AI – NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION

A NALAYATHIRAN PROJECT REPORT

Submitted by

S.NO	NAME	REGISTER NUMBER
1	R. Mahalakshmi	821319106009
2	A. Mathumitha	821319106010
3	M. Priyanka	821319106013
4	Roshan raji	821319106015
5	U. Sriram	821319106018

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

HX8001 – "PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTERNEPRENEURSHIP

PARISUTHAM INSTITUTE OF TECHNOLOGY AND SCIENCE

Ring road, Nanjikottai, Thanjavur -613 006

ANNA UNIVERSITY, CHENNAI 60025

2022-2023

ABSTRACT

Natural disasters are inevitable, and the occurrence of disasters drastically affects the economy, ecosystem and human life. Buildings collapse, ailments spread and sometimes natural disasters such as tsunamis, earthquakes, and forest fires can devastate nations. When earthquakes occur, millions of buildings collapse due to seismological effects. 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. To tackle this problem, we developed a multilayered deep convolutional neural network (CNN) model that classifies the natural disaster and tells the intensity of disaster of natural 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.INTRODUCTION

1.1 PROJECT OVERVIEW

As the population is growing rapidly, people need to acquire land to live on, and as a result the ecosystem is disturbed horrifically, which causes global warming and increases the number of natural disasters. Populations in underdeveloped countries cannot afford damages disasters cause to infrastructures. The aftermath of disasters leaves the humans in miserable situations, and sometimes the devastating effects cannot be detected; additionally, rescue operations cannot take place in most of the places and victims are unable to be identified due to geographical factors of the different areas. Disasters spread rapidly in dense areas, difficult to carry out; in this case, development of the strategy to predict such circumstances is crucial so that such disasters can be prevented beforehand. Data acquired helps to identify the intensity of the situation in a post disaster scenario. It helps to take actions and carry out necessary operations to tackle devastating scenarios. Raw images obtained from camera-equipped are processed and neural network-based feature extraction techniques are applied to analyze the intensity of the disaster.

1.2 PURPOSE

Disasters are difficult to carry out and spread quickly in dense places; in this situation, it is essential to establish a plan to anticipate these conditions so that disasters can be avoided in advance. Data collection aids in determining the severity of the issue in the aftermath of a disaster. To deal with disastrous situations, it helps to take action and conduct the required procedures. To determine the disaster's intensity, raw photos from cameras are processed and neural network-based feature extraction methods are used.

2. LITERATURE REVIEW

2.1 EXISTING PROBLEM

Damage mapping using U-Net convolutional network.

Year:2018

Author name : Yanbing Bai, Eric Mas, Schunichi Koshimura

Natural disasters cannot be prevented- but they can be detected, giving people precious time to get to safety. In this, a deep learning algorithm for the semantic segmentation of high-resolution remote-sensing images using the U-net convolutional network was proposed to map the damage rapidly. The algorithm was implemented within a Microsoft Cognitive Toolkit framework in the GeoAI platform provided by Microsoft. The study takes the 2011 Tohoku Earthquake-Tsunami as a case study, for which the pre- and post-disaster high-resolution WorldView-2 image is used. The performance of the proposed U-net model is compared with that of deep residual U-net. The comparison highlights the superiority U-net for tsunami damage mapping in this work. The deep learning method, which is represented by automatic feature extraction and selection, has achieved state-of-the-art performance in various remote-sensing-based damage assessment applications Finally, we output the damage-mapping result in the Arc GIS platform . The design of the framework considers the availability of data sources, the feasibility of model implementation, time cost and accuracy of the method immediately after the disaster.

Merits: AI can use the seismic data to analyse the magnitude and pattern earthquake.

Demerits : In CNN Large training data needed, don’t encode the position and orientation of object.

Natural disaster intensity analysis and classification based on multispectral images.

Year:2021.

Author name: Muhammad Aamir, Tariq, Irfan, Ahmad ,Azam, Adam ,Witold. Frantise Brumercik,

Natural disasters not only disturb the human ecological system but also destroy the properties. A deep learning method for the reconstruction of two-dimensional cardiac magnetic resonance images. Convolutional Neural Networks (CNNs) are 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 achieve highly accurate results. The proposed multi-layered deep convolutional neural network method works in two blocks of convolutional neural networks. Block-I Convolutional Neural Network (B-I CNN), detects the occurrence of a natural disaster and Block-II Convolutional Neural Network (B-II CNN), defines the intensity of the natural disaster. The first block consists of three mini convolutional blocks with four layers each and includes an image input and fully connected layers. The second block also consists of three mini convolutional blocks with two layers each, including an image input layer and fully connected layer.

Merits: Data acquired from UAVs helps to identify the facial expressions of victims, the intensity of their situation and their needs in a post disaster scenario.

Demerits: Unpredictable, Cant suitable for all type of disaster.

Natural Disaster Application on Big Data and Machine Learning

Year :2019

Author name **:** Rania rizki Arinta,Andi Wahju

Natural disasters are events that are difficult to avoid. There are several ways of reducing the risks of natural disasters. One of them is implementing disaster reduction programs. Big data, machine learning is mostly used. By utilizing this method , it facilitates tasks in visualizing, analyzing, and predicting natural disaster. Here is, the use of big data, machine learning, and deep learning in 6 disaster management area. This 6-disaster management area includes early warning damage, damage assessment, monitoring, detection, forecasting , predicting, and post-disaster coordination, response, long-term risk assessment and reduction. To find out whether the previous research solved the problem in the prediction area and early detection we must know the data source used already has 5'v characteristics, namely Velocity, Volume, Value, Variety, and Veracity. The performance level of the model made is good or not from the level of accuracy, precision, recall, and the execution time. The propose of this study to give an insight and the use of big data, machine learning, and deep learning from 6 disaster area which is early warning damage, damage assessment.

Merits: Big data in the evacuation process to plan accordingly during the time of disaster **.**

Demerits: Traditional storage can cost IoT money to store big data

Artificial Intelligence For Disaster Risk Reduction

Year: 2018

Author name: Monique Kuglitsch, ArifAlbayrak, Raúl Aquino, Allison Craddock.

Artificial intelligence, in particular machine learning (ML), is playing an increasingly important role in disaster risk reduction (DRR) – from the forecasting of extreme events and the development of hazard maps to the detection of events in real time, the provision of situational awareness and decision support, and beyond. AI refers to technologies that mimic or even outperform human intelligence when performing certain tasks. ML, which is a subset of AI that includes supervised,unsupervised or reinforcement learning, can be simplified as parsing data into algorithms that learn from data to make classifications or predictions. AI methods offer new opportunities related to applications in, for instance, observational data pre-processing as well as forecast model output post-processing. The methodological potential is strengthened by novel processor technologies that allow heavy-duty, parallel data processing.

MERITS

- It saves lives and money and future -proofs our development gains.
- It yields economic, social, environmental benefits that enhance the well-being and resilience of countries and communities.

DEMERITS

- In a disaster, you face the danger of death or physical injury. You may also lose your home, possessions, and community.

Natural disaster Monitoring with Wireless Sensor Networks

Year:2013

Author name: Dan Chen,Zhixin Liu,Lizhe Wang,Minggang Dou.

Abstract: The wireless sensor network (WSN) technology has applied in monitoring natural disasters for more than one decade. Disasters can be closely monitored by augmenting a variety of sensors Natural disaster monitoring with WSN is a well-known data intensive application for the high bandwidth requirements and stringent delay constraints. It manifests a typical paradigm of data-intensive application upon low-cost scalable system. By making representative works in this area by classifying those in the domains of application of WSNs for disasters and optimization technologies significantly. WSN technology inspired by the existing work with focuses on issues of supporting reliable data transmission, handling huge data of heterogeneous sources and types, andminimizing energy consumption. This study proposes a dynamic routing protocol, a method for net-work recovery, and a method for managing mobile nodes to enable real-time and reliable data transmission.. A distributed algorithm for joint optimal control of power and rate has been developed, which can improve utility of network more than 95% and to minimize the energy consumption. Experimental results indicate the potentials of the proposed approaches in terms of adapting to the needs of early warning on geo hazards.

Merits: Low cost, scalability, flexibility.

Demerits: As it is wireless in nature, it is prone to hacking by hackers.

3.1 REFERENCES

[1]. Ji-Myong Kim Development of Model to Predict the Natural Disaster Induced Financial Loses For Construction Projects Using Deep Learning

[2]. Hafiz Suliman Munawar, Remote Sensing Methods for Flood Prediction: A Review, 2022

[3]. Xianghai L, Construction of Urban Flood Disaster Emergency Management System Using Scenario Construction Technology, 2022

[4]. Dr. Uday N. Suryawanshi, Remote Sensing And Its Application in Disaster Management In India, 2022

