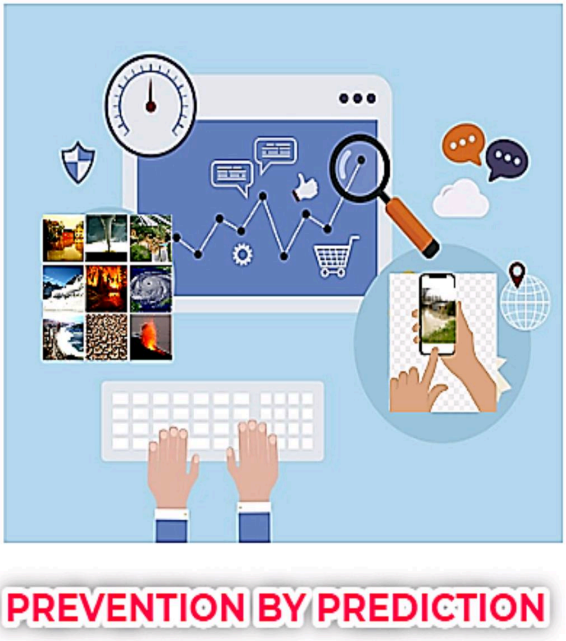
NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP



PROJECT TITLE: **Natural Disaster Intensity Analysis and Classification using Artificial Intelligence**

TEAM ID: **PNT2022TMID27942**

CONTRIBUTORS:

SIDHARTH M V (311519106088)

UTHRA T K (311519106104)

NITHYASREE H (311519106065)

VALLIAMMAI S (311519106105)

ABSTRACT

A natural disaster is "the negative impact following an actual occurrence of natural hazard in the event that it significantly harms a community". It can cause loss of life or damage property. The severity of the damage depends on the affected population's resilience and on the infrastructure available.Some of the natural disasters are cyclone, wildfire, earthquake and floods.There are many prediction methods to predict the occurrence of natural disaster, yet the level of damage due to disaster has not been reduced. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem, we have developed a multilayered deep convolutional neural network model that classifies the natural disaster and a reinforcement learning based model that tells the action to be performed based on the intensity of disaster . The model separates the key frames from the video for processing. The deployed CNN model works with an accuracy of 85.4% and the Reinforcement Learning models have lower deviation values ranging from minimum of '0' to maximum of '5'.This model is self –sufficient and can produce more efficient results in a longer run. The predictions has high accuracy rate and this model will be highly useful in real time.

PROJECT INDEX

1. INTRODUCTION

1.1 Project Inspiration

A natural disaster is characterized by the abnormal intensity of a natural agent (flood, mudslide, earthquake, avalanche, drought) when the usual measures to be taken to prevent this damage were not able to prevent their emergence or were not able to be taken.Natural Disasters are catastrophic events that result from any of the Earth's natural phenomena. These can range from floods and hurricanes to tsunamis and earthquakes.The Earth, over its 4.54 billion-year history, has seen many natural disasters. Some of these disasters have led to several mass extinctions and drastic repercussions for various surviving species.A natural disaster can be defined as:"A major event brought about by the natural processes of the Earth that causes widespread destruction to the environment and loss of life."A natural hazard often precedes a natural disaster.

The world is racked by disasters, some of which are awful yet controllable. Natural disasters, for instance, can be sudden events which cause significant destruction to property and lives. They can happen naturally, or humans can cause them. The emergency management process is essential to fix the damages caused by these catastrophes. With the process of disaster management, it is possible to limit the damage, and the dangers of the incident can be managed. The process is targeted at warding off disasters and reducing the consequences of events that cannot be avoided.

1.2 Purpose

There is no common system to help emergency responders measure the impact of natural disasters, to determine the proper allocation of resources, and to expedite mitigation processes. Therefore, a nation's ability to manage extreme events, including natural disasters and other perils, is difficult when there is no consistent method or mutual understanding among emergency management systems of different countries at all levels: international, continental, regional, national, provincial, and local. Therefore, a common severity classification system for all types of natural disasters for all stakeholder groups is required to understand, communicate, and educate the public on the nature of natural disasters.

2. LITERATURE SURVEY

2.1 Existing Problem

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

2.2 References

S NO	TITLE OF THE PAPER	AUTHOR	YEAR	METHODOLOGY
1	Hurricane Damage Detection using Machine Learning and Deep Learning Techniques: A Review	Swapandeep Kaur et al	2021	High volume data of social media and satellite imagery are analysed for damage detection of natural disasters using machine learning and deep learning techniques.
2	Natural Disasters Intensity Analysis and Classification Based on Multispectral Images Using Multi-Layered Deep on Multispectral Images Using Multi-Layered Deep Convolutional Neural Network	Muhammad Aamir, Tariq Ali, Muhammad Irfan, Ahmad Shaf, Muhammad Zeeshan Azam, Adam Glowacz, Frantisek Brumercik,Witold Glowacz,Samar Alqhtani and Saifur Rahman	2021	The proposed multi-layered deep convolutional neural network method works in two blocks of convolutional neural networks. The first block detects the occurrence of a natural disaster and the second defines the intensity of the natural disaster.

3	Disaster Intensity-Based Selection of Training Samples for Remote Sensing Building Damage Classification	Luis Moya , Christian Geiß , Member, IEEE, Masakazu Hashimoto, Erick Mas , Shunichi Koshimura , and Günter Strunz	2021	For the detection of severely damaged buildings a procedure that is based on the automatic selection of training samples for learning and calibrating the standard support vector machine classifier is and the use of two regularization parameters to define the support vectors.
4	Artificial Intelligence for Natural Hazards Risk Analysis:Potential, Challenges, and Research Needs	Seth Guikema	2020	This article is on artificial intelligence, machine learning, and statistical methods used for natural hazard risk assessment .
5	Deep Learning Benchmarks and Datasets for Social Media Image Classification for Disaster Response	Firoj Alam, Ferda Ofli, Muhammad Imran, Tanvirul Alam, Umair Qazi	2020	New datasets for disaster type detection, and informativeness classification, and damage severity assessment are proposed with benchmarking several state-of-the-art deep learning models and achieve promising results.
6	International Journal of Geo-Information,MDPI Journals	Huan Ning et al.	2020	For the detection of flood images obtained from social media, a CNN was developed. when more flooding images were available
7	Forest fire image recognition based on convolutional neural network	Wang Yuanbin, Dang Langfei, Ren Jieying	2019	To detect fire automatically, a forest fire image recognition method based on convolutional neural networks ,based on adaptive pooling method shows better performance and high

				recognition rate.
8	International Journal of Digital Earth, Taylor and Francis	Muham med Ali Sit et al.	2019	In order to identify content and time-related context of social media data related to the disasters, Deep learning, spatial temporal analysis and natural language processing were used.
9	Progress in Disaster Science, Elsevier	Brett W Robertso n et al.	2019	On the basis of urgency and time period, images were classified using Multilayer perceptron and VGG-16 CNN network
10	Neural Computing and Applications, Springer	Muham mad Dawood et al.	2019	Satellite images of hurricane were used to determine hurricane intensity using deep CNN method in an automatic manner.
11	MDPI journal Applied Science	Yundong Li et al.	2019	Damage caused to buildings during Hurricane Sandy during the year 2012 was detected using a deep learning method that is Single shot Multi-box detector by the use of aerial images.
12	ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-2, 2018	Dwarte et al.	2018	Flood severity was detected using Gated Recurrent Unit (GRU) - Convolutional Neural network(CNN).

13	IEEE Transactions on Image Processing	Ritesh Pradhan et al.	2018	Tropical cyclones occurrence during 1998-2012 was studied and the intensity of hurricanes were estimated using deep convolutional neural network.
14	Journal of Applied Remote Sensing	Yundong Li et al.	2018	Damage occurred during Hurricane Sandy in the year 2012 was determined using semi-supervised deep learning system applied on aerial images of the disaster.
15	Springer International Publishing AG, part of Springer Nature 2018	German Novikov et al.	2018	Damage caused by the wildfires in California in 2017 was determined by deep learning using satellite images.

2.3 Problem Statement

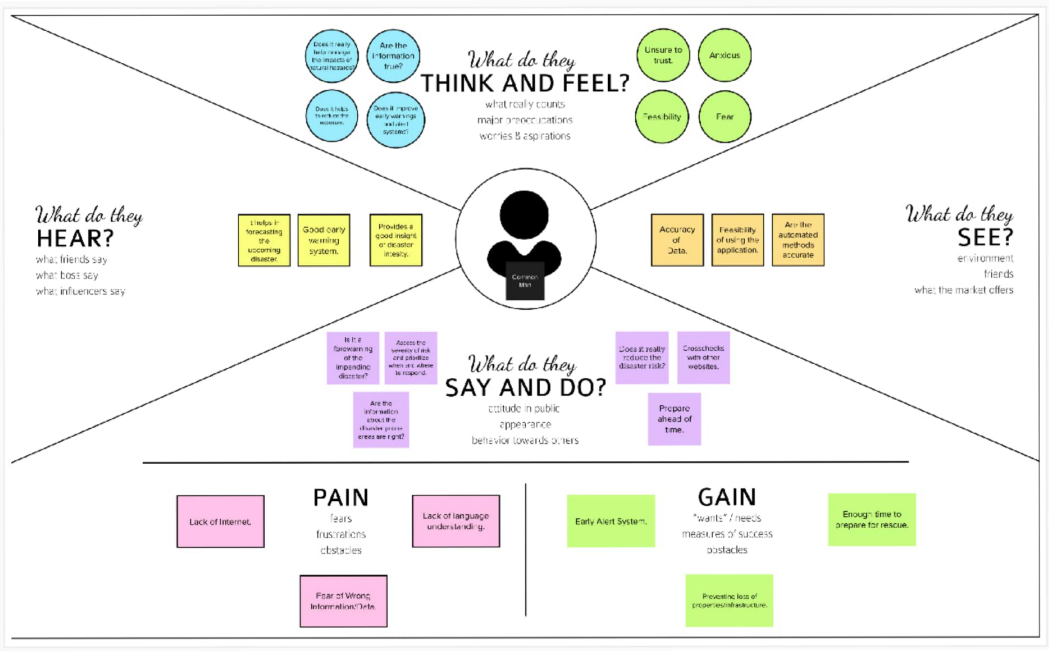
A **natural disaster** is "the negative impact following an actual occurrence of natural hazard in the event that it significantly harms a community". A natural disaster can cause loss of life or damage property. The severity of the damage depends on the affected population's resilience and on the infrastructure available.Some examples of the natural disasters are cyclone, wildfire, earthquake and floods.There are many prediction methods to predict the occurrence of natural disaster, yet the level of damage due to disaster couldn't be lowered.

