# ENVIROWATCH – MOBILIZING COMMUNITIES TO SOLVE VARIOUS ENVIROMENTAL ISSUES

**A PROJECT REPORT**

***Submitted by***

**BHARATH P 211420104037**

**BAALA B 211420104033**

**MARIA ROBIN ANDREW 211420104156**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**


**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**MARCH 2024**

# PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

## BONAFIDE CERTIFICATE

Certified that this project report **"ENVIROWATCH – MOBILIZING COMMUNITIES TO SOLVE VARIOUS ENVIRONMENTAL ISSUES"** is the bonafide work of **P.BHARATH, B. BAALA MARIA ROBIN ANDREW** who carried out the project work under my supervision.

**Signature of the HOD with date**

**Dr L. JABASHEELA M.E., Ph.D.,**
**Professor and Head,**
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai - 123

**Signature of the Supervisor with date**

**Dr P. J SATHISH KUMAR M.Tech., Ph.D.,**
**Professor,**
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123

Submitted for the Project Viva – Voce examination held on _____

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator Dr.N. PUGHAZENDI** and **Project Guide Dr. P. J. SATHISH KUMAR** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

**P. BHARATH**

**B. BAALA**

**MARIA ROBIN ANDREW**

# DECLARATION BY THE STUDENT

We BHARATH P (211420104037), BAALA B (211420104033), MARIA ROBIN ANDREW (211420104156) hereby declare that this project report titled "**ENVIROWATCH – MOBILIZING COMMUNITIES TO SOLVE VARIOUS ENVIROMENTAL ISSUES**", under the guidance of **Dr P. J SATHISH KUMAR M.Tech., Ph.D** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

# ABSTRACT

Understanding our various environmental issues like flood water stagnation, damaged roads prone to accident, garbage accumulation in public areas, post flood issues like electrical pole damage, etc, involves more than just knowing the immediate cause, effects, and damage; it also entails realizing the intricate interactions between environmental, social, and economic factors that lead to this recurring crisis. Disaster recovery is a challenging task that involves managing and prioritizing rescue efforts in addition to locating the impacted areas. The goal of this study is to limit the amount of resources needed for recovery in the aftermath of a disaster with the assistance of the people. The solution we propose will gather the bare minimum of information from the impacted individuals regarding the challenges and problems brought on by the tragedy using a mobile application. As data collected from affected individuals leads to overcome the reliance on the sensors, their maintenance and initial cost. The visual data is classified based on the problem using deep learning techniques which are then sent to the relevant departments to monitor the issue using our interface. We developed the deep learning model to utilize neural network technology for data classification and is trained on relevant data sets corresponding to different problems which include water stagnation, damaged electric pole, damaged roads and waste garbage accumulation. Our technology shows its ability to crowd source flood related issues from the community in tragedy and acts as a go-between for the affected people and the relevant authorities to accelerate the necessary work.

# TABLE CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 OVERVIEW

The proposed mobile application offers a comprehensive solution to address various environmental issues encountered by communities. Through a user-friendly interface built on Flutter, individuals can effortlessly report problems such as flood water stagnation, damaged roads, electrical pole damage, and accumulated garbage in public areas. This approach empowers citizens to actively participate in the enhancement of their surroundings by simply capturing and describing the issue they encounter. Upon submission, the data and images are transmitted to a Flask-based backend, where a deep learning algorithm comes into play. This algorithm efficiently classifies the received images into predefined categories corresponding to the identified environmental problems. This classification step serves as a pivotal component, ensuring that each complaint is directed to the appropriate government department responsible for its resolution. To streamline the process further, separate websites have been developed for each department, catering to flood management, road maintenance, electrical infrastructure, and sanitation. Upon classification, complaints are automatically redirected to the respective departmental websites, where officials can promptly review and take necessary action. For users, the application offers transparency and accountability. Through the mobile app, individuals can track the status of their submitted complaints, fostering a sense of engagement and ownership within the community. This feature not only ensures that users are kept informed about the progress of their reports but also encourages continuous participation in the improvement of environmental conditions. Overall, this integrated approach leverages technology to bridge the gap between citizens and governmental bodies, facilitating efficient communication and resolution of environmental concerns. By harnessing the power of mobile applications, deep learning algorithms, and dedicated departmental websites, the project aims to foster a collaborative environment where proactive community involvement contributes to the sustainable development and maintenance of a cleaner, safer, and more resilient urban landscape.

## 1.2 PROBLEM DEFINITION

The existing problem revolves around the inefficiency and lack of a streamlined process for addressing environmental issues faced by communities. Currently, there is no convenient mechanism for citizens to report environmental problems they encounter in their daily lives. This leads to underreporting and delays in addressing critical issues such as flood water stagnation, damaged roads, electrical pole damage, and garbage accumulation. Government departments responsible for environmental management often face a flood of complaints without clear categorization. This results in a significant amount of time and resources being wasted in sorting and assigning complaints to the appropriate departments.

Citizens often feel disconnected from the resolution process once they report an environmental issue. There is a lack of transparency regarding the status and progress of their complaints, leading to frustration and disengagement within the community. Communication between citizens and government departments is often fragmented and inefficient, resulting in delays and misunderstandings in resolving environmental concerns. Addressing these existing problems requires a comprehensive solution that facilitates easy reporting, efficient categorization, transparent communication, and effective resource allocation. By leveraging technology such as mobile applications, deep learning algorithms, and dedicated departmental websites, the proposed solution aims to overcome these challenges and create a more responsive and accountable system for managing environmental issues within communities.

# CHAPTER 2

# SYSTEM ANALSIS

## 2.1 EXISTING SYSTEM

The existing system for addressing environmental issues within the community suffers from several shortcomings, which hinder the efficient resolution of problems. Below is an overview of the existing system:

- **Manual Reporting Mechanisms:** Currently, citizens rely on manual methods such as phone calls, emails, or physical visits to government offices to report environmental issues. This process is cumbersome and time-consuming, leading to underreporting and delays in addressing critical problems.
- **Fragmented Communication Channels:** Communication between citizens and government agencies is often fragmented, with no centralized platform for submitting and tracking complaints. This fragmentation leads to inefficiencies and miscommunications, further delaying the resolution of environmental issues.
- **Limited Real-Time Data Collection:** Without sensor integration, the system relies heavily on manual reporting by citizens, which often results in delayed or incomplete data collection. Real-time information about environmental conditions, such as water levels during floods or air quality measurements, is not readily available. This limitation hampers the timely identification and response to emerging environmental threats.
- **Cost:** Installing and maintaining sensors can be expensive, especially if widespread coverage is desired. This cost may not be feasible for all communities, particularly those with limited resources.
- **Lack of Contextual Information:** Sensors typically provide quantitative data without contextual information or user insights. Direct data collection from users allows for qualitative information, such as detailed descriptions or photographs, which can provide valuable context for understanding and addressing environmental issues.