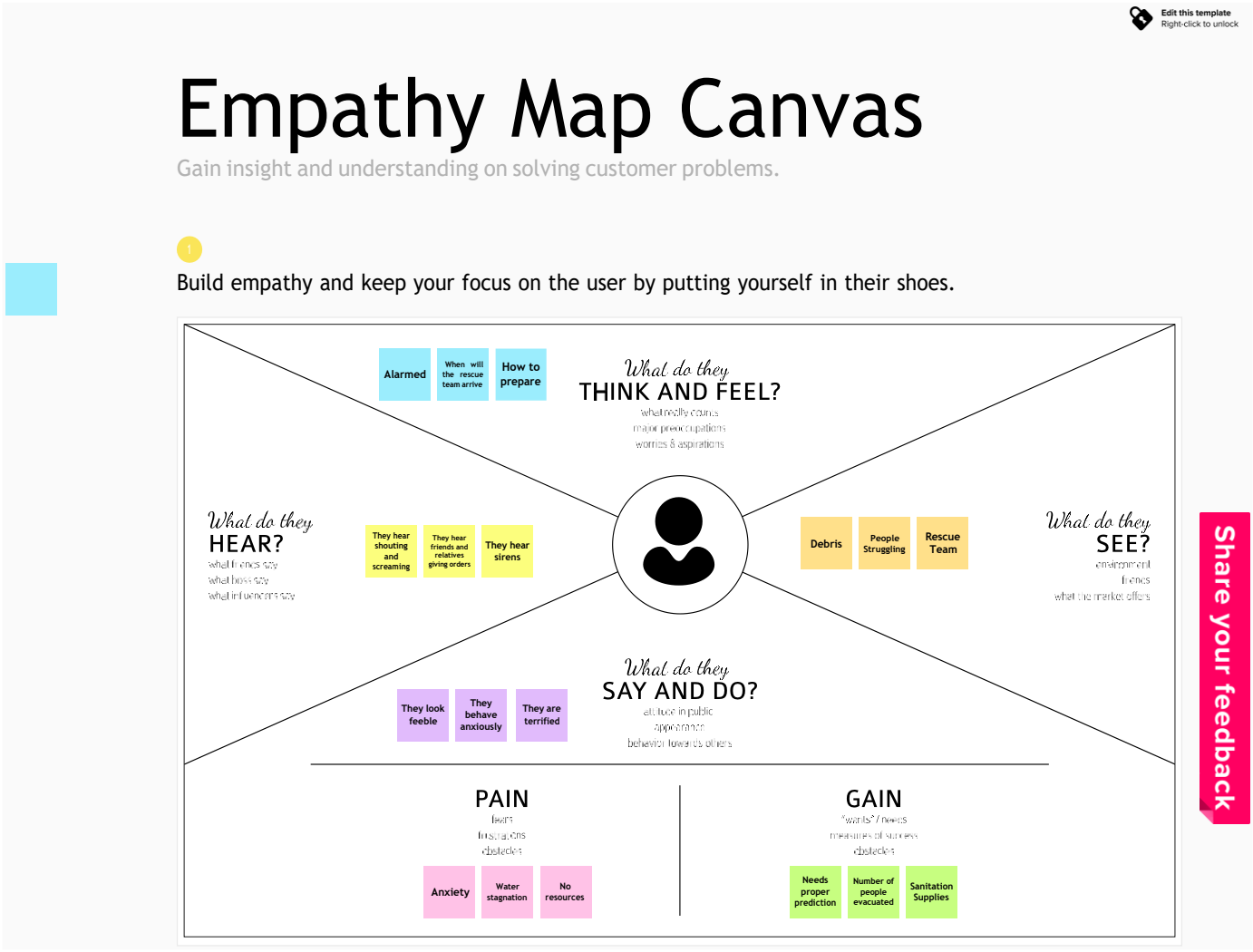
[5]. Evi Susanti Tasri, The effect of economic variables on natural disasters and the impact of disasters on economic variables, 2022

3.2 PROBLEM STATEMENT DEFINITION

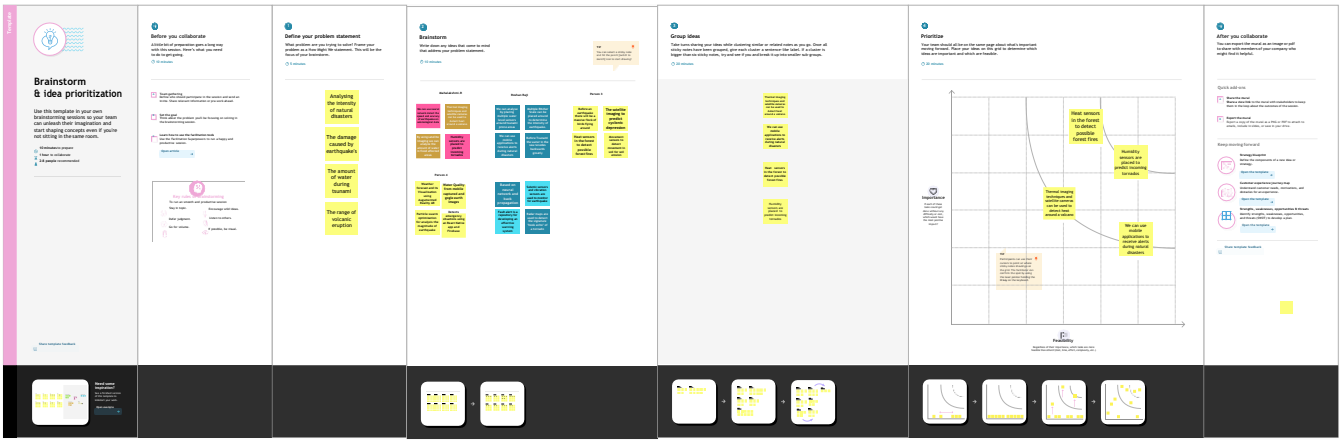
In existing system, use of manpower is difficult in case of natural disaster occurrence in hilly areas, and continuous electric power supply is highly affected in these areas due to maintenance issues of transmission lines. Therefore, in this case autopilot aerial equipment is used to gather images, and hidden content from aerial images needs to be identified in case of natural disasters such as landslides and heavy snowfall. Populations in underdeveloped countries cannot afford damages disasters cause to infrastructures. The aftermath of disasters leaves the humans in miserable situations, and sometimes the devastating effects cannot be detected; additionally, rescue operations cannot take place in most of the places and victims are unable to be identified due to geographical factors of the different areas.

4. IDEATION & PROPOSED SOLUTION

4.1 EMPATHY MAP CANVAS



4.2 IDEATION & BRAINSTORMING



4.3 PROPOSED SOLUTION

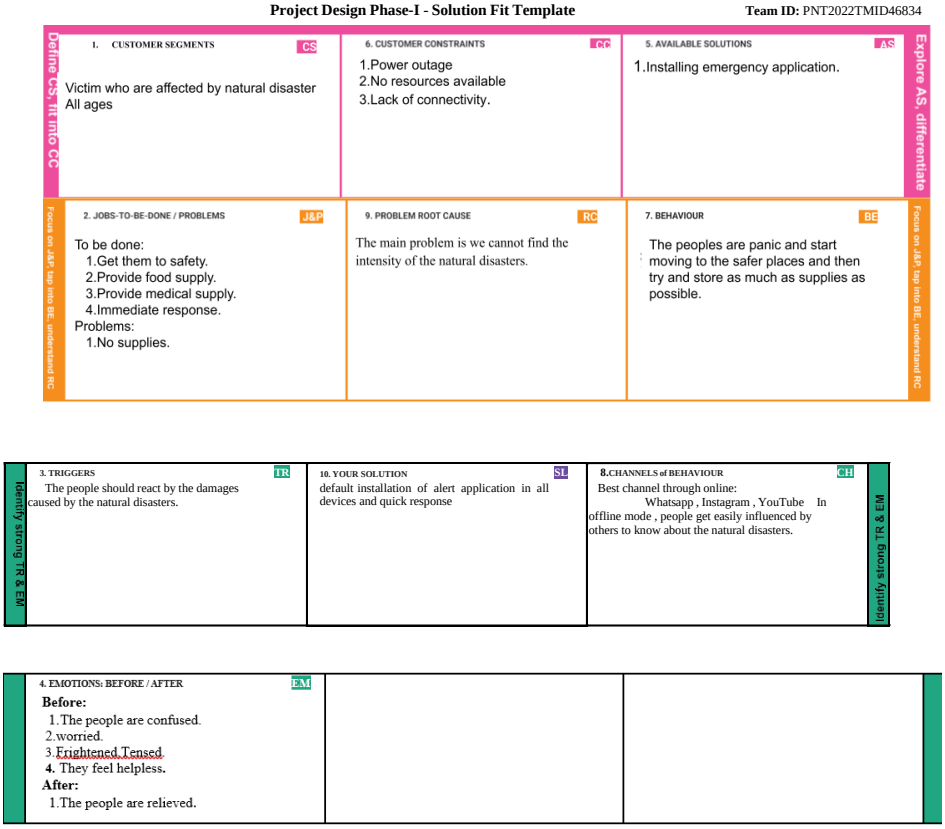
Natural disaster intensity analysis and classification based on multispectral images using a multilayered deep convolutional neural network. Moreover, this method consists of two blocks of a convolutional neural network. The first block detects a natural disaster occurring and the second one defines the intensity type of the natural disaster. Additionally, the first block consists of three mini convolutional blocks with four layers each, including 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 and includes an image input layer and fully connected layer. To evaluate the performance of the proposed multilayered deep convolutional neural network uses a train–test validation schema. To train the whole model, the training dataset was used, while for the fine-tuning of model the validation set was used. The performance of the whole framework was calculated on the basis of the test dataset. The proposed model works on an image dataset to detect and classify the natural disasters.

Propsed solution:

S.NO	PARAMETER	DESCRIPTION
1.	Problem statement (problem to be solved)	To find the solution for who are affected by the natural disasters by prediction.
2.	Idea/ solution description	1.By predicting the future data by analysis the previous data available. 2.Using sensors.
3.	Novelty/ Uniqueness	Predicting the future data by analysis the previous data available.
4.	Social impact/ Customer satisfaction	Ability to anticipate future events, could make a huge difference and help limit the human and material costs of such disasters.
5.	Business Model (Revenue Model)	Requirements like sensor, mobile app, image viewing.
6.	Scalability of the solution	Recurrent neural network.

4.4 PROBLEM SOLUTION FIT

There isn't a methodical approach to swiftly become alert and announce the disaster. The frames from the CCTV camera are compared to the pre-trained data. When the statistics are matched, an alert is sent to the response team, who then informs the general public. The proposed system should enable offline and online communication between the system and the response team.



5.REQUIREMENT ANALYSIS

4.5 FUNCTIONAL REQUIREMENT

Upload Images

In this module, we can upload the image or videos dataset to the system. The CCTV footage is used to capture the natural disaster image, but it has an increasing range of time resolution and space. Additionally, the information is kept in a database for later use. The dataset includes symptoms of disaster such as Cyclone, Earthquake, Tsunami, Fire Accidents, and Flood. These disasters symptoms are preserved as image or video and acquired from the Kaggle website.