How efficiently might we classify and analyse the risk of a Natural Disaster???

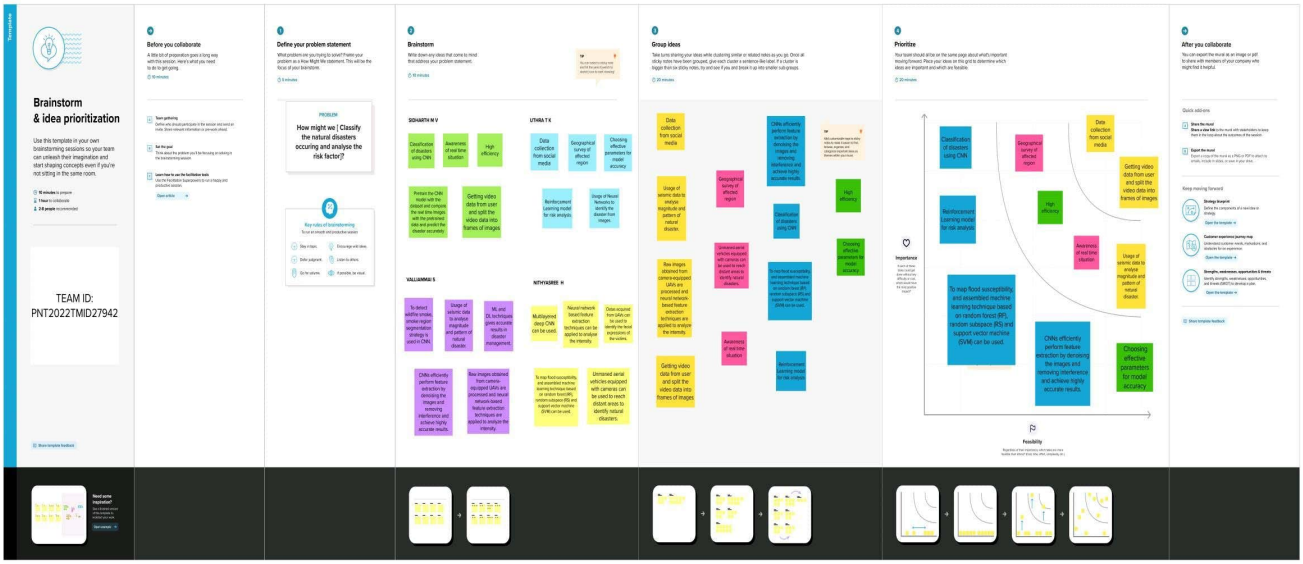
What is the problem?	Occurrence of natural disaster
Who does the problem affects?	Environment and all of its constituents such as human beings, plants, animals and infrastructures & economy
When the problem occurs?	Extreme climate conditions, human carelessness, seismic movements and greenhouse effect
Where does the problem occur?	Coastal regions, over populated regions
Why it is important to fix the problem?	To save the lives of living things, maintain ecological balance and financial security, forecast awareness, less losses due to damage

3. IDEATION & PROPOSED SOLUTION

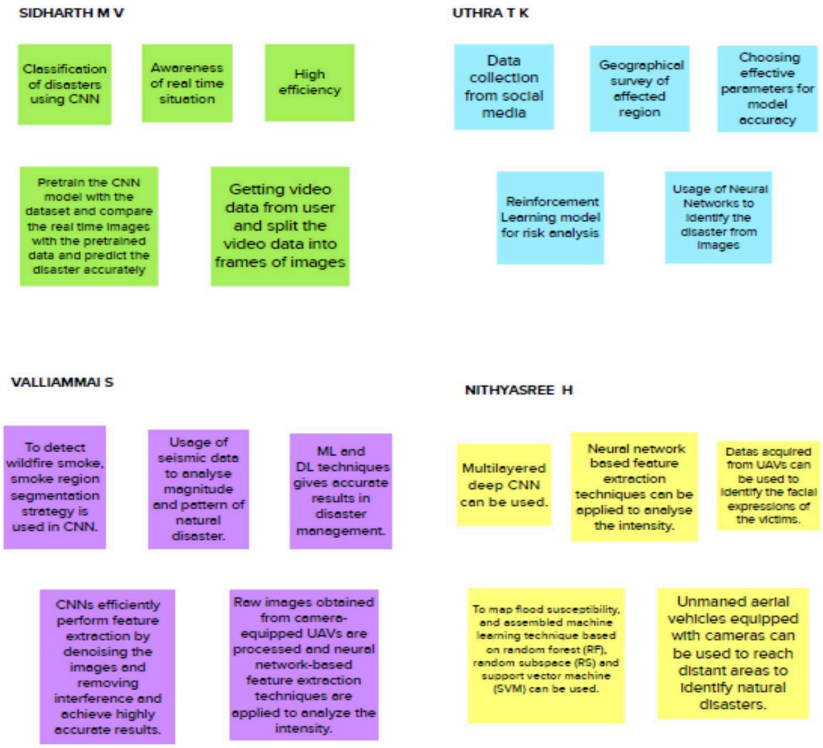
3.1 Empathy Map Canvas



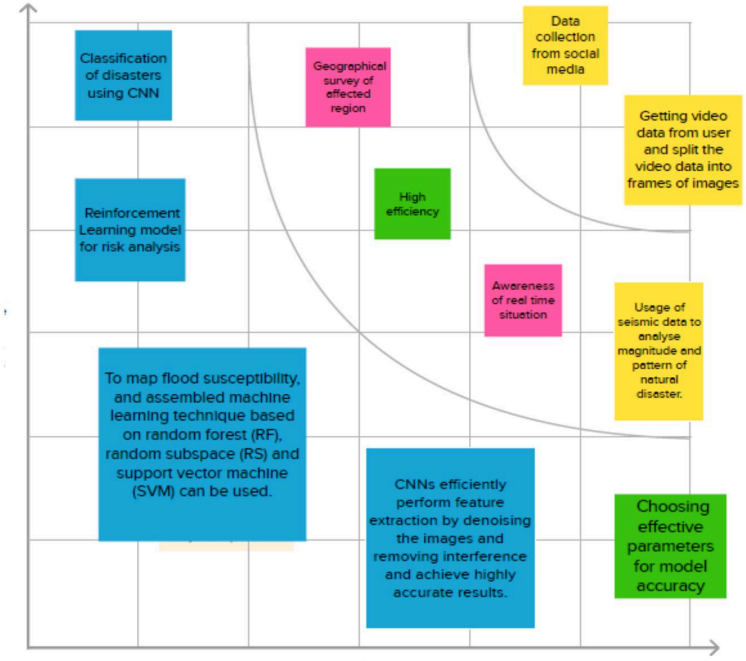
3.2 Ideation & Brainstorming



- ***Brainstorming***



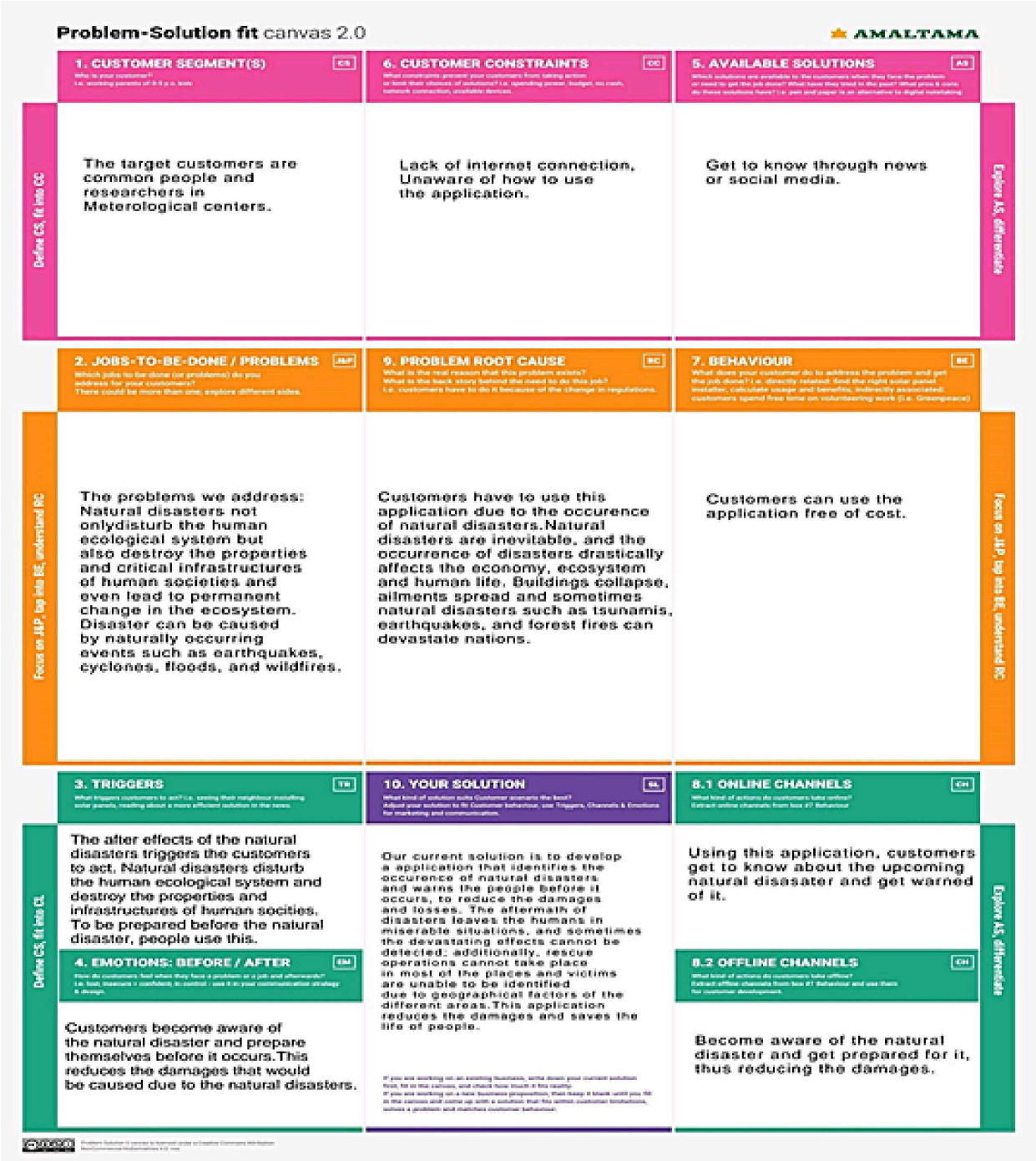
- ***Prioritize ideas***



3.3 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Classify and identify the occurring natural disaster and analyse its risk factor.
2	Idea / Solution description	Usage of AI-based multilayer network model for identification of the disaster.
3	Novelty / Uniqueness	Usage of self-sufficient AI model based on reinforcement learning for risk analysis.
4	Social Impact / Customer Satisfaction	It helps the user to know the risk factor at his location and mitigation suggestions support.
5	Business Model (Revenue Model)	It is a cheap software with affordable minimum requirements.
6	Scalability of the Solution	Better accuracy in intensity analysis and risk factor with mitigation measures is evaluated using this model.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

