# NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Project Report

*Submitted by*

**TEAM ID:** PNT2022TMID28524

|  |  |
| --- | --- |
| **TEAM LEADER:** PAVITHRA V | 312819104055 |
| **TEAM MEMBER:**MATHANKUMAR S | 312819104047 |
| **TEAM MEMBER:**LOGESH R | 312819104041 |
| **TEAM MEMBER:**LOGESHWARAN S | 312819104042 |

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **CHAPTER NO** | **TITLE** |
| 1. | **INTRODUCTION**   * 1. PROJECT OVERVIEW   2. PURPOSE |
| **2.** | **LITERATURE SURVEY**   * 1. EXISTING PROBLEM   2. REFERENCES   3. PROBLEM STATEMENT DEFINITION |
| **3.** | **IDEATION & PROPOSED SOLUTION**   * 1. EMPATHY MAP CANVAS   2. IDEATION & BRAINSTORMING   3. PROPOSED SOLUTION   4. PROBLEM SOLUTION FIT |
| **4.** | **REQUIREMENT ANALYSIS**   * 1. FUNCTIONAL REQUIREMENT   2. NON-FUNCTIONAL REQUIREMENTS |
| **5.** | **PROJECT DESIGN**   * 1. DATA FLOW DIAGRAMS   2. SOLUTION & TECHNICAL ARCHITECTURE   3. USER STORIES |
| **6.** | **PROJECT PLANNING & SCHEDULING**   * 1. SPRINT PLANNING & ESTIMATION   2. SPRINT DELIVERY SCHEDULE   3. REPORTS FROM JIRA |

|  |  |
| --- | --- |
| **7.** | **CODING & SOLUTIONING**   * 1. FEATURE 1   2. FEATURE 2 |
| **8.** | **TESTING**   * 1. TEST CASES   2. ACCEPTANCE TESTING |
| **9.** | **RESULTS**  9.1 PERFORMANCE METRICS |
| **10.** | **ADVANTAGES & DISADVANTAGES** |
| **11.** | **CONCLUSION** |
| **12.** | **FUTURE SCOPE** |
| **13.** | **APPENDIX**  13.1 SOURCE CODE |

**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

1.Natural Disasters Intensity Analysis and Classification Based on Multispectral Images

Using Multi-Layered Deep Convolutional Neural Network

Natural hazards pose significant risks throughout the world. They are among the deadliest

disasters. These events cause significant economic damage as well, with losses from a

large tropical cyclone impacting a developed nation approaching or, at times, exceeding

U.S. $100 billion. Risk analysis is, in broad terms, a systematic process aimed at

understanding the nature of risk in a given situation and expressing the risk together with

the underlying knowledge base. The primary focus is on artificial intelligence, machine

learning, and statistical methods. The proposed model works in two blocks: Block-I

convolutional neural network (B-I CNN), for detection and occurrence of disasters, and

Block convolutional neural network (B-II CNN), for classification of natural disaster

intensity types with different filters and parameters.

2.Tropical Cyclone Intensity Estimation Using Multidimensional Convolutional Neural Network

From Multichannel Satellite Imagery

Tropical Cyclone is a severe storm that occurs over the tropical ocean. TC intensity is one

of the key parameters for TC prediction and disaster prevention. Accurate estimation of

TC intensity is important to theoretical research studies and practical applications.

Inspired by the success of deep learning technology in various fields, recent attempts for

TC intensity estimation focus on designing effective convolutional neural network (CNN).

We design a deep learning model, called 3DAttentionTCNet, which is inspired by Alex Net.

Unlike Alex net, as the pooling layer compresses some important information resulting in

the loss of some intensity features, we remove the pooling layers. In addition, we remove

the dropout layer, the reason why we make this adjustment is that dropout regularization

technology randomly removes some neurons during the training process. It has been

confirmed that removing the dropout layer will cause negative deviations.

3.Designing Deep-Based Learning Flood Forecast Model With ConvLSTM Hybrid Algorithm

Early detection of natural disasters such as floods can greatly assist humans in reducing

the extent of the damage caused by such events. In the Fiji Islands, where this study is

focused, recent flood events resulted in major damages amounting to millions of dollars.

The loss of at least 225 lives during the 1931 flood event in Fiji was primarily due to the

unavailability of efficient flood warning systems. One simple, yet a robust mathematical

tool used to determine the flood state at a particular time for a given area is the Flood

Index (IF). A model is developed Develop multi-step predictive model using ConvLSTM, as

an objective model, with alternative methods of LSTM, CNN-LSTM and SVR that can also

determine the flood state.

4.A Conformal Regressor With Random Forests for Tropical Cyclone Intensity Estimation

Tropical Cyclone is an intense vortex system that originates over the tropical ocean and is

one of the most destructive natural disasters. TC intensity usually refers to the maximum

wind speed near the TC centre. TC intensity is an important indicator to quantify the

destruction potential. The basic idea of using satellite data to estimate the intensity is that

the cloud pattern strongly correlates with the TC Intensity in the image. It is considered an

excellent way to extract features from satellite images to estimate TC intensity. The most

common technique is the Dvorak technique. The Dvorak technique tried to estimate the

TC intensity using visible or infrared images based on the cloud structure. Various machine

learning models have also been applied to TC intensity estimation. Among them, the most

widely used was the linear regression model. A multiple linear regression (MLR) model

was constructed based on the extraction of the most significant signals and parameters

from satellite infrared images

## Problem Statement Definition

People needs a way to classify and analyse 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 analyse and detect natural disaster and protect them from such disaster.

