PROJECT REPORT

NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Submitted by

TEAM ID:PNT2022TMID42615

SREEPOORNA S IYER - 711119104063
BENISHA M - 711119104715
SUWETHAN M - 711119104068
JOTHI S - 711119104705

In partial fulfilment for the award of the degree
Of

BACHELOR OF ENGINEERING
In

ELECTRONICS AND COMMUNICATION ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING

TIRUNELVELI - 627 007

ANNA UNIVERSITY: CHENNAI - 600 025

NOVEMBER 2022

INDEX

1.INTRODUCTION

Natural disasters 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. It is usually thought of as being composed of at least risk assessment, risk management, risk communication, and risk governance among other aspects. Risk assessment, the process of understanding and characterizing the risk, often mathematically, is the primary focus.

Traditionally, risk assessment is thought of as answering the following three questions:
(1) What can go wrong?
(2) With what likelihood, and
(3) With what consequences?
More recently, there has been an increasing focus on the background state of knowledge underlying these assessments and the uncertainty in these assessments.

Risk analysis is critical for natural hazards, and a number of differe nt risk analysis methods exist for assessing risk to communities from natural hazards. One set of approaches is fragility-based models such as the widely used HAZUS model that are based on simulating physical loads to systems, estimating asset-level damage through fragility curves, and then estimating system performance, losses, and deaths. The second set of approaches is based on machine learning and artificial intelligence methods. These methods will be discussed more below, but briefly, they use past data and hazard loading information, information about the system and antecedent conditions, and other information to train a machine learning model that is then used to estimate system performance or losses for future events. The third type of approach that has seen widespread use in the academic literature, but less in practice, is network theory-based methods. These approaches focus on leveraging the topology of the system to estimate performance under hazard loading in a more computationally efficient manner, though the accuracy of the results has been questioned.

1.1 PROJECT OVERVIEW

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent changes in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Various researchers have applied many deep learning techniques to detect and classify natural disasters to overcome losses in ecosystems. However, the detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem, we developed a multi-layered deep convolutional neural network model that classifies the natural disaster and tells the intensity of the disaster of natural the model uses an integrated webcam to capture the video frame, and the video frame is compared with the pre-trained model, and the type of disaster is identified and showcased on the OpenCV window.

1.2 PURPOSE

A disaster management plan is a preventative plan designed to reduce the harmful effects of a disaster of any kind that falls under natural calamity. By creating a disaster management plan ahead of time, before a disaster strikes, we can prepare to meet a disaster as it comes.

AI methods have been used in both research and practice for natural hazards risk assessment. This existing work has generally focused on estimating one or more of the following:

(1) the physical loading due to the hazard given the occurrence of the hazard or
(2) physical damage or loss of system functionality given hazard loading.

Examples of each are provided below.

The intention is not to be exhaustive but to give representative examples. One example of the use of AI for estimating the properties of the physical hazard itself is in the area of flood modeling. Several start-ups are now working on approaches that ingest weather forecasts and predict the timing and extent of riverine flooding on fine spatial scales across watershed-scale domains using a combination of physical models and validated AI methods. A similar example is recent research that aims to estimate flooding without the use of computationally expensive fluid dynamics models. Often physical models are used to develop a training data set and then AI methods are trained and validated to predict what the flood map would look like given information about the storm and the geographic area. Models such as these that aim to predict hazard loading are, by themselves, not a full risk assessment. However, they can provide valuable information for a risk assessment by estimating physical hazards in a more computationally efficient manner.

The area where AI methods have had the largest use in natural hazards risk assessment is in estimating either damage or loss of system function given hazard loading. AI-based models have similarly been developed to estimate building damage given the occurrence of an earthquake (e.g., Suryanita & Adnan, 2012). These types of models take as input a spatial field of hazard loading (e.g., maps of predicted wind speeds, soil moisture levels, and other loading measures for a hurricane or a map of a ground motion measure for an earthquake) together with a set of information about the system, the area, and prevent conditions. The model then estimates a spatial field of impacts (e.g., a map of power outages due to the approaching hurricane or a map of building damage states due to an earthquake). Again, these models are not, by themselves, a full risk assessment, but, like the hazard loading predictions, provide critical information for a full risk assessment.

In both of these types of applications of AI methods for natural hazards, the goal is to improve predictive accuracy and/or reduce computational burden relative to more traditional physics-based and engineering-based models. These models have seen use in practice recently. For example, a large portion of electric power utilities report the use of some type of power outage prediction model, and some states have required power utilities in their state to have an outage prediction model. Start-ups such as One Concern, Inc. have built businesses on providing predictions of hazard events. Clearly, emergency managers, utility managers, and policymakers see value in AI-based models for natural hazards.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Natural disasters can cause great damage to the environment, property, wildlife, and human health. These events may include earthquakes, floods, hurricanes, tornadoes, tsunamis, landslides, wildfires, volcanic eruptions, and extreme temperatures. The major challenges associated with disaster response planning are the failure to strictly apply the law, the lack of public and staff education about disaster risks, poor urban planning, unstable security situation, citizen intervention, an endowment of equipment, tools, and infrastructure, and lack of finances. If we look at the average over the past decade, approximately 45,000 people globally died from natural disasters each year. This represents around 0.1% of global deaths. It distinguishes between effects in the immediate aftermath of the disaster – mortality and demographic recovery; land loss and capital destruction; economic crisis; and blame, scapegoating, and social unrest –and longer-term structural consequences–societal collapse; economic reconstruction; long-term demographic. The impact of disasters on the poor can, in addition to the loss of life, injury and damage, cause a total loss of livelihoods, displacement, poor health, and food insecurity, among other consequence.

2.2 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. doi: 10.1785/0220200021. [CrossRef] [Google Scholar]

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. doi: 10.3390/geosciences10030105. [CrossRef] [Google Scholar]

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 modeling using advanced ensemble machine learning models. Geosci. Front. 2021;12:101075. doi: 10.1016/j.gsf.2020.09.006. [CrossRef] [Google Scholar]

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. doi: 10.1109/TMI.2017.2760978. [PubMed] [CrossRef] [Google Scholar]

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 doi: 10.1155/2020/1245924. [CrossRef] [Google Scholar]

6. Ashiquzzaman A., Oh S.M., Lee D., Lee J., Kim J. Smart Trends in Computing and Communications, Proceedings of the SmartCom 2020, Paris, France, 29–31 December 2020. Springer; Berlin/Heidelberg, Germany: 2021. Context-aware deep convolutional neural network application for fire and smoke detection in a virtual environment for surveillance video analysis; pp. 459–467. [Google Scholar]

7. Li T., Zhao E., Zhang J., Hu C. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. Electronics. 2019;8:1131. doi: 10.3390/electronics8101131. [CrossRef] [Google Scholar]

