PROJECT REPORT

Team-id:PNT2022TMID00210

Natural Disasters Intensity Analysis and Classification Using Artificial Intelligence.

1.INTRODUCTION:

1.1 Project Overview:

Natural disasters are inevitable, and the occurrence of disasters drastically affects the economy, ecosystem and human life. 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.

Natural disasters are the results of a hazard overwhelming highly vulnerable community,often resulting in mortality and morbidity.Over the past decade,over 300 natural disasters occur yearly around the world affecting millions and cost billions.

The disaster cycle is a framework used to base a coordinated plan to respond,recover,prevent,and prepare for a disaster.Access to clean water,proper sanitation,food/nutrition,shelter,and the threat of communicable diseases are concerns that have potential to be detrimental to the management of a natural disaster,slowing the recovery process.

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.

To tackle this problem, we developed a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster of natural The model uses an integrated webcam to capture the video frame and the video frame is compared with the Pre-trained model and the type of disaster is identified and showcased on the OpenCV window.

1.2 Purpose:

Artificial intelligence (AI), in particular machine learning (ML&DL), 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.

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. However, there are several objectives that are integrated with it. Those are,

1. Identifying the hazard and its cause.

2. Reducing vulnerability and potential losses of hazard.

3. Assessing, reviewing and controlling the risk.

4. Applying efficient, effective, sustainable relief (food, shelter and money), medical and other facilities in disaster affected people thus they can survive.

5. Reducing the damage, death, sufferings and destruction of any natural and human induced disaster.

6. Giving protection to victims.

7. Increasing the strength among people to survive against disasters.

8. Building up capacity in every sector like- individual, social, economic, environmental, regional, national and international.

9. Ensuring the availability of local emergency equipment and transportation.

10.Promote the culture of disaster risk prevention and mitigation at all levels.

2. LITERATURE SURVEY:

2.1 Existing problem:

Studies analyzing the intensity of natural disasters have gained significant attention in the current decade. A. Ashiquzzaman et al. [1] utilized a video source for fire detection; processing video sources is a feasible task due to convolutional neural networks (CNNs), which require high performance computational resources including graphics hardware, and thus a smart and cost-effective fire detection network is proposed based on architecture of convolutional neural networks.

In convolutional neural networks, a model to detect wildfire smoke named wildfire smoke dilated dense net was proposed by Li et al. [2], consisting of a candidate smoke region segmentation strategy using an advanced network architecture. Mangalathu et al. [4] performed an evaluation of building clusters affected by earthquakes by exploring the deep learning method, which uses long short-term memory.

Natural disasters are unpredictable events, Hartawan et al. [5] enhanced multilayer perceptron algorithm by including convolutional neural network implemented on raspberry pi to find out the victims of natural disasters using streaming cameras and to aid the evacuation team to rescue the disaster victims. Amit et al. [6] proposed applying automatic natural disaster detection to a convolutional neural network using the features of disaster from resized satellite images of landslide and flood detections. Aerial images are able to show more specific and wider surface area of the ground, which helps acquire a vast amount of information about the occurrence of disaster.

2.2 References:

[1] Ashiquzzaman A., Oh S.M., Lee D., Lee J., Kim J. Springer; Berlin/Heidelberg, Germany: 2021. Context-aware deep convolutional neural network application for fire and smoke detection in virtual environment for surveillance video analysis; pp. 459–467. [Google Scholar]

[2] Li T., Zhao E., Zhang J., Hu C. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. [CrossRef] [Google Scholar]

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

[4] Hartawan D.R., Purboyo T.W., Setianingsih C. Disaster Victims Detection System Using Convolutional Neural Network Method; Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT); Bali. [Google Scholar]

[5]Amit S.N.K.B., Aoki Y. Disaster detection from aerial imagery with convolutional neural network; Proceedings of the 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC) [Google Scholar]

2.3 Problem Statement Definition:

The problem statement aims at developing a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of natural disaster. The model uses an integrated webcam to capture the video frame and the video frame is compared with the Pre-trained model and the type of disaster is identified and showcased on the OpenCV window.

Problem Statement is inclusive of below answers:

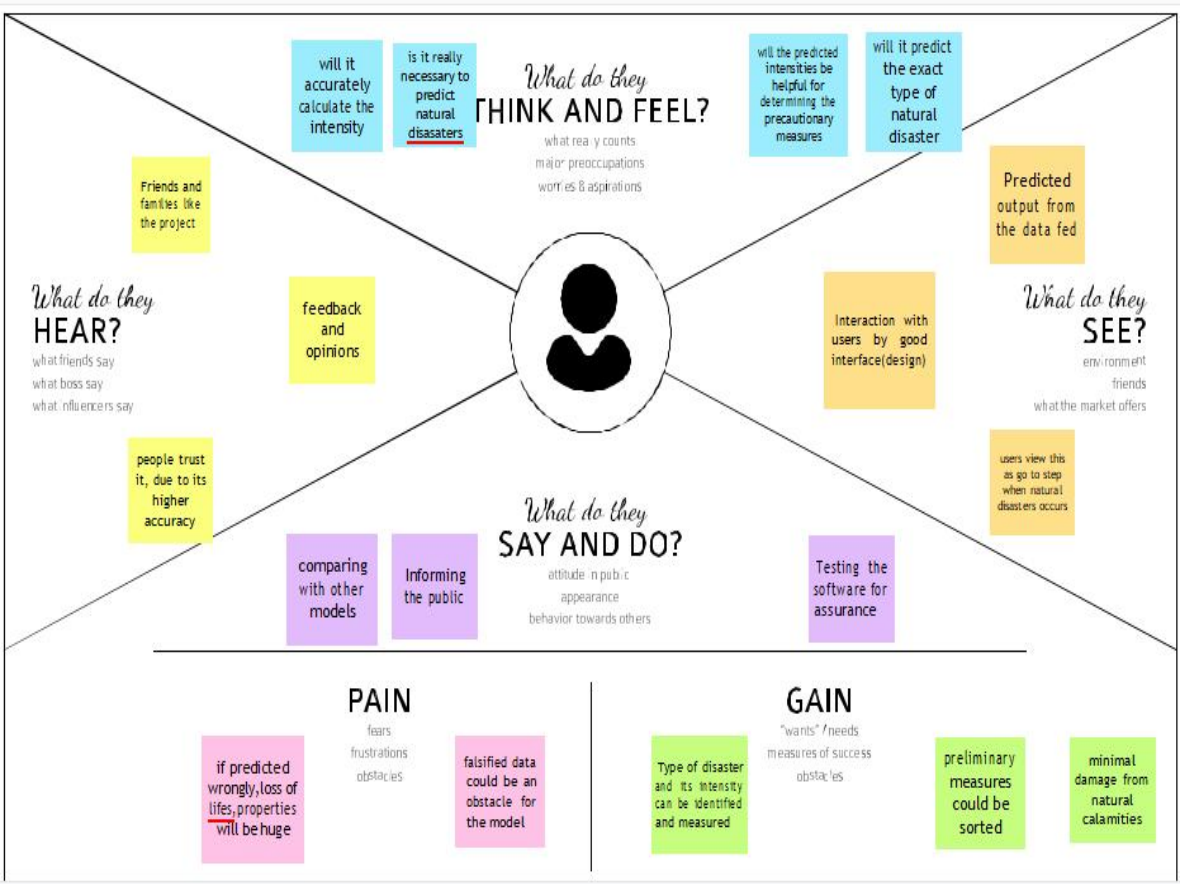
- Who does it affect/does not affect?
- What does it affect/does not affect?
- How does it affect/does not affect?
- When is it a problem/is not a problem.
- Where is it a problem/is not a problem.