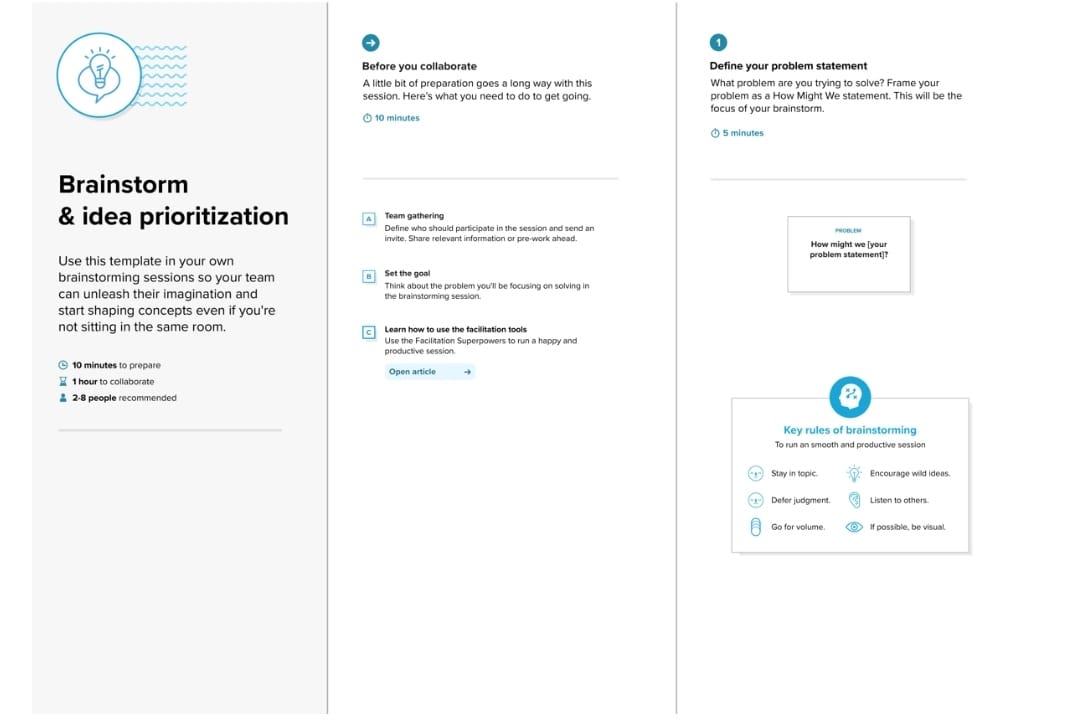
# CHAPTER 3

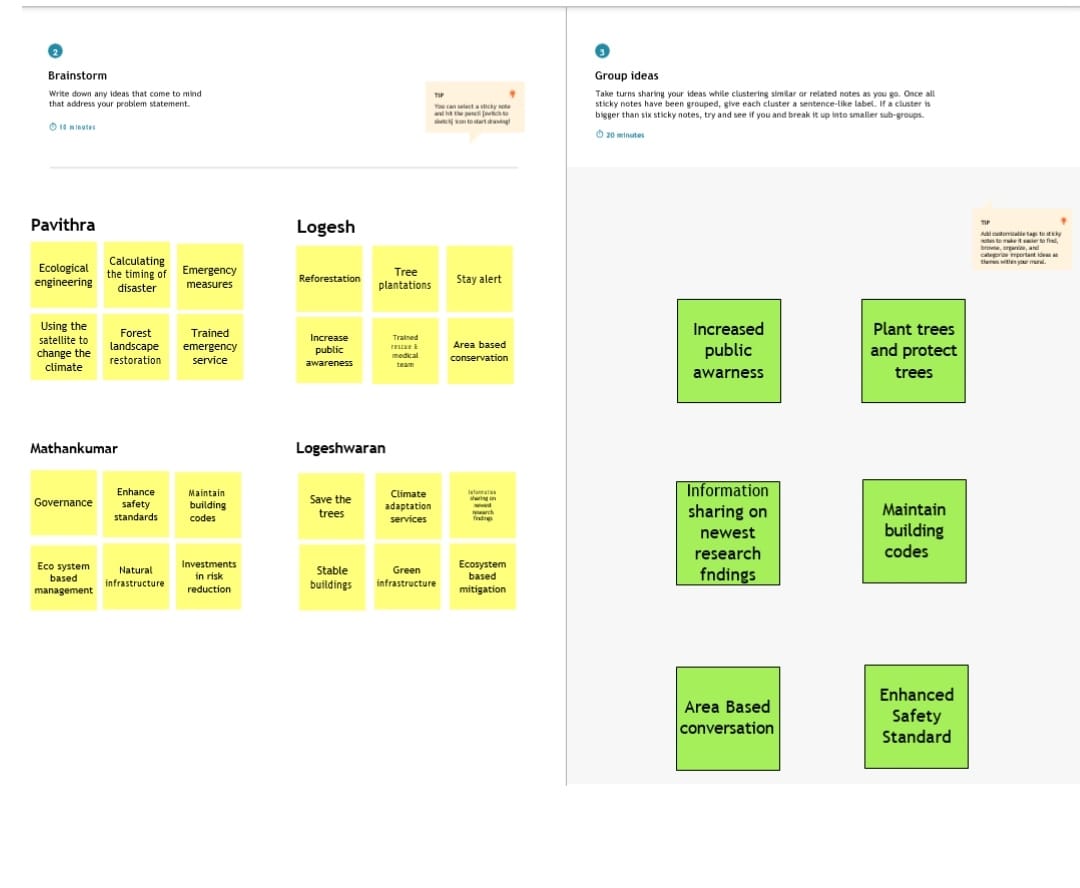
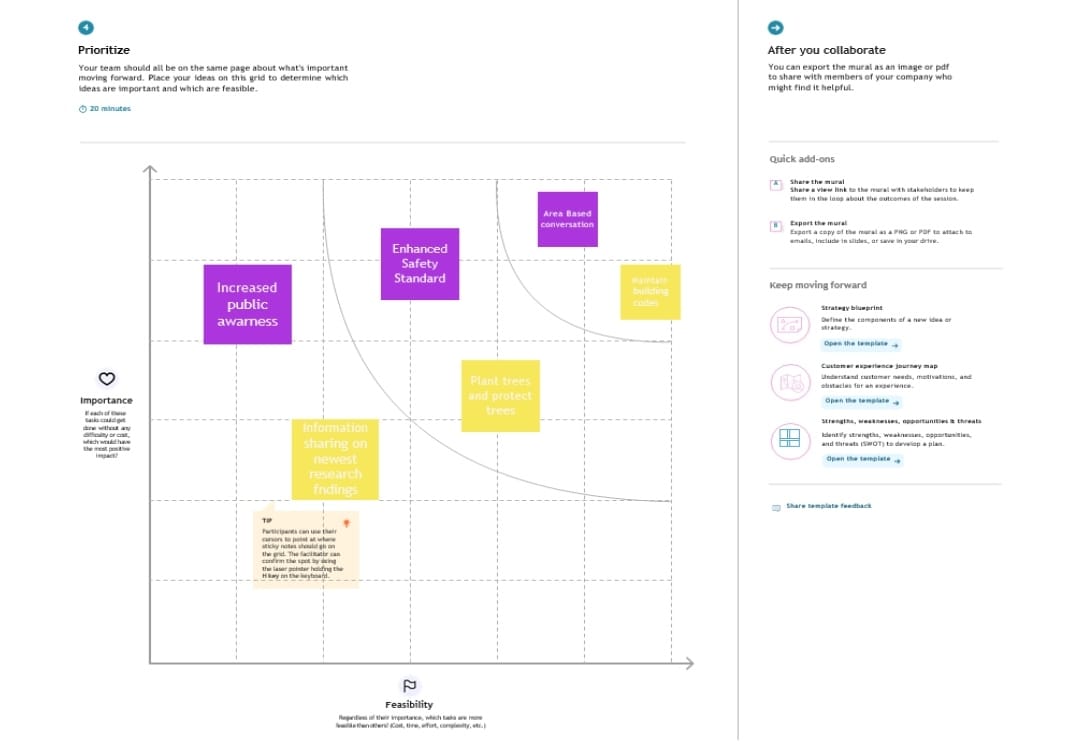
**IDEATION & PROPOSED SOLUTION**