8. Mangalathu S., Burton H.V. Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions. Int. J. Disaster Risk Reduce. 2019;36:101111. doi: 10.1016/j.ijdrr.2019.101111. [CrossRef] [Google Scholar]

9. Hartawan D.R., Purboyo T.W., Setianingsih C. Disaster Victims Detection System Using Convolutional Neural Network (CNN) Method; Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT); Bali, Indonesia. 1–3 July 2019; pp. 105–111. [Google Scholar]

10. Amit S.N.K.B., Aoki Y. Disaster detection from aerial imagery with the convolutional neural network; Proceedings of the 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC); Surabaya, Indonesia. 26–27 September 2017; pp. 239–245. [Google Scholar]

11. Yang S., Hu J., Zhang H., Liu G. Simultaneous Earthquake Detection on Multiple Stations via a Convolutional Neural Network. Seism. Res. Lett. 2021;92:246–260. doi: 10.1785/0220200137. [CrossRef] [Google Scholar]

12. Madichetty S., Sridevi M. Detecting informative tweets during disaster using deep neural networks; Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS); Bangalore, India. 7–11 January 2019; pp. 709–713. [Google Scholar]

13. Nunavath V., Goodwin M. The role of artificial intelligence in social media big data analytics for disaster management-initial results of a systematic literature review; Proceedings of the 2018 5th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM); Sendai, Japan. 4– December 2018; pp. 1–4. [Google Scholar]

14. Boonsuk R., Sudprasert C., Supratid S. An Investigation on Facial Emotional Expression Recognition Based on Linear-Decision-Boundaries Classifiers Using Convolutional Neural Network for Feature Extraction; Proceedings of the 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE); Pattaya, Thailand. 10–11 October 2019; pp. 1–5. [Google Scholar]

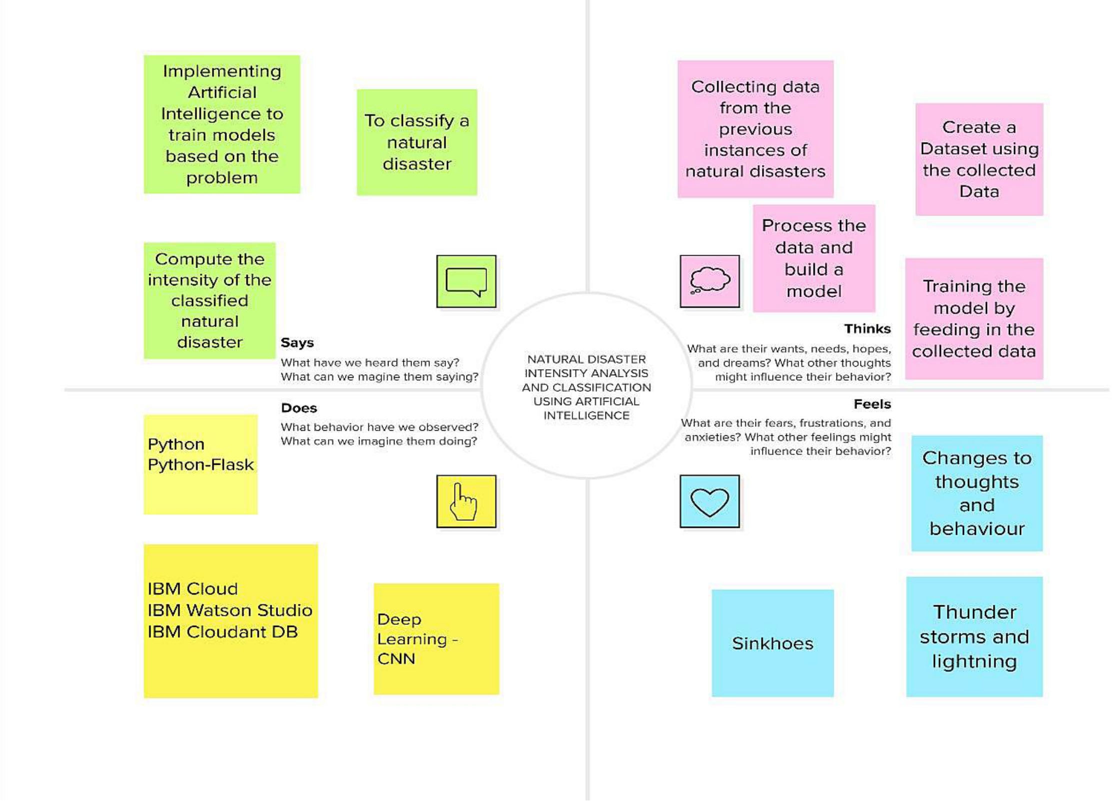
15. Zhou F., Huang J., Sun B., Wen G., Tian Y. Intelligent Identification Method for Natural Disasters along Transmission Lines Based on Inter-Frame Difference and Regional Convolution Neural Network; Proceedings of the 2019 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom); Xiamen, China. 16–18 December 2019; pp. 218–222. [Google Scholar]

2.3 PROBLEM STATEMENT DEFINITION

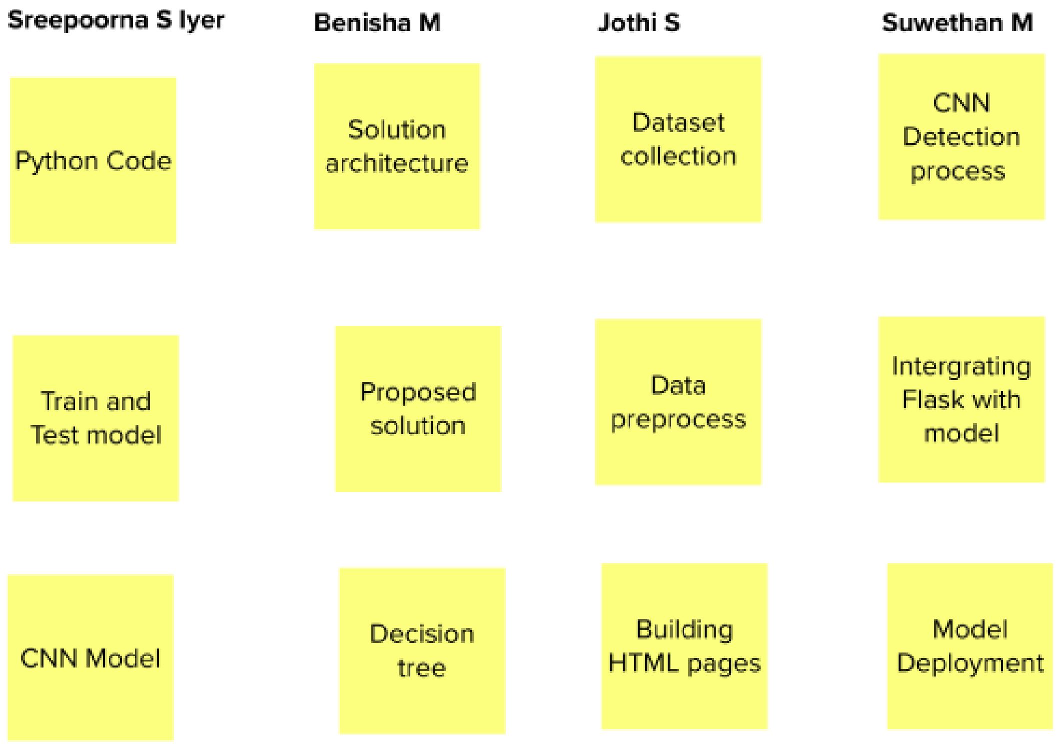
Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to a permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Various researchers have applied many deep learning techniques to detect and classify natural disasters to overcome losses in ecosystems. However, the detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem, we propose a multi-layered deep convolutional neural network. The proposed model works in two blocks: Block-I convolutional neural network (B-I CNN), for the detection and occurrence of disasters, and Block-II convolutional neural network (B-II CNN), for the classification of natural disaster intensity types with different filters and parameters.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