### 2.1.1 DISADVANTAGES

There are some potential disadvantages to consider:

- **Accuracy and Reliability:** Sensors may be prone to inaccuracies or malfunctions, leading to erroneous data collection. Factors such as calibration drift, environmental interference, or sensor degradation over time can impact the reliability of collected data.
- **Limited Flexibility:** Once sensors are installed, they are fixed in their locations and capabilities. Adapting sensor networks to accommodate changes in environmental conditions or community needs may require significant time and resources.
- **Manual Reporting Inefficiencies:** Relying on manual reporting methods such as phone calls or physical visits to government offices can lead to inefficiencies, including delays in reporting and processing complaints.
- **Inequitable Service Delivery:** Inconsistencies in the handling of complaints across different geographical areas or demographic groups may result in inequitable service delivery, with some communities receiving faster or more effective responses to their environmental concerns than others.
- **Dependency on Infrastructure:** Sensor-based data collection relies on infrastructure such as power sources and internet connectivity. In areas with unreliable infrastructure, sensor data may be incomplete or unavailable.
- **Inadequate Coverage:** Sensor networks may not provide complete coverage of all areas where environmental issues occur. Gaps in coverage can lead to incomplete or biased data, resulting in a skewed understanding of the overall situation.
- **Community Disengagement:** Relying solely on sensor data collection may exclude community members from actively participating in the identification and resolution of environmental issues. This lack of community engagement can result in a disconnect between decision-makers and the communities affected by environmental problems.

## 2.2 PROPOSED SYSTEM

The proposed system builds upon the existing framework to create a comprehensive and efficient solution for addressing environmental issues faced by communities. Leveraging

technology and user engagement, the proposed system aims to streamline the process of reporting, categorizing, and resolving environmental problems.

Central to the proposed system is a mobile application developed using Flutter, providing a user-friendly interface for citizens to report environmental issues they encounter. Through the application, users can easily capture images of the problems they observe, accompanied by textual descriptions if necessary. This streamlined reporting process encourages greater community involvement and ensures that environmental issues are promptly documented and communicated to the relevant authorities. Upon submission, the data and images are transmitted to a backend system developed with Flask and Python. Here, a deep learning algorithm is employed to analyze the images and classify them into predefined categories corresponding to the identified environmental problems. This classification step serves as a crucial component in the system, enabling automated routing of complaints to the appropriate government departments responsible for their resolution.

To facilitate the handling of complaints, separate websites are created for each government department tasked with addressing specific environmental issues. These departmental websites serve as centralized platforms for officials to review and manage incoming complaints. Upon classification of the images, the system automatically redirects the complaints to the respective departmental websites based on the identified problem category. This efficient routing mechanism ensures that complaints are directed to the appropriate authorities without delay, streamlining the process of problem resolution. For users, the proposed system offers transparency and accountability through a feature-rich mobile application. Users can track the status of their submitted complaints directly from the application, providing them with real-time updates on the progress of their reports. This feature enhances user engagement and fosters a sense of ownership within the community, encouraging continued participation in the improvement of environmental conditions. Overall, the proposed system represents a holistic approach to environmental management, integrating technology, user engagement, and government collaboration. By leveraging mobile applications, deep learning algorithms, and departmental websites, the system aims to bridge the gap between citizens and government authorities, facilitating efficient communication and resolution of environmental concerns. With its emphasis on transparency, accountability, and community involvement, the proposed system lays the foundation for a more responsive and sustainable approach to environmental stewardship within communities.

### 2.2.1 ADVANTAGES

The proposed system possess several advantages, few are listed below:

- **Efficient Reporting Process:** The mobile application streamlines the process of reporting environmental issues, allowing users to quickly capture and submit complaints with images and descriptions, reducing the time and effort required to report problems.
- **Enhanced User Engagement:** By empowering citizens to actively participate in the reporting and resolution of environmental issues, the system fosters a sense of community ownership and encourages greater involvement in environmental stewardship.
- **Real-time Communication:** The mobile application provides users with real-time updates on the status of their complaints, allowing them to track the progress of their reports and stay informed about the resolution process.
- **Automated Classification:** The use of deep learning algorithms enables automated classification of submitted images, allowing for efficient categorization of environmental issues and streamlined routing of complaints to the appropriate government departments.
- **Resource Allocation:** By automatically directing complaints to the relevant departments, the system optimizes resource allocation, ensuring that environmental issues are addressed promptly and efficiently.
- **Transparency and Accountability:** The system promotes transparency and accountability by providing users with visibility into the status of their complaints and facilitating open communication between citizens and government authorities.
- Centralized Management: Departmental websites serve as centralized platforms for officials to review and manage incoming complaints, providing a unified system for tracking and addressing environmental issues.
- **Improved Response Time:** With automated classification and routing of complaints, the system reduces response times for addressing environmental issues, leading to faster resolution and improved public safety.

## 2.3 DEVELOPMENT ENVIRONMENT

### SOFTWARE REQUIREMENT

- Operating System: Windows 10
- IDE: Visual Studio Code
- VS Code Plugins: Flutter, Dart, Python, html , css, javascript
- Web framework : Flask
- Map API: Google Maps
- API Backend: Firebase

### HARDWARE REQUIREMENT

- Processor: i5 6th gen or higher/ Ryzen 5 1600 or higher
- Graphics: Integrated Graphics or any External Graphics
- RAM: 8 GB or higher
- Proper Internet Connect

