

NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION USING **ARTIFICIAL INTELLIGENCE**

Project Report

Submitted by

Team leader

DHIYANESHWARI S

Team members

MARIYA DAYANA A

SIVANESAN A

YOKESH T

Faculty Mentor

GOWRI V

TEAM ID : PNT2022TMID45046

TABLE OF CONTENTS

CHAPTER NO	TITLE
1.	INTRODUCTION PROJECT OVERVIEW PURPOSE
2.	LITERATURE SURVEY EXISTING PROBLEM REFERENCES PROBLEM STATEMENT DEFINITION
3.	IDEATION & PROPOSED SOLUTION EMPATHY MAP CANVAS IDEATION & BRAINSTORMINGPROPOSED SOLUTION PROBLEM SOLUTION FIT
4.	REQUIREMENT ANALYSIS FUNCTIONAL REQUIREMENT NON-FUNCTIONAL REQUIREMENTS
5.	PROJECT DESIGN DATA FLOW DIAGRAMS SOLUTION & TECHNICAL ARCHITECTUREUSER STORIES
6.	PROJECT PLANNING & SCHEDULING SPRINT PLANNING & ESTIMATIONSPRINT DELIVERY SCHEDULE REPORTS FROM JIRA
7.	CODING & SOLUTIONING FEATURE 1 FEATURE 2

8.	TESTING TEST CASES 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

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 non-damaged 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 patterns 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 are 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-

- (i) Prevention of danger or threat of any disaster
- (ii) Preparedness to deal with any disaster
- (iii) Prompt response to any threatening disaster situation or disaster
- (iv) Assessing the severity or magnitude of effects of any disaster
- (v) Evacuation, rescue, and relief
- (vi) 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 Studyof the Liguria Region in Italy. *Geosciences* 2020, 10, 105. [CrossRef]
3. 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]
4. 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]
5. 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:

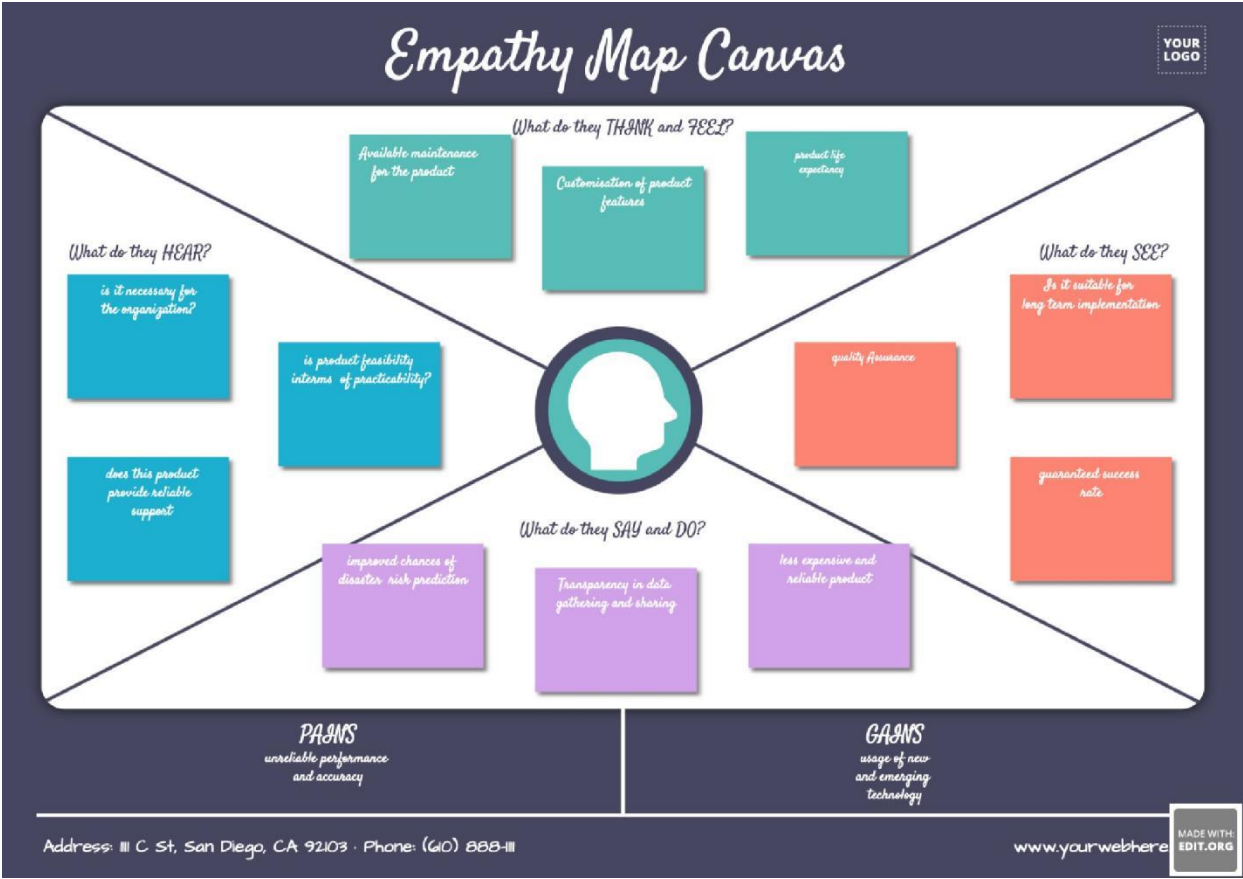
People need a way to classify and analyse the natural disaster so that they can prevent themselves from losses due to the disaster and millions of lives.