Problem Statement	**Description**
Who does the problem affect?	The users can mostly prevent them from being a victim for natural disasters as the model predict it's intensity.
Why is it important to use?	This problem is important to use because it could help us to save our lives and properties.
What are the benefits?	Safety measures are taken with respect to the intensity outputted.
How is it better than others?	Provides results with better accuracy
When to use?	In any place, if there seen any potential threat due to the natural disaster, we could use the model

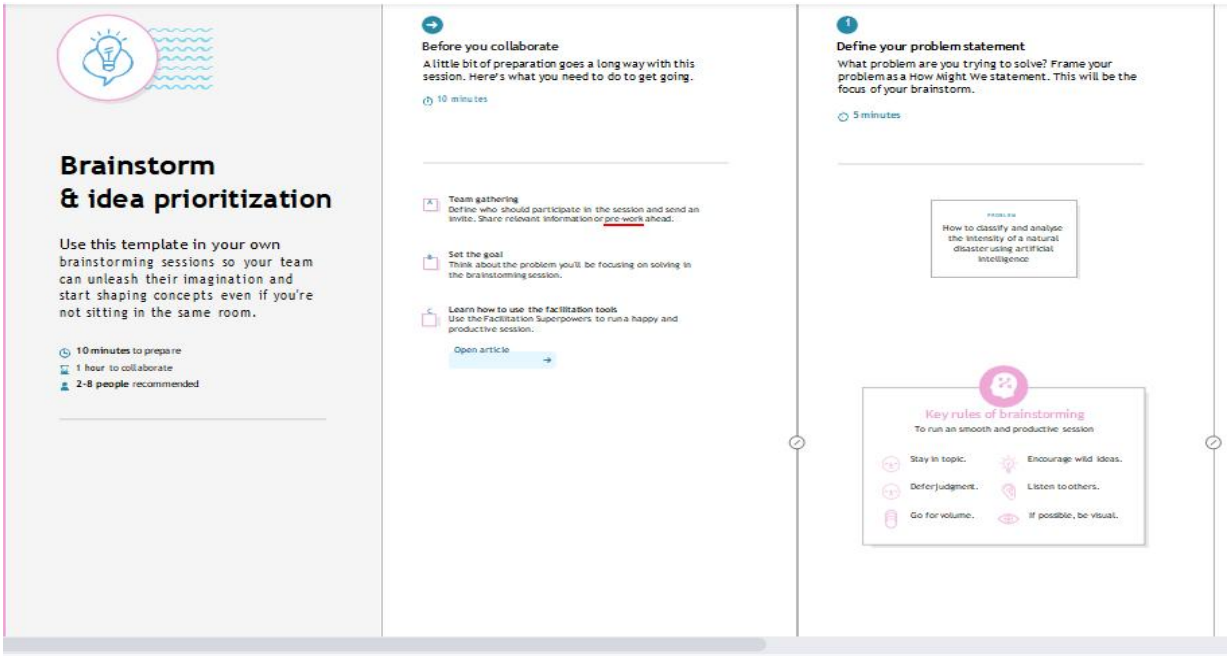
3. IDEATION & PROPOSED SOLUTION:

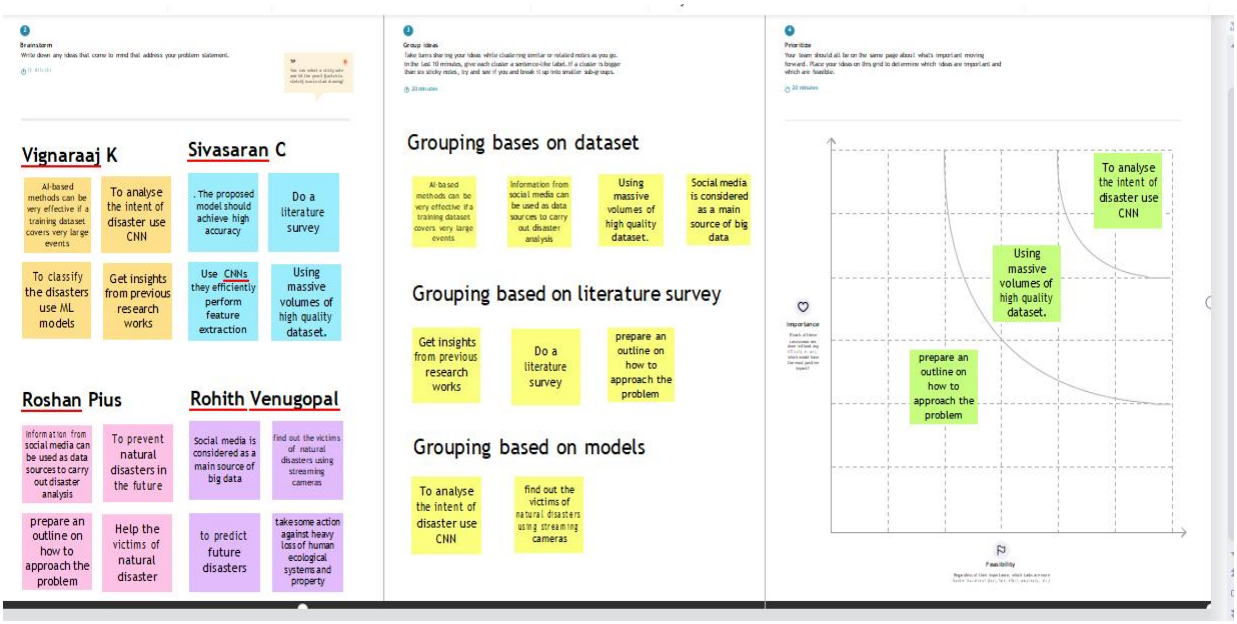
3.1 Empathy Map Canvas:

Gain insights and understanding on solving customer problems.