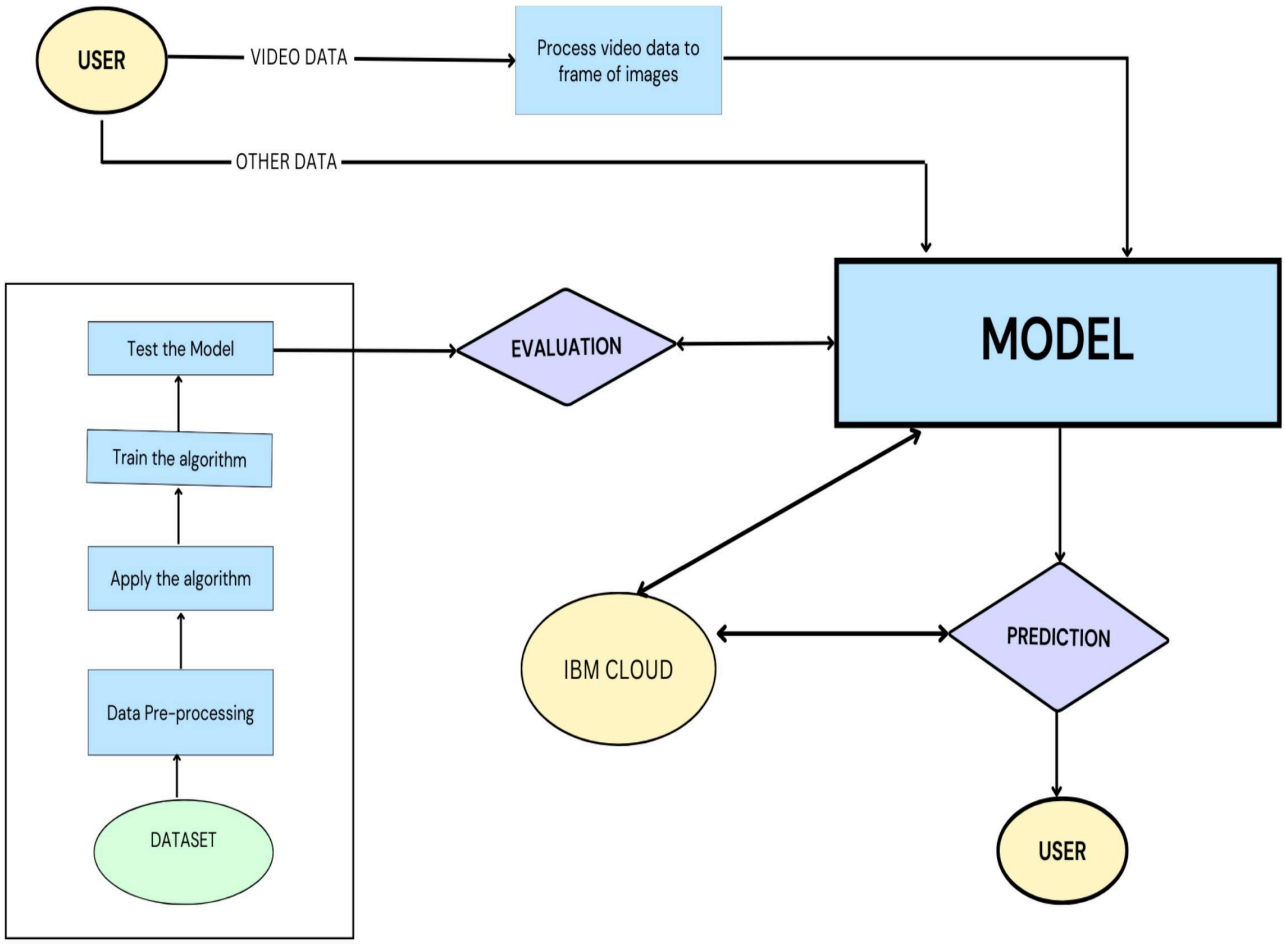
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form ,Registration through Gmail, Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Access to the location	Turn on the location.
FR-4	Input	Video feed from the camera.
FR-5	Processing	Processing the images and videos based on the available datas.
FR-6	Output	Alerts about the intensity of the disaster.

4.2 Non-Functional requirements

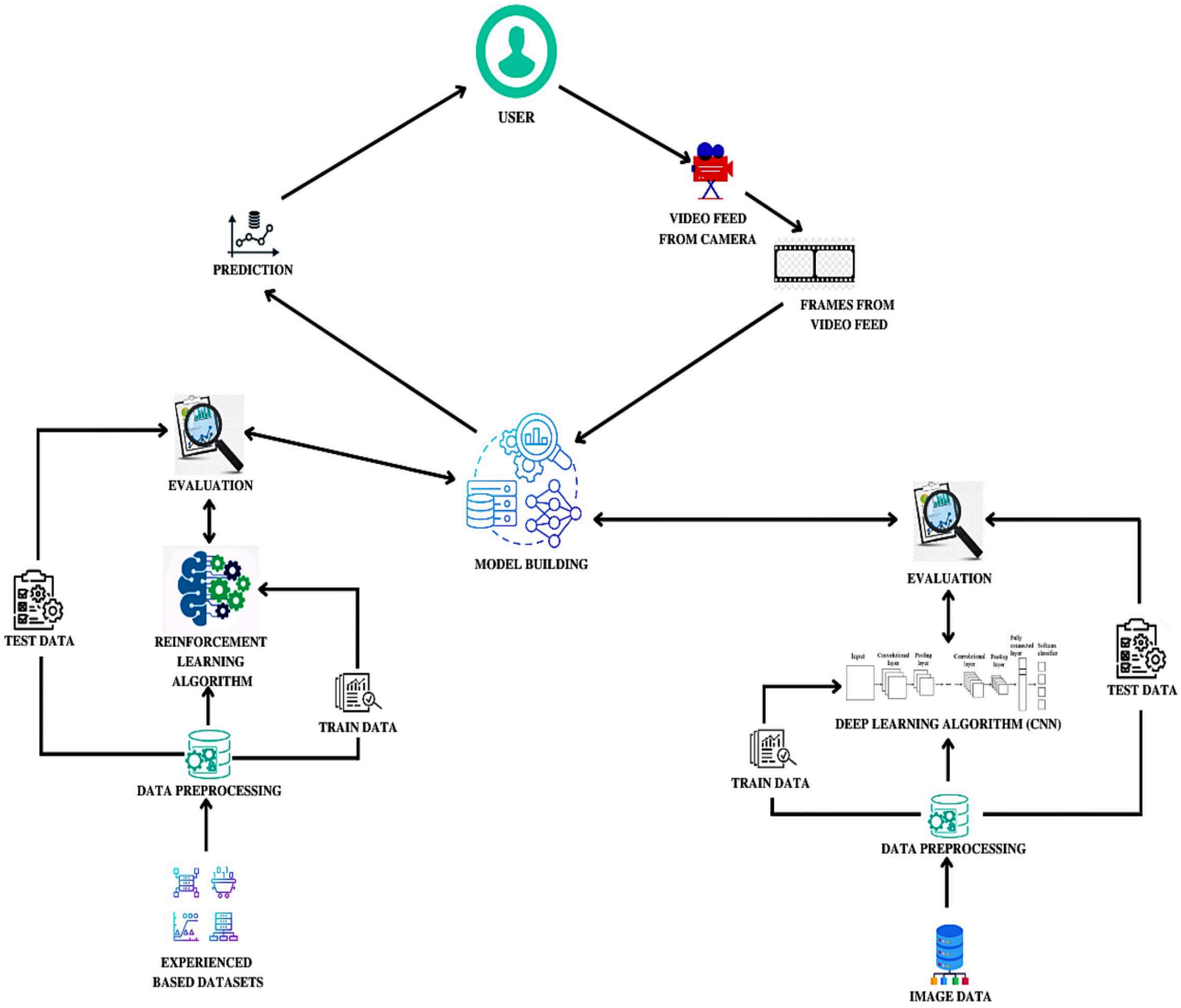
FR No.	Non-Functional Requirement	Description
NFR1	**Usability**	Can be used if there is proper internet connection and input.
NFR2	**Security**	Users must change the initially assigned login password immediately after the first successful login. Moreover, the initial should never be reused.
NFR3	**Reliability**	Gives accurate information and can be used any time.
NFR4	**Performance**	Good user experience and ease of operating the application. Latency is very low.
NFR5	**Availability**	The application is available all the time and can be accessed at any time.
NFR6	**Scalability**	Assesses the highest workloads. The application is scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance.