BRAINSTORMING

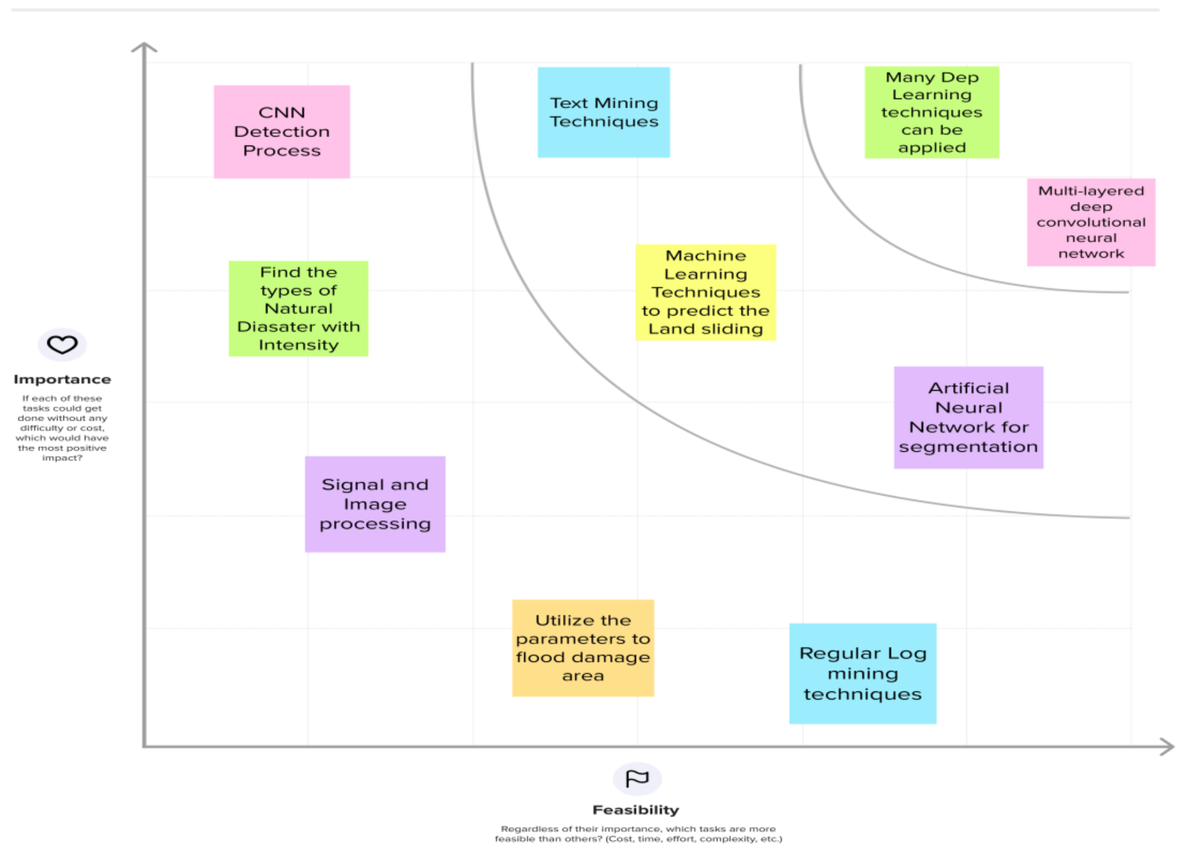
CLUSTER A

- Many Deep Learning Techniques can be applied
- Convolutional Neural Network
- Artificial Neural Network
- Signal and Image processing
- Find the types of natural disaster with intensity
- CNN Detection Process

CLUSTER B

- To predict magnitude of Earthquake
- Decision tree
- ANN used for multivariable pattern recognition at diff. level
- Detect Earthquake with speed and accuracy
- Random forest long,short-term model
- To evaluate the flood in terms of sensitivity, accuracy