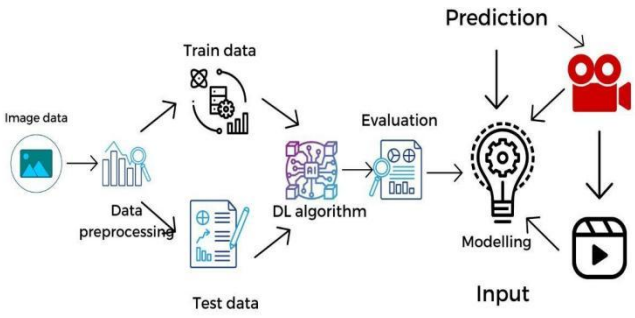


3.2 Ideation & Brainstorming:

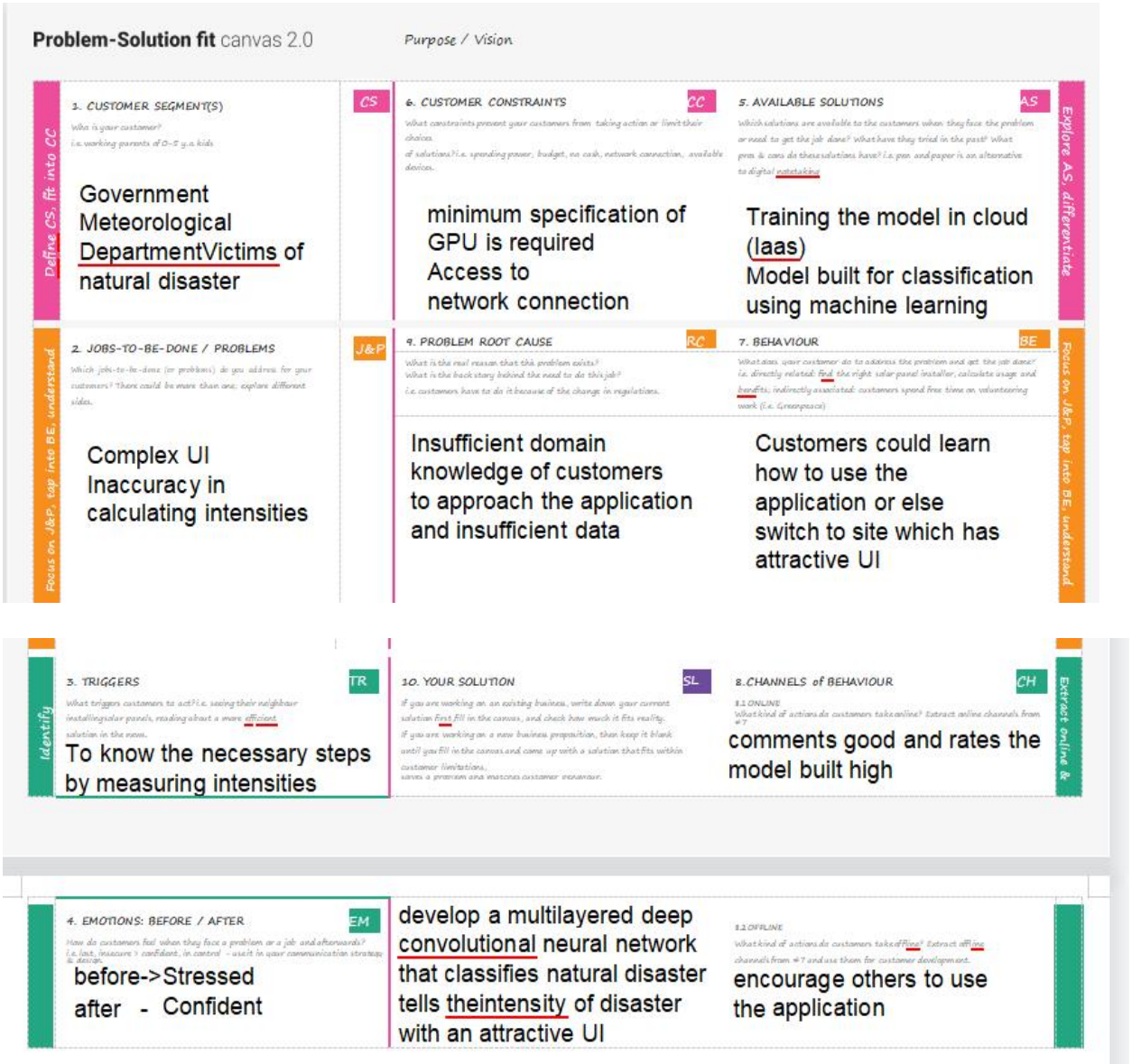




3.3 Proposed Solution:

S.NO.	PARAMETER	DESCRIPTION
1.	Problem statement(Problem to be solved)	To classify the natural disaster and calculate the intensity of the disaster.
2.	Idea/Solution description	To develop a multilayered deep convolutional neural network model that classifies the natural disaster and tells the intensity of disaster.
3.	Novelty/Uniqueness	We are implementing neural networks to train our model instead using machine learning algorithms which expected to provide with better accuracy.
4.	Social Impact/Customer Satisfaction	With better accuracy in predicting intensities precautions are taken respectively.
5.	Business Model(Revenue Model)	 The software is cheap, and the minimum requirements are affordable.
6.	Scalability of the Solution	Better accuracy in measuring the intensities of the natural disaster and in classifying it.

3.4 Proposed Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**Designation of Region**	• Ease of selection of necessary areas to be monitored • Versatile and Flexible operations an designated areas
FR-2	**Analysis of Required Phenomenon**	• Simple and easy analysis on the specific phenomenon to be observed
FR-3	**Accumulation of required Data**	• Fast and Efficient data gathering capabilities regarding past event analysis and future prediction
FR-4	**Organizing Unstructured data**	• Processing of raw and clustered data into clear and refined data which is useful for analysis and prediction tasks
FR-5	**Algorithm selection**	• The freedom to choose from several classes of algorithm to be used in the process • Customization of algorithm to suit the needs of a specific purpose.
FR 6	**Prediction and analysis of data the process**	• Advanced visualization techniques to help visualize the processed data for effective
FR-7	**Report generation**	• Restructuring of obtained results into clear and detailed report for future studies.

4.2 Non-functional Requirements:

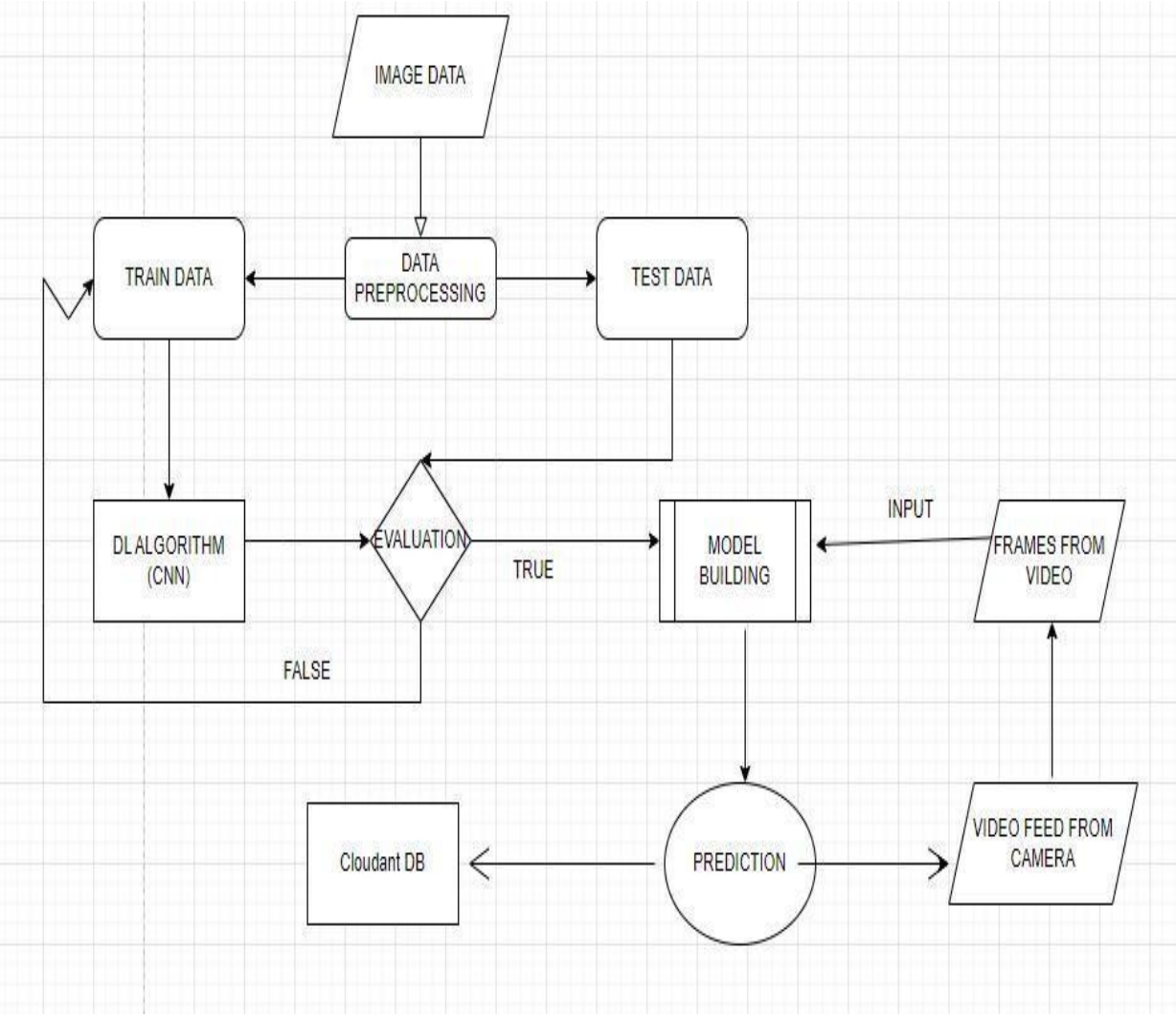
Following are the non-functional requirements of the proposed solution.

NFR No	Non-Functional Requirement	Description
NRF - 1	**Usability**	The system must be efficient and easy for the user to carry out tasks.
NRF - 2	**Security**	User details must be secured.
NRF - 3	**Reliability**	The product is robust and is capable of execution of processes even in the most difficult and unpredictable environments.
NRF - 4	**Performance**	The system should be able to handle many users without performance deterioration.
NRF - 5	**Availability**	The system should be accessible to a user at a given point in time
NRF -6	**Scalability**	The website pages should load fast with the total number of simultaneous users.