## Empathy Map Canvas

****

* 1. **Ideation & Brainstorming**

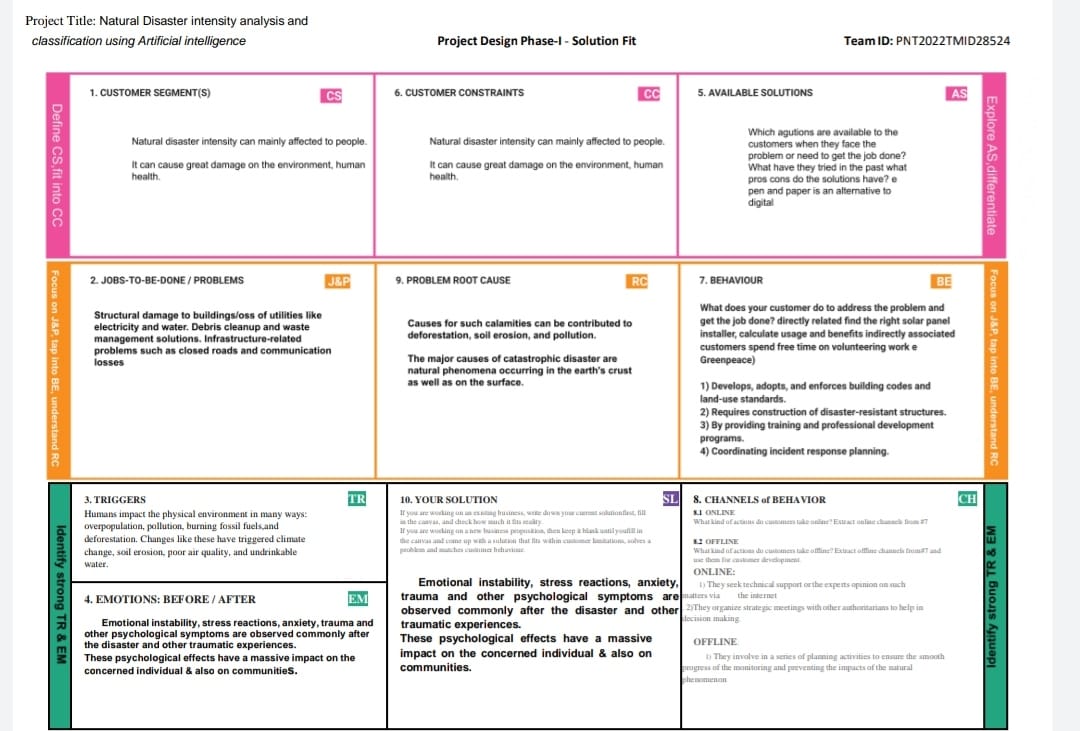
****



## Proposed solution

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | People needs a way to classify and analyse 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 isclassified. |
| **FR-3** | Accuracy | Since the training and testing images are huge,  theaccuracy is higher. |
| **FR-4** | Speed | The generation of results from the input  imagesare faster. |
| **FR-5** | Resolution | The resolution of the integrated web camerashould be high enoughto capture the video  frames. |
| **FR-6** | User Interface | Maximizing the interaction in Web  DesigningService. |

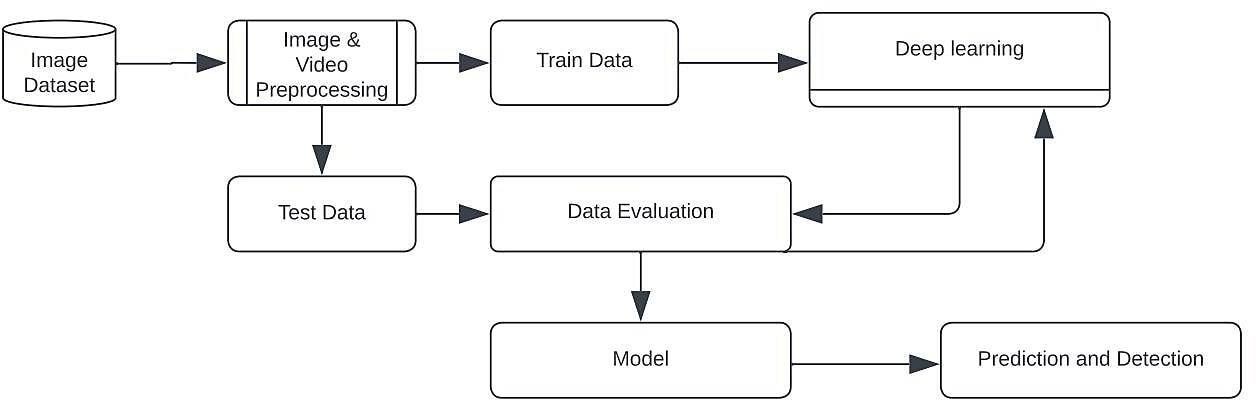
* 1. **Non-Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **NFR. No.** | **Non- Functional**  **Requirement** | **Description** |
| **NFR-1** | Usability | User friendly and classify the disaster easily. |
| **NFR-2** | Security | The modelis secure due to the cloud  deploymentmodels and also thereis no login issue. |
| **NFR-3** | Reliability | Accurate prediction of the natural disaster  and thewebsite can also be fault tolerant. |
| **NFR-4** | Performance | It is shown thatthe model givesalmost 95  percentaccuracy 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 Googlechrome, Microsoft edge and also it can be  extended to the NDRFand customers. |