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:



Ideation & Brainstorming

The image displays a collection of digital templates for collaborative brainstorming and idea prioritization. The templates are organized into several sections:

- Before you collaborate:** Includes instructions for group preparation and a time allocation of 10 minutes.
- Define your problem statement:** Provides a space for writing the problem statement and a time allocation of 5 minutes.
- Brainstorm:** Features a grid for brainstorming ideas, with a time allocation of 10 minutes.
- Group ideas:** Includes a section for grouping ideas and a time allocation of 10 minutes.
- Prioritize:** Features a 2x2 matrix for prioritizing ideas based on importance and feasibility, with a time allocation of 10 minutes.

The templates are designed to be used in a collaborative environment, with instructions for group activities and time allocations. The 'Prioritize' section includes a 2x2 matrix with axes labeled 'Importance' and 'Feasibility'. The matrix is divided into four quadrants: 'High Importance / Low Feasibility', 'High Importance / High Feasibility', 'Low Importance / Low Feasibility', and 'Low Importance / High Feasibility'. A curved line separates the 'High Importance / High Feasibility' quadrant from the others, indicating the most promising ideas.

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 (pretrained) 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 robustnes

CHAPTER 4

REQUIREMENT ANALYSIS

Functional Requirement:

FR No.	Functional Requirement(Epic)	Functional Requirement(Epic)
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 capture the video frames
FR-6	User Interface	Maximizing the interaction in Web Designing Service.

Non-Functional Requirement:

NFR. No.	NonFunctional 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 Googlechrome, Microsoft edge and also it can be extended to the NDRF and customers.