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:

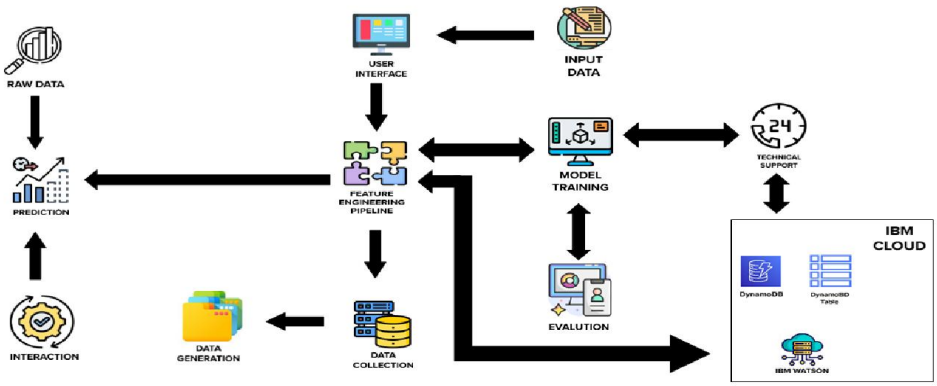
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

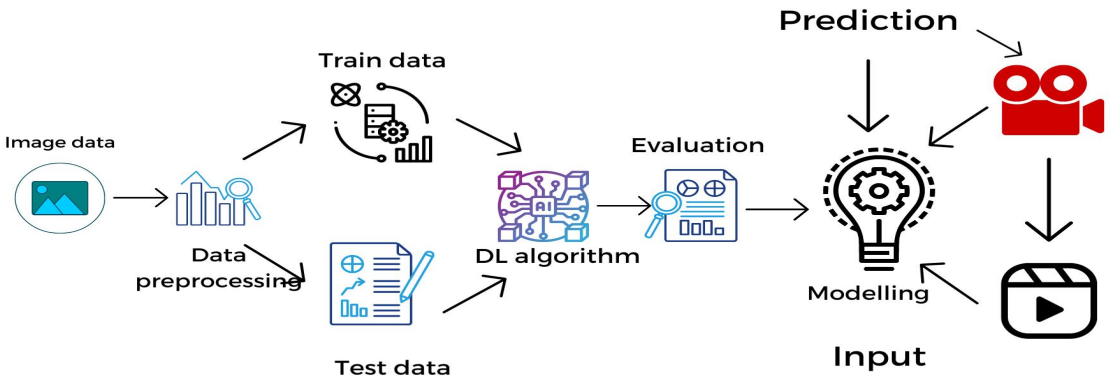


5.2 Solution & Technical Architecture:

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

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.





Components & Technologies:

1. User Interface

Description

User interacts with the application for the prediction of Any Natural disaster which will happen in future minutes.

Technology

HTML, CSS, Flask, Python.

2. Feature Engineering Pipeline

Description

Algorithms can't make sense of raw data. We have to select, transform, combine, and otherwise prepare our data so the algorithm can find useful patterns.

Technology

Image processing, pattern extraction, etc.

3. Model Training kit

Description

It learns patterns from the data. Then they use these patterns to perform particular tasks

Technology

deep learning multilayer convolutional neural network(CNN).

4. Prediction unit

Description

This function is used to predict outcomes from the new trained data to perform new tasks and solve new problems.

Technology

Neural networks.

5. Evaluation system

Description

It monitors that how Algorithm performs on data as well as during training

Technology

Chi-Square, Confusion Matrix, etc.

6. Interactive services

Description
To interact with our model and give it problems to solve. Usually this takes the form of an API. a user interface or a command-line interface
Technology
Application programming interface, etc.

7. Data collection unit

Description

Data is only useful if it's accessible, so it needs to be stored ideally in a consistent structure and conveniently in one place.
Technology
IBM Cloud, SQL Server.

8. Data generation system

Description

Every machine learning application lives off data. That data has to come from somewhere. Usually, it's generated by one of your core business functions
Technology
Synthetic data generation.

9. IBM Cloud services
Description
Processed data stored in cloud service which can be access by the admin anywhere over the internet.
Technology
IBM Cloud etc.

Application Characteristics:

1. Open-Source Frameworks

Description
An open source framework is a template for software development that is designed by a social network of software developers. These frameworks are free for public use and provide the foundation for building a software application.
Technology
Keras, tensor flow.

2. Application interface

Description

User uses mobile application and web application to interact with model
Technology
Android and Web Development (PhoneGap, ReactNative, and NativeScript).

3. Availability (both live camera update and upload video)

Description

It includes both the above mentioned work. A good internet connection is needed for online work to explore the software perfectly. Offline work includes the saved data to explore for later time.
Technology
Caching, backend server.

4. Regular Updates

Description

The truly excellent software product needs a continuous process of improvements and updates. Maintain your server and make sure that your content is always up-to-date. Regularly update an app and enrich it with new features.

Technology

- Waterfall Approach
- Incremental Approach
- Spiral Approach

5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Upload Data	USN-1	As a user, I can upload either a live stream, video or photo of the disaster	I can upload the data.	High	Sprint-1
Customer	Dashboard	USN-2	As a user, I can access the services and information provided in the dashboard	I can create interactive interfaces using html,css	Medium	Sprint-2
Functional Requirement (Epic)	Obtain Output	USN-3	As a user, I can receive the classification and the intensity of the disaster	I can receive the information about the disaster	High	Sprint-3
Customer care	Help Desk	USN-4	As a user, I can get the guidance from the customer care	I can get help from the customer care for carryingout my tasks	Low	Sprint-4

6.PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Landing page	USN-1	As an admin, I can upload the saved video to the application	2	High	Vignaraaj, Sivasaran
Sprint-1		USN-2	As an admin, I will access the live camera and feed the input to the application	1	High	Roshan, Rohith
Sprint-2		USN-3	As an admin, I can develop code to predict output from live camera	2	Low	Sivasaran, Rohith
Sprint-2		USN-4	As an admin, I can develop code to predict output from uploaded video	2	Medium	Roshan, Vignaraaj
Sprint-1		USN-5	As a user, I can access the information about the natural disasters	1	High	Vignaraaj, Rohith
Sprint-1	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	2	High	Sivasaran, Vignaraaj
Sprint-1		USN-7	As an admin,I can create the interactive interfaces	2	High	Vignaraaj, Sivasaran
Sprint-4	Helpdesk	USN-8	As a user, I can get the guidance from thecustomer care	1	High	Sivasaran, Roshan, Rohith
Sprint-3	Management	USN-9	As an administrator, I can collect new datasetsand keep the model trained	2	High	Vignaraaj
Sprint-3		USN-10	As an administrator, I can update other	2	Medium	Sivasaran, Roshan

			features of the application			
Sprint-3		USN-11	As an administrator, I can maintain the information about the user	2	medium	Rohith, Sivasaran
Sprint-4		USN-12	As an administrator, I can maintain third-partyservices	1	Low	Roshan

6.2 Sprint Delivery Schedule:

Sprint	Total Story Point	Duration	Start Date	End Date (Planned)	Story Points Completed	Sprint Release Date
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	4	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	19 Nov 2022

7.CODING & SOLUTIONING:

A convolutional neural network is a class of artificial neural networks. It is a Deep Learning algorithm that can take in an input image, assign importance to various objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.

The advantage of CNN is to provide an efficient dense network which performs the prediction or identification efficiently.

Feature 1:

One of the features enabled in this application is uploading the data e.g(video,image), and saving the data for further prediction. The predicted output is displayed in the screen.

Code is attached below:

```
@app.route('/uploader',methods=['GET','POST'])
def uploader():
  if request.method == "POST":
    f = request.files['filename']
    f.save("Flask/videos/save.mp4")
  cap=cv2.VideoCapture("Flask/videos/save.mp4")
  while(True):
    _,frame = cap.read()
    frame=cv2.flip(frame,1)
    while(True):
      (grabbed,frame) = cap.read()
      if not grabbed:
        break
      output = frame.copy()
      frame = cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
      frame = cv2.resize(frame,(64,64))
      x=np.expand_dims(frame,axis=0)
      result = np.argmax(model.predict(x),axis=1)
      index=['Cyclone','Earthquake','Flood','Wildfire']
      result = str(index[result[0]])
      #print(result)
      cv2.putText(output,"activity:
{}".format(result),(10,120),cv2.FONT_HERSHEY_PLAIN,1,(0,25,255),1)
      cv2.imshow("Output",output)
    if cv2.waitKey(0) & 0xFF==ord('q'):
      break
  print("[INFO]cleaning up...")
  cap.release()
  cv2.destroyAllWindows()
  return render_template("upload.html")
```

Feature 2:

Another feature enabled in this application is accessing the live camera using Open CV framework and processing the live video frame by frame for further prediction. The predicted output is displayed in the screen.