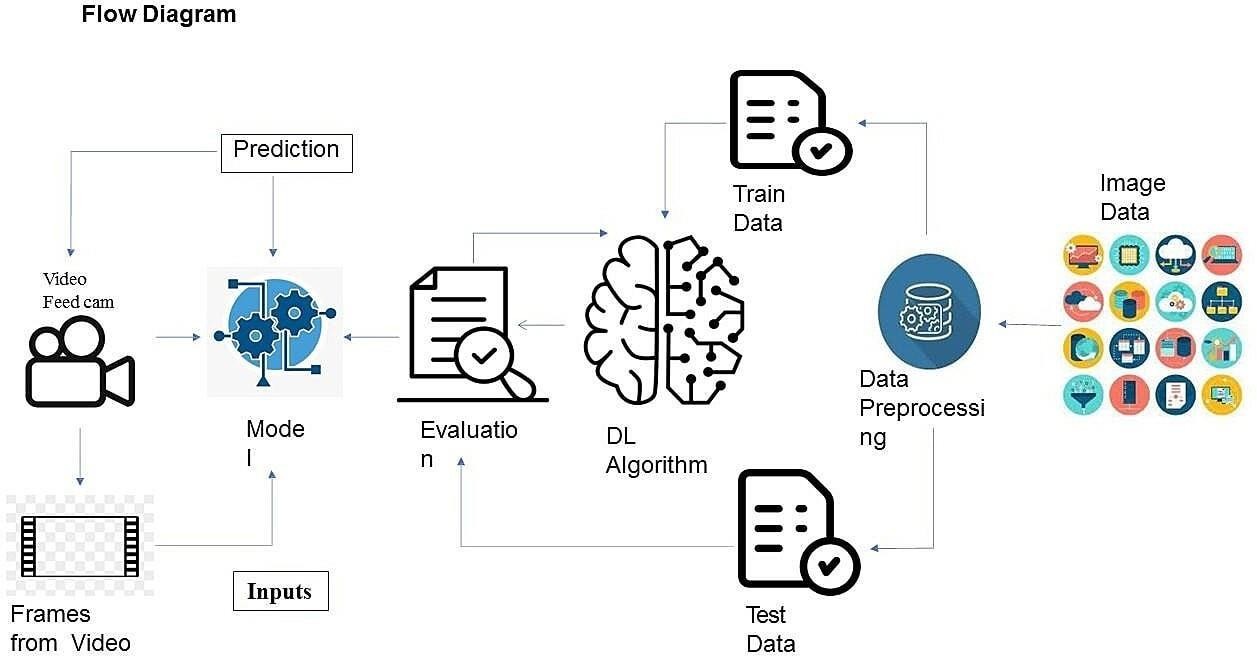
# CHAPTER 5 PROJECT DESIGN

## 5.2 Data Flow Diagrams

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



## Flow Diagram

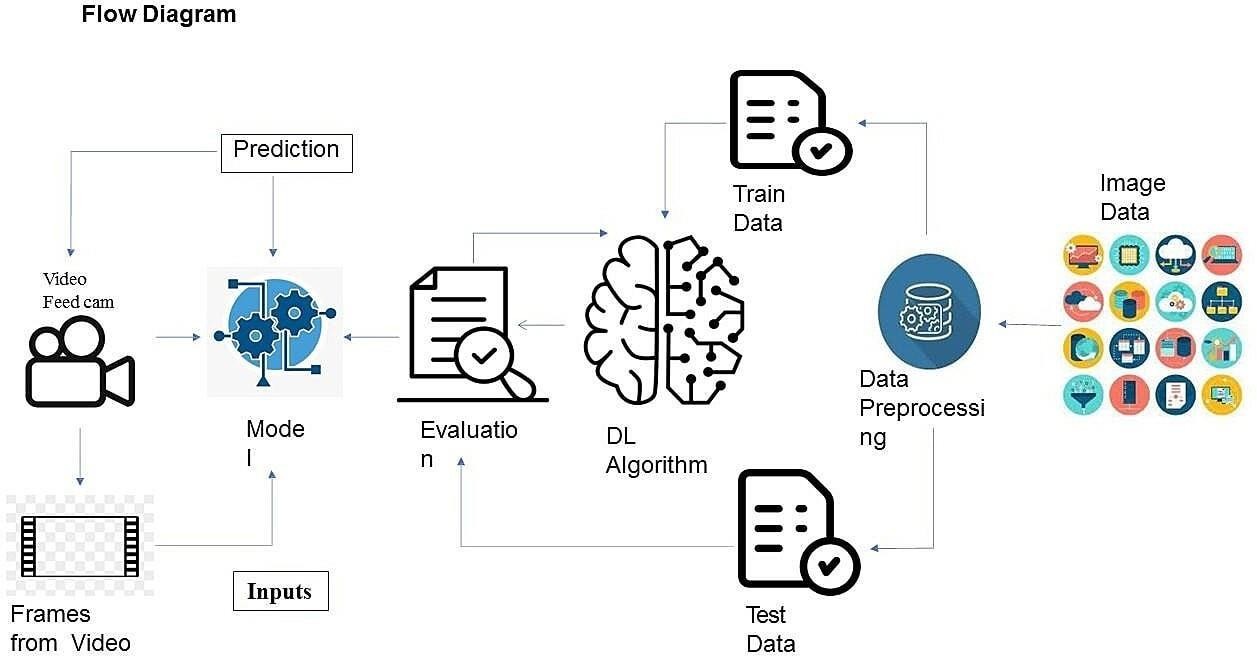


* 1. **Solution & Technical Architecture Solution Architecture**

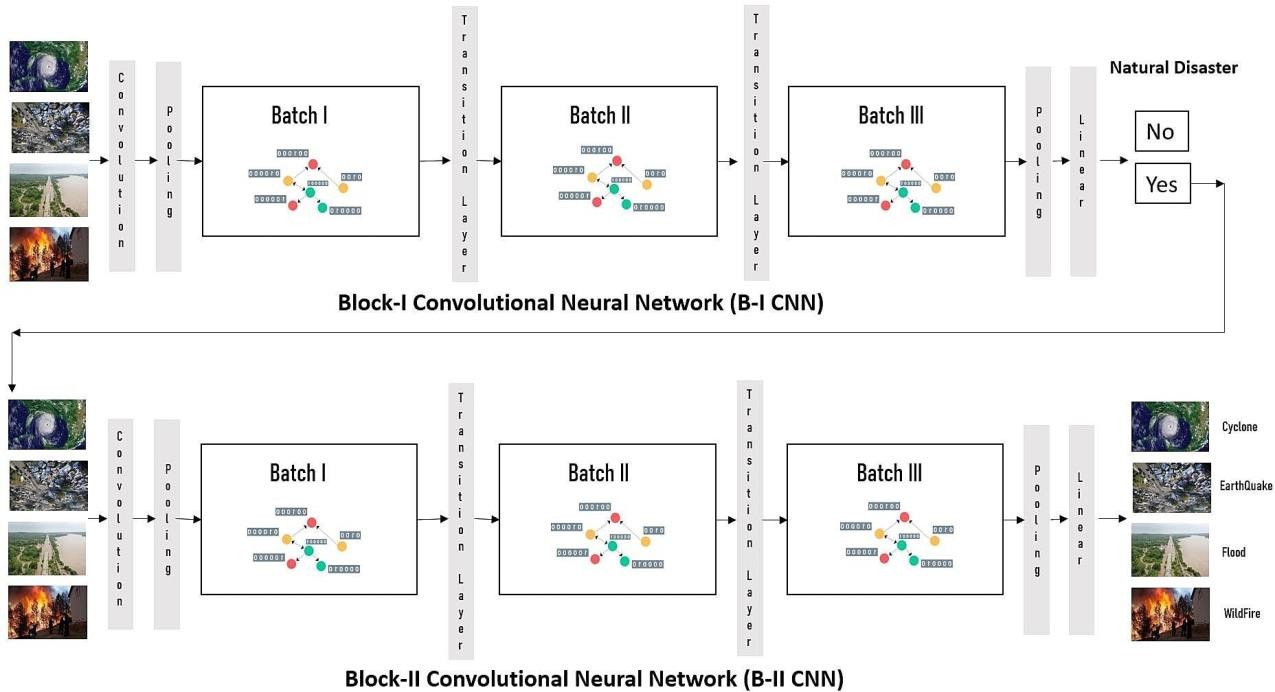
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