IDEA LISTING AND GROUPING

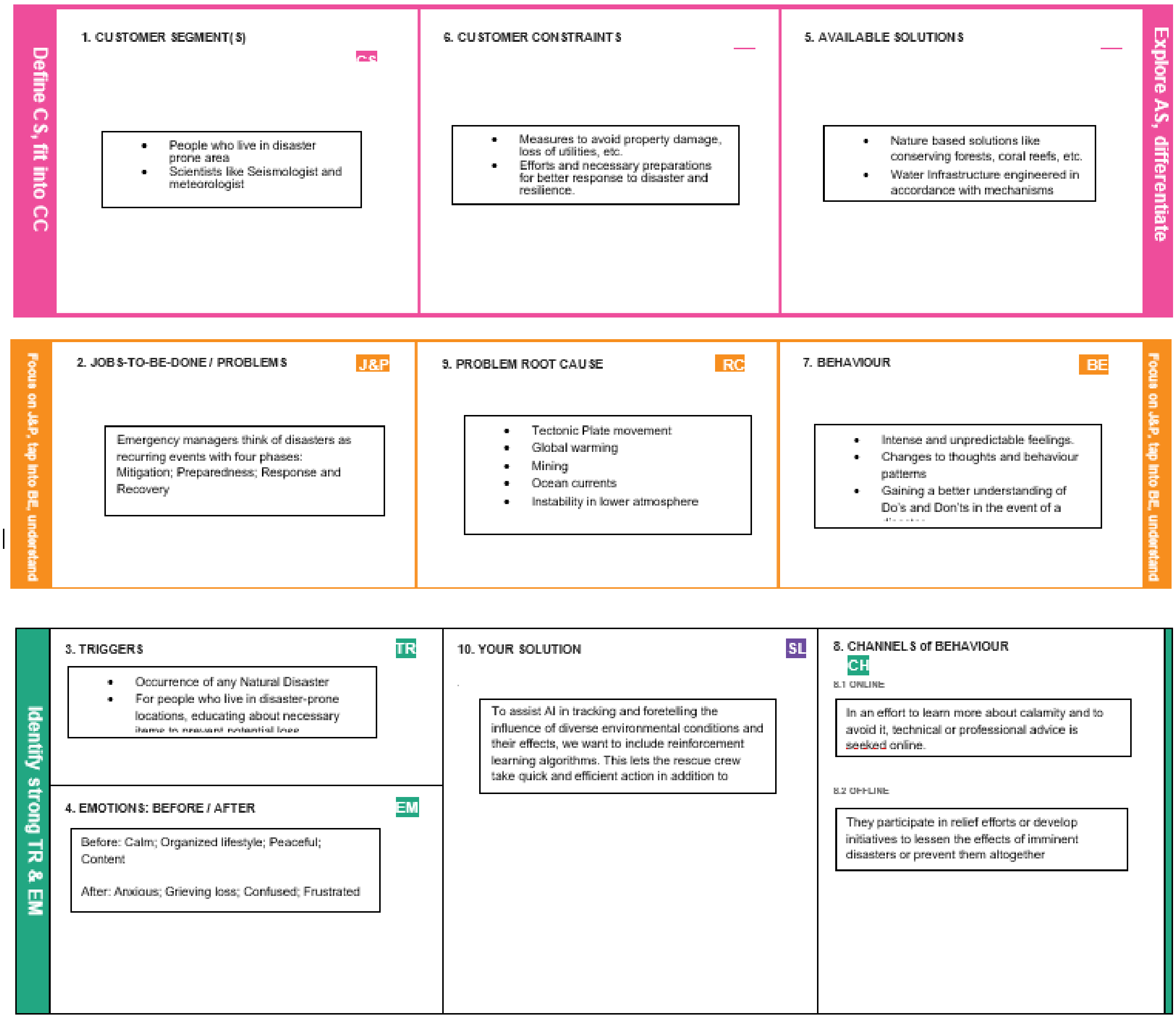


IDEA PRIORITISING

3.3 PROPOSED SOLUTION

Neural networks provide multilevel network architectures, where Convolutional Neural Networks (CNNs) are the most frequently implemented architecture as the direct input of multidimensional vector images, speech recognition, and image processing can be carried out with low complexity. CNNs efficiently perform feature extraction by denoising the images and removing interference and achieving highly accurate results. The proposed multi-layered deep convolutional neural network method works in two blocks of convolutional neural networks. The first block, known as Block-I Convolutional Neural Network (B-I CNN), detects the occurrence of a natural disaster and the second one, known as Block-II Convolutional Neural Network (B-II CNN), defines the intensity of the natural disaster. Additionally, the first block consists of three mini convolutional blocks with four layers each and includes an image input and fully connected layers. On the other hand, the second block also consists of three mini convolutional blocks with two layers each, including an image input layer and a fully connected layer.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

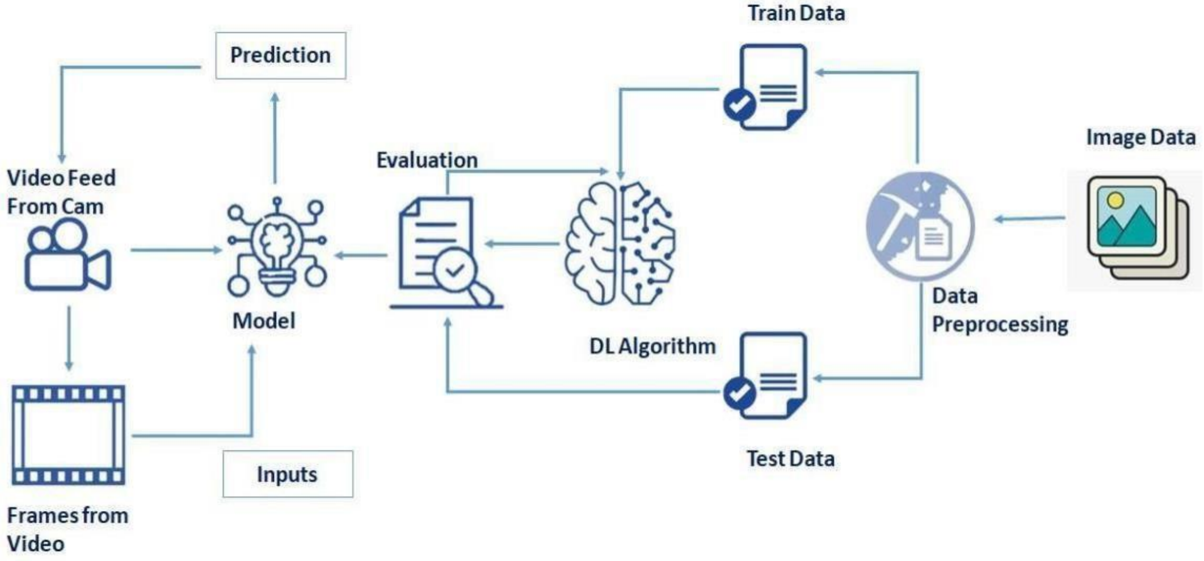
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Request permission	Access permission from Web camera
FR-2	Disaster prediction	Based on the webcam image, natural disaster is classified
FR-3	Accuracy	Since, the training and testing images are huge, the accuracy is higher
FR-4	Speed	The generation of results from the input images is 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

4.2 NON-FUNCTIONAL REQUIREMENTS

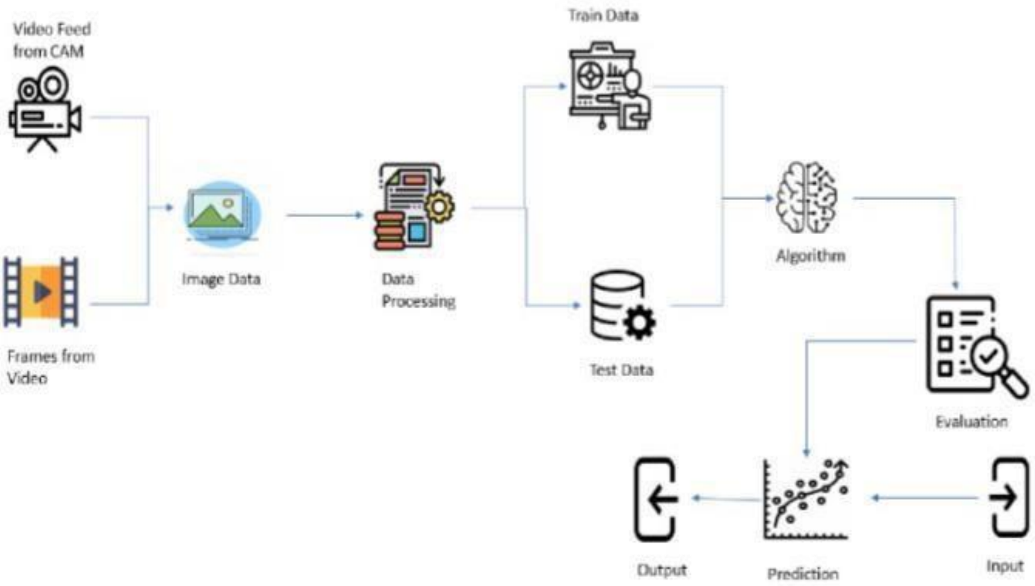
FR No.	Non-Functional Requirement	Description
NRF-1	Usability	User friendly and classify the disaster easily
NRF-2	Security	The model is secure due to the cloud deployment models and also there is no login issue
NRF-3	Reliability	Accurate prediction of the natural disaster and the website can also be fault tolerant
NRF-4	Performance	It is shown that the model gives almost 90 percent accuracy after continuous training
NRF-5	Availability	The website will be made available for 24 hours
NRF-6	Scalability	The website can run on web browsers like Google Chrome, and Microsoft Edge, and also it can be extended to the NDRF and customers