5. PROJECT DESIGN

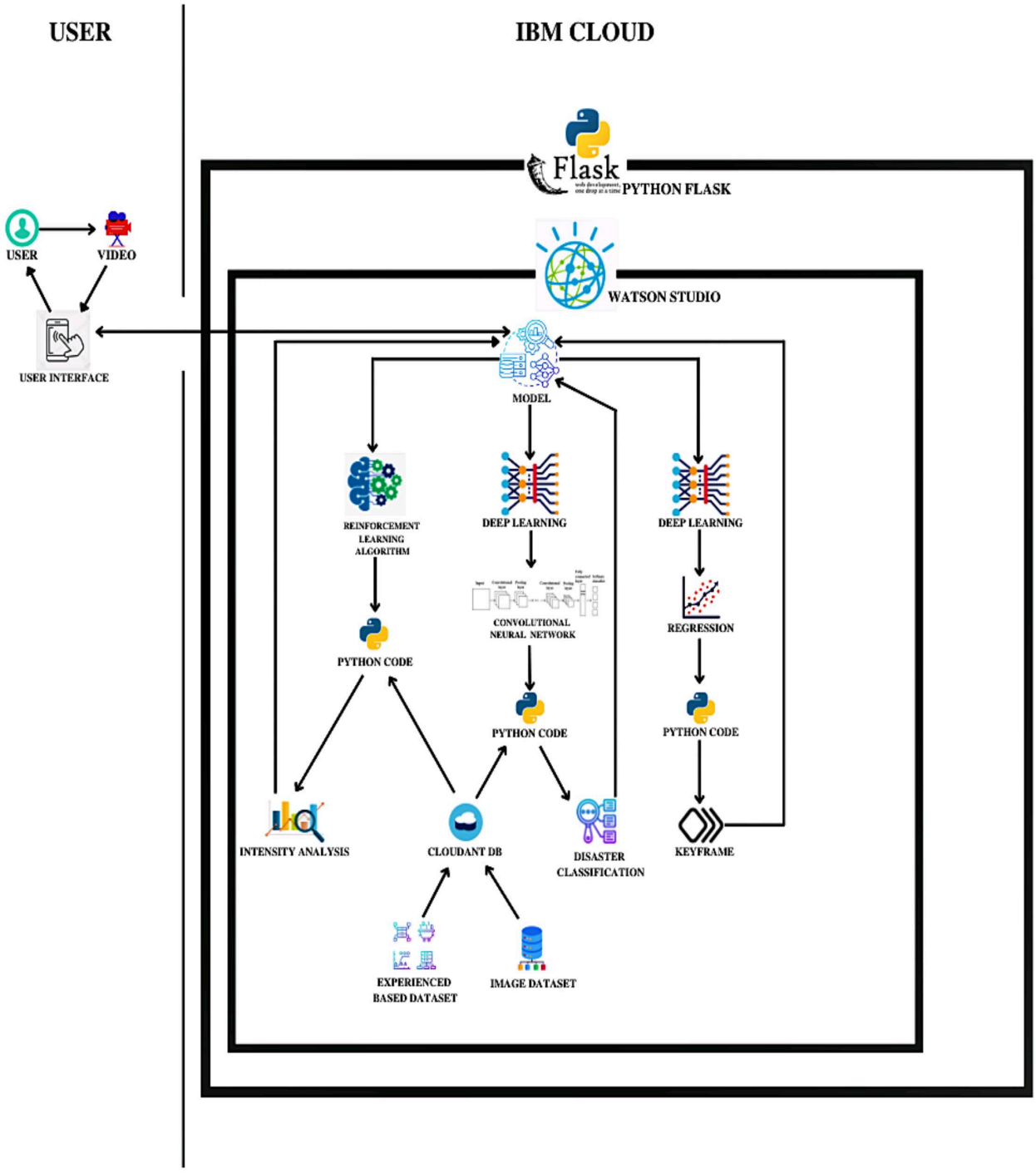
5.1 Data Flow Diagrams



5.2 Solution Architecture



5.3 Technical Architecture



5.4 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I will receive confirmation OTP once I have registered for the application	I can enter the OTP & click confirm	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can access my account / dashboard	Medium	Sprint-1
		USN-5	As a user, I can register for the application through LinkedIN	I can access my account / dashboard	Medium	Sprint-3
	Login	USN-6	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-7	As a user, I can access the information and services provided in the dashboard	I can access my dashboard	High	Sprint-2
Customer (Web user)	Login	USN-8	As a user, I can login to the web application and accessmy dashboard	I can access my account / dashboard using the registered login credentials	High	Sprint-1
Customer Care Executive	Help desk	USN-9	As a user, I can get the help from customer care	I can reach out to the help desk if there is any grievances	Medium	Sprint-3
Administrator	Management	USN-10	As an administrator, I can collect new datasets and keep the model trained	I can collect and train model with new dataset	Medium	Sprint-3
		USN-11	As an administrator, I can update the features of the application	I can feature to the application	Medium	Sprint-3
		USN-12	As an administrator, I can collect the data of user and maintain user history	I can maintain details of the user such as name, gmail, location etc	High	Sprint-1

		USN-13	As an administrator, I can maintain 3rd - party services	I can maintain services such as GPS etc	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

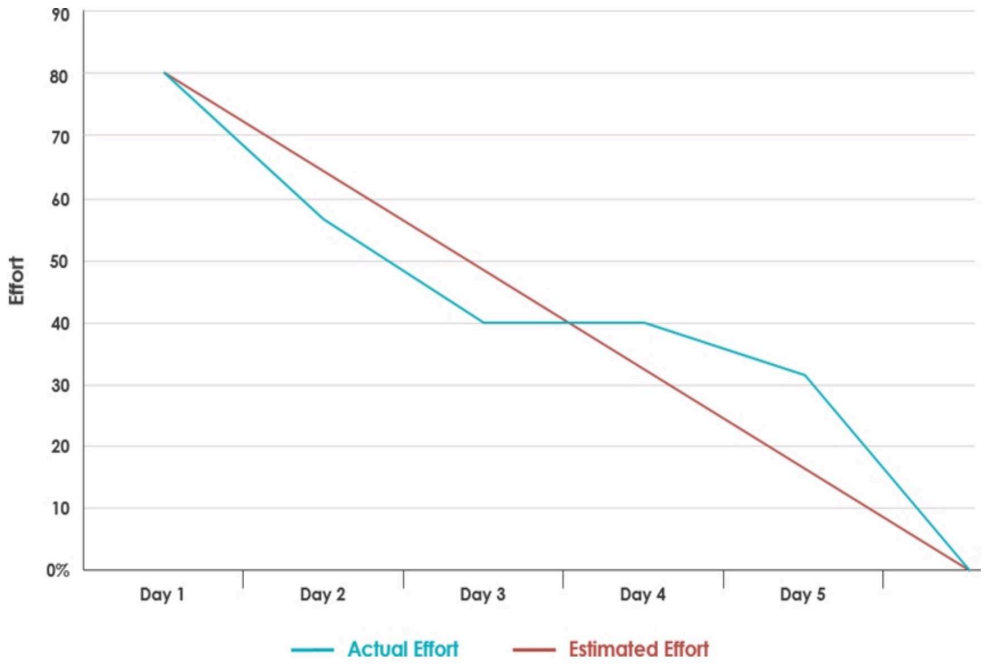
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Registration	USN-1	As a user, I can register for the website by entering my email, password, and confirming my password.	1	High	Nithyasree, Valliammai
Sprint-3		USN-2	As a user, I will receive confirmation email once I have registered for the application.	1	High	Nithyasree, Valliammai
Sprint-3		USN-3	As a user, I can register for the application through mobile number.	2	Medium	Nithyasree, Valliammai
Sprint-3		USN-4	As a user, I will receive confirmation SMS (OTP)	2	Medium	Nithyasree, Valliammai
Sprint-3	Login	USN-5	As a user, I can log into the application by entering email & password.	1	High	Nithyasree, Valliammai
Sprint-3		USN-6	As a user, I can upload image/video.	2	Medium	Nithyasree, Uthra
Sprint-3		USN-7	As a user, I get predicted output.	1	High	Nithyasree, Uthra
Sprint-3		USN-8	As a user, I can logout of the website.	3	Low	Nithyasree, Sidharth
Sprint-3		USN-9	As a developer, I can collect data and maintain user history.	3	Low	Sidharth, Nithyasree
Sprint-1	HTML pages	USN-10	As a developer, Build registration page using HTML	1	High	Uthra
Sprint-1		USN-11	As a developer, Build login page using HTML	1	High	Uthra

Sprint-1		USN-12	As a developer, Build home page using HTML	1	High	Uthra, Sidharth
Sprint-2		USN-13	As a developer, Build upload page using HTML	1	High	Uthra, Sidharth
Sprint-2		USN-14	As a developer, Build redirecting pages using HTML	3	Low	Uthra, Sidharth
Sprint-2		USN-15	As a developer, Build analysis report page using HTML	1	High	Uthra, Sidharth
Sprint-1		USN-16	As a developer, Build logout page using HTML	2	Medium	Uthra
Sprint-3		USN-17	As a developer, Getting access to location from user.	1	High	Nithyasree
Sprint-1	Dashboard	USN-18	As a developer, Build dashboard page using HTML	1	High	Uthra, Sidharth
Sprint-1		USN-19	As a developer, collect dataset to train and test model.	1	High	Uthra, Sidharth, Nithyasree, Valliammai
Sprint-1	Image preprocessing	USN-20	As a developer, import imagedatagenerator library to perform image augmentation	2	Medium	Sidharth, Nithyasree
Sprint-1		USN-21	As a developer, configure imagedatagenerator library to augment images	2	Medium	Sidharth
Sprint-1		USN-22	As a developer, applying imagedatagenerator library to trainset and testset images	1	High	Sidharth
Sprint-2	Model building	USN-23	As a developer, Train-test and save model.	1	High	Uthra, Sidharth
Sprint-2	Application building	USN-24	As a developer, Build python code.	1	High	Nithyasree, Uthra
Sprint-3		USN-25	As a developer, Run the application.	1	High	Nithyasree, Sidharth
Sprint-4	Cloud integration	USN-26	As a developer, Integrate the application with IBM cloud.	1	High	Nithyasree, Valliammai
Sprint-4		USN-27	As a developer, Train the model on IBM	1	High	Uthra,

			Cloud.			Sidharth, Nithyasree,
Sprint-4		USN-28	As a developer, Upload datasets to Cloudant DB	3	Low	Sidharth, Valliammai
Sprint-4		USN-29	As a developer, Fetch data from Cloudant DB to perform manipulations on code	2	Medi um	Nithyasree, Valliammai
Sprint-4		USN-30	As a developer, Get feedback from the user	1	High	Valliammai
Sprint-4		USN-31	As a developer, update features on website	3	Low	Uthra, Nithyasree
Sprint-4		USN-32	As a developer, work on bugs	2	Medi um	Sidharth, Valliammai
Sprint-4		USN-33	As a developer, track the progress of team using JIRA Software	1	High	Sidharth
Sprint-4	Analysis	USN-34	As a developer, perform user acceptance test	1	High	Sidharth, Valliammai
Sprint-4		USN-35	As a developer, performance testing is done to check the efficiency of the code	1	High	Uthra, Valliammai
Sprint-4		USN-36	As a developer, collect & analyse testcase report	1	High	Uthra, Nithyasree
Sprint-4		USN-37	As a developer, Submitting the overall project report	1	High	Uthra, Sidharth