Noise Filtering

By using filter techniques to minimise noise in image or video frames, it is possible to identify the signs of a natural disaster. The filter's objective is to eliminate noise, which degrades the appearance of images. This claim is supported by statistics. The usual frequency response of a filter is built. To remove "salt and pepper" noise, image processing often uses the nonlinear approach of filtering. A median filter is preferred than convolution when edge preservation and noise reduction are the primary considerations. Similar to photo binarization practise, document picture binarization is a technique used in the pre-processing phase of document analysis to distinguish the text in the foreground from the background of the document. A speedy and accurate binarization strategy is needed for the following document image processing activities.

Classification

Classification is the process of dividing data into various categories. The method starts by determining the class of the given data points. Classification is achievable for both structured and unstructured data. The terms target, label, and classes are occasionally used to describe the classes. The frames captured by the CCTV footage will be compared to the trained dataset in the system database for the features obtained in the feature extraction stage in the classification process. The specific image will be recognised once the ideal match is discovered based on the symptoms matched. The detected disaster name with its type will be displayed over the image. Here, a convolution neural network approach is employed to classify data.

Disaster Detection

The classification is the final step of the system. After analyzing the structure, each section individually evaluated for the probability of true positives. The CNN varies in how the convolutional and max pooling layers are realized and how the nets are trained. Finally classify the image regions using deep learning algorithm and improve the accuracy in classification. In this module, the system receives the image after the model has identified the disaster and extracts its types. And the responsive team will receive a warning message to protect the surrounding and alert people.

Alert System

The rapid growth of increasing the population and urbanization has led to the outbreak of disaster. Disaster is a natural hazard to the environment and the interference of the atmosphere system; the environment affects living organisms. In this module, send alert to the authority in terms of SMS at the time of fire detection. It can be useful to provide earlier detection.

NON FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a

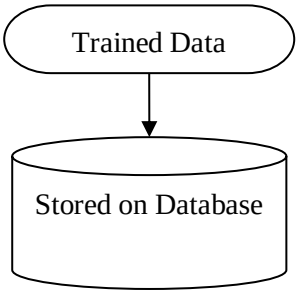
PROJECT DESIGN

4.6 DATA FLOW DIAGRAMS

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

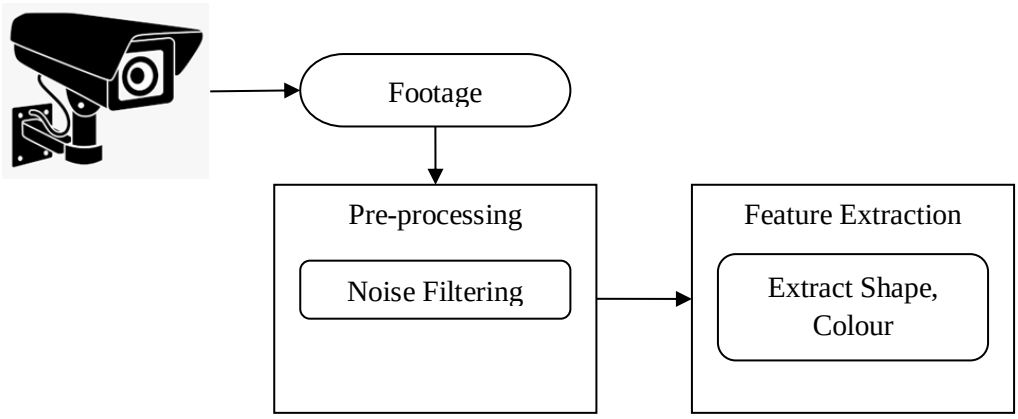
LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



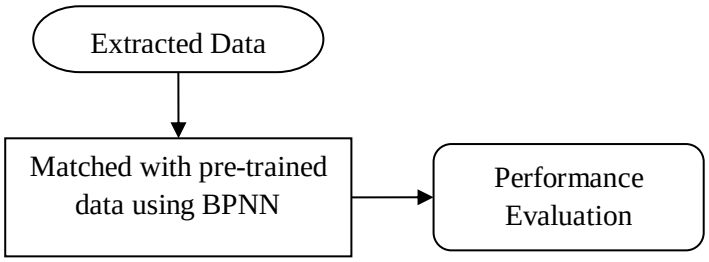
LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.



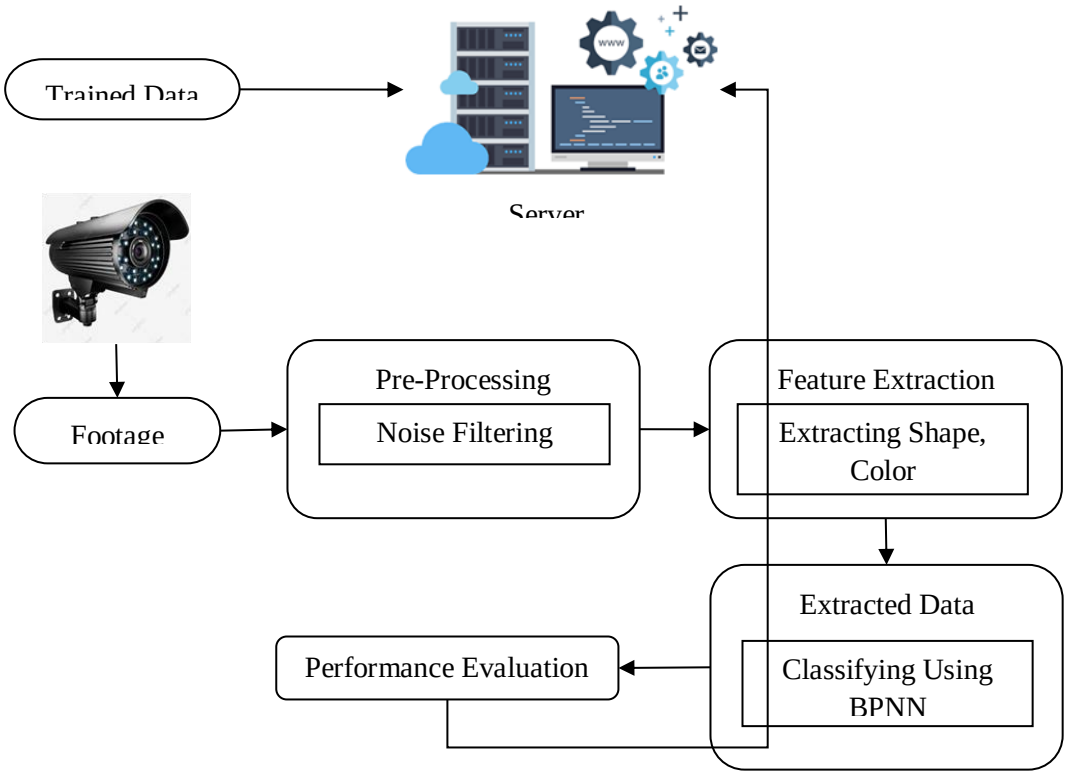
LEVEL 2

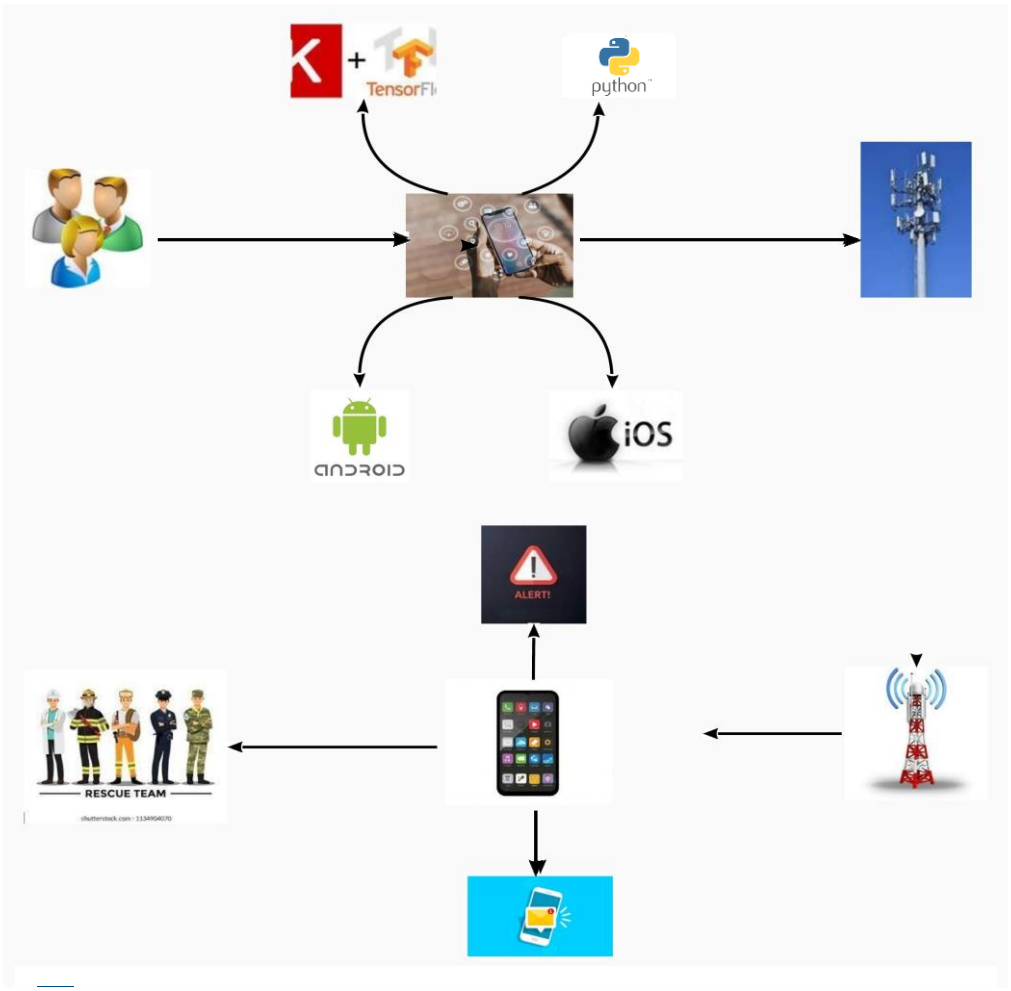
A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data within the defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modelling and one of the oldest.