# CHAPTER 3
# SYSTEM DESIGN

## 3.1 UML DIAGRAMS

### Use Case Diagram

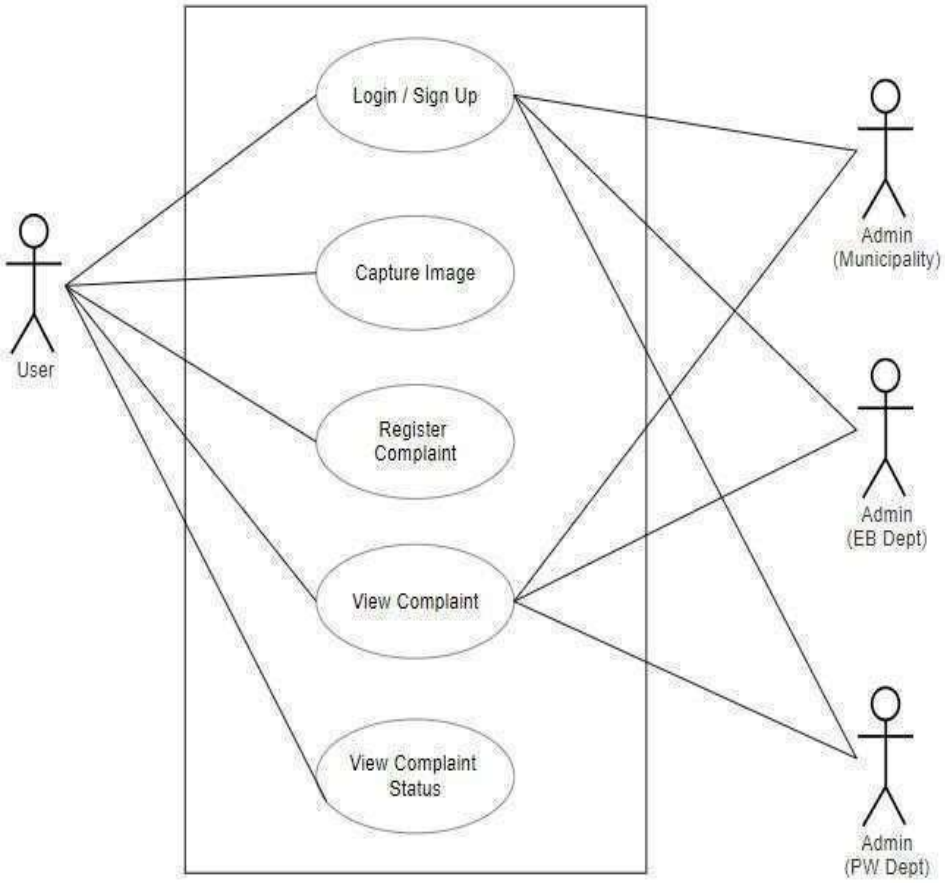



**Fig 3.1.1 Use case diagram for Tour planning application**

The Usecase diagram in the figure shows the main use cases of the enivrowatch system and the stakeholders involve in the system. The Stakeholders involved in the sytem are the users, and the admins of the corresponding departments who are responsible. This UML diagram depicts the use cases of the system in the overview in relevants to the user.This use case diagram refers to activities done by user and system and their corresponding usecases.

**Class Diagram:**

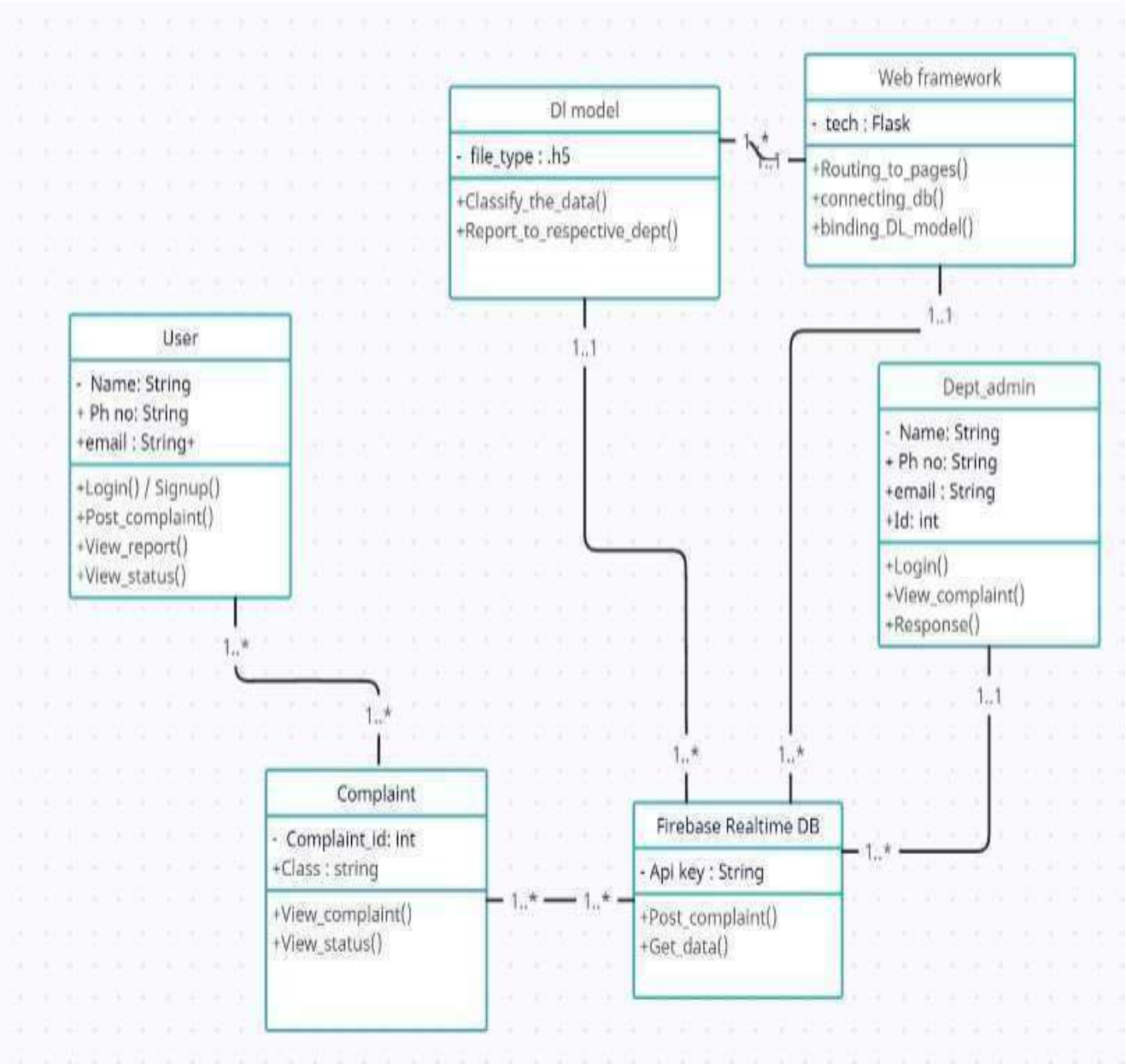



**Fig 3.1.2 Class Diagram for Envirowatch application**

The Class diagram in the figure shows the different class models of the system along with their relationship among each other. Each class may involve a single system or a data model such as complaint and user data model.