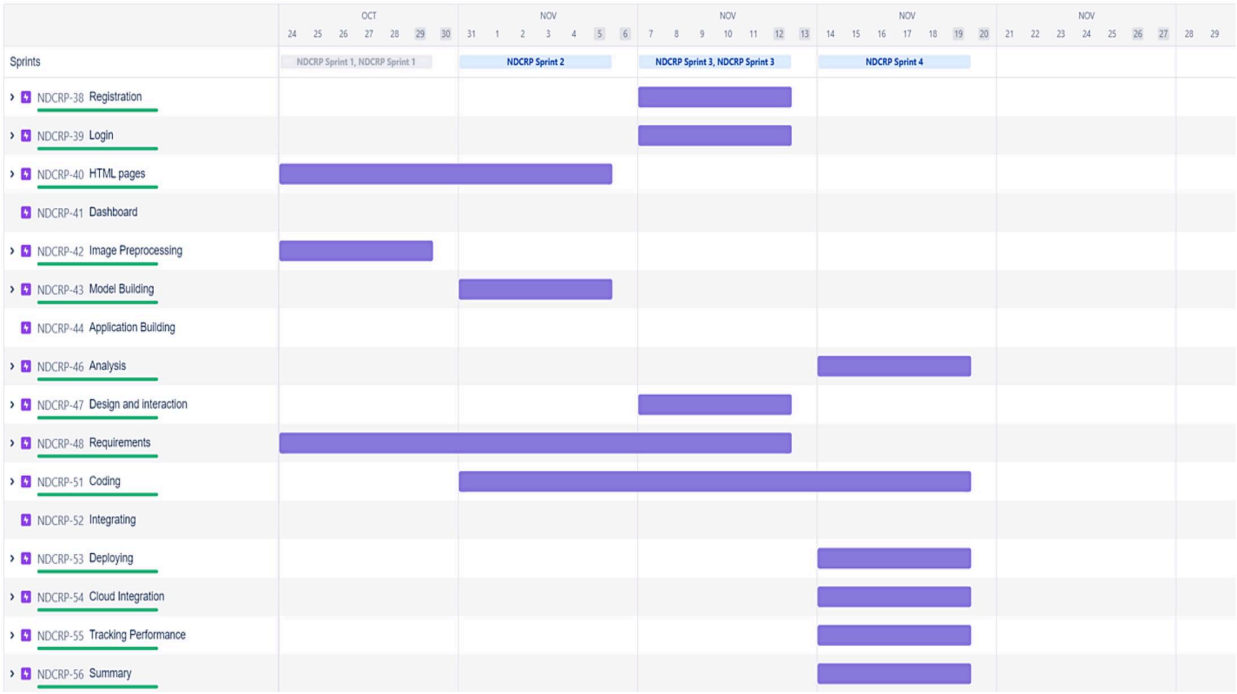
6.2 Sprint Delivery Schedule

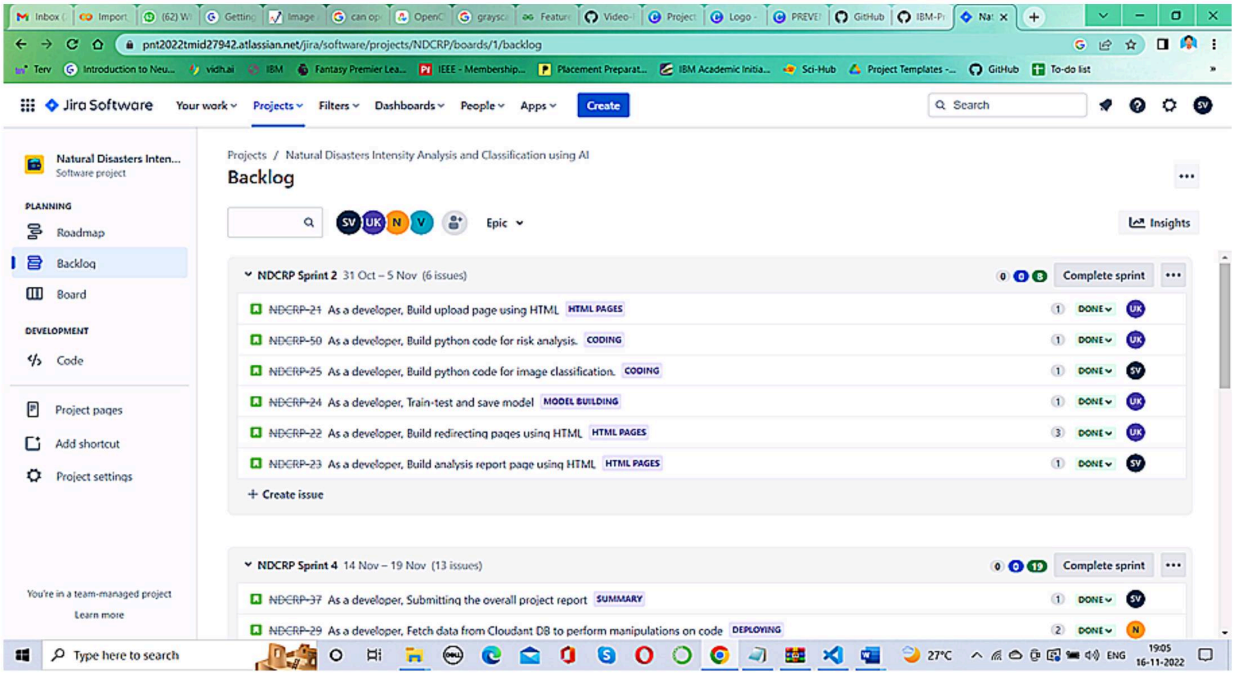
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