4.7 SOLUTION & TECHNICAL ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).




TensorFl
python
iOS
android
ALERT!
RESCUE TEAM

4.8 USERSTORIES

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Mahalakshmi.R Mathumitha Priyanka Roshan Raji Sriram
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Mahalakshmi
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Mathumitha
Sprint-2		USN-4	As a user, I can register for the application through Gmail	2	Medium	Priyanka
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Sriram
Sprint-1	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	2	High	Roshan Raji
Sprint-1	login	USN-7	As a user, I can log into the web application and access the dashboard	2	High	Mahalakshmi
Sprint-4	Helpdesk	USN-8	As a user, I can get the guidance from the customer care	1	High	Mathu mitha

5. PROJECT PLANNING & SCHEDULING

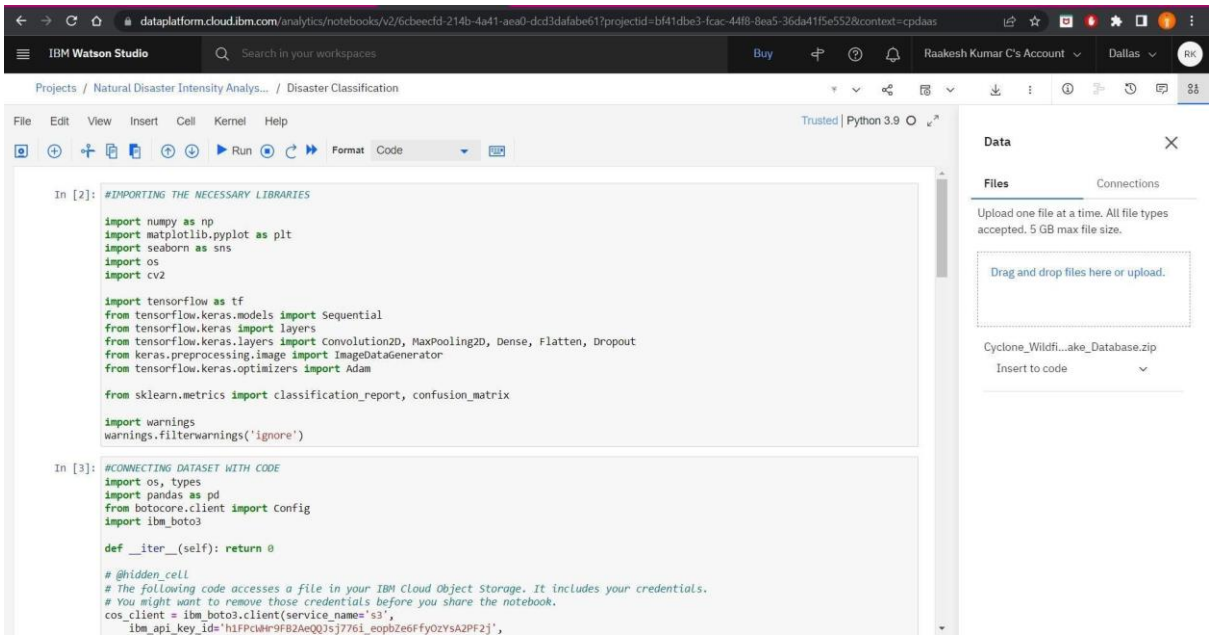
PROJECT PLANNING

MILESTONE AND ACTIVITY LIST

PLANNING	Planning the modules and features which are going to implement.
REQUIREMENTS	We need software's and tools for develop and to install all required resources
DESIGN	In this we design the modules like dashboard, details and input etc...,
DEVELOPMENT	We are going to develop the predictor which uses the previous dataset. Here, we use some algorithm for prediction.
TESTING	We are going to test the model if we face any error, we debug the error
DEPLOYMENT	At last, we submit the project in GitHub.

5.1 SPRINT PLANNING & ESTIMATION

Jupyter Notebook asset in IBM Watson Studio feature of IBM Cloud:



Sprint - I Milestones:

1. Import the Necessary Libraries

```
In [2]: #IMPORTING THE NECESSARY LIBRARIES

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import cv2

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Dense, Flatten, Dropout
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam

from sklearn.metrics import classification_report, confusion_matrix

import warnings
warnings.filterwarnings('ignore')
```

2. Upload and Connect Dataset with notebook

```
In [3]: #CONNECTING DATASET WITH CODE
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='h1FPcWHr9FB2AeQQJsj776i_eopbZe6FfyOzYsA2PF2j',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'naturaldisasterintensityanalysisa-donotdelete-pr-pwiuxy2i5hirv2'
object_key = 'Cyclone_Wildfire_Flood_Earthquake_Database.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

3. Extracting the Dataset using BytesIO unzip function

In [4]:
```
#EXTRACTING THE DATASET USING BytesIO unzip function
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [5]:
```
ls
```

```
Cyclone_Wildfire_Flood_Earthquake_Database/
```

In [6]:
```
pwd
```

Out[6]: '/home/wsuser/work'

4. Listing out the Disaster Classes

In [8]:
```
#LISTING OUT THE DISASTER CLASSES
for i in os.listdir(dir):
    print(i)
```

```
readme.txt
Earthquake
Cyclone
Wildfire
Flood
```

In [9]:
```
path=os.path.join(dir,'readme.txt')
os.remove(path)
```

In [10]:
```
for i in os.listdir(dir):
    print(i)
```

```
Earthquake
Cyclone
Wildfire
Flood
```

5. Configuring ImageDataGenerator Class

In [18]:
```
dir=r'/home/wsuser/work/Cyclone_Wildfire_Flood_Earthquake_Database'
#CONFIGURING THE ImageDataGenerator CLASS
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

6. Split the dataset into training, testing, and validation data

In [11]:
```
pip install split_folders
```

```
Collecting split_folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
Note: you may need to restart the kernel to use updated packages.
```

In [12]:
```
#SPLIT THE DATASET INTO TRAINING, TESTING AND VALIDATION DATA
import splitfolders
splitfolders.ratio(dir,output="dataset",seed=42,ratio=(.7,.2,.1),group_prefix=None)
```

```
Copying files: 4428 files [00:02, 1678.84 files/s]
```

In [13]:
```
for i in os.listdir(dir):
    print(i)
```

```
Earthquake
Cyclone
Wildfire
Flood
```

In [14]:
```
dir1=r'/home/wsuser/work/dataset'
```

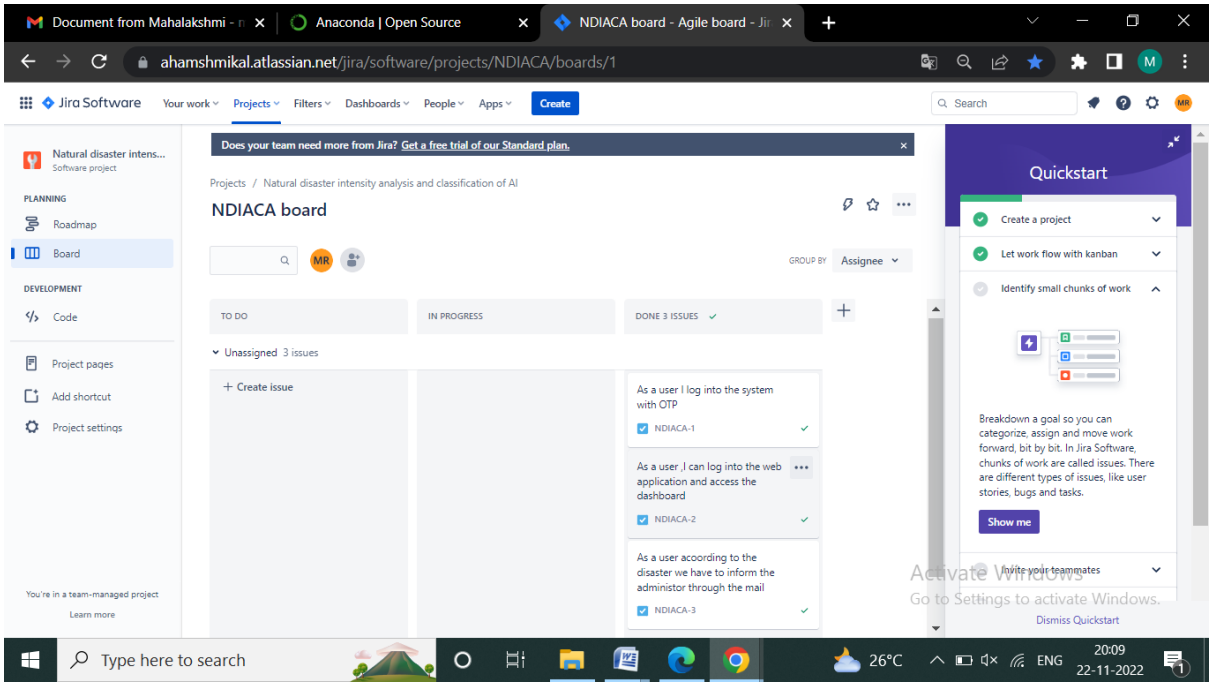
In [15]:
```
for i in os.listdir(dir1):
    print(i)
```