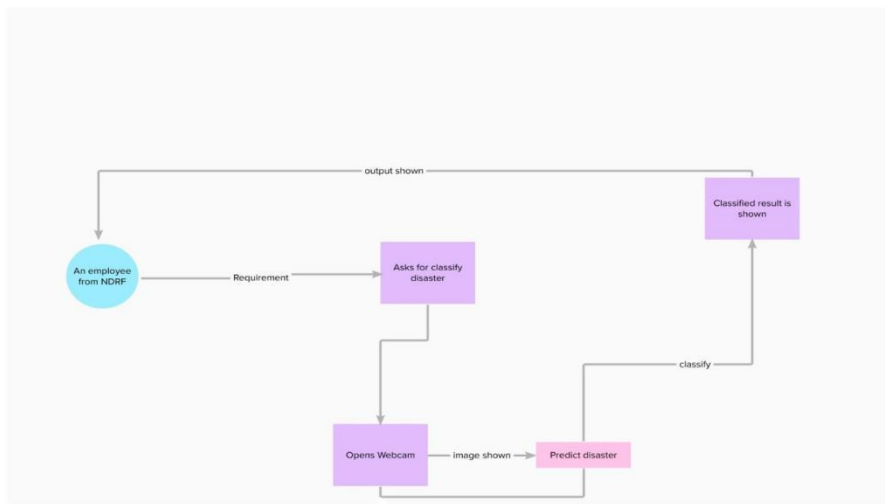
CHAPTER 5

PROJECTDESIGN

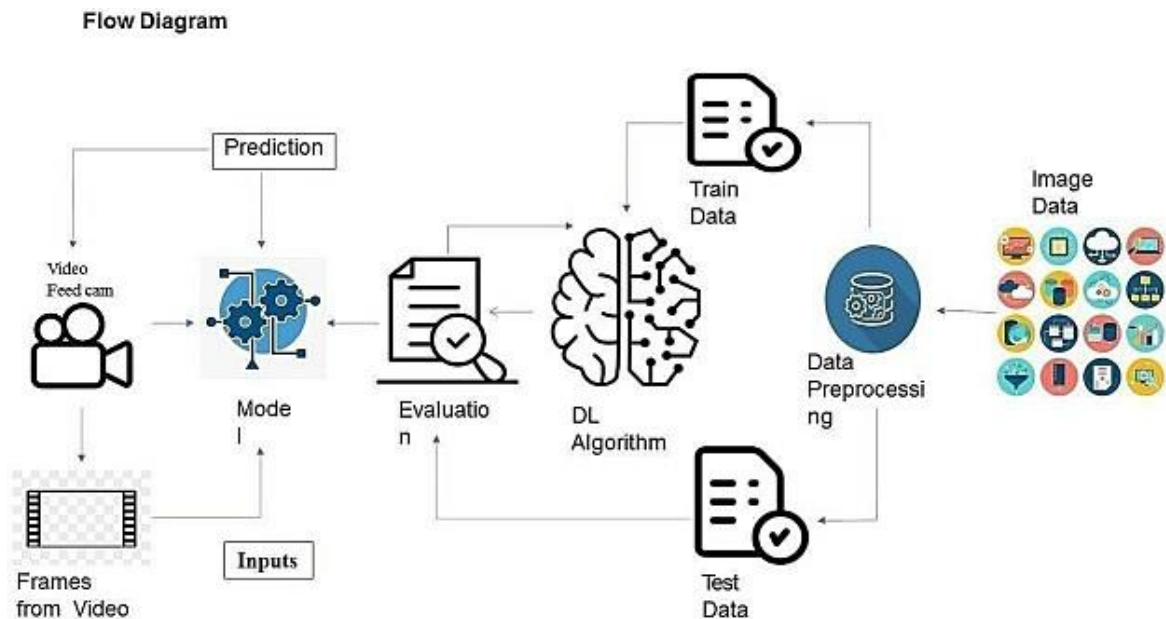
Data Flow Diagrams:

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.

Data Flow Diagram for “Natural Disasters Intensity Analysis and Classification using Artificial Intelligence”:



Flow Diagram:



Solution & Technical Architecture:

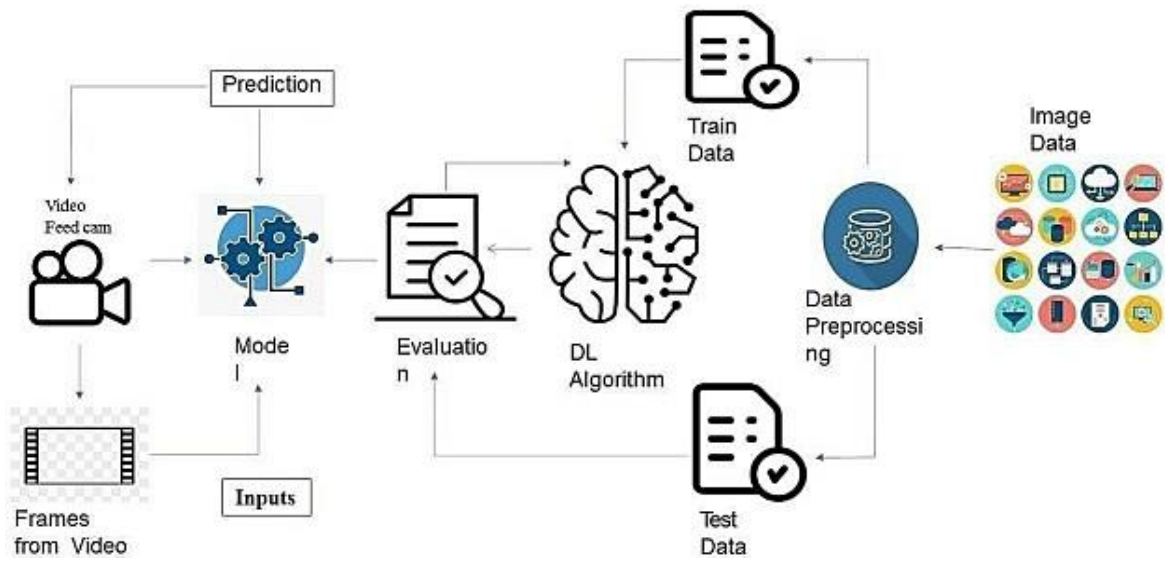
Solution Architecture:

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:

Flow Diagram



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.
2.	Disaster Detection	This function is used to detect, Decision Outcomes, the new trained data to perform tasks and solve new problems.	Decision trees, Regression, Convolutional Neural networks
3.	Evaluation system	It monitors that how Algorithm performs on data as well as during training.	Chi-Square, Confusion Matrix, etc.
4.	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
5.	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.
6.	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 Development (HTML, CSS)
4.	Availability (both Online and Offline work)	It includes both online and offline work. As good internet connection is needed 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, colour themes, etc. which make a software interface look good and functional.	CSS

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 can collect the dataset for monitoring and analyzing	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 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-10	As a user, I can use Flask app to use model easily through web app.	10
Sprint-4	Model deployment	USN-11	As an administrator, I can deploy the AI model in IBM Cloud.	10

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)

$$\text{Average velocity} = \text{Sprint duration} / \text{velocity}$$

$$= 20/6$$

$$= 3$$

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 methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CHAPTER 7

CODING & SOLUTIONING

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 Open webcam button	Working as expected	Pass
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 Intro page	Application should show below UI elements: Home page Intro page Open webcam button	Working as expected	Pass
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

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	3	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

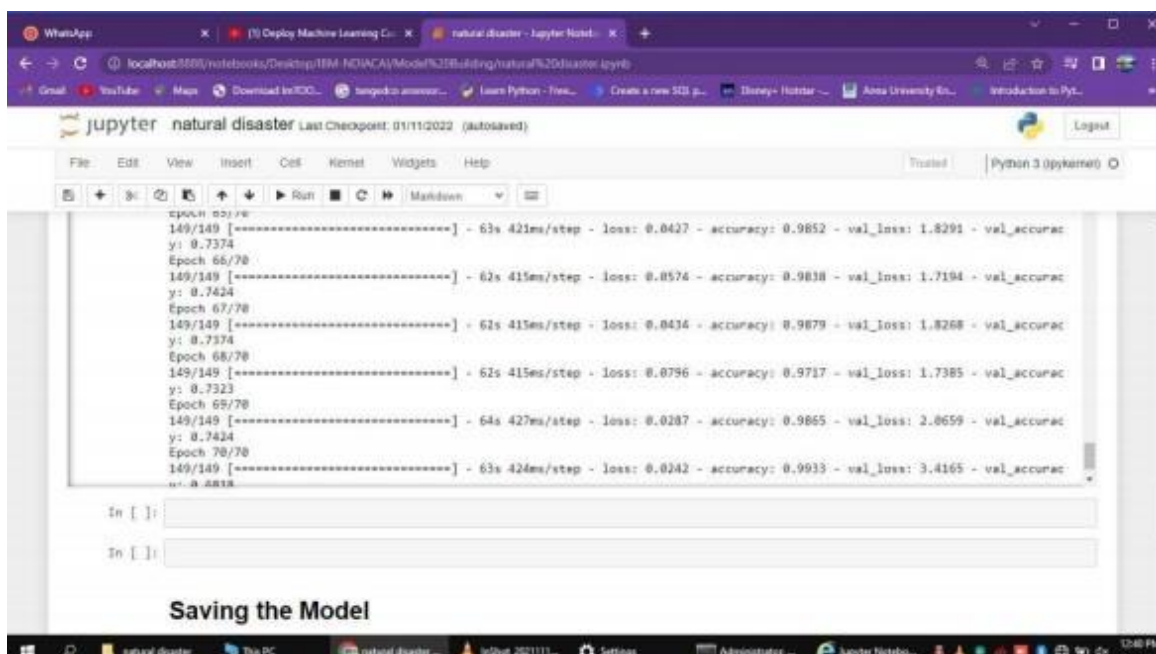
CHAPTER 9

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:



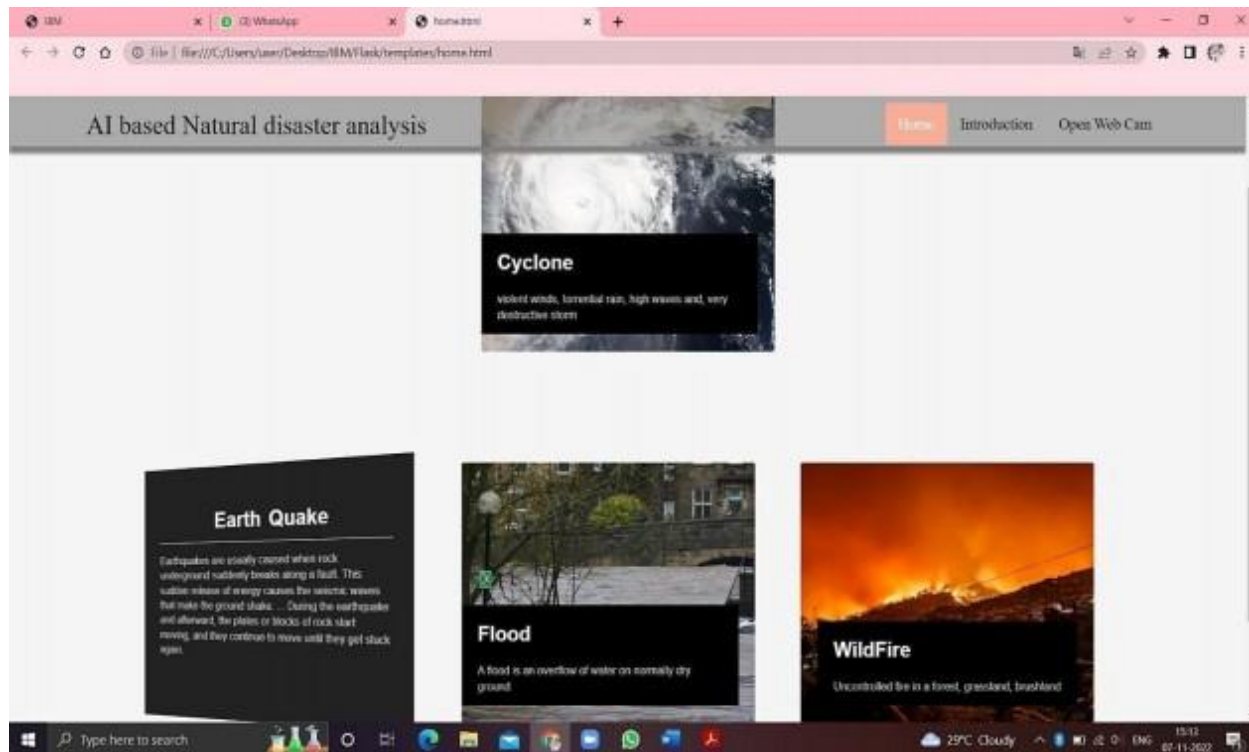
The screenshot shows a Jupyter Notebook interface with a terminal output window displaying training metrics for epochs 66 through 70. The metrics include training loss, accuracy, validation loss, and validation accuracy. The accuracy is consistently high, reaching approximately 99.33% by epoch 70. Below the terminal output, there are two empty input fields for code execution, and a button labeled 'Saving the Model'.

```
Epoch 66/70  
149/149 [=====] - 63s 421ms/step - loss: 0.0427 - accuracy: 0.9852 - val_loss: 1.8291 - val_accuracy: 0.9734  
Epoch 67/70  
149/149 [=====] - 62s 415ms/step - loss: 0.0574 - accuracy: 0.9838 - val_loss: 1.7194 - val_accuracy: 0.97424  
Epoch 68/70  
149/149 [=====] - 62s 415ms/step - loss: 0.0434 - accuracy: 0.9879 - val_loss: 1.8268 - val_accuracy: 0.97374  
Epoch 69/70  
149/149 [=====] - 62s 415ms/step - loss: 0.0796 - accuracy: 0.9717 - val_loss: 1.7385 - val_accuracy: 0.97323  
Epoch 70/70  
149/149 [=====] - 64s 427ms/step - loss: 0.0287 - accuracy: 0.9865 - val_loss: 2.0659 - val_accuracy: 0.97424  
149/149 [=====] - 63s 424ms/step - loss: 0.0242 - accuracy: 0.9933 - val_loss: 3.4165 - val_accuracy: 0.99333
```