**Sequence Diagram:**

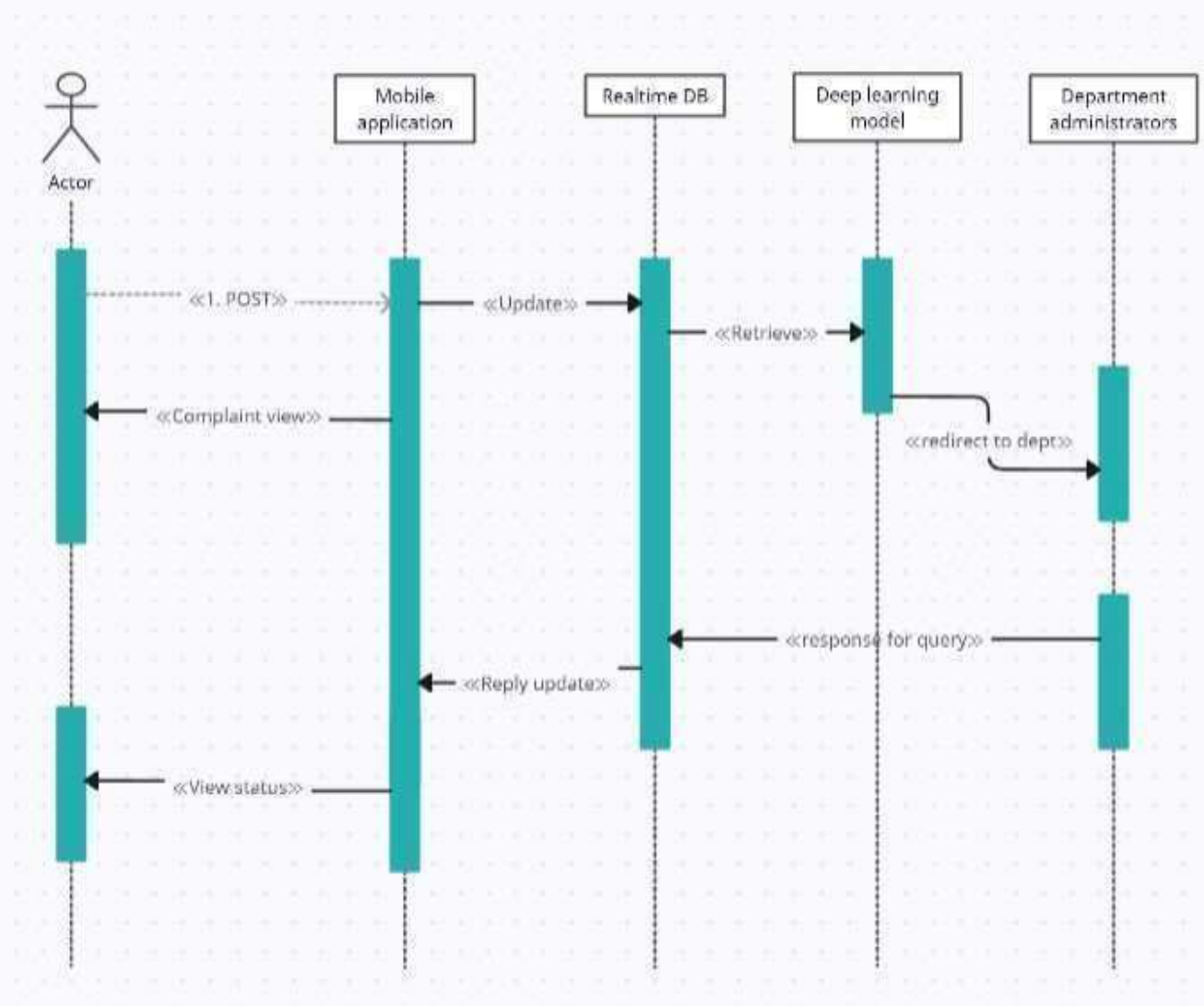



**Fig 3.1.3 Sequence Diagram for Envirowatch application**

The Sequence diagram of the system is shown in the figure. The sequence diagram shows the flow of request and reply of different queries between each module of the envirowatch system and their lifetime. It starts from reporting an issue to addressing the issue to corresponding departments.