```
val
train
test
```

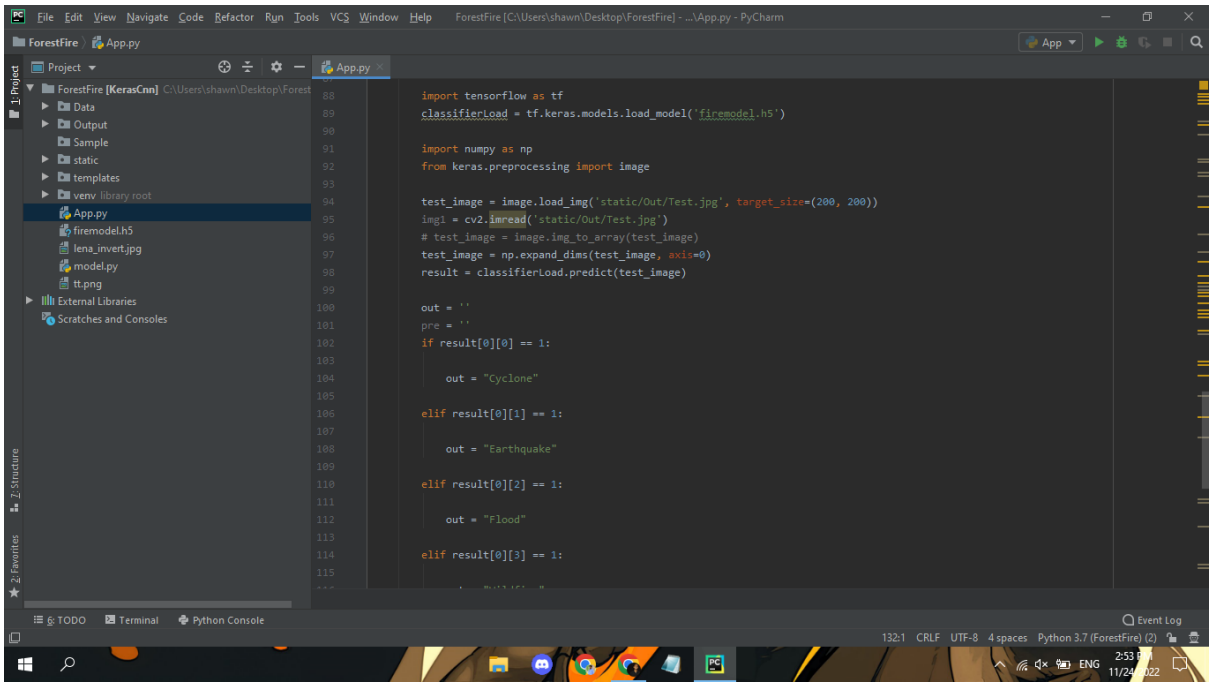
5.2 SPRINT DELIVERY SCHEDULE

Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)	Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)
8	30 Oct 2022	Sprint-1	8	10 Days	27 Oct 2022	5 Nov 2022
4	07 Nov 2022	Sprint-2	4	8 Days	6 Nov 2022	13 Nov 2022
6	11Nov 2022	Sprint-3	6	4 Days	14 Nov 2022	17 Nov 2022
2	22Nov 2022	Sprint-4	2	3 Days	18 Nov 2022	20 Nov 2022

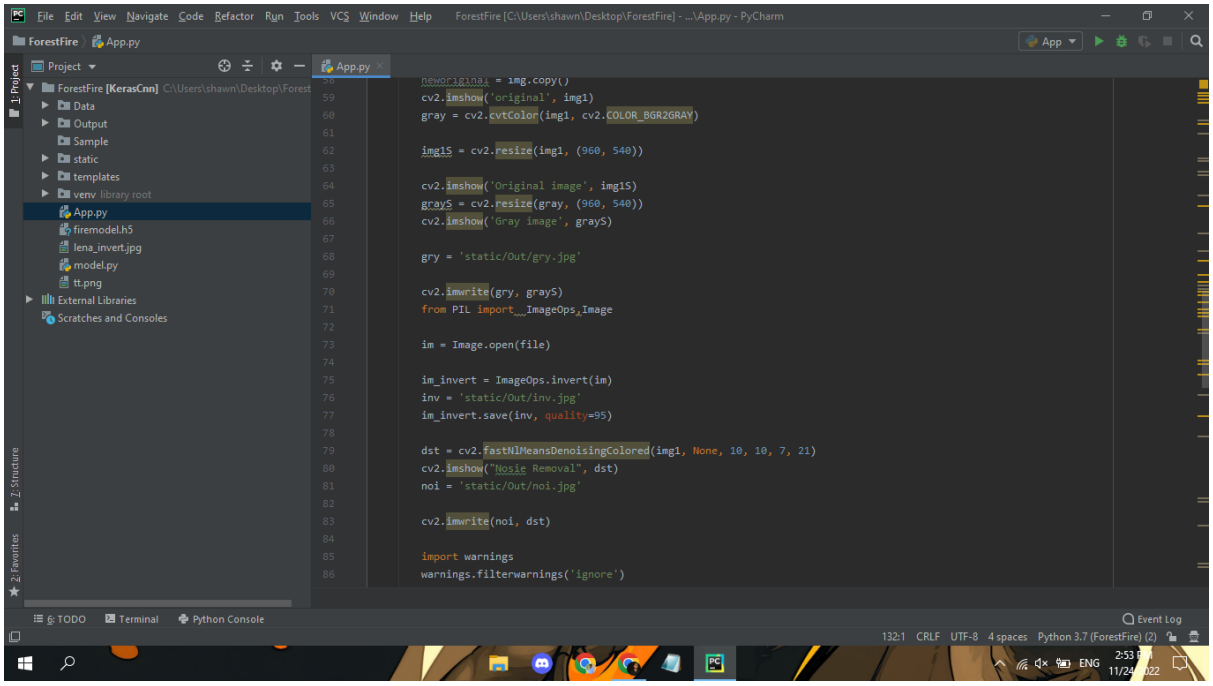
5.3 REPORTS FROM JIRA



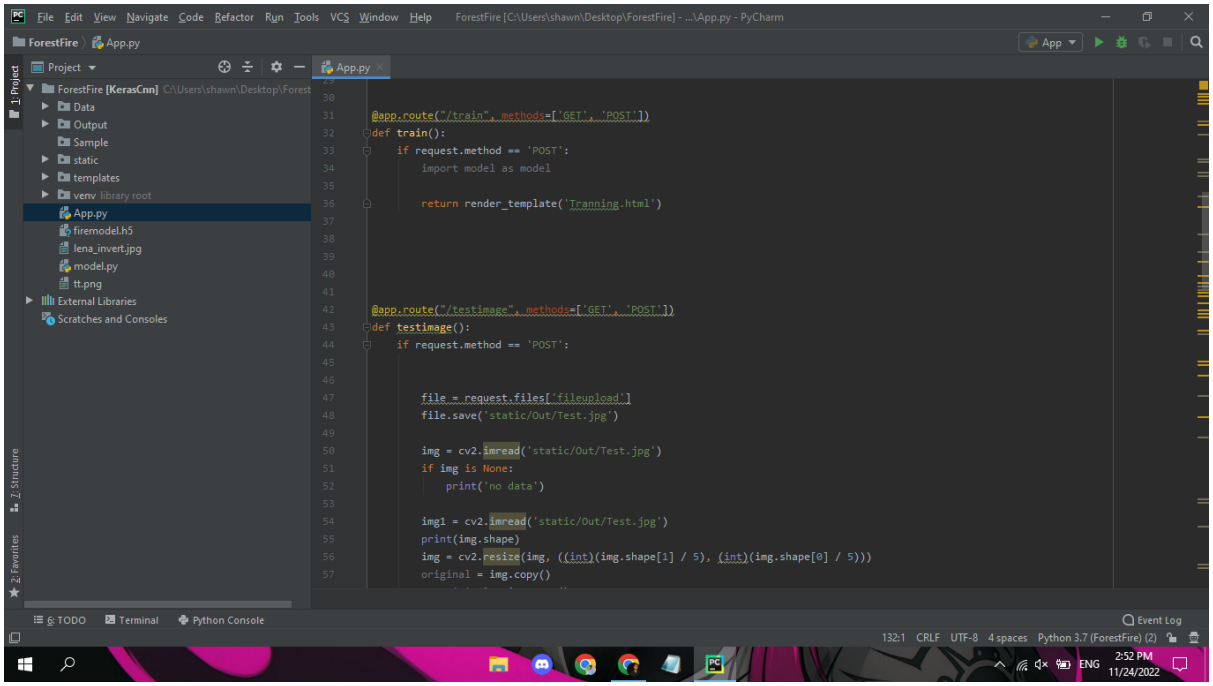
6 .4 CODING & SOLUTIONING

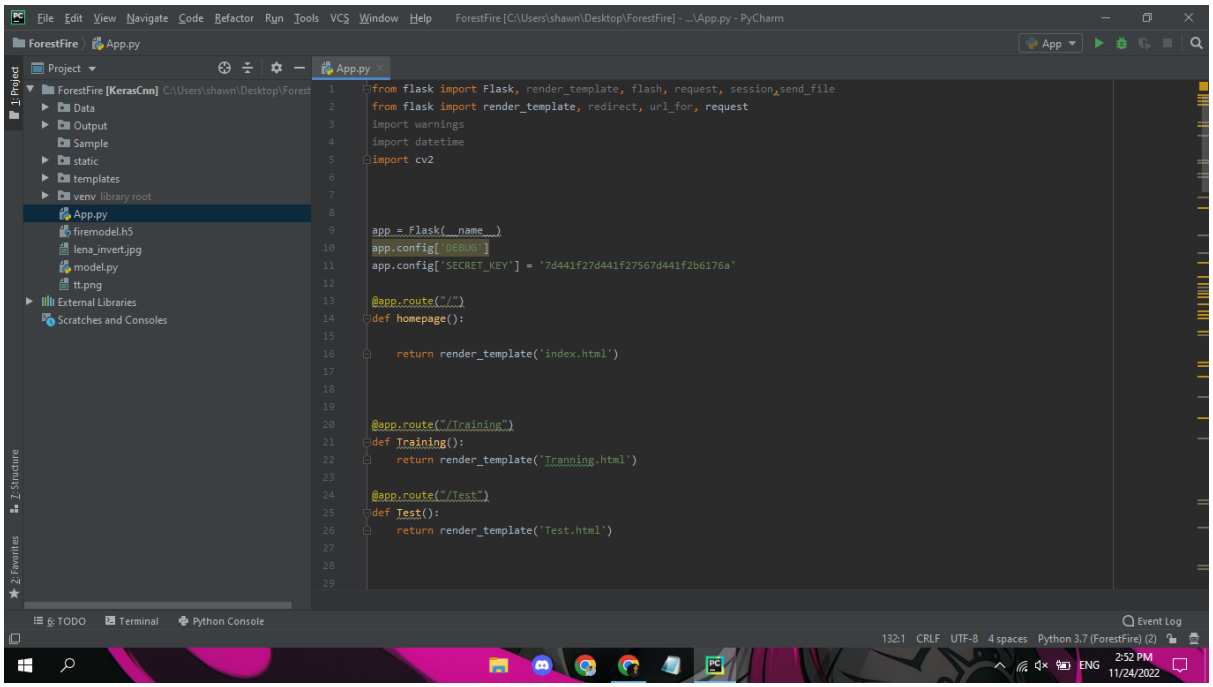


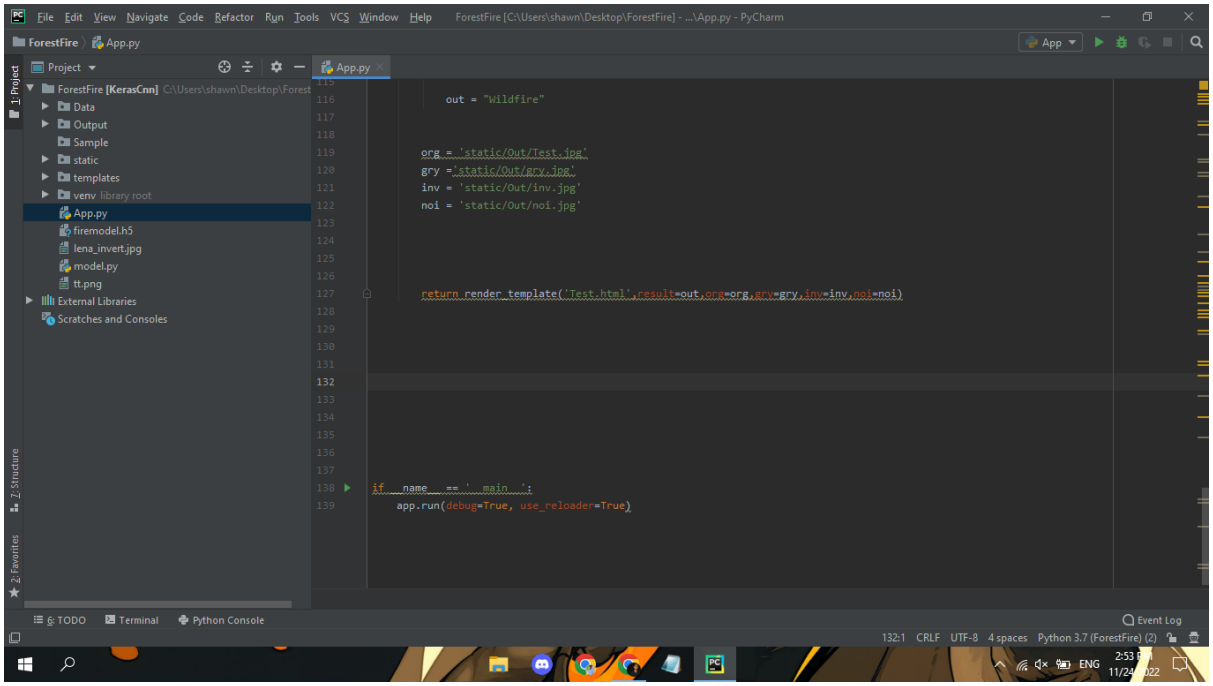
```
import tensorflow as tf
classifierLoad = tf.keras.models.load_model('firemodel.h5')