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer	Collection of datasets	USN-1	As a user, I can collect the dataset for monitoring & analysing.	The database is connected.	Medium	Sprint-1
	Home Page	USN-2	As a user, I want to know about the basics of frequently occurring disasters.	Basics of frequently occurring disasters are known.	High	Sprint-1
	Intro Page	USN-3	As a user, I want to know about the introduction of disasters in particular areas.	Introduction of disasters in particular areas are known.	High	Sprint-1
	Open Webcam	USN-4	As a user, I adapted the webcam to analyse and classify the disaster from video capturing.	Video capturing is running successfully.	High	Sprint-1
	Analysis of required phenomenon	USN-5	As a user, I can regulate certain factors influencing the action and report on past event analysis.	I can regulate and report the analysis.	High	Sprint-2

	Algorithm Selection	USN-6	As a user, I can choose the required algorithm for specific analysis.	Algorithms for specific analysis are chosen.	High	Sprint-2
	Training & Testing	USN-7	As a user, I can train & test the model using the algorithm.	The model is tested and trained successfully.	High	Sprint-2
	Detection & analysing of data	USN-8	As a user, I can detect and visualise the data effectively.	Detection and visualisation of data is done successfully.	High	Sprint-3
	Model Building	USN-9	As a user, I can build web applications.	I can build a web application.	High	Sprint-3
	Integrate the web app with AI model	USN-10	As a user, I can use the Flask app to use models easily through a web app.	I can use the flask app.	High	Sprint-4
	Model Deployment	USN-11	As an administrator, I can deploy the AI model in IBM cloud.	I can deploy the IBM cloud.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collection of datasets	USN-1	As a user, I can collect the dataset for monitoring & analysing.	5	Medium	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-1	Home Page	USN-2	As a user, I want to know about the basics of frequently occurring disasters.	5	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-1	Intro Page	USN-3	As a user, I want to know about the introduction of disasters in particular areas.	5	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-1	Open Webcam	USN-4	As a user, I adapted the webcam to analyse and classify the disaster from video capturing.	5	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S

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	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-2	Algorithm Selection	USN-6	As a user, I can choose the required algorithm for specific analysis.	5	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-2	Training & Testing	USN-7	As a user, I can train & test the model using the algorithm.	10	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-3	Detection & analysing of data	USN-8	As a user, I can detect and visualise the data effectively.	10	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-3	Model Building	USN-9	As a user, I can build web applications.	10	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-4	Integrate the web app with AI model	USN-10	As a user, I can use the Flask app to use models easily through a web app.	10	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S
Sprint-4	Model Deployment	USN-11	As an administrator, I can deploy the AI model in IBM cloud.	10	High	SreePoorna S Iyer, Suwethan M, Benisha M, Jothi S

6.2 SPRINT DELIVERY SCHEDULE

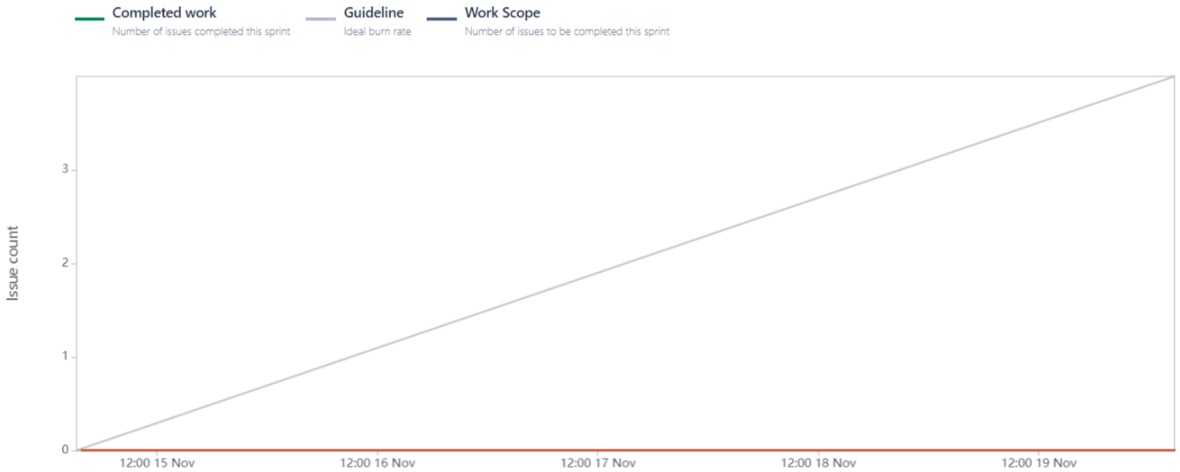
Name	Milestone Number	Description	Mandatory	Optional
Project Objectives	M-001	We will be able to learn how to get and prepare the dataset. We will be able to know how to do image processing. We will understand how CNN layers are work. Classify images using a Convolutional Neural Network. We will be able to know what are the activation functions can be used. We will be able to know how to read images using OpenCV and Convolutional Neural Networks for Computer vision AI problems.	Yes	-
Project Flow	M-002	To accomplish project flow, we should have complete Data collections, image processing, model building, video streaming and altering.	Yes	-
Pre-Requisites	M-003	To accomplish this project, we should have known the following software and packages such as anaconda navigator, tensor flow, Keras, etc.	Yes	-
Prior Knowledge	M-004	One should have knowledge on Artificial Neural Networks supervised and unsupervised learning, CNN and Regression Classification and clustering, ANN.	Yes	-
Data Collection	M-005	We can collect datasets from different open sources like kaggle.com,etc.	Yes	-