Code is attached below.

```
@app.route('/livecam',methods=['GET','POST'])
def livecam():
    cap=cv2.VideoCapture(0)
    while(True):
        (grabbed,frame) = cap.read()
        if not grabbed:
            break
        output = frame.copy()
        frame = cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
        frame = cv2.resize(frame,(64,64))
        x=np.expand_dims(frame,axis=0)
        result = np.argmax(model.predict(x),axis=-1)
        index=['Cyclone','Earthquake','Flood','Wildfire']
        result = str(index[result[0]])
        cv2.putText(output,"activity:
          {}".format(result),(10,120),cv2.FONT_HERSHEY_PLAIN,1,(0,25,255),1)
        cv2.imshow("Output",output)
        if cv2.waitKey(0) & 0xFF==ord('q'):
            break
    print("[INFO]cleaning up...")
    cap.release()
    cv2.destroyAllWindows()
    return render_template("upload.html")
```

8. TESTING:

8.1 Test Cases

Test ID	Test Description	Test Input	Expected Result	Actual Result	Status
ID-01	Load the trained Model into the application	Analysis.h5	Prompts to next line	Prompts to the next line	Pass
ID-02	Upload the video as a customer	Disaster.mp4	File uploaded and ready for prediction	File uploaded and sent for prediction	Pass
ID-03	Open integrated camera to capture live video	Cv2.Videocapture(0)	Opens integrated live camera and records	Opens camera and split the recording frame by frame	pass
ID-04	Train and test the model	Dataset.csv	Dataset trains with a good accuracy	Provides Trained Model With high accuracy	Pass

8.2 User Acceptance Testing

The purpose of this document is to briefly explain the test coverage and open issues of the Natural disaster intensity analysis and classification using artificial intelligence project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

This shows how many bugs were fixed or closed at each severity level and how they were fixed.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	3	2	4	14
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	10	2	4	14	30
Not Reproduced	0	0	1	0	1
Skipped	5	1	1	0	7
Won't Fix	0	5	2	1	8
Totals	23	14	13	20	70

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	7	0	0	7
Client Application	30	0	0	24
Security	0	0	0	0
Out-source Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS

9.1 Performance Metrics:

S.No.	Parameter	Values
1.	Model Summary	**CNN-** {optimizer:Adam,loss:categorical_ crossentropy,metrics:accuracy}
2.	Accuracy	Training Accuracy - 91.51% Validation Accuracy -73.74%

Our Project marks the successive performance by implementing in order to be cost effective and more reliable to use and to predict the future from the natural disaster that we are ahead of. The successive way includes the objectives, activities and the approaches for the project. It mainly includes the trained dataset which gives an excessive measure of success which helps to overcome the future from this natural disaster.

10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- The use of AI to forecast natural disasters would save millions of lives. Furthermore, the information evaluated by AI-powered systems can aid in understanding the scale and patterns of natural catastrophes such as floods, earthquakes, and tsunamis, which would aid in improved infrastructure development in disaster-prone areas.

DISADVANTAGES:

- Sometimes the prediction may fail and result in huge loss.

11.CONCLUSION

Many researchers have attempted to use different deep learning methods for detection of natural disasters. However, the detection of natural disasters by using deep learning techniques still faces various issues due to noise and serious class imbalance problems.

To address these problems, we proposed a 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.

12. FUTURE SCOPE

The overall proposed model works on an image dataset to detect and classify the natural disasters. As the model is evaluated on a simple central processing unit (CPU)-based system, it only detects disaster types and then classifies them into cyclone, earthquake, flood and wildfire classes.

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.

13. APPENDIX:

Source Code

home.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
   <title>Document</title>
</head>
<body>
   <div class="card text-center">
      <div class="card-header">
        <ul class="nav nav-tabs card-header-tabs">
          <li class="nav-item">
            <a class="nav-link active" aria-current="true" href="{{ url_for('home') }}" style="font-size:
24px;">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('intro') }}" style="font-size: 24px;">Introduction</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('upload') }}" style="font-size: 24px;">Upload</a>
          </li>
        </ul>
        <h3 style="float: right;">AI based Natural Disaster Analysis</h3>
      </div>
    </div>
    <div class = "container" style="text-align: center;">
    <div class="card" style="width: 18rem; padding: 10px; margin: 40px; margin-left: 40px;display:inline-
block">
      <img class="card-img-top" src="{{ url_for('static', filename='cyclone.jpg') }}" alt="Card image cap">
      <div class="card-body" >
        <h5 class="card-title">Cyclone</h5>
```

```
        <p class="card-text">cyclone, large system of winds that circulates counterclockwise direction north of the Equator and clockwise direction to the south.</p>
        <a href="https://en.wikipedia.org/wiki/Cyclone" class="btn btn-primary">Know more</a>
       </div>
      </div>
     <div class="card" style="width: 18rem; padding: 10px; margin: 40px; margin-left: 40px;display:inline-block">
       <img class="card-img-top" src="{{ url_for('static', filename='earthquake.jpg') }}" alt="Card image cap">
       <div class="card-body" >
        <h5 class="card-title">Earthquake</h5>
        <p class="card-text">A sudden violent shaking of the ground, causing great destruction, as a result of movements within the earth's crust.</p>
        <a href="https://en.wikipedia.org/wiki/Earthquake" class="btn btn-primary">Know more</a>
       </div>
       </div>
     </div>
     <div class = "container" style="text-align: center;">
       <div class="card" style="width: 18rem; padding: 10px; margin: 40px; margin-left: 40px;display:inline-block">
        <img class="card-img-top" src="{{ url_for('static', filename='flood.jpg') }}" alt="Card image cap">
        <div class="card-body" >
         <h5 class="card-title">Flood</h5>
         <p class="card-text">An overflow of a large amount of water beyond its normal limits, especially over what is normally dry land.</p>
         <a href="https://en.wikipedia.org/wiki/Flood" class="btn btn-primary">know more</a>
        </div>
       </div>
       <div class="card" style="width: 18rem; padding: 10px; margin: 40px; margin-left: 40px;display:inline-block">
        <img class="card-img-top" src="{{ url_for('static',filename='wildfire.jpg') }}" alt="Card image cap">
        <div class="card-body" >
         <h5 class="card-title">Wild Fire</h5>
         <p class="card-text">A wildfire is an unplanned, uncontrolled and unpredictable fire in an area of combustible vegetation starting in rural and urban areas.</p>
         <a href="https://en.wikipedia.org/wiki/Wildfire" class="btn btn-primary">Know more</a>
        </div>
        </div>
       </div>
```

```
</body>
</html>
```

intro.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
   <title>Document</title>
</head>
<body>
   <div class="card text-center">
      <div class="card-header">
        <ul class="nav nav-tabs card-header-tabs">
          <li class="nav-item">
            <a class="nav-link" aria-current="true" href="{{ url_for('home') }}" style="font-size:
24px;">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" href="{{ url_for('intro') }}" style="font-size: 24px;">Introduction</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('upload') }}" style="font-size: 24px;">Upload</a>
          </li>
        </ul>
        <h3 style="float: right;">AI based Natural Disaster Analysis</h3>
      </div>
    </div>
    <h2 style="padding: 50px; margin: 50px; word-spacing: 15px; text-align: center ;line-height: 1.6;">
      China, India and the United States are among the countries in the world most
affected by natural disasters.
Natural disasters have the potential to wreck and even end the lives of those people,
who stand in their way. <br><br> However, whether or not you are likely to be
```

affected by a natural disaster dramatically depends on where in the world you live, The objective of the project is to human build a web application to detect the type of disaster. The input is taken from the in-built webcam, which in turn is given to the pre-trained model.
The model predicts the type of disaster and displayed on UI.