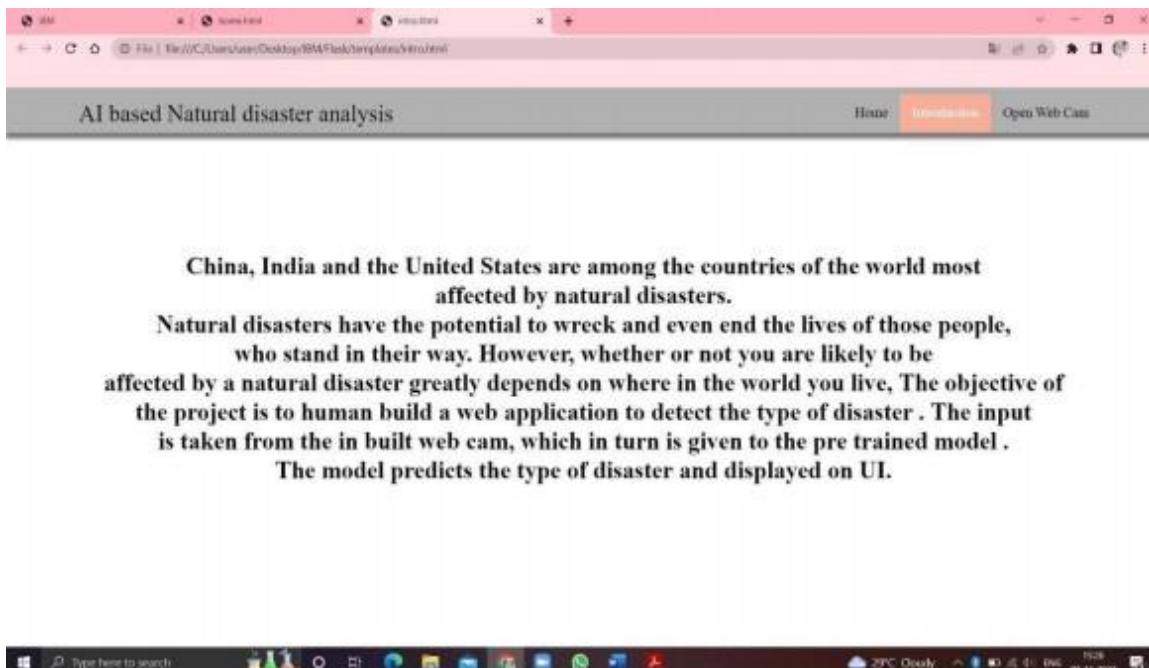
fn []:
fn []:

Saving the Model

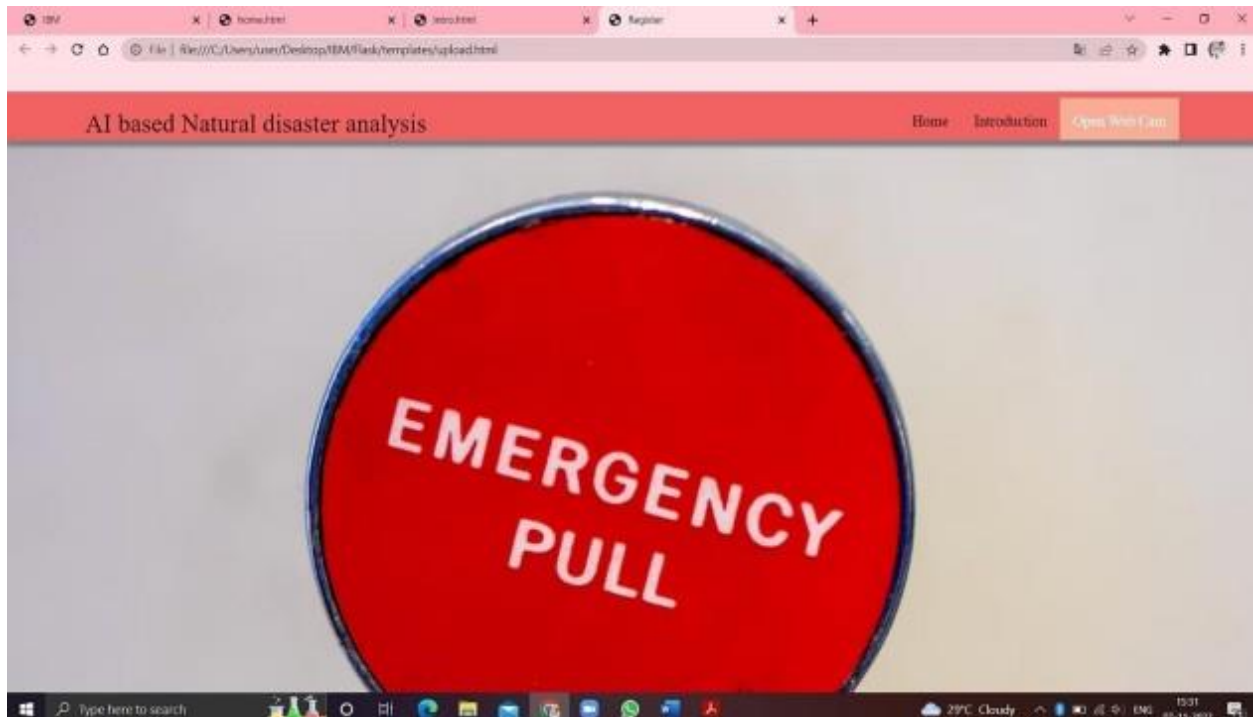
HOME PAGE:



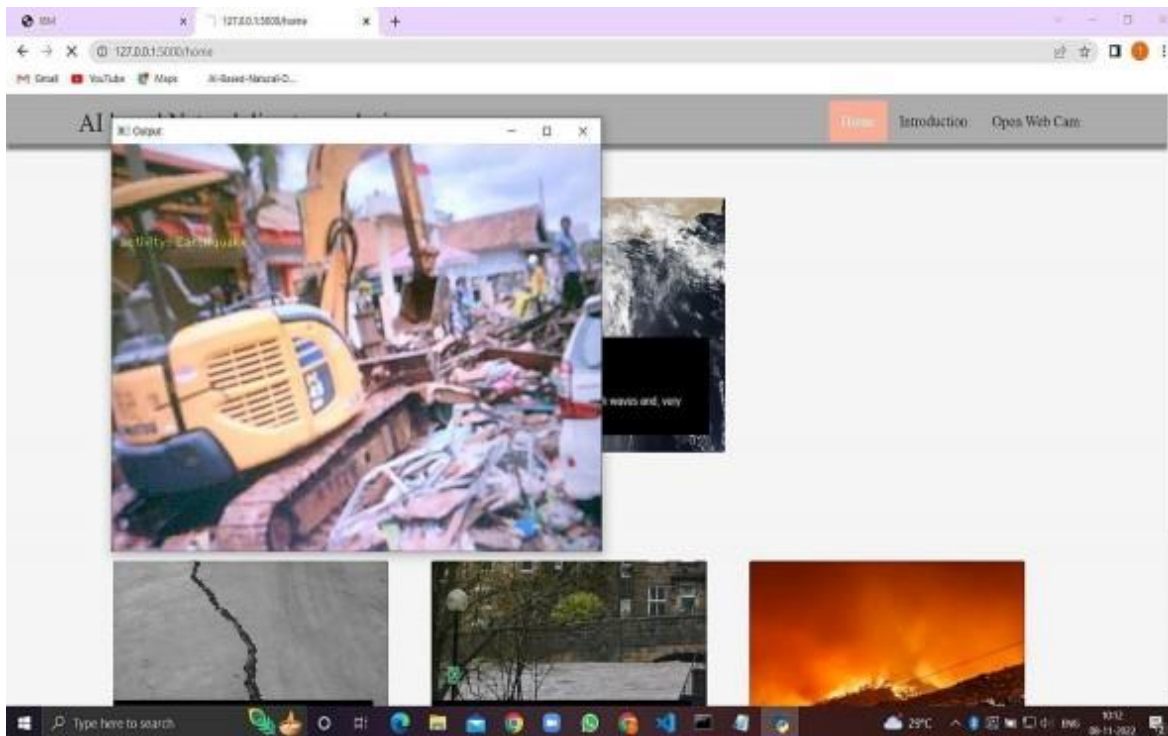
INTRODUCTION PAGE

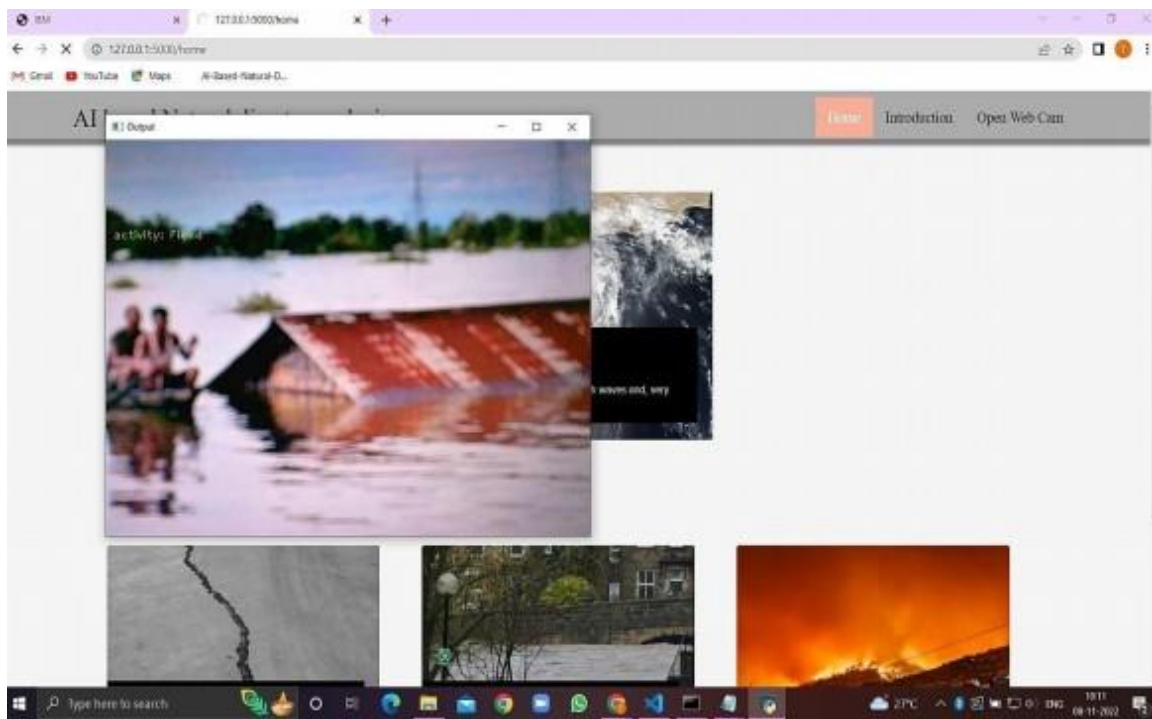


WEB CAM:



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
```

```
#Flatten-used for 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 Augmentation

```
#setting parameter for Image Data augmentation 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\rajeshwari\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\rajeshwari\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 pooling

```
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), act
```

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 testset
```

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 tensorflow.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\Cyclone\921.jpg",grayscale=False,target_size= (64,64)) #loading of the image  
  
x = image.img_to_array(img)#image to array  
  
x = np.expand_dims(x,axis = 0)#changing the shape  
  
pred = model.predict_classes(x)#predicting the classes  
  
pred  
  
index=['Cyclone','Earthquake','Flood','Wildfire']
```

```
result=str(index[pred[0]]) result
```

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

GitHub: <https://github.com/IBM-EPBL/IBM-Project-47730-1660801833>