import numpy as np
from keras.preprocessing import image

test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
img1 = cv2.imread('static/Out/Test.jpg')
# test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = classifierLoad.predict(test_image)

out = ''
pre = ''
if result[0][0] == 1:

    out = "Cyclone"

elif result[0][1] == 1:

    out = "Earthquake"

elif result[0][2] == 1:

    out = "Flood"

elif result[0][3] == 1:
```



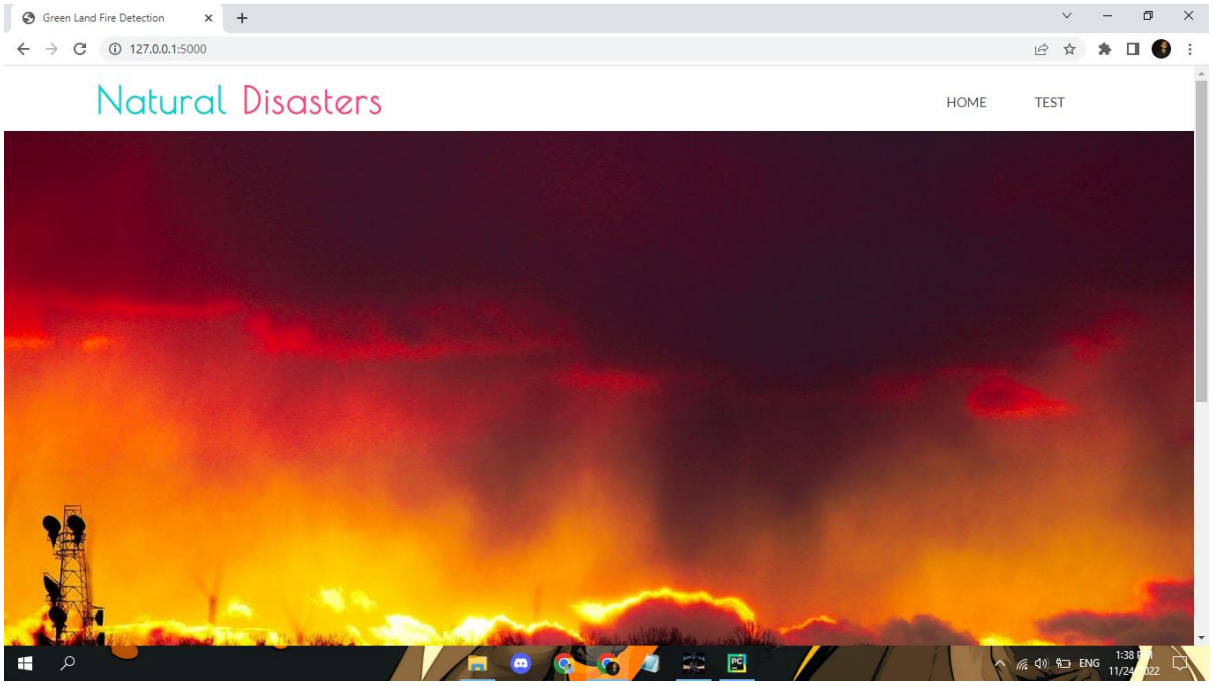
```
neworiginal = img.copy()
cv2.imshow('original', img1)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)

img1S = cv2.resize(img1, (960, 540))

cv2.imshow('Original image', img1S)
grayS = cv2.resize(gray, (960, 540))
cv2.imshow('Gray image', grayS)

gry = 'static/Out/gry.jpg'

cv2.imwrite(gry, grayS)
from PIL import ImageOps,Image

im = Image.open(file)

im_invert = ImageOps.invert(im)
inv = 'static/Out/inv.jpg'
im_invert.save(inv, quality=95)

dst = cv2.fastNlMeansDenoisingColored(img1, None, 10, 10, 7, 21)
cv2.imshow("Nosie Removal", dst)
noi = 'static/Out/noi.jpg'

cv2.imwrite(noi, dst)