**Collaboration diagram:**

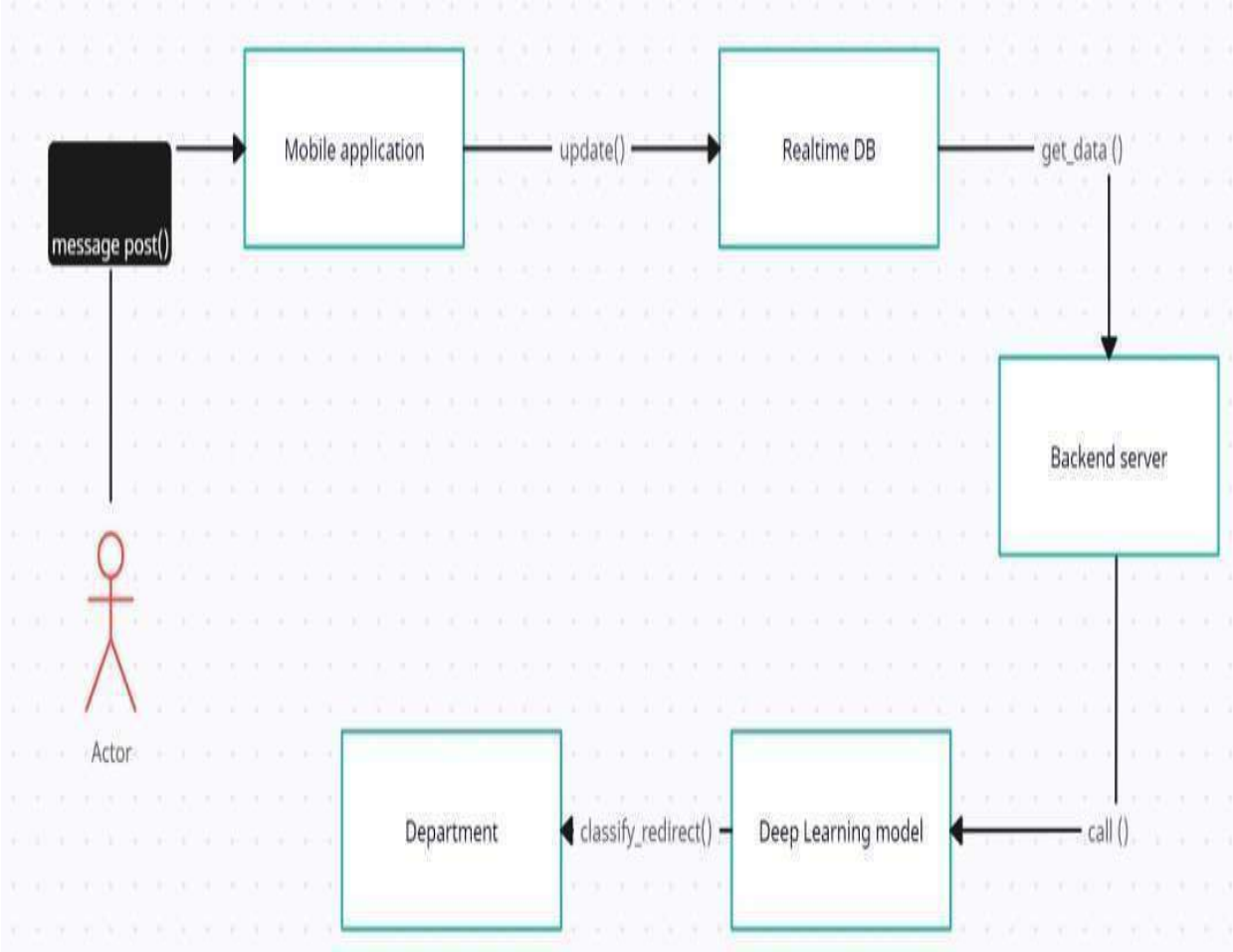



**Fig 3.1.4 Collaboration diagram for Envirowatch application**

The collaboration diagram in the figure shows the collaborative approach and the relationship of the models in the system.The UML diagram depicts the directional dynamic flow of the actions perform in between the model. It shows the activities of the stakeholders in the system.

**Activity diagram:**

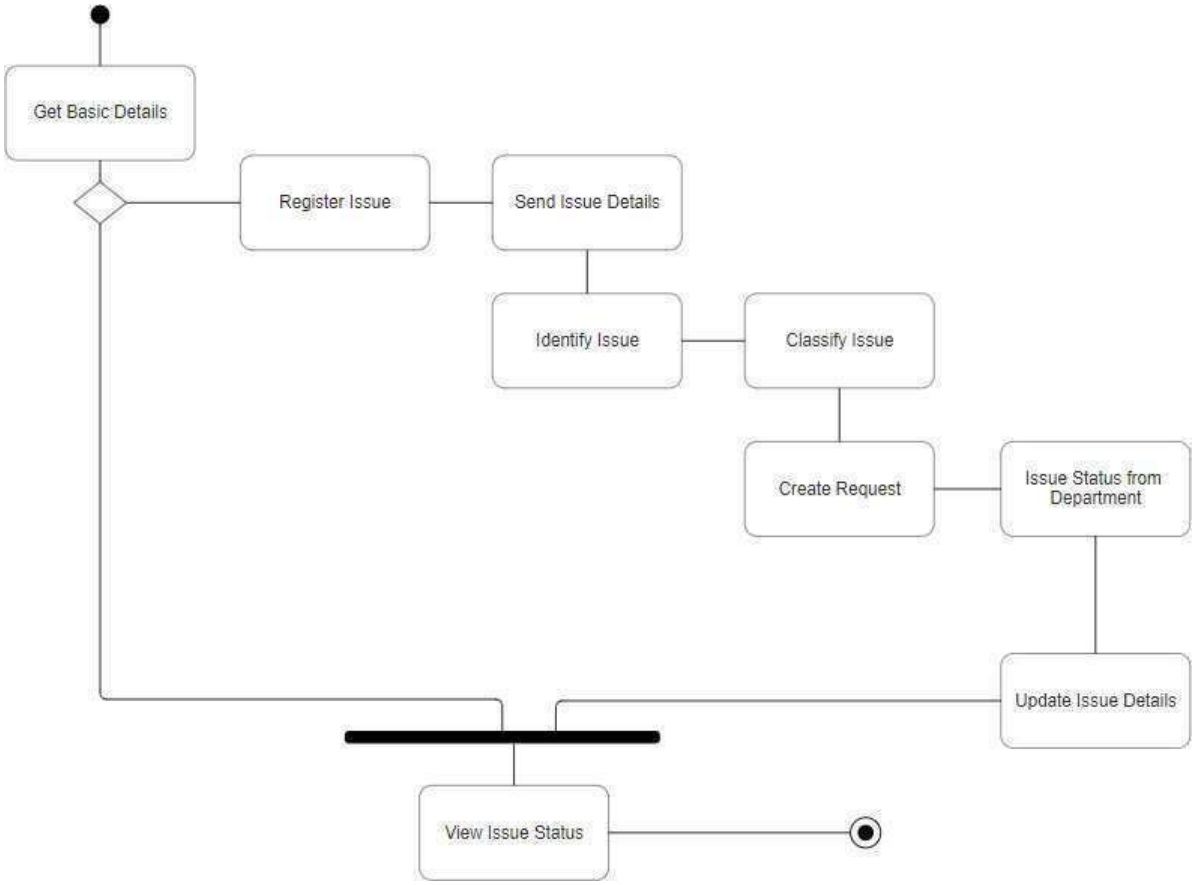



**Fig 3.1.5 Activity Diagram for ENVIROWATCH**

The Activity diagram in the figure shows the activity flow in the system. The Uml diagram shows the directional flow the activity which is done from registering an issue to the system, storing in the database, classification and sending a complaint to the responsible departments.