Image Processing	M-006	Importing the ImageDataGenerator libraries, define Parameters/Arguments for ImageDataGenerator class, applying Image Data Generator Functionality to train set and test set.	Yes	-
Model Building	M-007	Importing the model building libraries, Initialising the model, Adding CNN layers, Adding dense layers, Configuring the learning process, Train the model, Save the model and Predictions.	Yes	-
Video Analysis	M-008	OpenCV for video processing, creating an account in Twill service and sending alert messages.	Yes	-
Train CNN model	M-009	Register for IBM Cloud and train Image Classification model.	Yes	-
Ideation Phase	M-010	Prepare Literature Survey on the selected project and Information Gathering, Empathy map and Ideation.	Yes	-
Project Design Phase-I	M-011	Prepare Proposed Solution, Problem solution fit and Solution Architecture.	Yes	-
Project Design Phase-II	M-012	Prepare Customer Journey, Functional Requirements, Data flow Diagrams and Technical Architecture.	Yes	
Project Planning Phase	M-013	Prepare Milestone and Activity list, Sprint Delivery Plan.	Yes	
Project Development Phase	M-014	Project Development delivery of Sprint-1, Sprint-2, Sprint-3 and Sprint-4.	Yes	

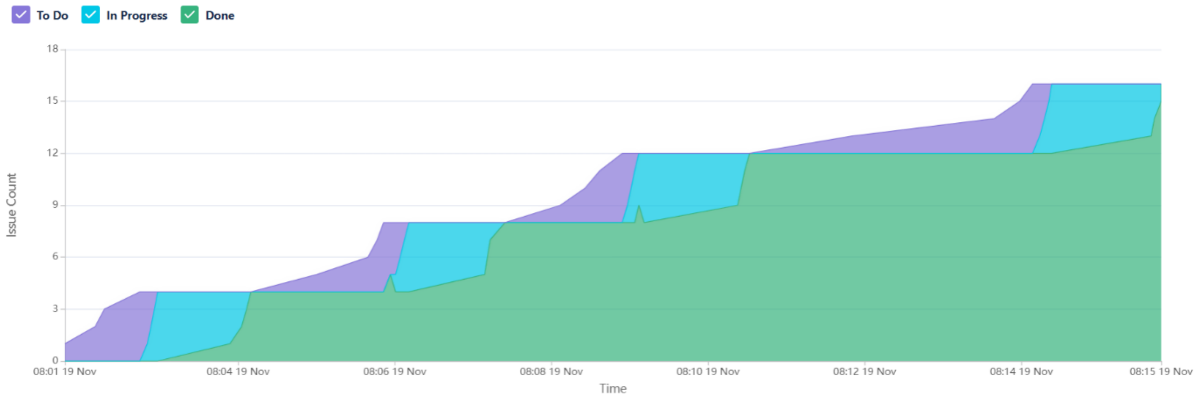
6.3 REPORTS FROM JIRA

1.Velocity Report

2.Burnup Report



3.Cumulative Flow Diagram



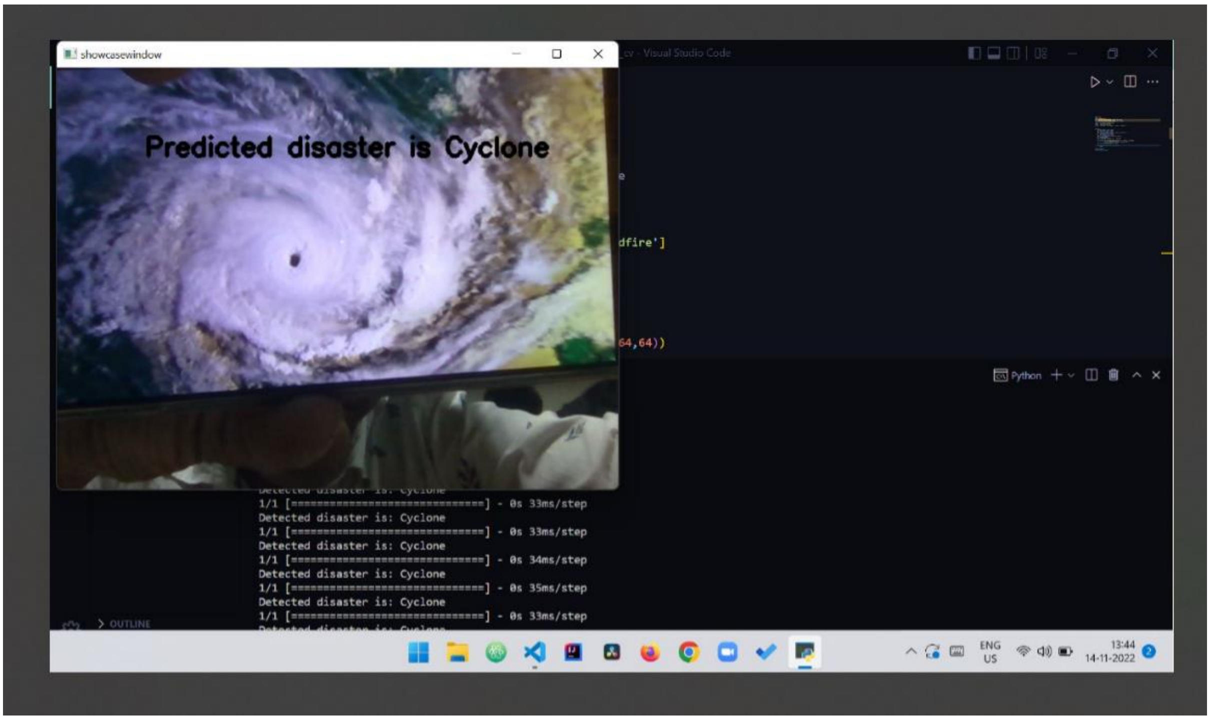
7.

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

7.1 FEATURE 1

```
#Importing required libraries
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img("F:/Workspace/IBM/Natural Disaster Intensity Analysis and Cassification/dataset/test_set/Cyclone/927.jpg", target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
model.predict(x)
pred = np.argmax(model.predict(x))
op = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
print(pred)
print(op[pred])
```

7.2 FEATURE 2



8. TESTING

8.1 TEST CASES

Section	Total Cases	Not Tested	Fail	Pass
Home Page	7	0	0	7
Intro Page	51	1	0	50
Predict Page	2	0	0	2
Model	3	0	0	3

8.2 USER ACCEPTANCE TESTING

Defect Analysis

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	7

9. RESULTS

9.1 PERFORMANCE METRICS

To evaluate the performance of the proposed multilayered deep convolutional neural network, use a train–test validation schema. To train the whole model, the training dataset was used, while for the fine-tuning of the model the validation set was used. The performance of the whole framework was calculated on the basis of the test dataset. For the evaluation of the proposed model on the given dataset of classification for positive and negative values, four types of data were accrued: true positive (TP), the number of correctly positive classified images; true negative (TN) the number of correctly negative classified images; false positive (FP), the number of incorrectly positive classified images; and false negative (FN), the number of images that are incorrectly classified as negative images. To calculate the performance of the model, the specificity (SP), sensitivity (SE), accuracy rate (RR), precision (PRE) statistical values were adopted as a criterion. The F1 score was used when a conflict occurred between accuracy and sensitivity to evaluate the performance.