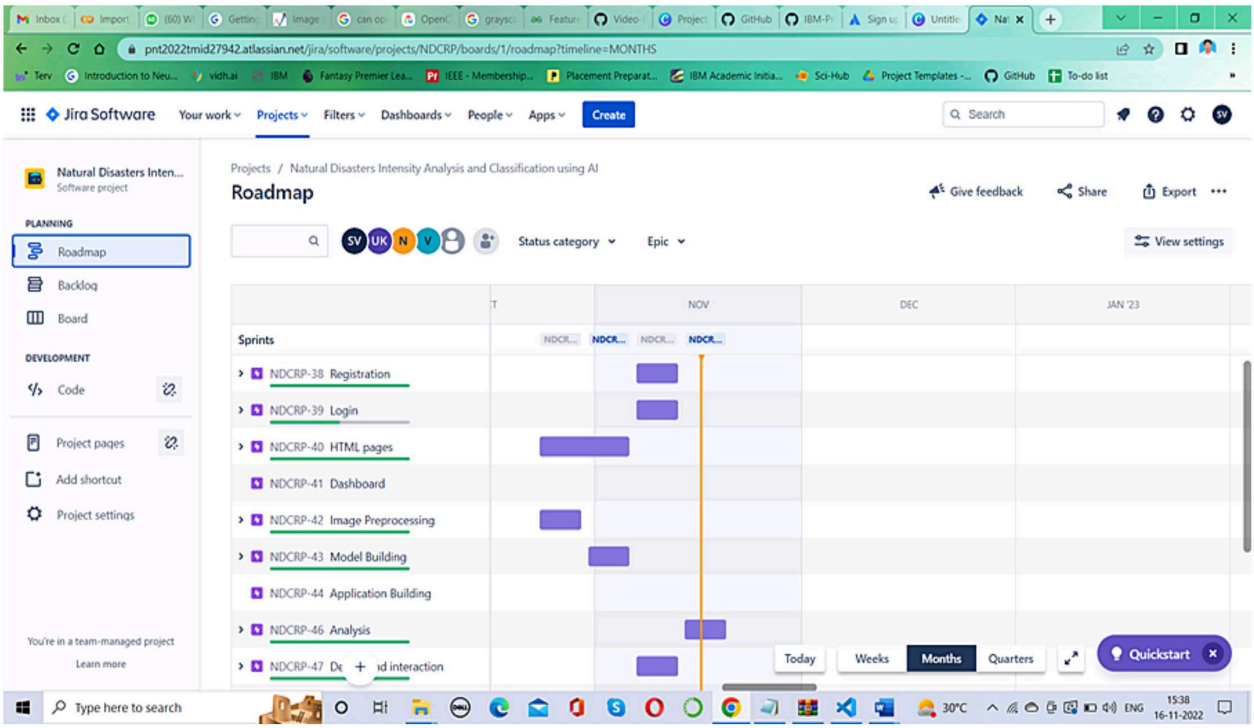


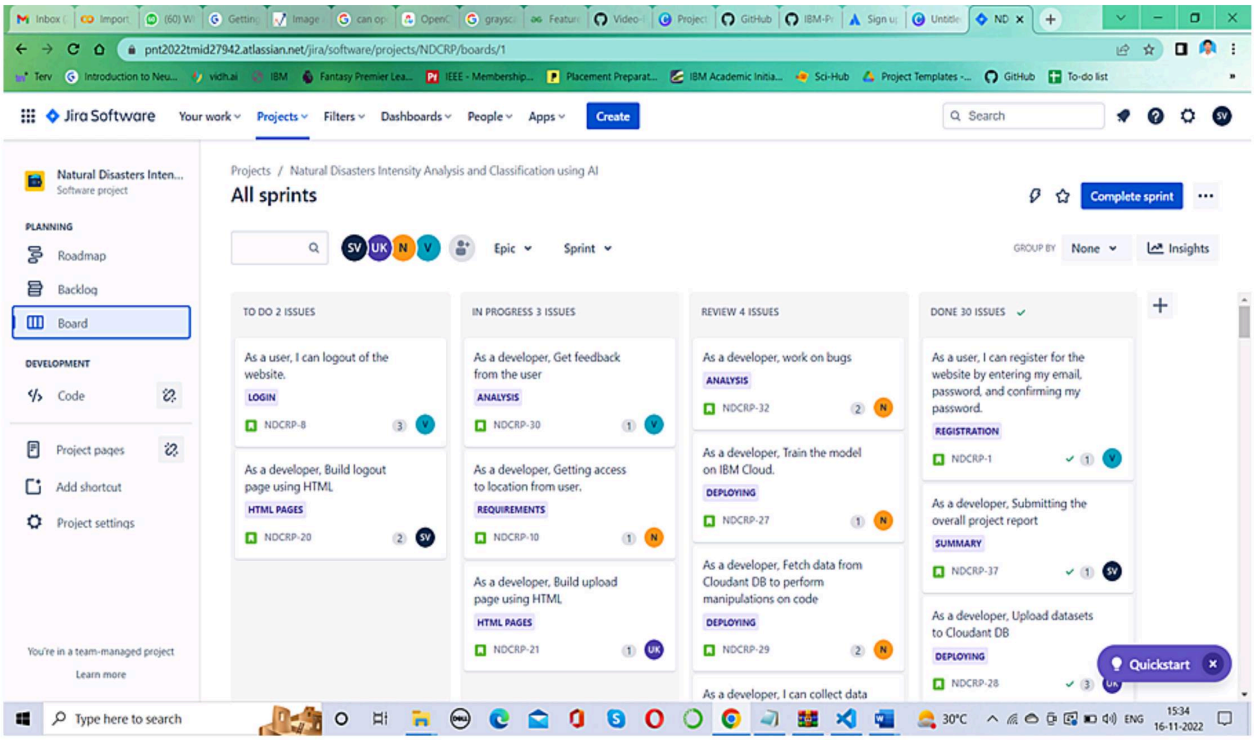
6.3 Reports from JIRA

Road Map






pnt2022tmid27942.atlassian.net/jira/software/projects/NDCRP/boards/1/roadmap?timeline=MONTHS
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Natural Disasters Inten...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Natural Disasters Intensity Analysis and Classification using AI
Roadmap
Give feedback
Share
Export
Status category
Epic
View settings
NOV
DEC
JAN '23
Sprints
NDCRP-38 Registration
NDCRP-39 Login
NDCRP-40 HTML pages
NDCRP-41 Dashboard
NDCRP-42 Image Preprocessing
NDCRP-43 Model Building
NDCRP-44 Application Building
NDCRP-46 Analysis
Today
Weeks
Months
Quarters
Quickstart
Type here to search
30°C
ENG
15:38
16-11-2022

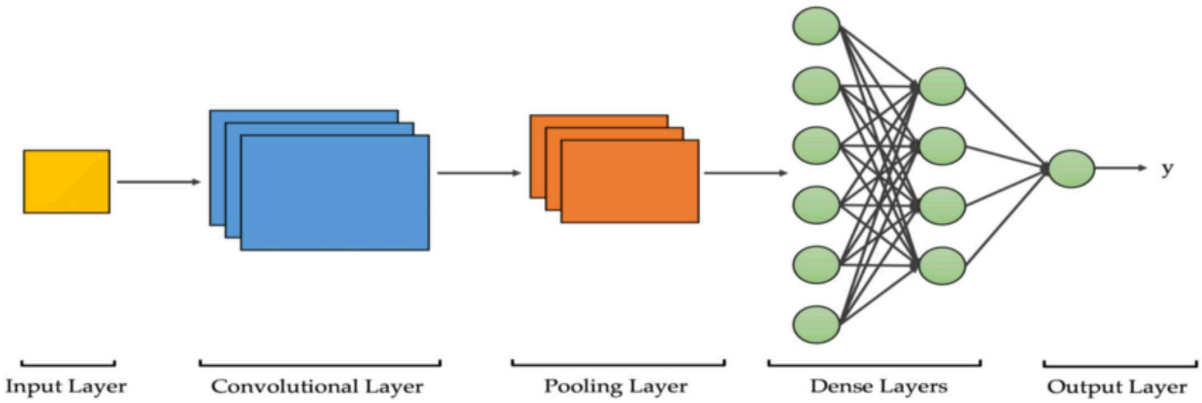

pnt2022tmid27942.atlassian.net/jira/software/projects/NDCRP/boards/1
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Natural Disasters Inten...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Natural Disasters Intensity Analysis and Classification using AI
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO 2 ISSUES
As a user, I can logout of the website.
LOGIN
NDCRP-8
As a developer, Build logout page using HTML
HTML PAGES
NDCRP-20
IN PROGRESS 3 ISSUES
As a developer, Get feedback from the user
ANALYSIS
NDCRP-30
As a developer, Getting access to location from user.
REQUIREMENTS
NDCRP-10
As a developer, Build upload page using HTML
HTML PAGES
NDCRP-21
REVIEW 4 ISSUES
As a developer, work on bugs
ANALYSIS
NDCRP-32
As a developer, Train the model on IBM Cloud.
DEPLOYING
NDCRP-27
As a developer, Fetch data from Cloudant DB to perform manipulations on code
DEPLOYING
NDCRP-29
DONE 30 ISSUES
As a user, I can register for the website by entering my email, password, and confirming my password.
REGISTRATION
NDCRP-1
As a developer, Submitting the overall project report
SUMMARY
NDCRP-37
As a developer, Upload datasets to Cloudant DB
DEPLOYING
NDCRP-28
Quickstart
Type here to search
30°C
ENG
15:34
16-11-2022

7. CODING & SOLUTIONING

7.1 Feature 1 (Convolutional Neural Network)

Feature Overview:

A Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification. Deep Learning thus recognizes objects in an image by using a CNN. CNNs are playing a major role in diverse tasks/functions like image processing problems, computer vision tasks like localization and segmentation, video analysis, to recognize obstacles in self-driving cars, as well as speech recognition in natural language processing.




```
#-----------Import libraries
#import the necessary libraries
import tensorflow
from tensorflow import keras
from keras import Sequential
from keras.layers import
Convolution2D,MaxPooling2D,Flatten,Dense,Dropout
from keras.preprocessing.image import ImageDataGenerator as idm
import numpy as np
import warnings
warnings.filterwarnings('ignore')

#------------Mount google drive
```

```
from google.colab import drive
drive.mount('/content/drive')

#----------Unzip dataset
pwd
ls
cd /content/drive/MyDrive/IBM_DATASET/
ls
!unzip training_set.zip

#----------Configure and apply imagedatagenerator
#Configure train_datagen
train_datagen = idm(rescale=1./255,zoom_range=0.2,horizontal_flip=True,
vertical_flip=False)
#Configure test_datagen
test_datagen = idm(rescale=1./255)
Xtrain =
train_datagen.flow_from_directory('/content/drive/MyDrive/IBM_DATASET/tr
ain_set',target_size=(76,76), batch_size=100,class_mode='categorical',
color_mode = "rgb")
Xtest =
test_datagen.flow_from_directory('/content/drive/MyDrive/IBM_DATASET/tes
t_set',target_size=(76,76), batch_size=100,class_mode='categorical',
color_mode = "rgb")

#----------Display class indices
Xtrain.class_indices
Xtest.class_indices

#----------Create Model
#Defining model
model=Sequential()

#----------Build and Configure model
#Convolution Layer 1
```

```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(76,76,3
)))
model.add(MaxPooling2D(pool_size=(2,2)))
#Convolution Layer 2
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(76,76,3
)))
model.add(MaxPooling2D(pool_size=(2,2)))
#Convolution Layer 3
model.add(Convolution2D(64,(3,3),activation='relu',input_shape=(76,76,3
)))
model.add(MaxPooling2D(pool_size=(2,2)))
#Convolution Layer 4
model.add(Convolution2D(64,(3,3),activation='relu',input_shape=(76,76,3
)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(3,activation='softmax'))

#-----------Model Summary
model.summary()

#-----------Compile the model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=
['accuracy'])

#-----------Train Model
#Model fitting - training and validation
history = model.fit_generator(Xtrain, steps_per_epoch= len(Xtrain),
epochs=15, validation_data=Xtest, validation_steps= len(Xtest))

#-----------Test Model
from tensorflow.keras.preprocessing import image

test_img=image.load_img('/content/drive/MyDrive/IBM_DATASET/test_set/Ea
```

```
rthquaqe/Copy of 22.jpg',target_size=(76,76))
test_img

x=image.img_to_array(test_img)
x=np.expand_dims(x,axis=0)
predicted=np.argmax(model.predict(x))
Prediction_category=['Cyclone','Earthquake','Flood']
Prediction_category[predicted]

#----------Save Model
model.save('CNN_Model_for_Disaster_Classification.h5')

#----------Plot Accuracy graph
#importing dependencies to plot the graph
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
#Training and Validation Accuracy Plots
epochs_range = range(15)

plt.figure(figsize=(6,6))
plt.plot(epochs_range, history.history['accuracy'], label='Training
Accuracy')
plt.plot(epochs_range, history.history['val_accuracy'],
label='Validation Accuracy')
plt.legend()
plt.title('Training and Testing Accuracy')
plt.show()

#----------Plot Loss graph
#Training and Validation Loss Plot
plt.figure(figsize=(6,6))
plt.plot(epochs_range, history.history['loss'], label='Training Loss')
plt.plot(epochs_range, history.history['val_loss'], label='Validation
Loss')
plt.legend()
```

```
plt.title('Training and Testing Loss')
plt.show()

#-----------Confusion Matrix
from sklearn.metrics import classification_report, confusion_matrix
Y_pred = model.predict_generator(Xtest,500 // 100)
y_pred = np.argmax(Y_pred, axis=1)
print('Confusion Matrix')
print(confusion_matrix(Xtest.classes, y_pred))

#-----------Classification Report
print('Classification Report')
target_names = ['Cyclone', 'Earthquake', 'Flood']
print(classification_report(Xtest.classes, y_pred,
target_names=target_names))
```

This CNN model has comparatively more layers (i.e. 3layers) which increases the efficiency of the model to classify accurately and the capacity of the model gets increased as the number of layers get increased. The image input is processed through the different network layers of the model and the class of which image belongs to is been predicted. Here we have 3 classes: Cyclone, Flood , Earthquake. The model predicts and reports any one of the class.