**Component Diagram:**

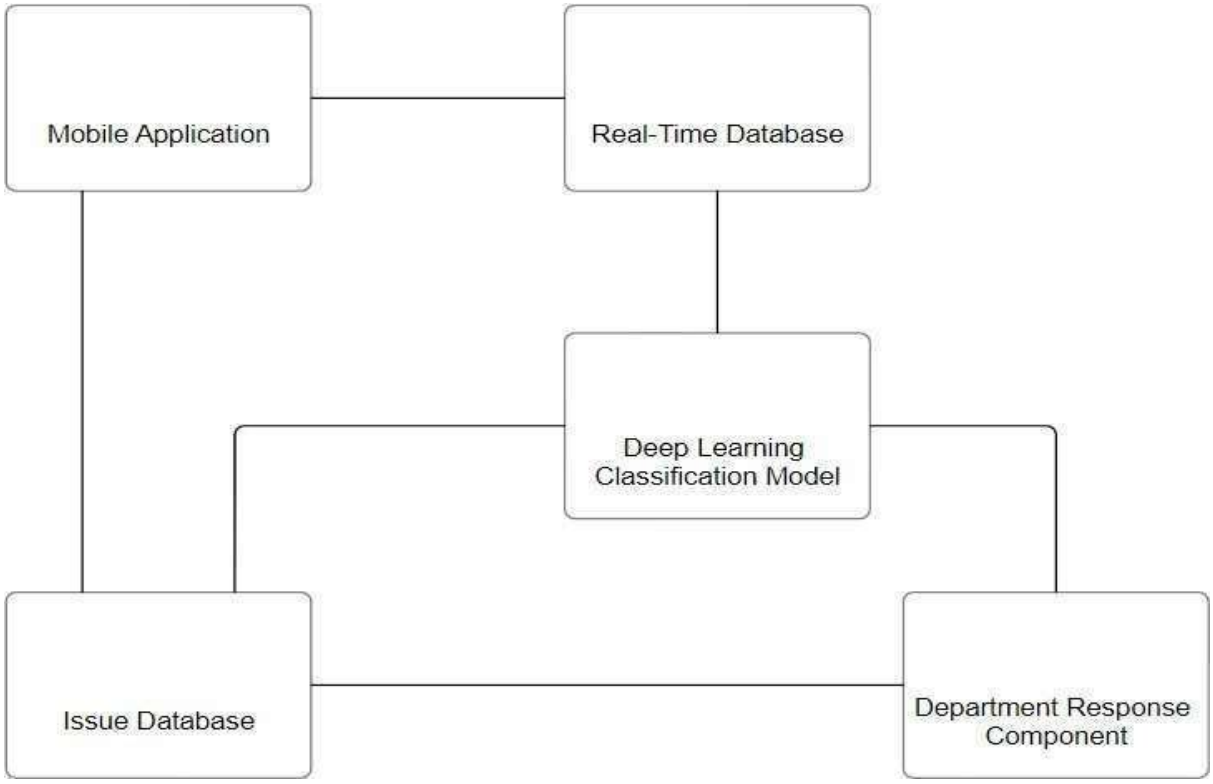



**Fig 3.1.6 Component Diagram for ENVIROWATCH**

The Component diagram in the figure shows different components in the system. This UML diagram depicts the link between the different main components of the envirowatch. The components in the envirowatch system includes the mobile application, Firebase real-time and NoSQL databases, deep learning model components and web service components for the department.

**Deployment Diagram**

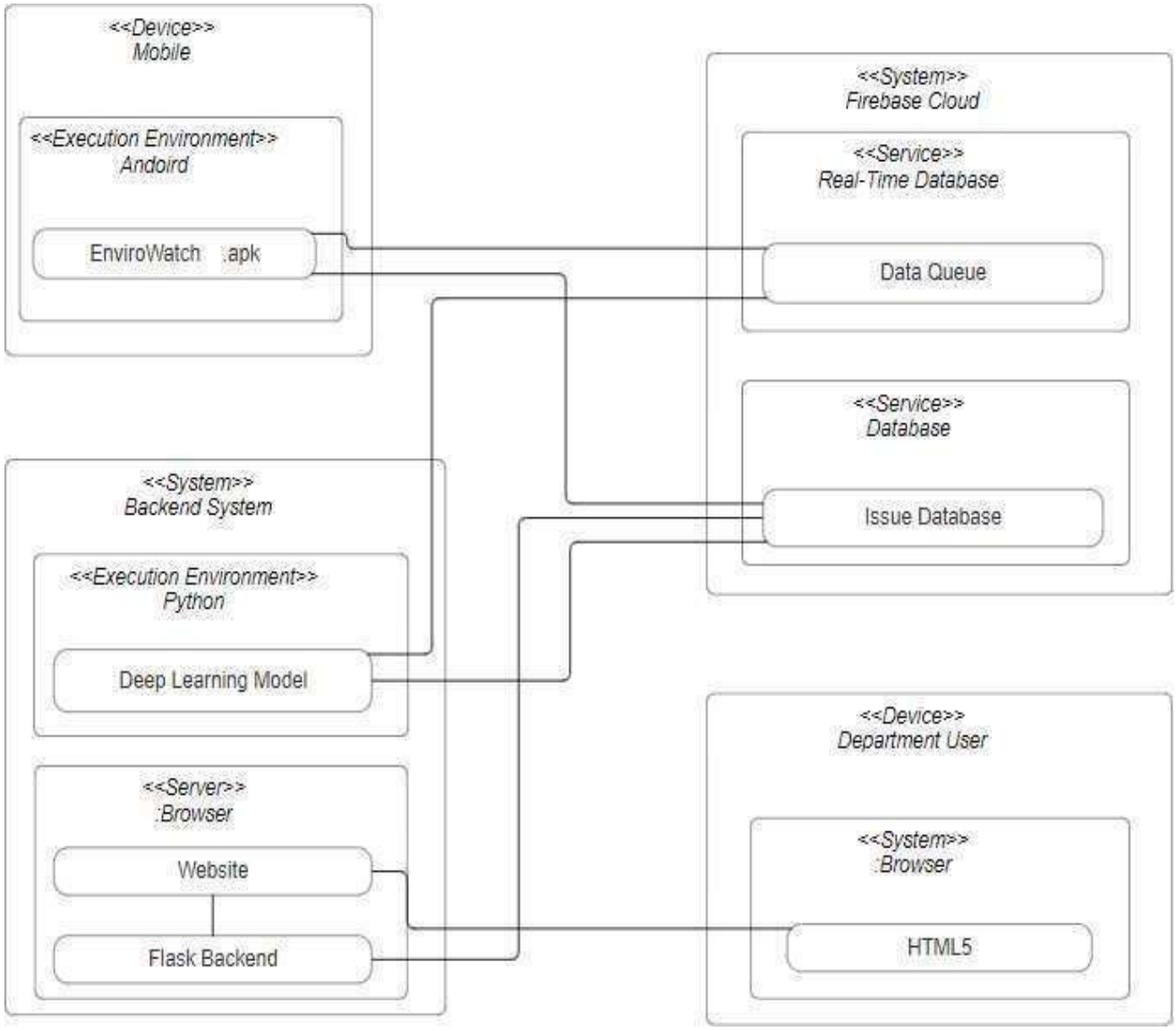



**Fig 3.1.7 Deployment Diagram for ENVIROWATCH**

The deployment diagram shows the environment of the envirowatch is being deployed. The Deployment diagram in the figure shows the deployment model to the system. The UML diagram depicts the harware models in the sytem architecture. The mobile application requires an android architecture to run the application. The database retrieval and storage is done in the firebase SaaS console. The Deep learning model runs in the server along with the backend services for the website UI.