</h2>
</body>
</html>

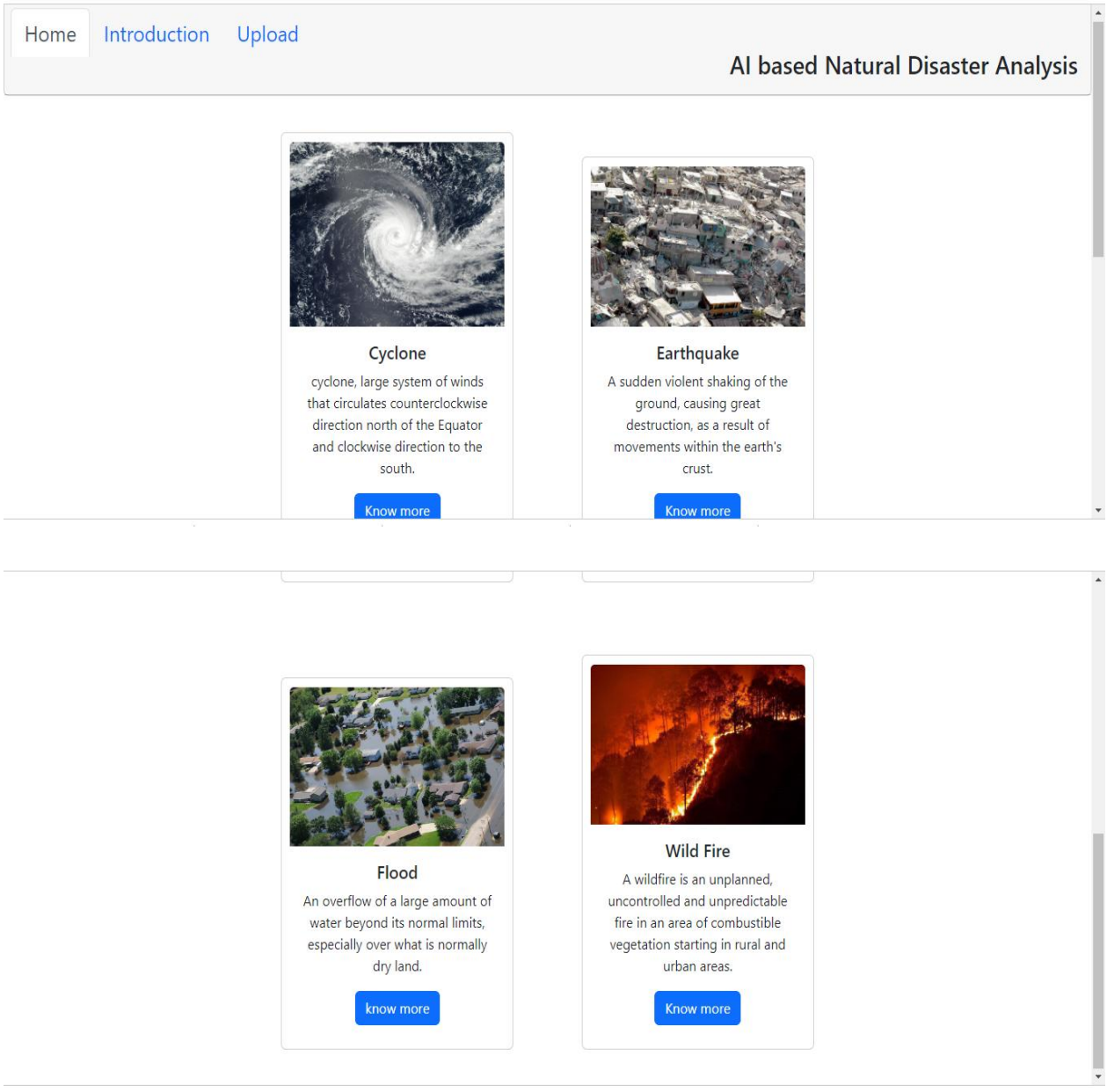
upload.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
  <title>Document</title>
  <style>
    table {
      max-width: 100%;
    }
    tr:nth-child(odd) {
      background-color: #eee;
    }
    th {
      background-color: #555;
      color: #fff;
    }
    th,
    td {
      text-align: left;
      padding: 0.5em 1em;
    }
    .center {
      margin-left: auto;
      margin-right: auto;
```

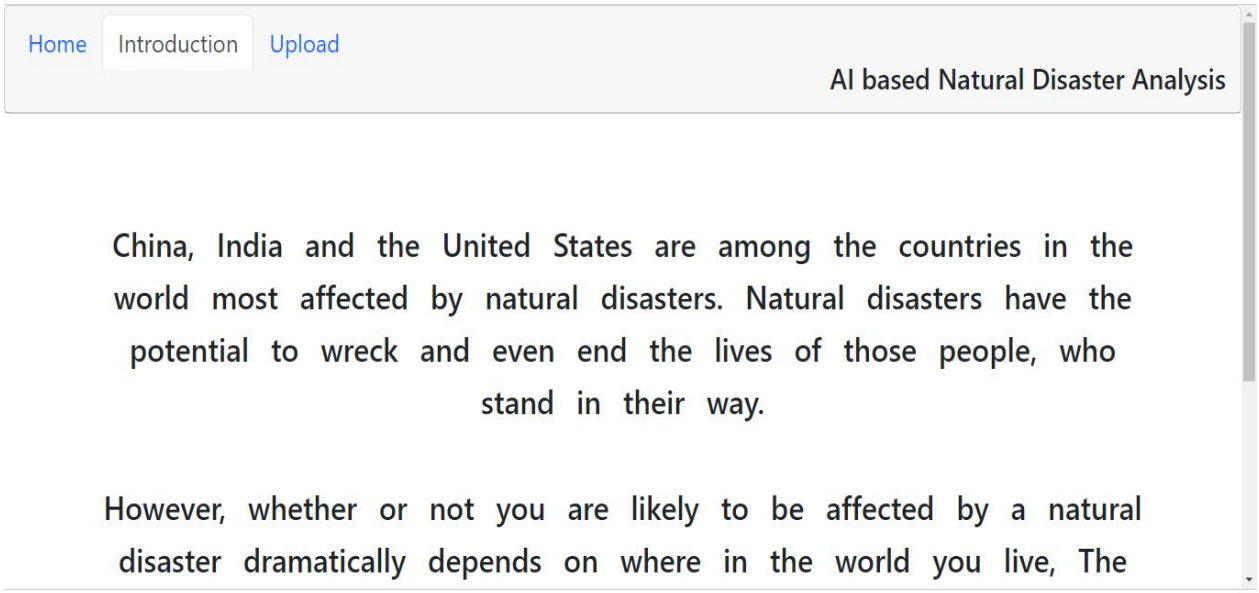
```
    }
    table, th, td {
      border: 1px solid black;
    }

  </style>
</head>
<body>
  <div class="card text-center">
    <div class="card-header">
      <ul class="nav nav-tabs card-header-tabs">
        <li class="nav-item">
          <a class="nav-link" aria-current="true" href="{{ url_for('home') }}" style="font-size:
24px;">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{{ url_for('intro') }}" style="font-size: 24px;">Introduction</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" href="{{ url_for('upload') }}" style="font-size: 24px;">Upload</a>
        </li>
      </ul>
      <h3 style="float: right;">AI based Natural Disaster Analysis</h3>
    </div>
  </div><br><br><br><br><br>
  <table class="center">
    <tr>
      <th>UPLOAD VIDEO</th>
      <th>LIVE DETECT</th>
    </tr>
    <tr>
      <td>
        <form action ="{{ url_for('uploader') }}"  method = "POST" enctype = "multipart/form-data">
          <input type = "file" name = "filename" />
          <input type = "submit" value="Submit"/>
        </form>
      </td>
      <td>
```