* + - 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 classifywhich happened just  before. | HTML, CSS,  JavaScript, Django, Python. |
| 3. | Disaster | This functionis used to detect, | Decision |
| Detection | outcomesfrom | trees,Regression, |
|  | the new trained data to performnew | Convolutio |
|  | tasks andsolve new problems. | nal 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 interactwith our model and give it problems  to solve.Usually this takesthe form of  an API, auserinterface, or a command- line interface. | Application programming interface, etc. |

|  |  |  |  |
| --- | --- | --- | --- |
| 6. | Data collectionunit | Data is only usefulif it’s accessible, so it needsto be stored ideally in a consistent structureand conveniently in one place. | IBM Cloud,SQL Server. |
| 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 keepsour models secureand makes sureonly those who have  permission can use them. | Encryption and Decryption  (OTP). |

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | Application interface | User uses mobile application and web applicationto interact withmodel | Web Develop ment (HTML,C SS) |
| 4. | Availability (both Online and Offline work) | Its include both online and offline work. As goodinternet connection is need for online work to explore the software perfectly. Offline work includes thesaved data to explore for  later time. | Caching, backend server. |
| 5. | Regular Updates | Thetruly excellent software product needs a continuous processof improvements and updates. Maintain your server andmake sure thatyour 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, colour themes, etc. whichmake 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 touse 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 analyse 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 visualise 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:**

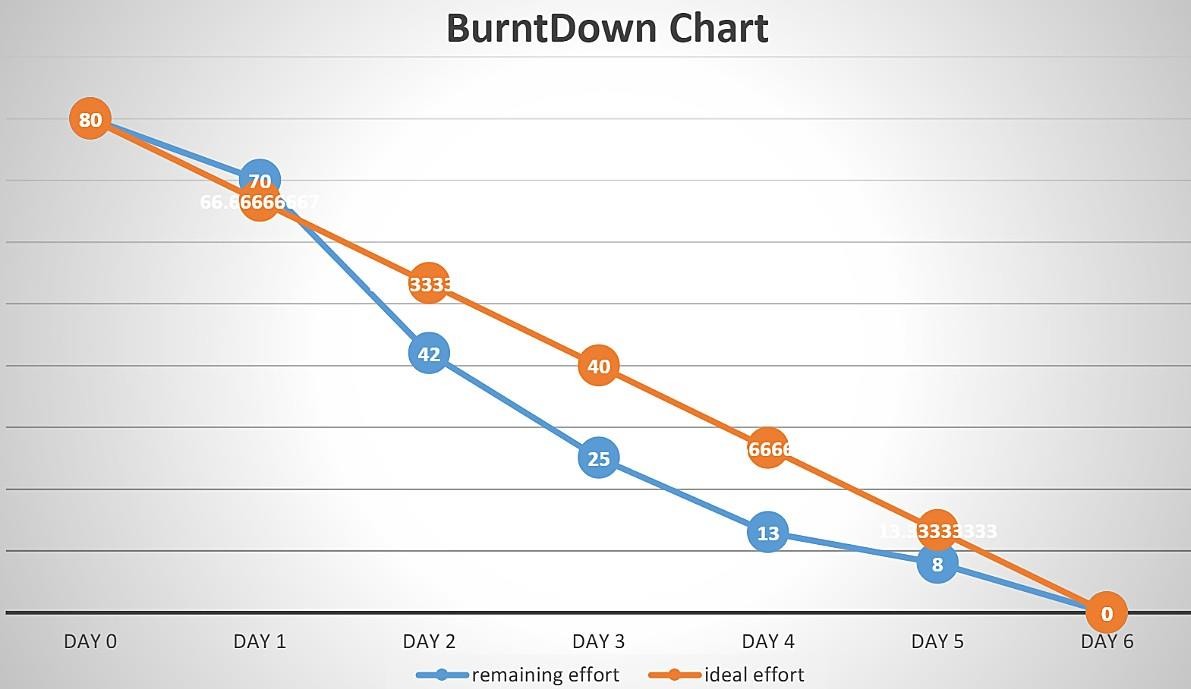
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

## 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 elementsin Intro page | 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 outputwindow | 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.

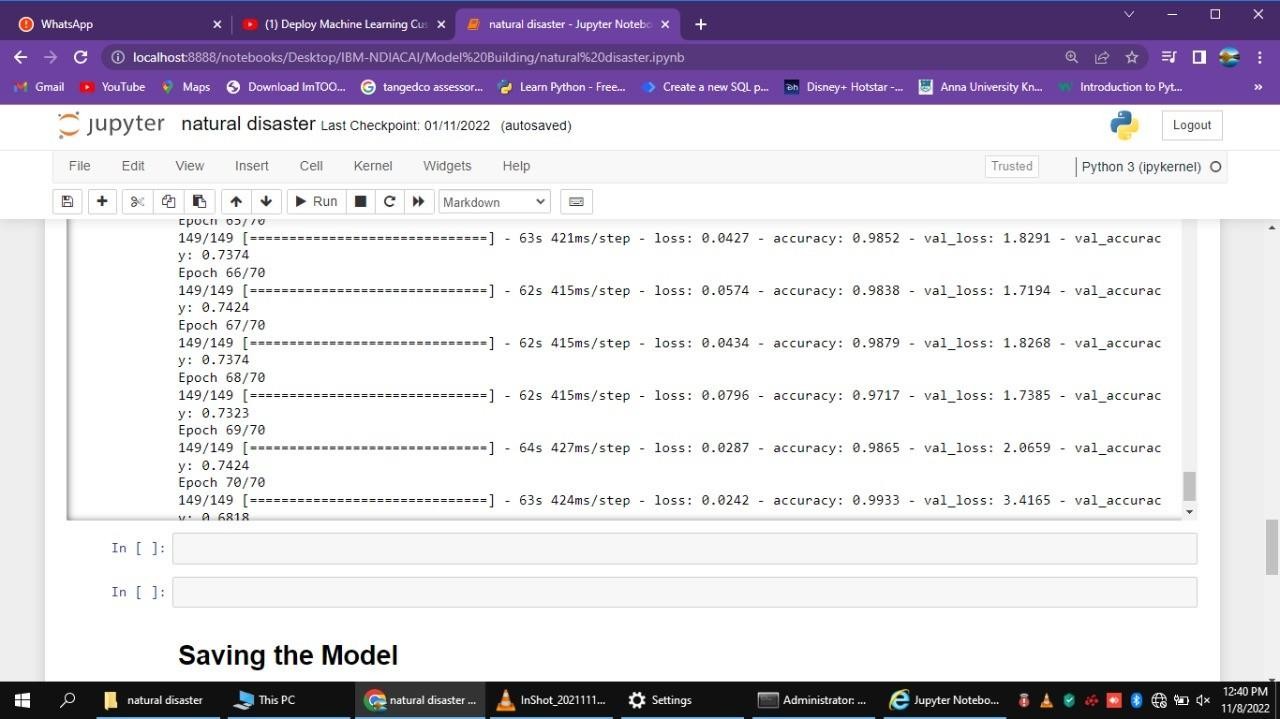
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Test Cases** | **Not Tested** | **Fail** | **Pass** |
| Home Page | 4 | 0 | 0 | 4 |
| Intro Page | 4 | 0 | 0 | 4 |
| Open Webcam | 3 | 0 | 0 | 3 |