7.2 Feature 2 (Key Frame Separation Model)

Feature Overview:

In accessing large collections of digitized videos, it is often difficult to find both the appropriate video file and the portion of the video that is of interest. We use keyframes to summarize videos, and to provide access points into them . Hence, the classification becomes more accurate.

```
#------------Mount google drive
from google.colab import drive
drive.mount('/content/drive')
```

```
#------------Import Dependencies
import os
import cv2
import subprocess

#------------Function
def get_frame_types(video_fn):
    command = 'ffprobe -v error -show_entries frame=pict_type -of
default=noprint_wrappers=1'.split()
    out = subprocess.check_output(command + [video_fn]).decode()
    frame_types = out.replace('pict_type=','').split()
    return zip(range(len(frame_types)), frame_types)

def save_i_keyframes(video_fn):
    frame_types = get_frame_types(video_fn)
    i_frames = [x[0] for x in frame_types if x[1]=='I']
    if i_frames:
        basename = os.path.splitext(os.path.basename(video_fn))[0]
        cap = cv2.VideoCapture(video_fn)
        for frame_no in i_frames:
            cap.set(cv2.CAP_PROP_POS_FRAMES, frame_no)
            ret, frame = cap.read()
            outname = 'Key_frame_'+str(frame_no)+'.jpg'
            cv2.imwrite(outname, frame)
            print (outname)
        cap.release()

    else:
        print ('No I-frames in '+video_fn)

if __name__ == '__main__':
    save_i_keyframes(filename)
```

7.3 Feature 3 (Reinforcement Learning Model)

Feature Overview:

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience. The RL process is as follows:

- Observe ---> Environment of the Agent
- Decide ---> Decision as per Observations
- Act ---> Action on the Decision
- Receive ---> Getting Rewarded or Penalised as per the Action
- Learn ---> from previous Actions and improve
- Iterate ---> Repeat the entire process till Success

7.3.1 RL Model for Earthquake

```
#------------Mount google drive
from google.colab import drive
drive.mount('/content/drive')

#------------Read data
df=pd.read_csv( path)

#------------Plot Earthquake prone locations
sns.scatterplot(x='latitude',y='longitude',data=df)
plt.title('Earthquake Prone Locations ')

#------------Import Dependencies
```

```
import gym
from gym import Env
from gym.spaces import Discrete, Box, Dict, Tuple, MultiBinary,
MultiDiscrete
import numpy as np
import random
import os
from stable_baselines3 import PPO
from stable_baselines3.common.vec_env import VecFrameStack
from stable_baselines3.common.evaluation import evaluate_policy

#----------Building an Environment
class EarthquakeEnv(Env):
    def __init__(self):
        # Actions we can take: hide, stay, vacate
        self.action_space = Discrete(3)
        # Plate movement array
        self.observation_space = Box(low=np.array([0]),
high=np.array([100]))
        # Set start magnitude
        self.state = 5 + random.randint(-3,3)
        # Set decision duration
        self.decision_duration = 60

    def step(self, action):
        # Apply action
        # 0 -1 = -1 magnitude
        # 1 -1 = 0
        # 2 -1 = 1 magnitude
        self.state += action -1
        # Reduce decision duration by 1 second
        self.decision_duration -= 1

        # Calculate reward
        if self.state >=5 and self.state <=20:
            reward =1
```

```
        else:
            reward = -1

        # Check if time is over
        if self.decision_duration <= 0:
            done = True
        else:
            done = False

        # Apply magnitude error
        #self.state += random.randint(-1,1)
        # Set placeholder for info
        info = {}

        # Return step information
        return self.state, reward, done, info

    def render(self):
        # Implement viz
        pass

    def reset(self):
        # Reset magnitude
        self.state = np.array([15 + random.randint(-3,3)]).astype(float)
        # Reset decision duration
        self.decision_duration = 60
        return self.state
env = EarthquakeEnv()
env.observation_space.sample()

#----------Test Environment
episodes = 80
for episode in range(1, episodes+1):
    state = env.reset()
    done = False
    score = 0
```

```
    while not done:
        env.render()
        action = env.action_space.sample()
        n_state, reward, done, info = env.step(action)
        score+=reward
    print('Episode:{} Score:{}'.format(episode, score))

#-----------Train Model
log_path = os.path.join('Training', 'Logs')
model = PPO("MlpPolicy", env, verbose=1, tensorboard_log=log_path)
model.learn(total_timesteps=10000)

#-----------Save Model
model.save('PPO')
evaluate_policy(model, env, n_eval_episodes=80)
```

The seismic magnitude is initialised '5' and the decision duration being 60sec , for a wrong desicion at that episode for the concerned state, after the action the agent is penalised and gets 59 seconds only for the decision making during the next state.For the correct decision the agent is rewarded +1 sec for decision making. Here the 3 actions are: Hide(-1) , Stay(0) , Evacuate(+1). On a longer run the model becomes self sufficient and improves the decision making during risk.

7.3.2 RL Model for Flood & Cyclone

```
#-----------Import Dependencies
import gym
from gym import Env
from gym.spaces import Discrete, Box, Dict, Tuple, MultiBinary,
MultiDiscrete
import numpy as np
import random
import os
from stable_baselines3 import PPO
from stable_baselines3.common.vec_env import VecFrameStack
```

```
from stable_baselines3.common.evaluation import evaluate_policy

#-----------Building an Environment
class Water_DisasterEnv(Env):
    def __init__(self):
        # Actions we can take: closed room, stay, vacate
        self.action_space = Discrete(3)
        # Temperature array
        self.observation_space = Box(low=np.array([0]),
high=np.array([100]))
        # Set start rainfall
        self.state = 10 + random.randint(-3,3)
        # Set decision duration
        self.decision_duration = 60

    def step(self, action):
        # Apply action
        # 0 -1 = -1 rainfall
        # 1 -1 = 0
        # 2 -1 = 1 rainfall
        self.state += action -1
        # Reduce decision duration by 1 minute
        self.decision_duration -= 1

        # Calculate reward
        if self.state >=10 and self.state <=20:
            reward =1
        else:
            reward = -1

        # Check if time is over
        if self.decision_duration <= 0:
            done = True
        else:
            done = False
```

```
            # Apply rainfall fluctuations
            #self.state += random.randint(-1,1)
            # Set placeholder for info
            info = {}

            # Return step information
            return self.state, reward, done, info

        def render(self):
            # Implement viz
            pass

        def reset(self):
            # Reset rainfall
            self.state = np.array([15 + random.randint(-3,3)]).astype(float)
            # Reset decision duration
            self.decision_duration = 60
            return self.state
env = Water_DisasterEnv()
env.observation_space.sample()

#-----------Test Environment
episodes = 100
for episode in range(1, episodes+1):
    state = env.reset()
    done = False
    score = 0

    while not done:
        env.render()
        action = env.action_space.sample()
        n_state, reward, done, info = env.step(action)
        score+=reward
    print('Episode:{} Score:{}'.format(episode, score))

#-----------Train Model
```

```
log_path = os.path.join('Training', 'Logs')
model = PPO("MlpPolicy", env, verbose=1, tensorboard_log=log_path)
model.learn(total_timesteps=20000)

#----------Save Model
model.save('PPO')
evaluate_policy(model, env, n_eval_episodes=100)
```

The rainfall measure is initialised '10" and the decision duration being 60sec , for a wrong desicion at that episode for the concerned state, after the action the agent is penalised and gets 59 seconds only for the decision making during the next state.For the correct decision the agent is rewarded +1 sec for decision making. The reward is granted for right decision only when the rainfall measure is between 10 and 20. Here the 3 actions are: Hide(-1) , Stay(0) , Evacuate(+1). On a longer run the model becomes self sufficient and improves the decision making during risk.

7.4 Database Schema

Training Image dataset link:
https://drive.google.com/file/d/1JCl8bDZQNOYdSdCUcJHgUWlRlKvoekfD/view?usp=drive_web

Testing Image dataset link:
https://drive.google.com/file/d/1T34RSRag3M8lS58SqoluGdmXNWnc_Ez/view?usp=drive_web

CSV files:
https://drive.google.com/file/d/1Gn55xF2C8lMHleZ6pD5J2UE5FruLhbAQ/view?usp=share_link
https://drive.google.com/file/d/1s37TRvWmFomrG15zEEiEtTW5bvrg-3Gu/view?usp=share_link

8. TESTING

8.1 Test Cases

https://docs.google.com/spreadsheets/d/132Lf2Efx5a6CRVcB1k07XXoFiXjuPU8gyZgdFevWNw/edit?usp=sharing

8.2 User Acceptance Testing

1.PURPOSE OF DOCUMENT:

The purpose of this document is to briefly explain the test coverage and open issues of the project at the time of the release to User Acceptance Testing (UAT).

2.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	15	6	4	5	30
Duplicate	0	0	4	1	5
External	2	4	0	1	7
Fixed	20	5	4	10	39
Not Reproduced	0	1	0	0	1
Skipped	0	0	1	0	1
Won't Fix	1	4	3	1	9
Totals	38	20	16	18	92

3. TEST CASE ANALYSIS:

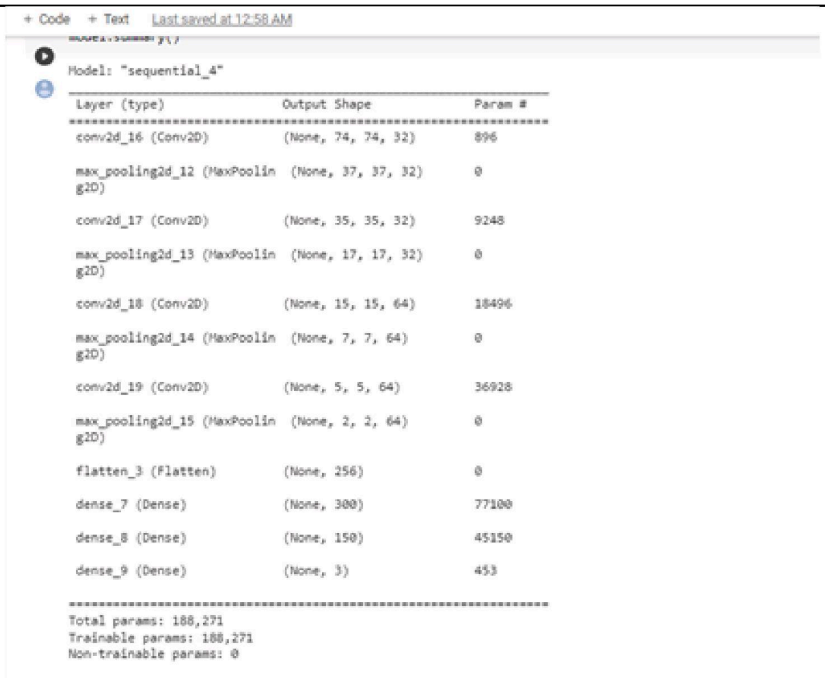
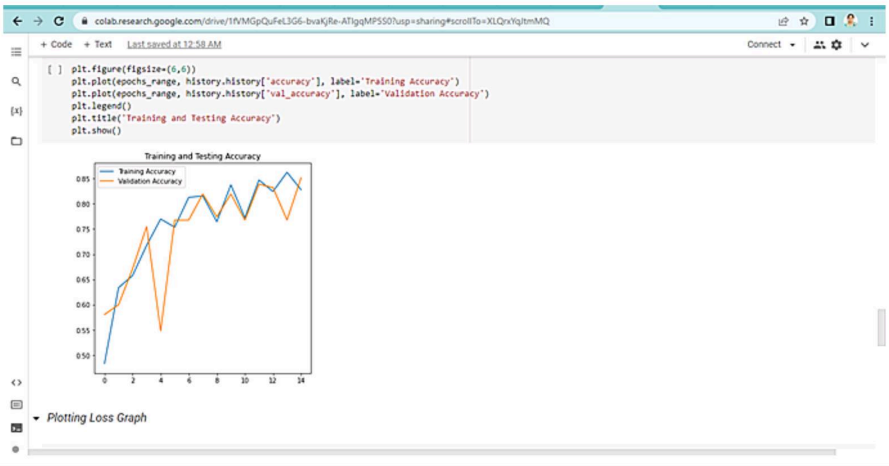
This report shows the number of test cases that have passed, failed, and untested