```
        <form action="{{ url_for('livecam') }}" method="POST">
            <input type="submit" value="Open camera"/>
        </form>
       </td>
     </tr>
   </table>  <br><br><br>
   <p style="text-align:center">Enter 'q' to quit</p>
  </body>
</html>
```

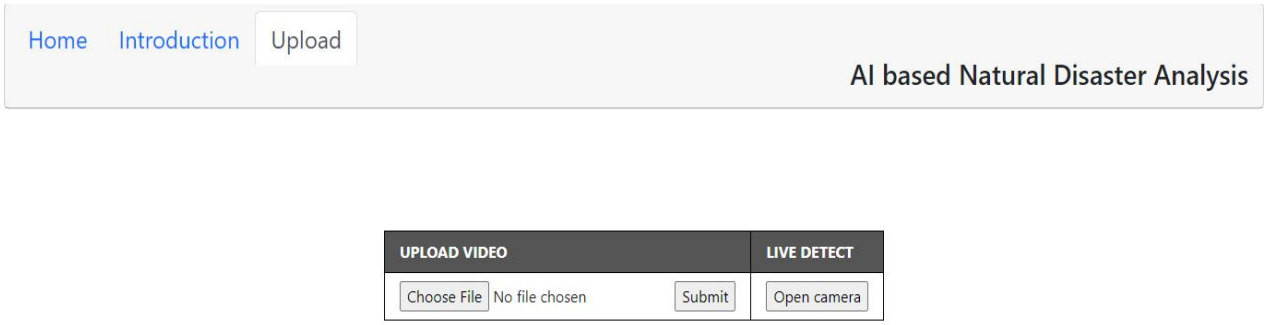
Home.html



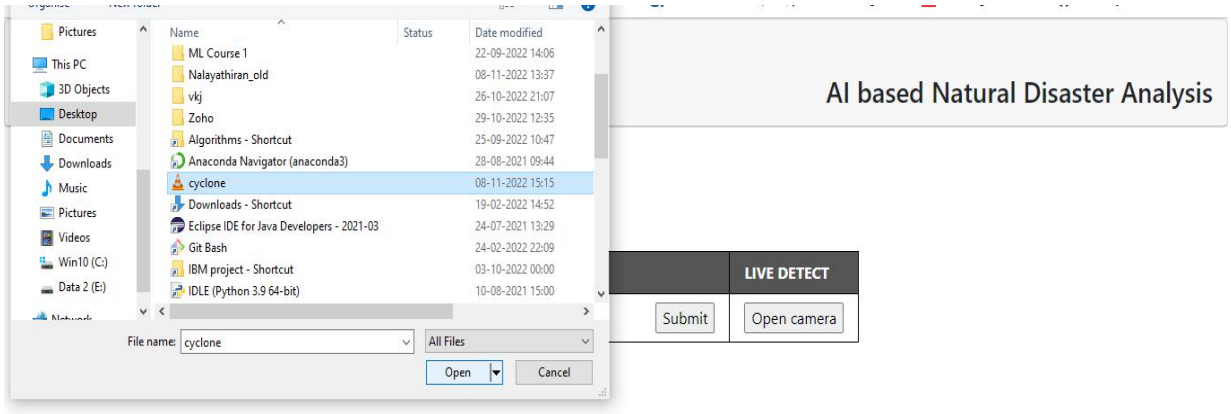
Intro.html:



Upload.html:

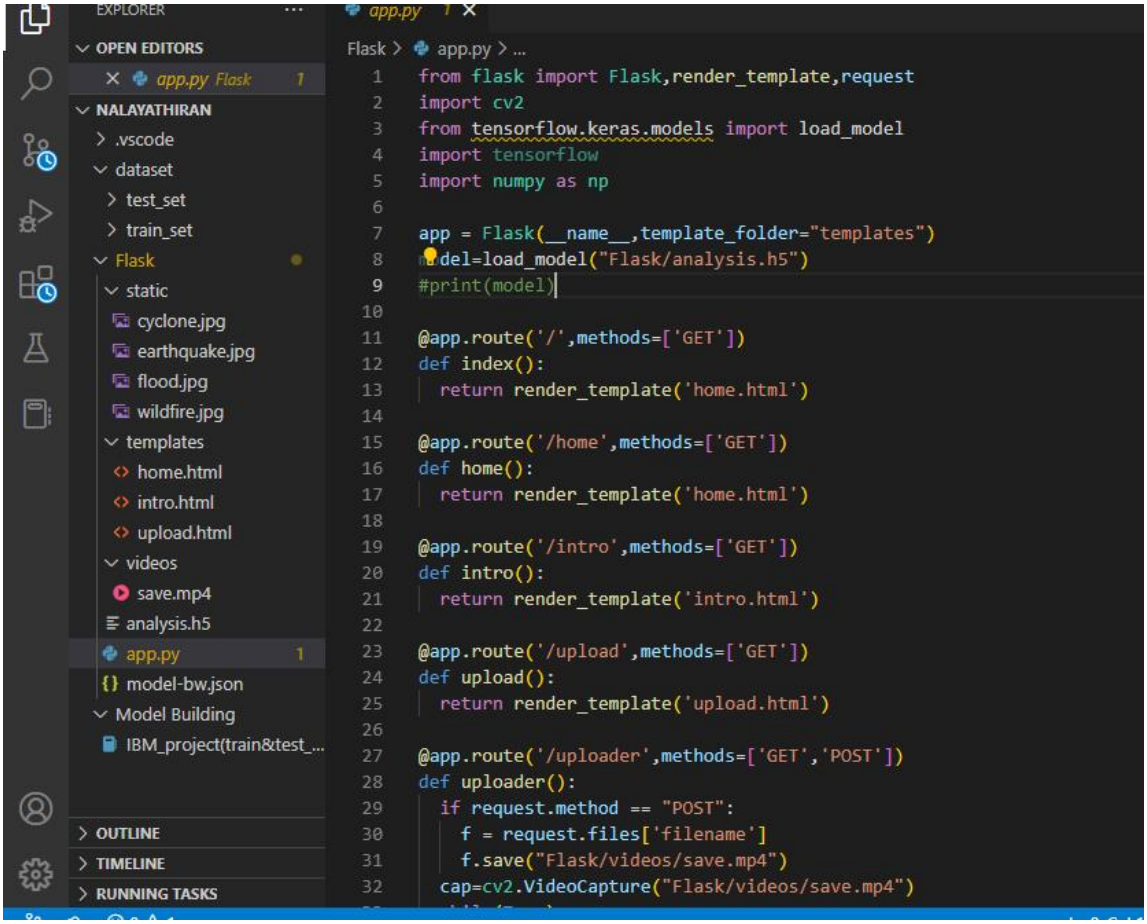


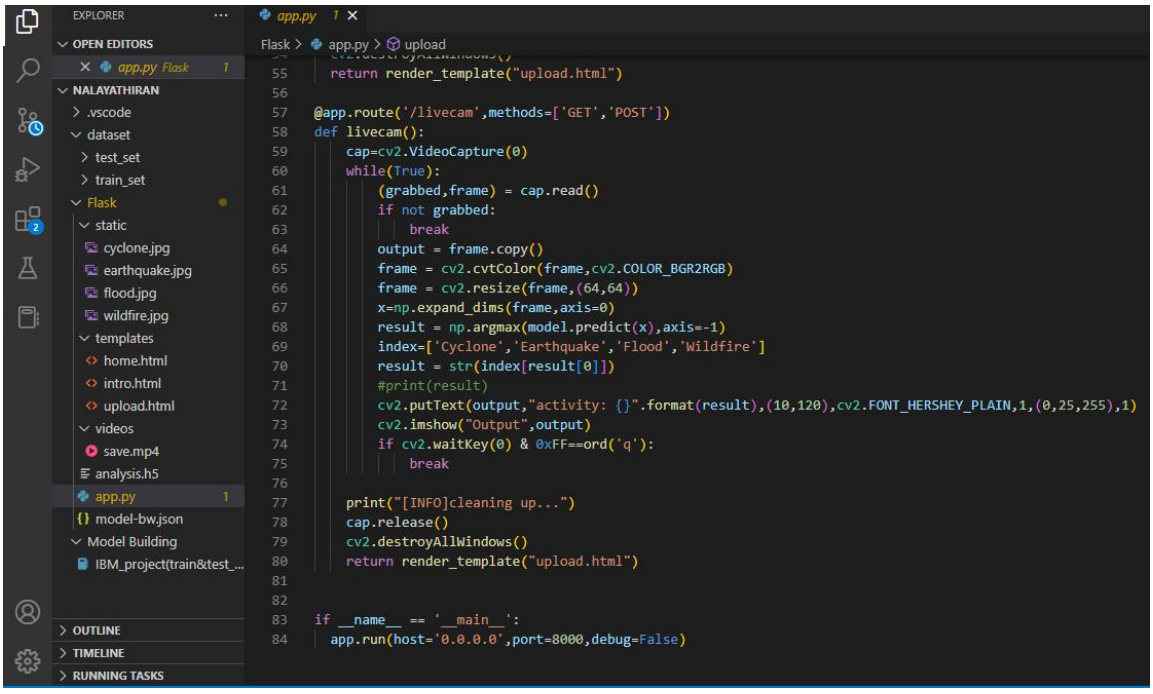
Enter 'q' to quit



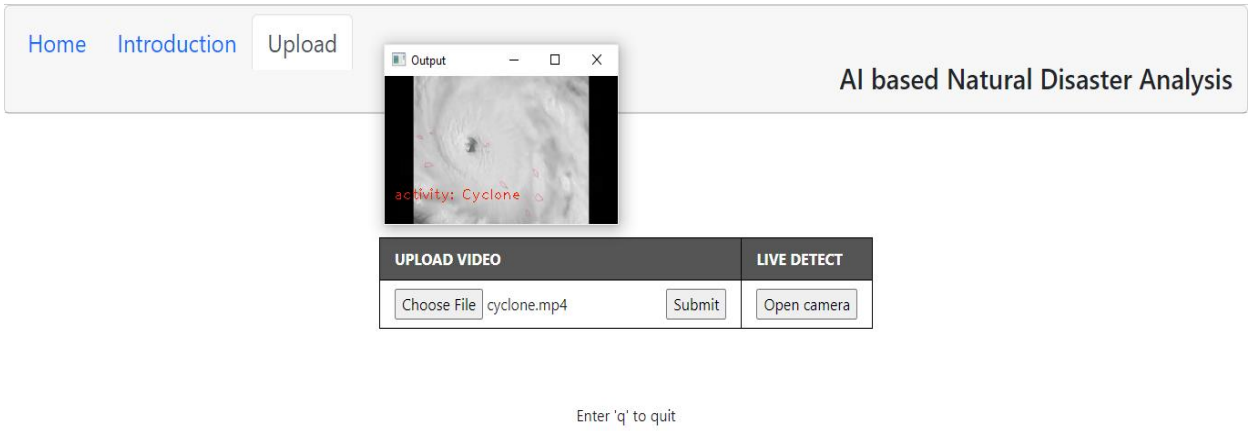
Enter 'q' to quit

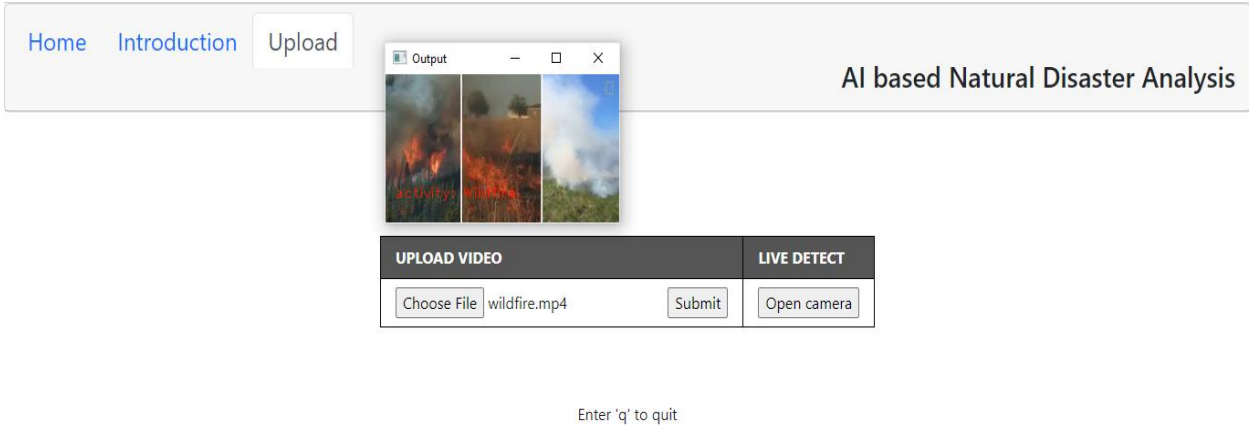
Flask(application.py)





OUTPUT:


Home
Introduction
Upload
Output
activity: Cyclone
AI based Natural Disaster Analysis
UPLOAD VIDEO
LIVE DETECT
Choose File
cyclone.mp4
Submit
Open camera
Enter 'q' to quit


Home
Introduction
Upload
Output
AI based Natural Disaster Analysis
UPLOAD VIDEO
LIVE DETECT
Choose File
wildfire.mp4
Submit
Open camera
Enter 'q' to quit

AI based Natural disaster analysis.ipynb

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,shear_range=0.2)
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
x_train=train_datagen.flow_from_directory("/content/drive/MyDrive/dataset/train_set",target_size=(64,64),class_mode='categorical',batch_size=5,color_mode='rgb')
x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/dataset/test_set",target_size=(64,64),class_mode='categorical',batch_size=5,color_mode='rgb')
```

```
Found 742 images belonging to 4 classes.
Found 198 images belonging to 4 classes.
```

```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Conv2D,MaxPooling2D,Flatten
```

```
model=Sequential()
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

+ Code + Text