Sensitivity (SE) = TP / TP+FN - [1]

The sensitivity (*SE*) in Equation (1) is the true positive measurement, the ratio of correctly identified values.

Specificity (SP) = TN / TP+FP - [2]

Equation (2) shows the value of specificity (*SP*), the ratio of negatives which are correctly classified.

Accuracy Rate (AR) = TP+TN / TP+TN+FP+FN - [3]

Equation (3) gives the value of accuracy rate (*AR*), which is equal to the actual measurement of specified values.

Precision (PRE) = TP / TP+FP - [4]

The precision (*PRE*) in Equation (4) explains the proportion of closeness in measurement values.

$$F1-\text{Score } (F1) = 2(SE \times PRE) / SE + PRE \qquad - [5]$$

The $F1$–Score ($F1$) in Equation (5) is the proportion of recall and precision which actually measure the model accuracy for the dataset.

10. ADVANTAGES & DISADVANTAGES

Advantages

1.More accurate prediction of natural disasters.

2.Predict magnitude of earthquake.

3.Predict magnitude of earthquake.

4.Detect earthquake with speed and accuracy on seismological data.

5.Utilize some parameters to access the model for flood damage area detection.

6.Sum-up good results as compared to the already existing techniques in the southeast Asia.
7.Classify the natural disasters on various parameters.

Disadvantages

1.Limited statistical parameters for prediction

2.Work only for prediction on seismic dataset

3.Limited parameters used for prediction

4.Depends on public feedback to detect earthquake

5.Parametric limitation for the detection of flood damaging regions

6.Work for monsoon floods in June and September for specific regions in India for time series data

7.Limited for only early stages of natural disasters

11. CONCLUSION

Many researchers have attempted to use different deep learning methods for detection of natural disasters. However, the detection of natural disasters by using deep learning techniques still faces various issues due to noise and serious class imbalance problems. To address these problems, we proposed a multi-layered deep convolutional neural network for detection and intensity classification of natural disasters. The proposed method works in two blocks—one for detection of natural disaster occurrence and the second block is used to remove imbalanced class issues. The results were calculated as average statistical values: sensitivity, 97.54%; specificity, 98.22%; accuracy rate, 99.92%; precision, 97.79%; and F1-score, 97.97% for the proposed model. The proposed model achieved the highest accuracy as compared to other state-of-the-art methods due to its multilayered structure. The proposed model performs significantly better for natural disaster detection and classification, but in the future the model can be used for various natural disaster detection processes.

12. FUTURE SCOPE

Disaster resilience efforts may look very different tomorrow from how they appear today. Once an advancing cyclone or hurricane is identified, for example, geo-spatial, weather and previous disaster data could be used to predict how many people will be displaced from their homes and where they will likely move. Such insights could help emergency personnel identify how much aid (water, food, medical care) will be needed and where to send it. AI algorithms could instantaneously assess flooding, building and road damage based on satellite images and weather forecasts, allowing rescuers to distribute emergency aid more effectively and identify those still in danger and isolated from escape routes.

McKinsey’s Noble Intelligence is just one example of an initiative trying to harness AI’s potential to support humanitarian causes. For instance, the team is developing an algorithm that will reduce the time it takes to assess damage to buildings such as schools from weeks to minutes, using a combination of satellite, geo-spatial, weather and other data. This information can then be used to identify the best places to set up temporary school tents and where to prioritize reconstruction efforts.

13. APPENDIX

13.1 Source Code

Model Building
<u>AI based Natural disaster analysis.ipynb</u>

```
"""
Data Augmentation
"""
#Import the ImageDataGenerator library
from tensorflow.keras.preprocessing.image import ImageDataGenerator
"""
Configure ImageDataGenerator class
"""
#Train_set configuration...
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
#Test_set configuration...

test_datagen = ImageDataGenerator(rescale=1./255)
"""
Applying ImageDataGenerator functionality
"""
#Train_set
xtrain = train_datagen.flow_from_directory('F:/Workspace/IBM/Natural Disaster Intensity Analysis and Cassification/dataset/train_set',
                                       target_size=(64,64),
                                       class_mode='categorical',
                                       batch_size=100)
#Test_set
xtest = test_datagen.flow_from_directory('F:/Workspace/IBM/Natural Disaster Intensity Analysis and Cassification/dataset/test_set',
                                       target_size=(64,64),
                                       class_mode='categorical',
                                       batch_size=100)
"""
CNN model
"""
#Importing the required library
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
#Initializing the model
model = Sequential()
"""
CNN layers
"""
#Convolutional layer
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
#Pooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#Flatten layer
model.add(Flatten())
#Hidden layers
model.add(Dense(300,activation='relu'))
```

```
model.add(Dense(150,activation='relu'))
model.add(Dense(4,activation='softmax'))
#Compiling the model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
#Training the model
model.fit(xtrain,
        steps_per_epoch=len(xtrain),
        epochs=20,
        validation_data=xtest,
        validation_steps=len(xtest))
#Saving the model in Hierarchical Data Format
model.save('disaster.h5')
#Testing the model
#Importing required libraries
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img("F:/Workspace/IBM/Natural Disaster Intensity Analysis and Cassification/dataset/test_set/Cyclone/927.jpg", target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
model.predict(x)
pred = np.argmax(model.predict(x))
op = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
print(pred)
print(op[pred])
```