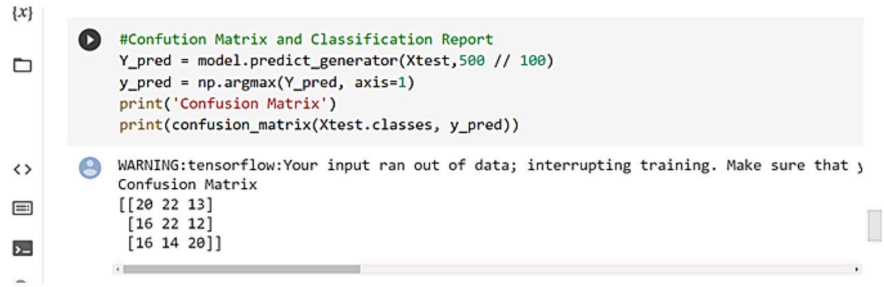
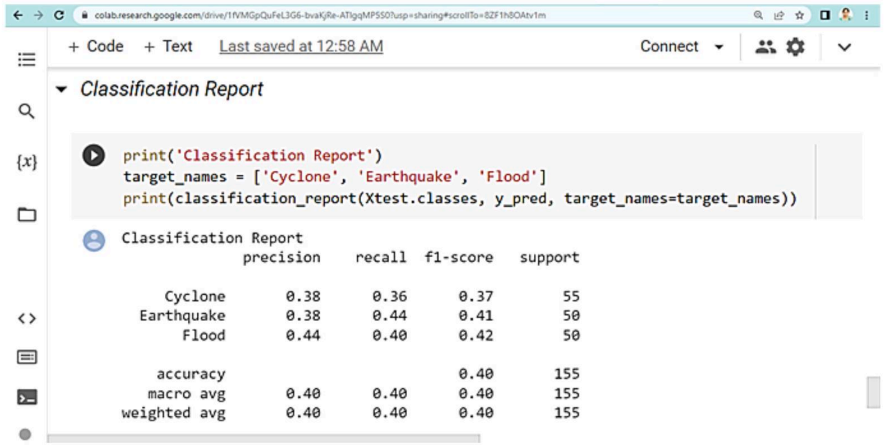
SECTION	TOTAL CASES	NOT TESTED	FAIL	PASS
Print Engine	5	0	0	5
Client Application	25	0	2	23
Security	3	0	0	3
Exception Reporting	7	0	0	7
Final Report Output	4	0	0	4
Version Control	1	0	0	1

9. RESULTS

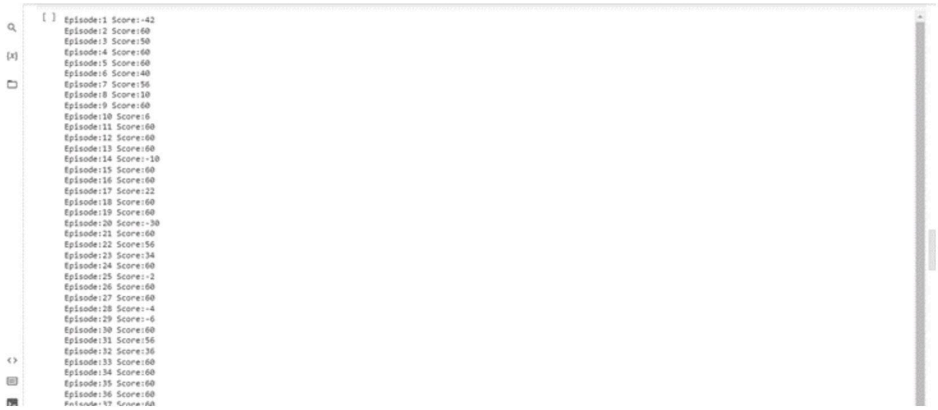
9.1 Performance Metrics

- *For CNN Model*

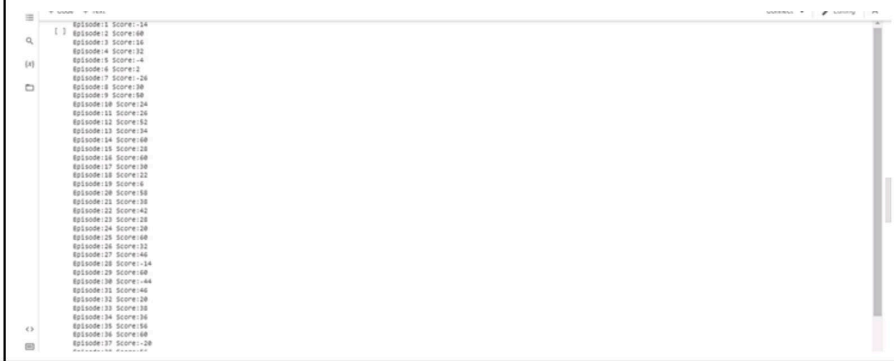
S.No	Parameter	Screenshot
1.	Model Summary	
2.	Accuracy	

3.	Loss	
4.	Confusion Matrix	
5.	Classificati on Report	

- For RL Model-1 (Earthquake)

S.No.	Parameter	Screenshot
1.	Episode and Score	
2.	Evaluation Policy	

- For RL Model-2 (Flood and Cyclone)

S.No.	Parameter	Screenshot
1.	Episode and Score	
2.	Evaluation Policy	

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- This model supports more accurate classification of the natural disaster.
- It does highly efficient prediction of the type of disaster due to increased capacity.
- It also supports in finding the level of risk and next mitigated action based on the self-sufficiency model.
- The model will be able to guide in selection of the next action to be performed by the user during occurrence of a disaster.

10.2 Disadvantages

- The scope of the Reinforcement learning model has to be widened to more number of actions which will happen by test and trials on its own in a longer run.
- In real time there will be more number of special cases which takes time to be handled.

11. CONCLUSION

To tackle the problem of prediction of the classification and risk analysis of occurring Natural Disaster more efficiently , we have developed a multi-layered deep convolutional neural network model that classifies the natural disaster and a reinforcement learning based model that tells the action to be performed based on the intensity of disaster . The model separates the key frames from the video for processing. The deployed CNN model works with an accuracy of 85.4% and the Reinforcement Learning models have lower deviation values ranging from minimum of '0' to maximum of '5'. This model is self –sufficient and can produce more efficient results in a longer run. The predictions has high accuracy rate and this model will be highly useful in real time.

12. FUTURE SCOPE

This model can be used by adding more number of layers to increase the model capacity to classify more number of natural disasters most efficiently. The Reinforcement Model will be self- sufficient and thus will add more number of states and actions and will be able to guide more appropriately. This project can be extended for other natural disaster classifications and become more generic world-wide.

13. APPENDIX

13.1 Source Code

https://github.com/IBM-EPBL/IBM-Project-18936-1659691440/tree/main/Project/Final%20Deliverables/6.%20Deployment/6.1%20DevOps%20code

13.2 GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-18936-1659691440

https://github.com/IBM-EPBL/IBM-Project-18936-1659691440/tree/main/Project/Final%20Deliverables/1.%20Project%20Report

14. REFERENCES

Papers:

[1] Pi,Y., Nath, N.D. and Behzadan, A.H., 2020. Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. Advanced Engineering Informatics, 43,p.101009.

[2] Bui, D.T., Hoang, N.D., Martínez-Álvarez, F., Ngo, P.T.T., Hoa, P.V., Pham, T.D., Samui, P. and Costache, R., 2020. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. Science of The Total Environment, 701, p.134413.

[3] Ahmad, K., Riegler, M., Pogorelov, K., Conci, N., Halvorsen, P. and De Natale, F., 2017, June. Jord: a system for collecting information and monitoring natural disasters by linking social media with satellite imagery. In Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing (pp. 1-6).

[4] Kovordányi, R. and Roy, C., 2009. Cyclone track forecasting based on satellite images using artificial neural networks. ISPRS Journal of Photogrammetry and Remote Sensing, 64(6),pp.513-521.

[5] Dawood, M. and Asif, A., 2019. Deep-PHURIE: deep learning based hurricane intensity estimation from infrared satellite imagery. Neural Computing and Applications, pp.1-9.

[6] Cao, Q.D. and Choe, Y., 2020. Building damage annotation on post-hurricane satellite imagery based on convolutional neural networks. Natural Hazards, pp.1-20.

[7] Doshi, J., Basu, S. and Pang, G., 2018. From satellite imagery to disaster insights. arXiv preprintar Xiv:1812.07033.

[8] Pritt, M. and Chern, G., 2017, October. Satellite image classification with deep learning. In 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR) (pp. 1-7). IEEE.

[9] Brinker, T.J., Hekler, A., Utikal, J.S., Grabe, N., Schadendorf, D., Klode, J., Berking, C., Steeb, T., Enk, A.H. and von Kalle, C., 2018. Skin cancer classification using convolutional neural networks: systematic review. Journal of medical Internet research, 20(10), p.e11936.

Links:

https://docs.opencv.org/3.4/d4/da8/group__imgcodecs.html

https://stackoverflow.com/questions/42798634/extracting-keyframes-python-opencv

https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/

https://stackoverflow.com/questions/67303001/plot-confusion-matrix-with-keras-data-generator-using-sklearn

https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy

https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy

https://medium.com/@myworldsharma.jay/key-frame-extraction-from-video-9445564eb8ed

https://towardsdatascience.com/beginners-guide-to-custom-environments-in-openai-s-gym-989371673952

https://towardsdatascience.com/create-your-own-reinforcement-learning-environment-beb12f4151ef

https://towardsdatascience.com/openai-gym-from-scratch-619e39af121f