# CHAPTER 9 RESULTS

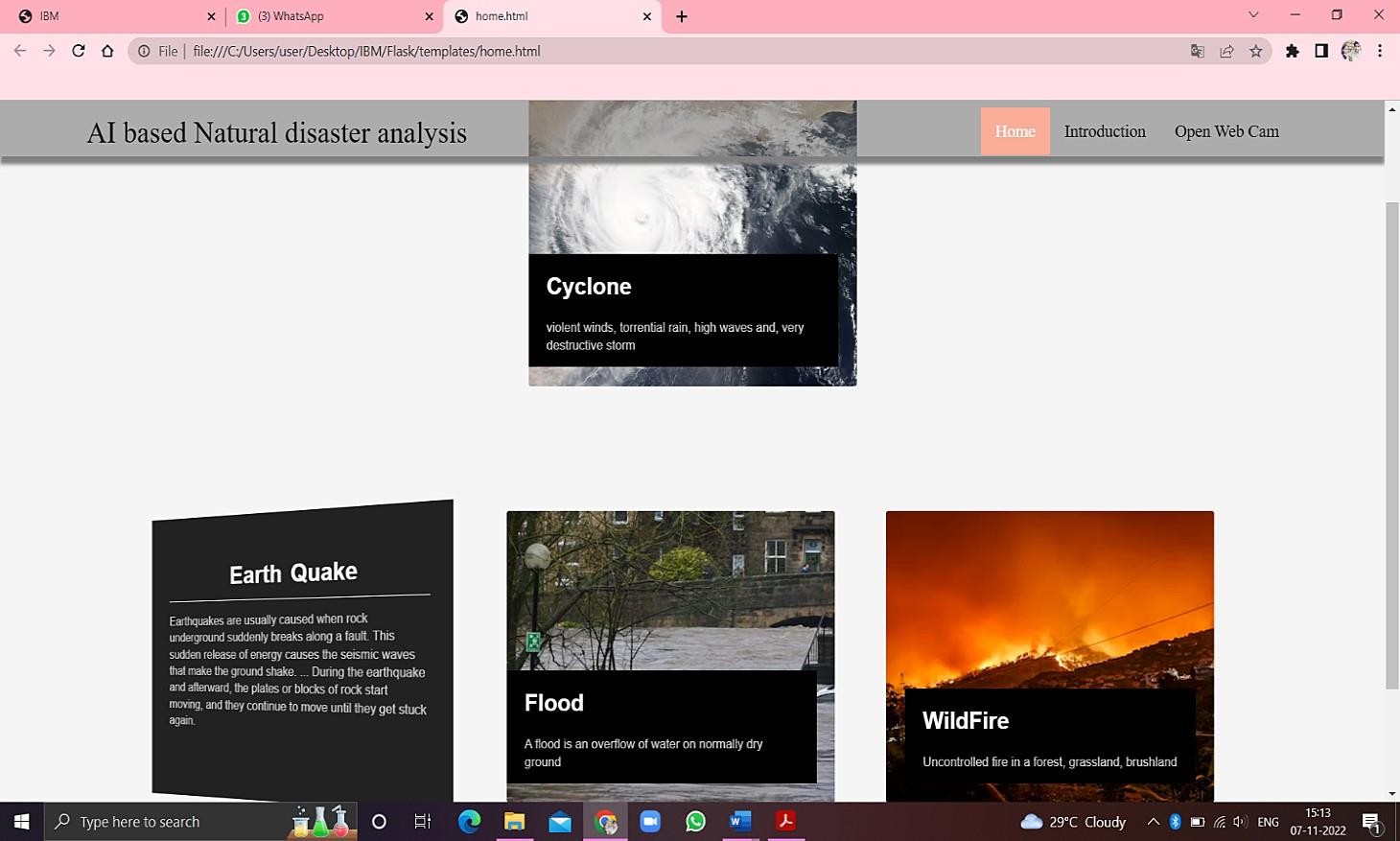
## Performance Metrics

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 testedwiththe data which is separate from the trained data and has predicted the data well.

