-padding-40 w3-center w3-opacity‖>

<div class=‖w3-xlarge w3-padding-20‖>

<h1>A Database project </h1>

</footer>

<script>

// Used to toggle the menu on small screens when clicking on the menu buttonFunctionmyFunction() {

Var x = document.getElementById(―navDemo‖);

If (x.className.indexOf(―w3-show‖) == -1) { x.className += ― w3-show‖;

} else {

x.className = x.className.replace(― w3-show‖, ―‖);

}

}

</script>

</body>

</html>

## Links to find files, documents and result related to this project,

**GitHub:** [**https://github.com/IBM-EPBL/IBM-Project-50748-1660923078**](https://github.com/IBM-EPBL/IBM-Project-50748-1660923078)

## Project Demo Link:

## Drive: <https://drive.google.com/file/d/15y99hH-LAq_bcmFg3Sh6MXsPqq5GsOKJ/view?usp=sharing>