import warnings
warnings.filterwarnings('ignore')
```

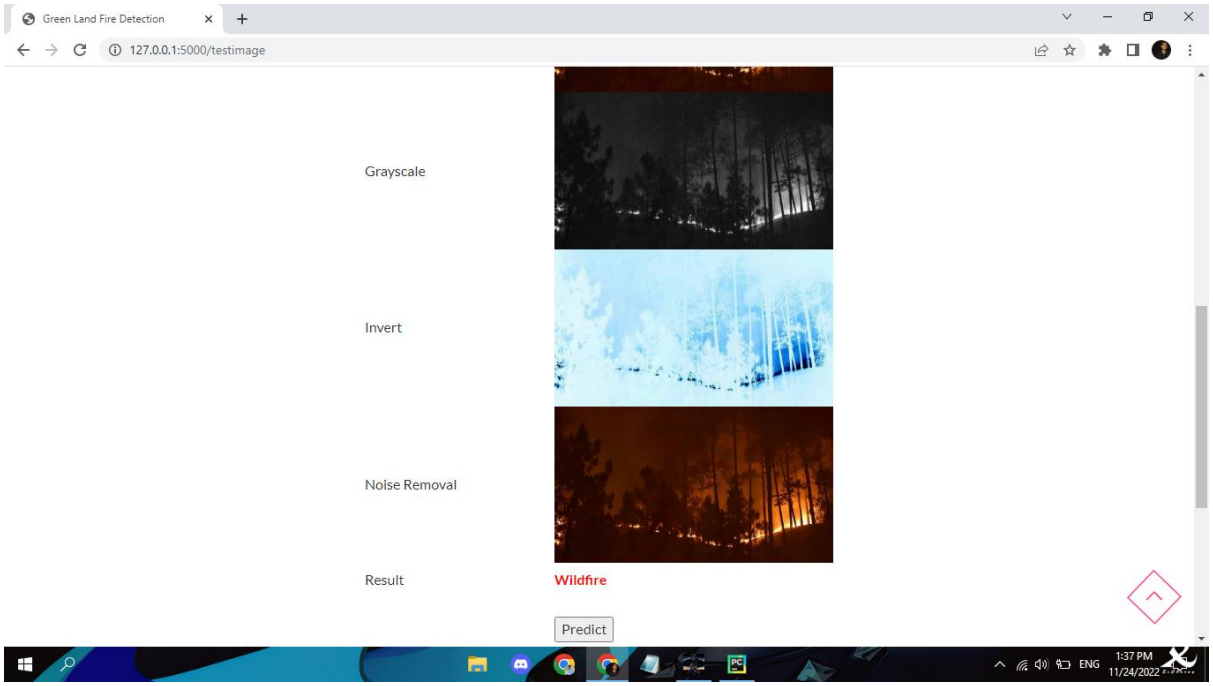

File Edit View Navigate Code Refactor Run Tools VCS Window Help ForestFire [C:\Users\shawn\Desktop\ForestFire] - ...\App.py - PyCharm
ForestFire App.py
App
Project
App.py
ForestFire [KerasCnn] C:\Users\shawn\Desktop\Forest
Data
Output
Sample
static
templates
venv library root
App.py
firemodel.h5
lena_invert.jpg
model.py
tt.png
External Libraries
Scratches and Consoles
@app.route("/train", methods=['GET', 'POST'])
def train():
if request.method == 'POST':
import model as model
return render_template('Tranning.html')
@app.route("/testimage", methods=['GET', 'POST'])
def testimage():
if request.method == 'POST':
file = request.files['fileupload']
file.save('static/Out/Test.jpg')
img = cv2.imread('static/Out/Test.jpg')
if img is None:
print('no data')
img1 = cv2.imread('static/Out/Test.jpg')
print(img.shape)
img = cv2.resize(img, ((int)(img.shape[1] / 5), (int)(img.shape[0] / 5)))
original = img.copy()
6: TODO
Terminal
Python Console
Event Log
132:1 CRLF UTF-8 4 spaces Python 3.7 (ForestFire) (2)
ENG
2:52 PM
11/24/2022


File Edit View Navigate Code Refactor Run Tools VCS Window Help ForestFire [C:\Users\shawn\Desktop\ForestFire] - ...\App.py - PyCharm
ForestFire App.py
App
Project
App.py
ForestFire [KerasCnn] C:\Users\shawn\Desktop\Forest
Data
Output
Sample
static
templates
venv library root
App.py
firemodel.h5
lena_invert.jpg
model.py
tt.png
External Libraries
Scratches and Consoles
from flask import Flask, render_template, flash, request, session,send_file
from flask import render_template, redirect, url_for, request
import warnings
import datetime
import cv2
app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
return render_template('index.html')
@app.route("/Training")
def Training():
return render_template('Tranning.html')
@app.route("/Test")
def Test():
return render_template('Test.html')
6: TODO
Terminal
Python Console
Event Log
132:1 CRLF UTF-8 4 spaces Python 3.7 (ForestFire) (2)
ENG
2:52 PM
11/24/2022



Solutioning:





7 TESTING

5.4 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on “HOW” to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

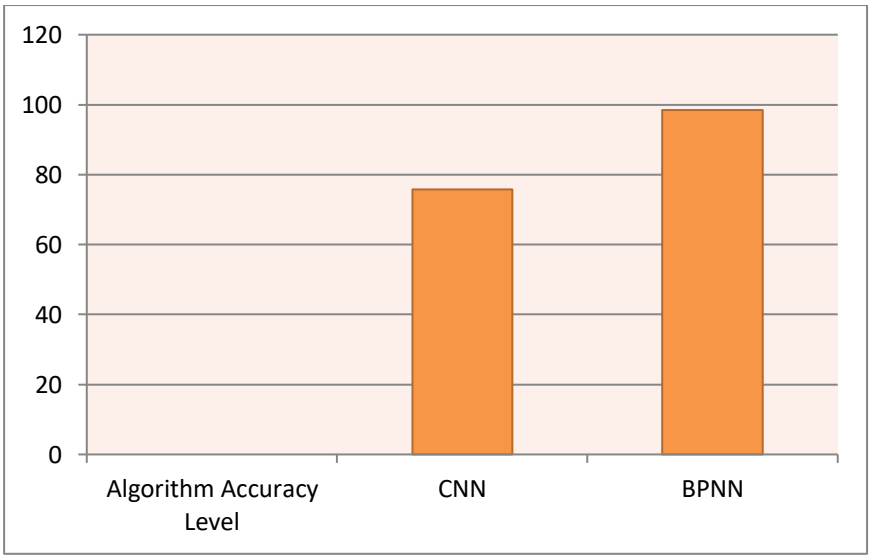
S.NO	Scenario	Input	Excepted output	Actual output
1	Admin	User name and password	Login	Login success.
2	Upload Dataset	Upload symptoms of disaster and its types as csv file	Upload successfully	Details stored in database.
3	CCTV footages	Predict and Classify the disaster and its types	Predict and classify the disaster and its types	Predicted successfully.

5.5 USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

RESULTS

5.6 PERFORMANCE METRICS



6. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Automated analysis of natural disaster
- Reduce the time and computational complexity
- Relevant features are extraction
- Improved accuracy rate

DISADVANTAGE

- Manual analysis of natural disaster
- Need additional sensors to detect
- Time complexity is high
- Accuracy is less

7. CONCLUSION

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 multilayered 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 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.

8. FUTURE SCOPE

However, if this model is run on a graphic processing unit (GPU)-based system in the future with real time sensors and monitoring power, then the proposed model will be used as a real time natural disaster detection model and provide some upcoming predictions for future disasters. The main purpose of this model is to detect and classify the type of disaster with a high accuracy rate. To prevent natural disasters in the future, said model can be used to predict future disasters and take some action against heavy loss of human ecological systems and property

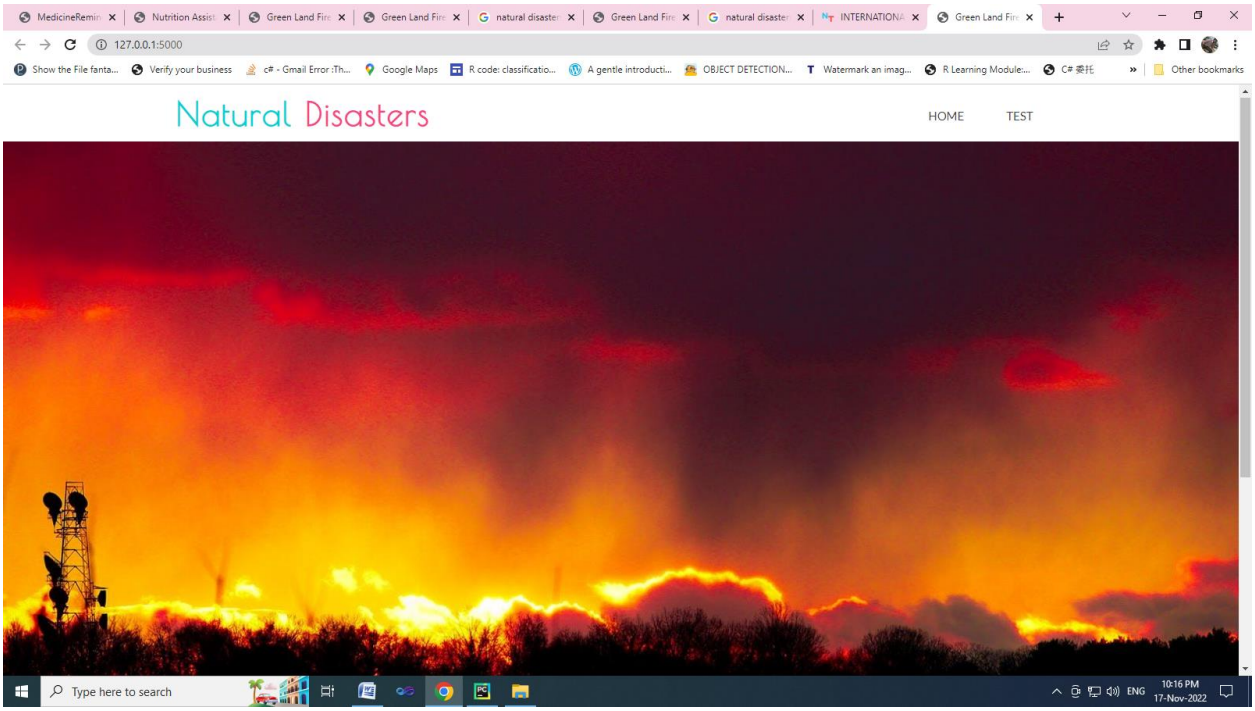
9. APPENDIX

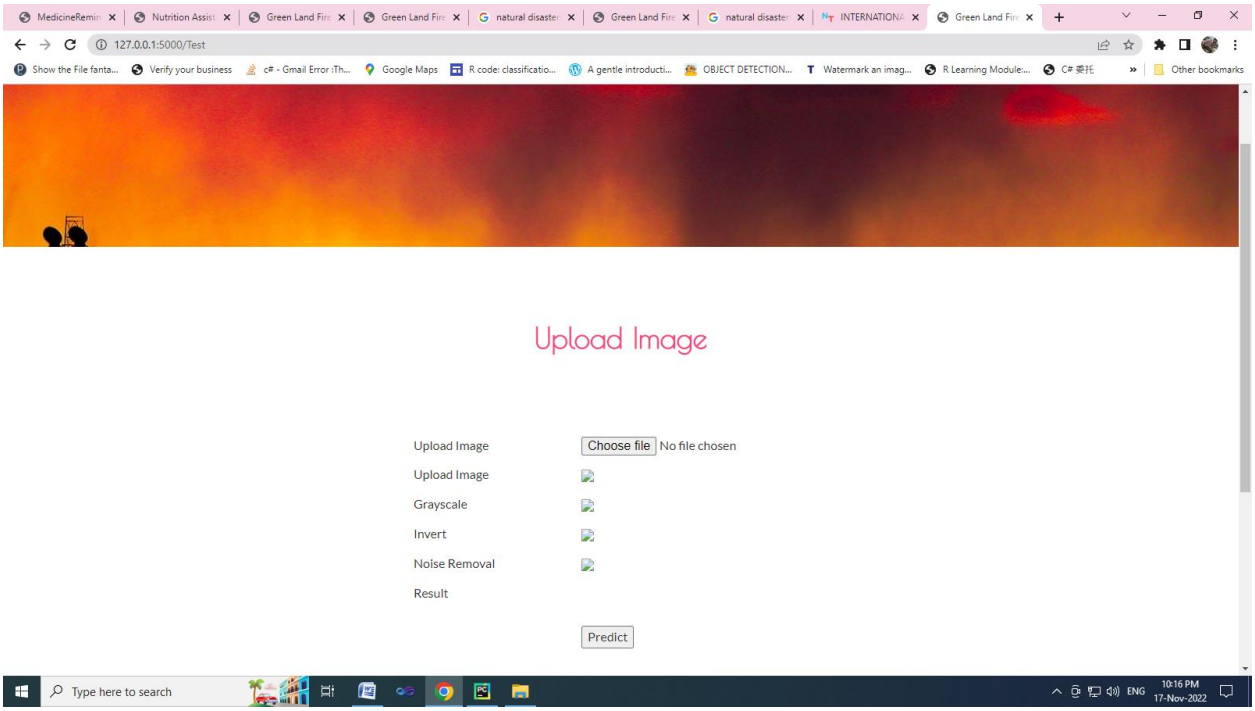
SOURCE CODE

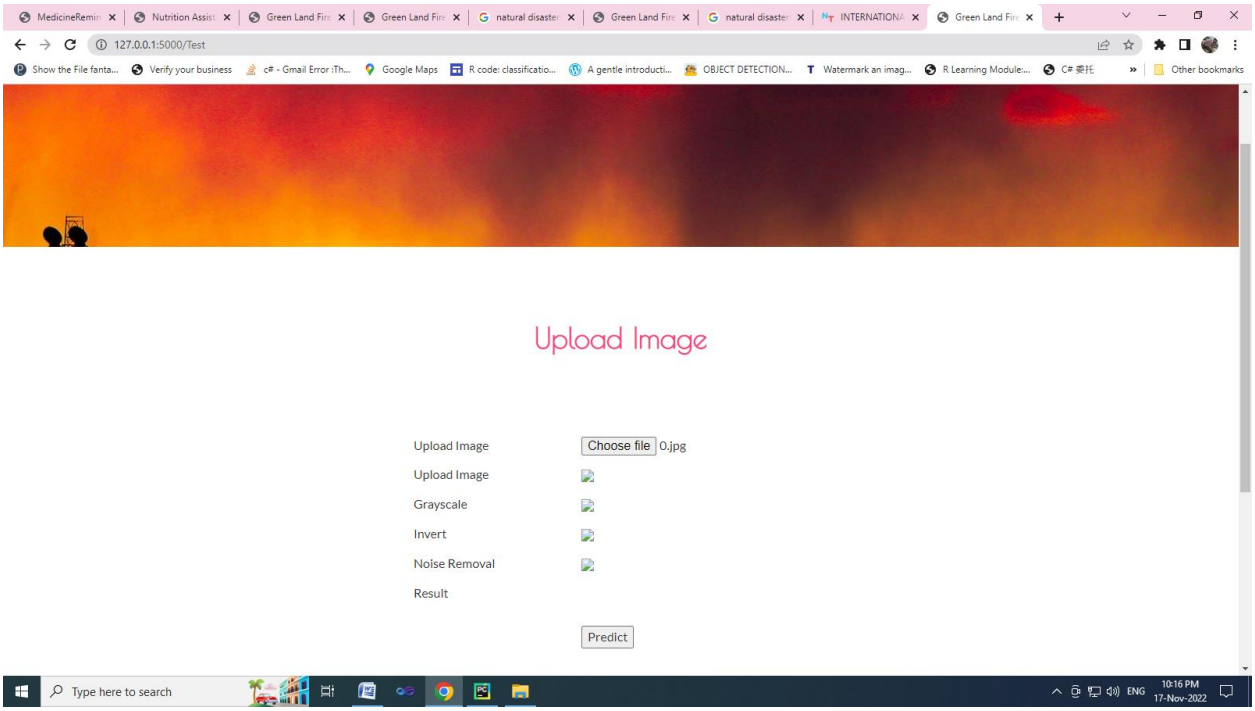
```
import Flask, render_template, flash, request, session,send_file
from flask import render_template, redirect, url_for, request
import warnings
import datetime
import cv2