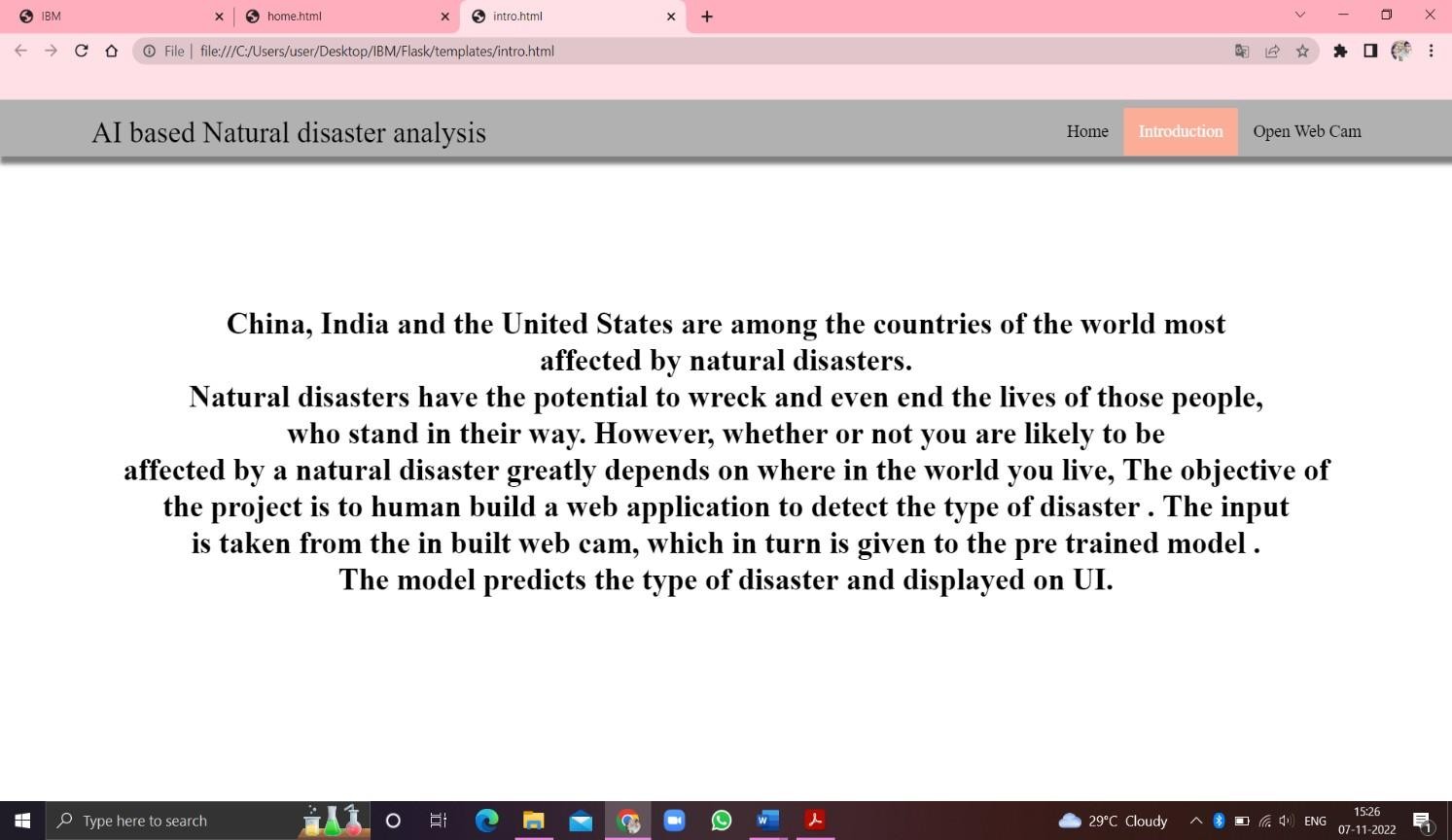
**Output of application**



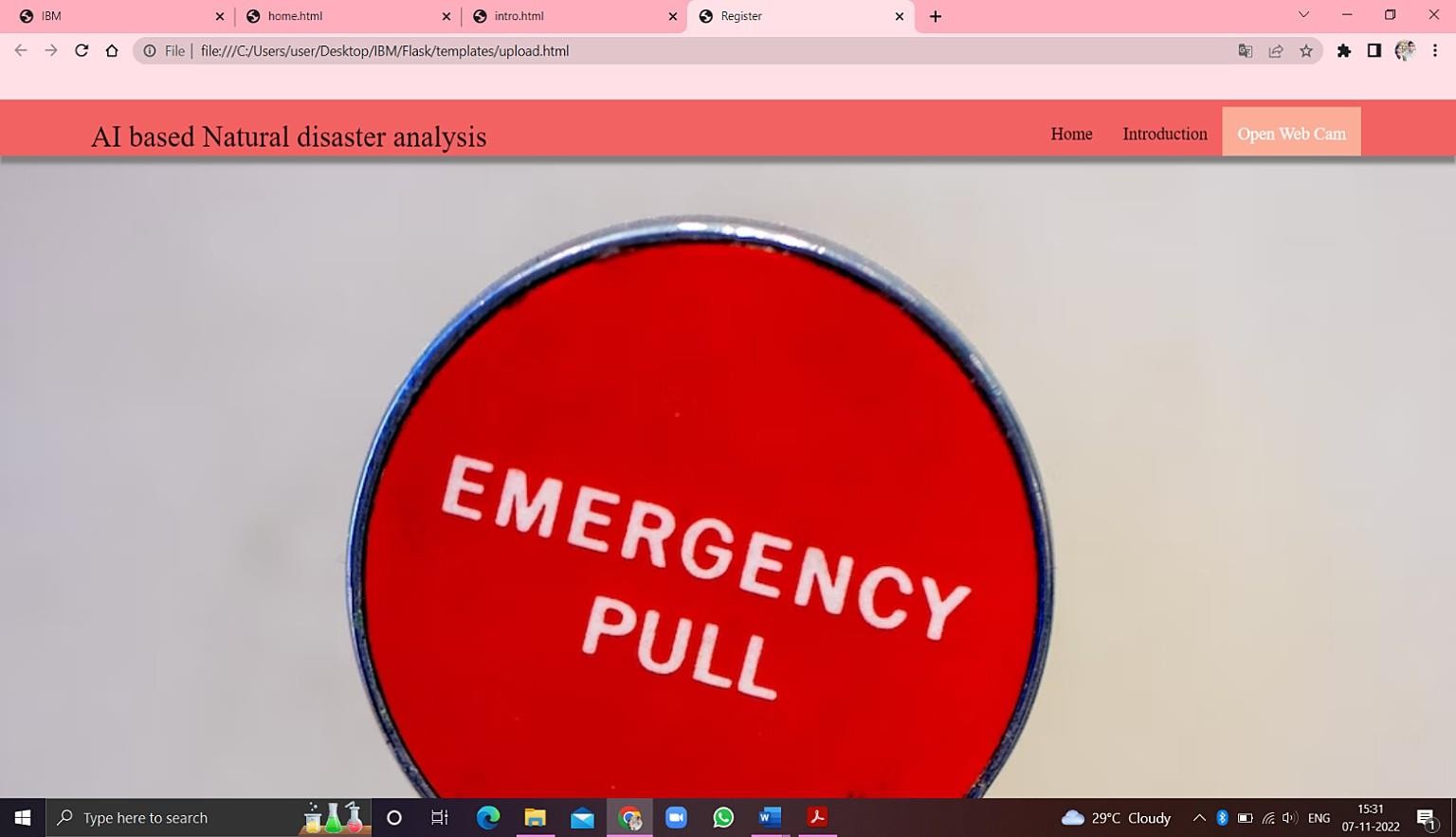
# HOME PAGE



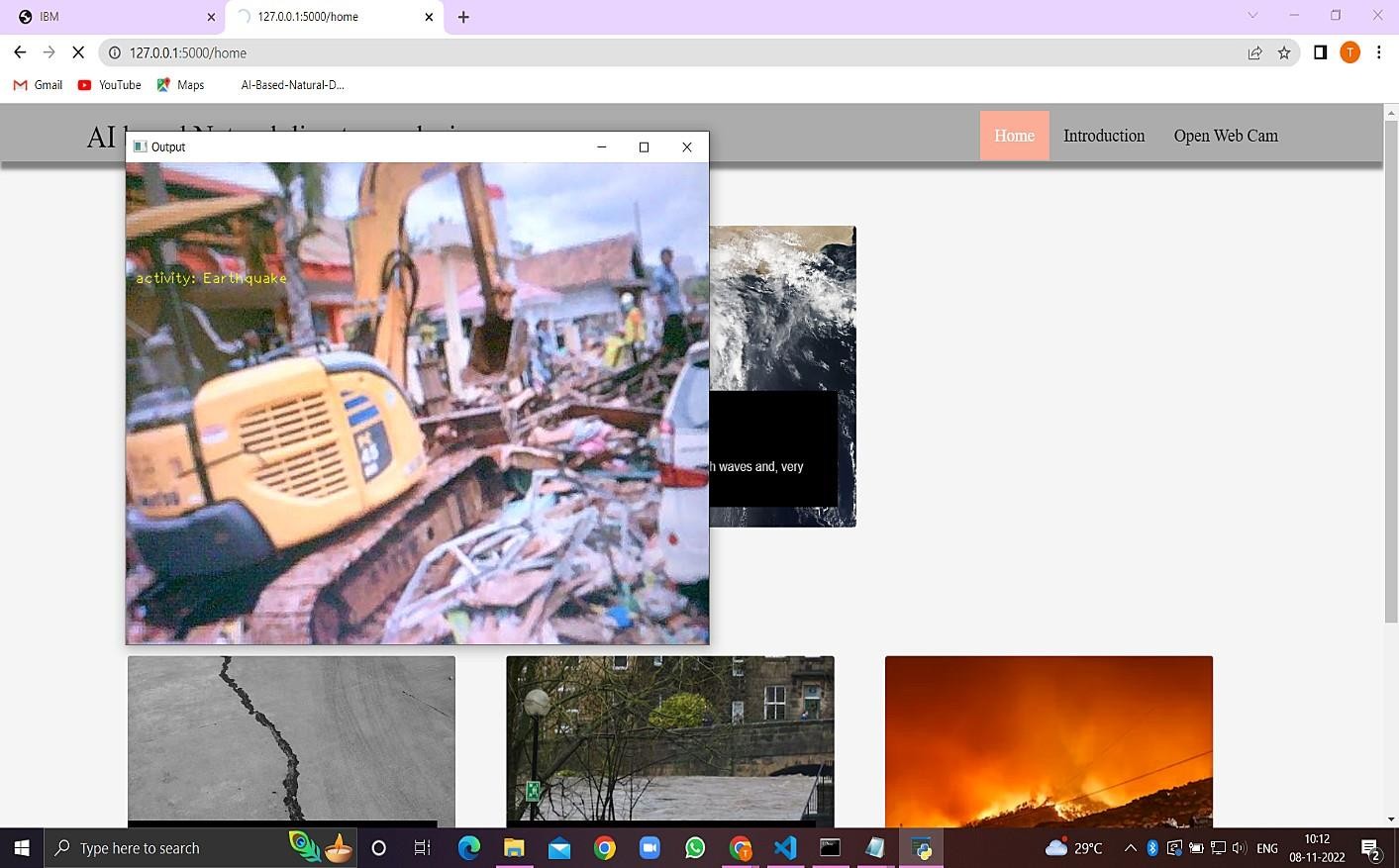