## 3.2 DATA DICTIONARY

This is normally represented as the data about data. It is also termed as metadata some times which gives the data about the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes. Following are some rules, which defines the construction of data dictionary entries:

- Words should be defined to understand what they need and not the variable need by which they may be described in the program.
- Each word must be unique. We cannot have two definitions of the same client.
- Aliases or synonyms are allowed when two or more entries show the same meaning. For example, a vendor number may also be called a customer number. 10
- A self-defining word should not be decomposed. It means that the reduction of any information into subparts should be done only if it is really required, that is it is not easy to understand directly.
- Data dictionary includes information such as the number of records in a file, the frequency a process will run, security factors like pass word which the user must enter to get excess to the information

### 3.2.1 SIGN IN TABLE

| Column name | Data type | Description | Constraint |
|---|---|---|---|
| Email | varchar | Username of the user | Not null |
| Password | varchar | Password of the user | Not null |

**3.2.1 Sign in table for Envirowatch application**

### 3.2.2 SIGN UP TABLE

| Column name | Data type | Description | Constraint |
|---|---|---|---|
| Username | varchar | Full name of the user | Not null |
| Email | varchar | Username name of the user | Not null |
| Password | varchar | Password of the user | Not null |

**3.2.2 Sign up table for Envirowatch application**

### 3.2.3 MESSAGE TABLE:

| Column name | Data type | Description | Constraint |
|---|---|---|---|
| Image Url | varchar | Name of place | Not null, unique primary key |
| Latitude | Double | Latitude coordinate of the place | Not null |
| Longitude | Double | Longitude coordinate of the place | Not null |
| Complaint ID | String | Id of complaint given by user | Default |
| Email ID | String | Email id of user | Default |
| Name | String | Name of user | Default |
| PH NO | String | Mobile no of user | Default |
| Message | String | Message in which user describes the prevailing situation | Default |
| Status | String | Status of the compli-ant issue | Default |
| Date | Date | Date of the issue post | Not null |

**3.2.3 Message table for Envirowatch application**

### 3.2.4 RESPONSE TABLE

| Column name | Data type | Description | Constraint |
|---|---|---|---|
| Reply | String | Replying to the user about the problem clearance. | Not null |
| Image url | String | Photo of the problem clearance. | Not null |

**3.2.4 Response table for Envirowatch application**

### 3.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a visual depiction of how data moves through a system. It shows processes, data stores, data flows, and external entities, providing a clear overview of the system's data architecture. It's a valuable tool for system design, analysis, and communication among stakeholders. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

The data-flow diagram is a tool that is part of structured analysis and data modeling. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data-flow plan.

**0 LEVEL DFD**

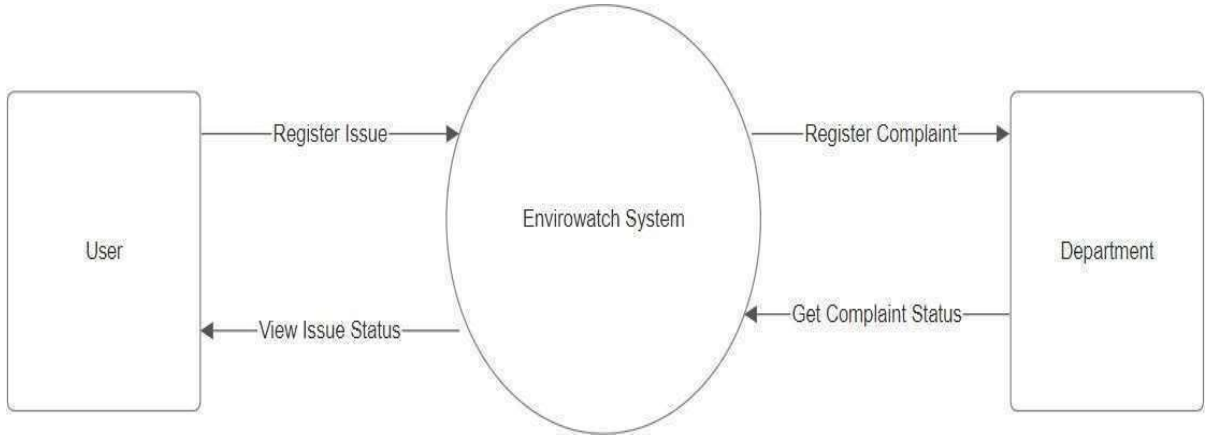



**Fig 3.4.1 Dataflow diagram level 0**

The figure shows the level-0 Data Flow Diagram of the envirowatch system. It is the simpliest form of data flow shown between the system and stakeholders. The User send the issue data to the system which then classified and stored in the system. The final data is viewed by the both the user and corresponding department.