Application Building

app.py

```
import cv2
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import subprocess
from asyncio import subprocess
from flask import Flask, render_template, request, url_for

app=Flask(__name__)

model=load_model("F:/Workspace/IBM/Natural Disaster Intensity Analysis and Cassification/Flask/disaster.h5")

@app.route('/')
def home():
    return render_template("home.html")

@app.route('/intro')
def intro():
    return render_template("intro.html")

@app.route('/predict')
def predict():
    video = cv2.VideoCapture(0)
    index = ['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
```

```
    while(1):
        success,frame = video.read()
        cv2.imwrite("1.jpg", frame)
        img = image.load_img("1.jpg", target_size=(64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(model.predict(x),axis=1)
        p = index[pred[0]]
        cv2.putText(frame,"Predicted disaster is " + str(p), (100,100),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 4)
        cv2.imshow("Prediction window ('Press 'Q' to quit')", frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            print("Detected disaster is: " + str(p))
            break
    video.release()
    cv2.destroyAllWindows()
    return render_template("upload.html", disaster= str(p))


if __name__ == '__main__':
    app.run(debug = False)
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Natural Disaster Classification</title>
<link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='images/favicon1.png')}}">
<style>

    body {
        margin: 30px;
        background-color: #f5ece4;
    }
    .nav_bar {
        position:relative;
        height: 50px;
        width: 100%;
        border-radius: 10px;
        /* background-color: rgb(105, 237, 138); */
        background-image: linear-gradient(to right, #f6f0ea, #f5ece4, #f3e8dd, #f2e3d7, #f1dfd1);
        box-shadow: -2px -2px 5px #cacaca;
        transition: transform .5s;
    }

    .nav_bar:hover {
        transform: scale(1.02);
    }
    .txt {
        /* border: 1px solid black; */
```

```
    display: inline-block;
    min-width: 70%;
    text-align: center;
    left: 20px;
    top: 12px;
    font-size: 1.5rem;
}

a, span {
    position: absolute;
    border-radius: 5px;
    /* border: 1px solid black; */
}

table {
    margin-top: 20px;
    margin-left: 20px;
    margin-right: 20px;
    height: 100%;
}

.cam {
    top: 5px;
    right: 50px;
    /* border: 3px solid black; */
}
.home  {
    right: 250px;
    top: 15px;
    font-size: 1.2rem;
    text-decoration:none;
    color: black;
}
.intro {
    right: 130px;
    top: 15px;
    font-size: 1.2rem;
    text-decoration:none;
    color: black;
}

.home:hover, .intro:hover {
    cursor: default;
    color: white;
}

.cam > img {
    /* right: 130px; */
    height: 40px;
    width: 40px;
}

td > img {
    display: block;
    height: 200px;
```

```
        width: 300px;
      }
</style>
</head>
<body>
<div class="nav_bar">
<span class="txt">Natural Disaster Intensity Analysis and Classification using AI</span>

<a href="/" class="home">Home</a>
<a href="/intro" class="intro">Introduction</a>
<a href="/predict" class="cam">
<img src="{{ url_for('static', filename='images/camera.png') }}">
</a>
</div>
</div>

<table>
<tr>
<td>
<img class="cycl" src="{{ url_for('static', filename='images/Cycl.jpg') }}" alt="">
<b>Cyclone:</b><br>
Voilent winds, torrential rain, high waves and very destructive strom
</td>
<td>
<img class="eq" src="{{ url_for('static', w of water on normally dry ground
</td>
<td>
<img class="wf" src="{{ url_for('static', filename='images/Wf.jpg') }}" alt="">
<b>Wildfire:</b>
Uncontrolled fire in a forest, grassland, brushland
</td>
</tr>
</table>

</body>
</html>
```

intro.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Natural Disaster Classification</title>
<link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='images/favicon2.png') }}">
<style>

      body {
        margin: 30px;
        background-color: #f5ece4;
      }
      .nav_bar {
        position:relative;
```

```
   height: 50px;
   width: 100%;
   border-radius: 10px;
   /* background-color: rgb(105, 237, 138); */
   background-image: linear-gradient(to right, #f6f0ea, #f5ece4, #f3e8dd, #f2e3d7, #f1dfd1);
   box-shadow: -2px -2px 5px #cacaca;
   transition: box-shadow 1s;
}

.nav_bar:hover {
   box-shadow: 3px 3px 5px #cacaca;
}
.txt {
   /* border: 1px solid black; */
   display: inline-block;
   min-width: 70%;
   text-align: center;
   left: 20px;
   top: 12px;
   font-size: 1.5rem;
}

a, span {
   position: absolute;
   border-radius: 5px;
   /* border: 1px solid black; */
}

table {
   margin-top: 20px;
   margin-left: 20px;
   margin-right: 20px;
   height: 100%;
}

.cam {
   top: 5px;
   right: 50px;
   /* border: 3px solid black; */
}
.home  {
   right: 250px;
   top: 15px;
   font-size: 1.2rem;
   text-decoration:none;
   color: black;
}
.intro {
   right: 130px;
   top: 15px;
   font-size: 1.2rem;
   text-decoration:none;
   color: black;
}

.home:hover, .intro:hover {
```

```
        cursor: default;
        color: white;
    }

    .cam > img {
        /* right: 130px; */
        height: 40px;
        width: 40px;
    }

    td > img {
        display: block;
        height: 200px;
        width: 300px;
    }

    .content {
        /* padding: 20px; */
        height: 400px;
        width: 100%;
        border: 1px solid black;
        font-size: 1.3rem;
        bottom: 200px;
    }

    .cont-con {
        position: relative;
        /* border: 1px solid black; */
        height: 500px;
        width: 100%;

    }
    .cont {
        position: absolute;
        background-color: #f7e5d5;
        font-size: 20px;
        /* border: 1px solid black; */
        height: 400px;
        width: 700px;
        left: 250px;
        top: 50px;
        border-radius: 10px;
        padding: 10px;
        box-shadow: 5px 5px 5px #807272;
        object-fit: cover;
        transition: transform .5s;
        }

    .cont:hover {
        box-shadow: 5px 5px 5px #807272;
        transform: scale(1.02);
    }
</style>
</head>
<body>
```

```
<div class="nav_bar">
<span class="txt">Natural Disaster Intensity Analysis and Classification using AI</span>

<a href="/" class="home">Home</a>
<a href="/intro" class="intro">Introduction</a>
<a href="/predict" class="cam">
<img src="{{ url_for('static', filename='images/camera.png') }}" alt="">
</a>
</div>
<div class="cont-con">
<div class="cont">
<b> Abstract: </b> Natural disasters not only disturb the human ecological system but also destroy
the
properties and critical infrastructures of human societies and even lead to permanent change in the
ecosystem. Disaster can be caused by naturally occurring events such as earthquakes,
cyclones,floods, and wildfires. Many deep learning techniques have been applied by various
researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection
of naturaldisasters still faces issues due to the complex and imbalanced structures of images. To
tackle thisproblem, we propose a multilayered deep convolutional neural network. The proposed
model works in two blocks: Block-I convolutional neural network (B-I CNN), for detection and
occurrence of disasters, and Block-II convolutional neural network (B-II CNN), for classification of
natural disaster intensity types with different filters and parameters. The model is tested on 4428
natural images and performance is calculated and expressed as different statistical values:
sensitivity (SE), 97.54%;specificity (SP), 98.22%; accuracy rate (AR), 99.92%; precision (PRE),
97.79%; and F1-score (F1), 97.97%. The overall accuracy for the whole model is 99.92%, which is
competitive and comparable with state-of-the-art algorithms. <br>
<b> Keywords: </b>deep learning; natural disasters intensity and classification; convolutional
neural network
</div>
</div>

</body>
</html>
```

upload.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Alert!!!</title>
<link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='images/favicon3.png') }}">
<style>
    img {
      height: 100%;
      width: 100%;
    }
</style>
</head>
<body>
<!-- <img src="{{ url_for('static', filename='images/warning.png')}}" alt=""> -->
<h1><center>The Detected disaster is {{ disaster }}</center></h1>
```

</body>
</html>

13.2 GitHub & Project Demo Link

GitHub repository link:
https://github.com/IBM-EPBL/IBM-Project-21521-1659782726

Project Demo (video) link:
https://youtu.be/ZtHBZUKLu4s