## INTRODUCTION PAGE

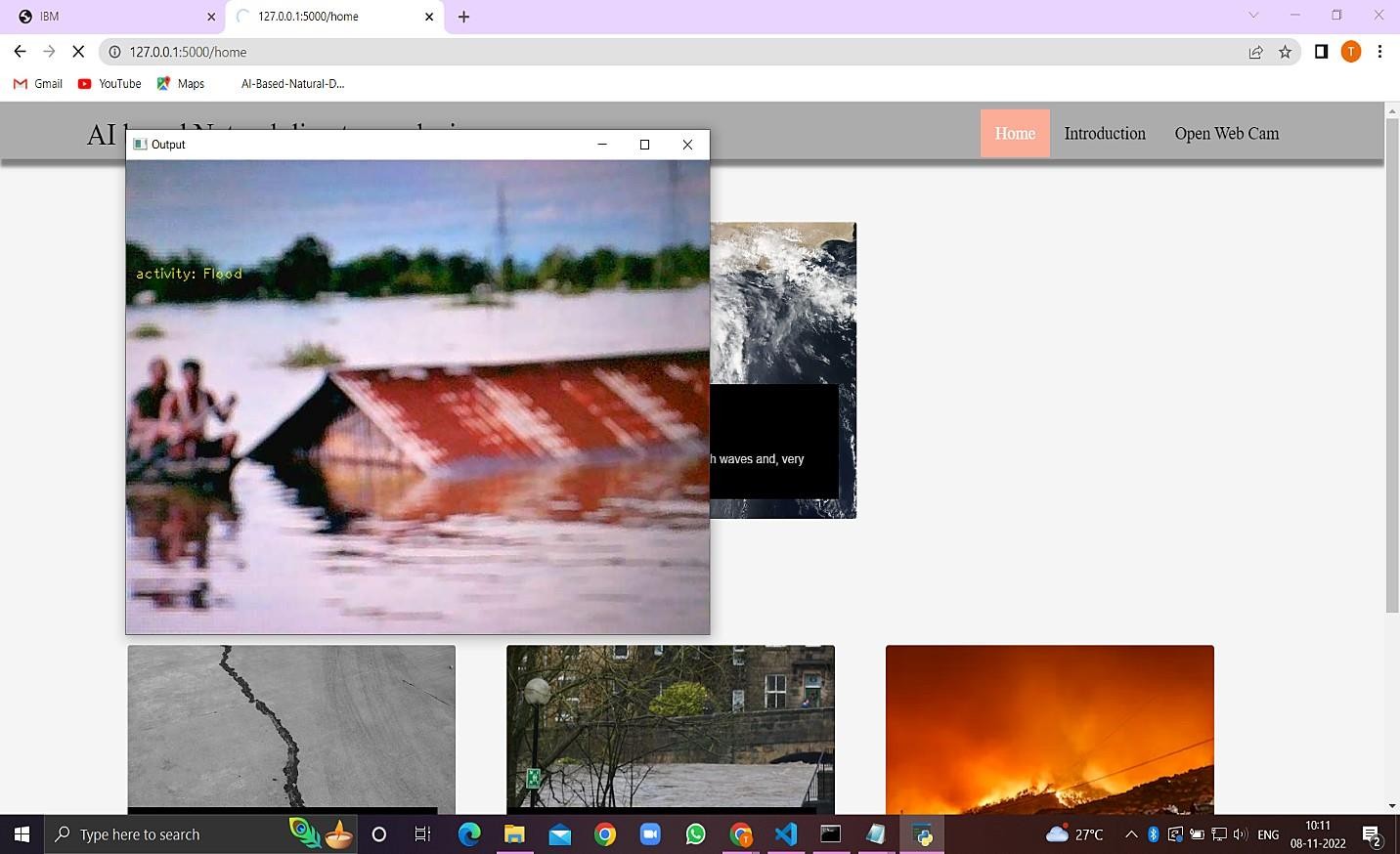


**WEB CAM**



**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