```
[ ] model.add(Dense(units=128,activation='relu'))
    model.add(Dense(units=4,activation='softmax'))
    model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
[ ] model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 128)               802944
```

```
[ ] model.fit_generator(generator=x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=20)

    Epoch 1/20
    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a futu
      """Entry point for launching an IPython kernel.
    149/149 [==============================] - 44s 293ms/step - loss: 1.1635 - accuracy: 0.4798 - val_loss: 0.9364 - val_accuracy: 0.6566
    Epoch 2/20
    149/149 [==============================] - 41s 273ms/step - loss: 0.8416 - accuracy: 0.6429 - val_loss: 0.8283 - val_accuracy: 0.6717
    Epoch 3/20
    149/149 [==============================] - 42s 284ms/step - loss: 0.6678 - accuracy: 0.7655 - val_loss: 0.7795 - val_accuracy: 0.7323
    Epoch 4/20
    149/149 [==============================] - 41s 273ms/step - loss: 0.6775 - accuracy: 0.7493 - val_loss: 0.6493 - val_accuracy: 0.7626
    Epoch 5/20
    149/149 [==============================] - 41s 273ms/step - loss: 0.5995 - accuracy: 0.7749 - val_loss: 0.6781 - val_accuracy: 0.7879
    Epoch 6/20
    149/149 [==============================] - 41s 273ms/step - loss: 0.5397 - accuracy: 0.7817 - val_loss: 0.8131 - val_accuracy: 0.7172
    Epoch 7/20
    149/149 [==============================] - 42s 285ms/step - loss: 0.4696 - accuracy: 0.8275 - val_loss: 0.6780 - val_accuracy: 0.7879
    Epoch 8/20
    149/149 [==============================] - 41s 272ms/step - loss: 0.4959 - accuracy: 0.8194 - val_loss: 0.8018 - val_accuracy: 0.7576
    Epoch 9/20
    149/149 [==============================] - 41s 273ms/step - loss: 0.3969 - accuracy: 0.8544 - val_loss: 0.6865 - val_accuracy: 0.7828
    Epoch 10/20
    149/149 [==============================] - 41s 273ms/step - loss: 0.3885 - accuracy: 0.8652 - val_loss: 0.8218 - val_accuracy: 0.7677
```

```
[ ]

    {'Cyclone': 0, 'Earthquake': 1, 'Flood': 2, 'Wildfire': 3}
```

```
img = image.load_img(r"/content/drive/MyDrive/dataset/test_set/Earthquake/1347.jpg",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])

1/1 [==============================] - 0s 82ms/step
Earthquake
```

```
img = image.load_img(r"/content/drive/MyDrive/dataset/test_set/Cyclone/918.jpg",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
index=['Cyclone','Earthquake','Flood','Wildfire']
y=np.argmax(model.predict(x),axis=1)
print(index[int(y)])

1/1 [==============================] - 0s 23ms/step
Cyclone
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-11024-1659254459

DEMOLINK:

https://drive.google.com/file/d/1BUykEheLec-Tug4fUwEnAi5C_l3b0zvL/view