app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

@app.route("/")
def homepage():

    return render_template('index.html')

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

@app.route("/Test")
def Test():
    return render_template('Test.html')
@app.route("/train", methods=['GET', 'POST'])
def train():
    if request.method == 'POST':
        import model as model
        return render_template('Tranning.html')
@app.route("/testimage", methods=['GET', 'POST'])
def testimage():
    if request.method == 'POST':
```

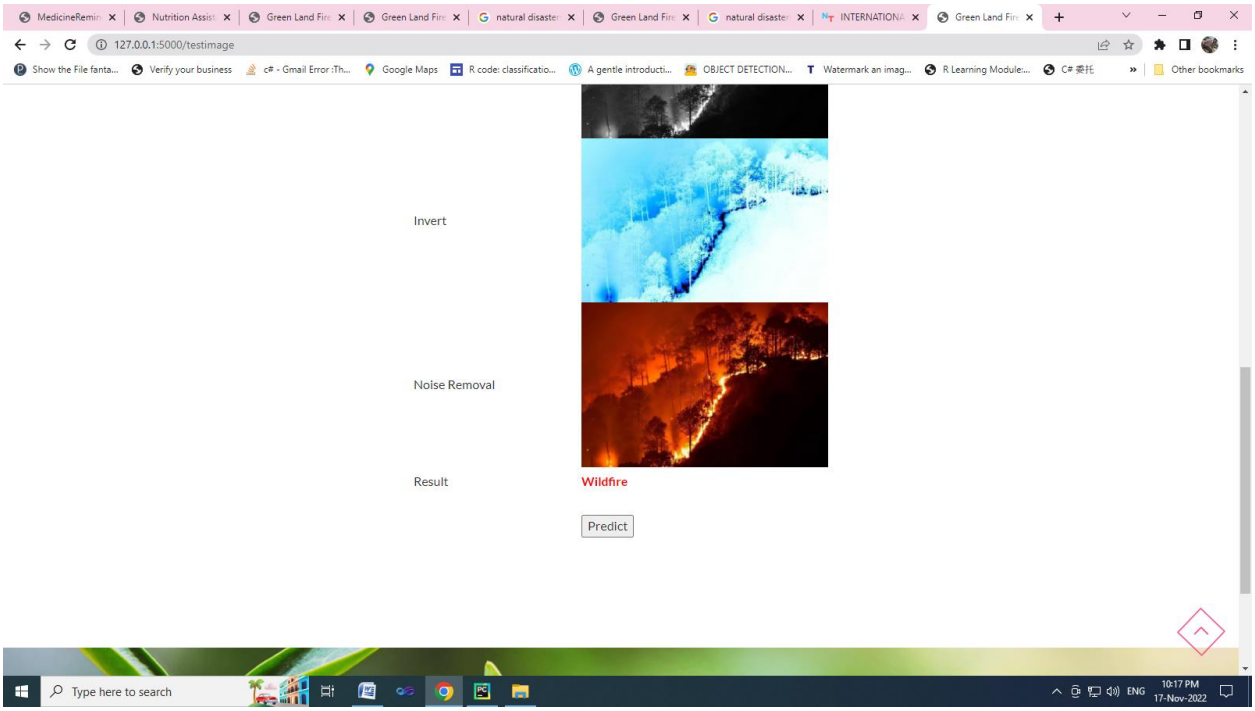
```
file = request.files['fileupload']
file.save('static/Out/Test.jpg')

img = cv2.imread('static/Out/Test.jpg')
if img is None:
   print('no data')

img1 = cv2.imread('static/Out/Test.jpg')
print(img.shape)
img = cv2.resize(img, ((int)(img.shape[1] / 5), (int)(img.shape[0] / 5)))
original = img.copy()
neworiginal = img.copy()
cv2.imshow('original', img1)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)

img1S = cv2.resize(img1, (960, 540))

cv2.imshow('Original image', img1S)
grayS = cv2.resize(gray, (960, 540))
cv2.imshow('Gray image', grayS)

gry = 'static/Out/gry.jpg'

cv2.imwrite(gry, grayS)
from PIL import  ImageOps,Image

im = Image.open(file)

im_invert = ImageOps.invert(im)
inv = 'static/Out/inv.jpg'
im_invert.save(inv, quality=95)
dst = cv2.fastNlMeansDenoisingColored(img1, None, 10, 10, 7, 21)
cv2.imshow("Nosie Removal", dst)
```

```
            noi = 'static/Out/noi.jpg'
            cv2.imwrite(noi, dst)
            import warnings
            warnings.filterwarnings('ignore')
            import tensorflow as tf
            classifierLoad = tf.keras.models.load_model('firemodel.h5')
            import numpy as np
            from keras.preprocessing import image
            test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
            img1 = cv2.imread('static/Out/Test.jpg')
            # test_image = image.img_to_array(test_image)
            test_image = np.expand_dims(test_image, axis=0)
            result = classifierLoad.predict(test_image)

            out = ''
            pre = ''
            if result[0][0] == 1:
               out = "Cyclone"
            elif result[0][1] == 1:
               out = "Earthquake"
            elif result[0][2] == 1:

               out = "Flood"
            elif result[0][3] == 1:
               out = "Wildfire"
            org = 'static/Out/Test.jpg'
            gry ='static/Out/gry.jpg'
            inv = 'static/Out/inv.jpg'
            noi = 'static/Out/noi.jpg'

            return render_template('Test.html',result=out,org=org,gry=gry,inv=inv,noi=noi)

        if __name__ == '__main__':
app.run(debug=True, use_reloader=True)
```


Natural Disasters
HOME
TEST
Type here to search
ENG
10:16 PM
17-Nov-2022


127.0.0.1:5000/Test
Upload Image
Upload Image
Choose file
No file chosen
Upload Image
Grayscale
Invert
Noise Removal
Result
Predict
Type here to search
ENG
10:16 PM
17-Nov-2022


127.0.0.1:5000/Test
Upload Image
Upload Image
Choose file
0.jpg
Upload Image
Grayscale
Invert
Noise Removal
Result
Predict
Type here to search
ENG
10:16 PM
17-Nov-2022


127.0.0.1:5000/testimage
Invert
Noise Removal
Result
Wildfire
Predict
Type here to search
ENG
10:17 PM
17-Nov-2022