**FIRST LEVEL DFD**

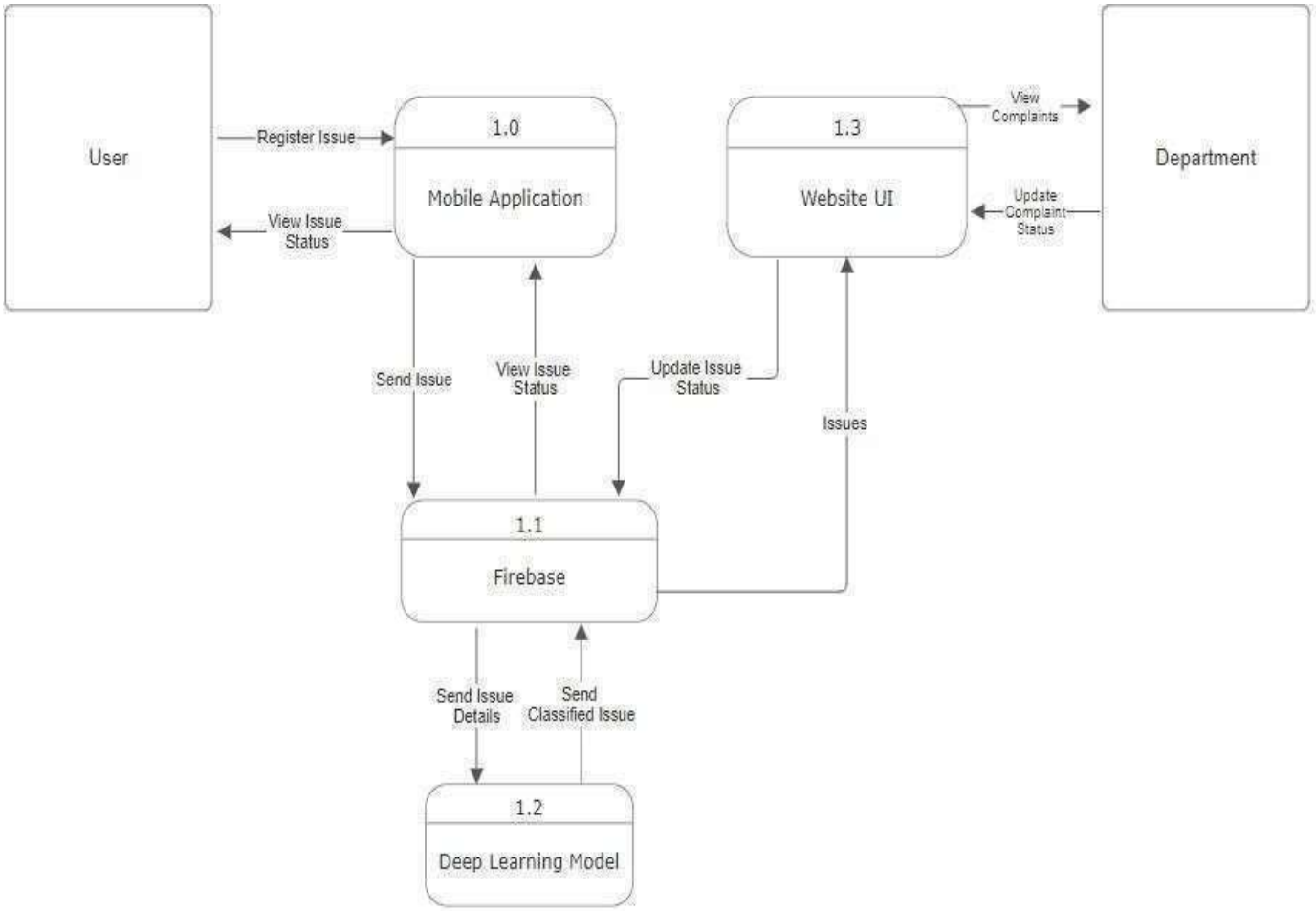
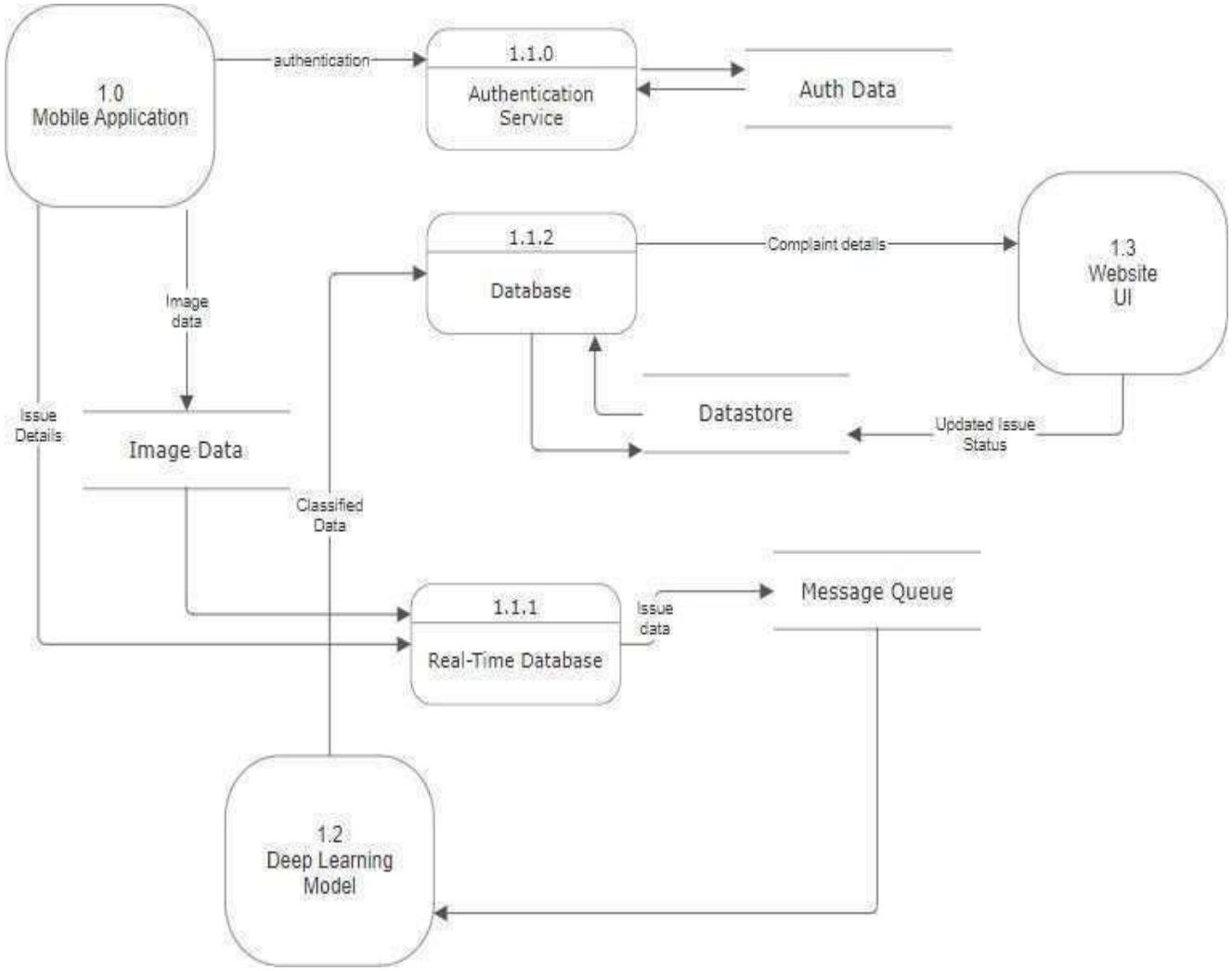



**Fig 3.4.2 Dataflow diagram level 1**

The figure shows the Level-1 Data Flow Diagram of the envirowatch system. This UML diagram shows the flow of data in different modules within the system design. This diagram shows the data transaction between the different models and processes out a different data.

**SECOND LEVEL DFD**




The figures shows the Level-2 Data Flow Diagram of the envirowatch system. This level of the DFD diagrams includes the the internal flow of data in different modules of the system. The directional data flow and processed data can be also seen the diagram.

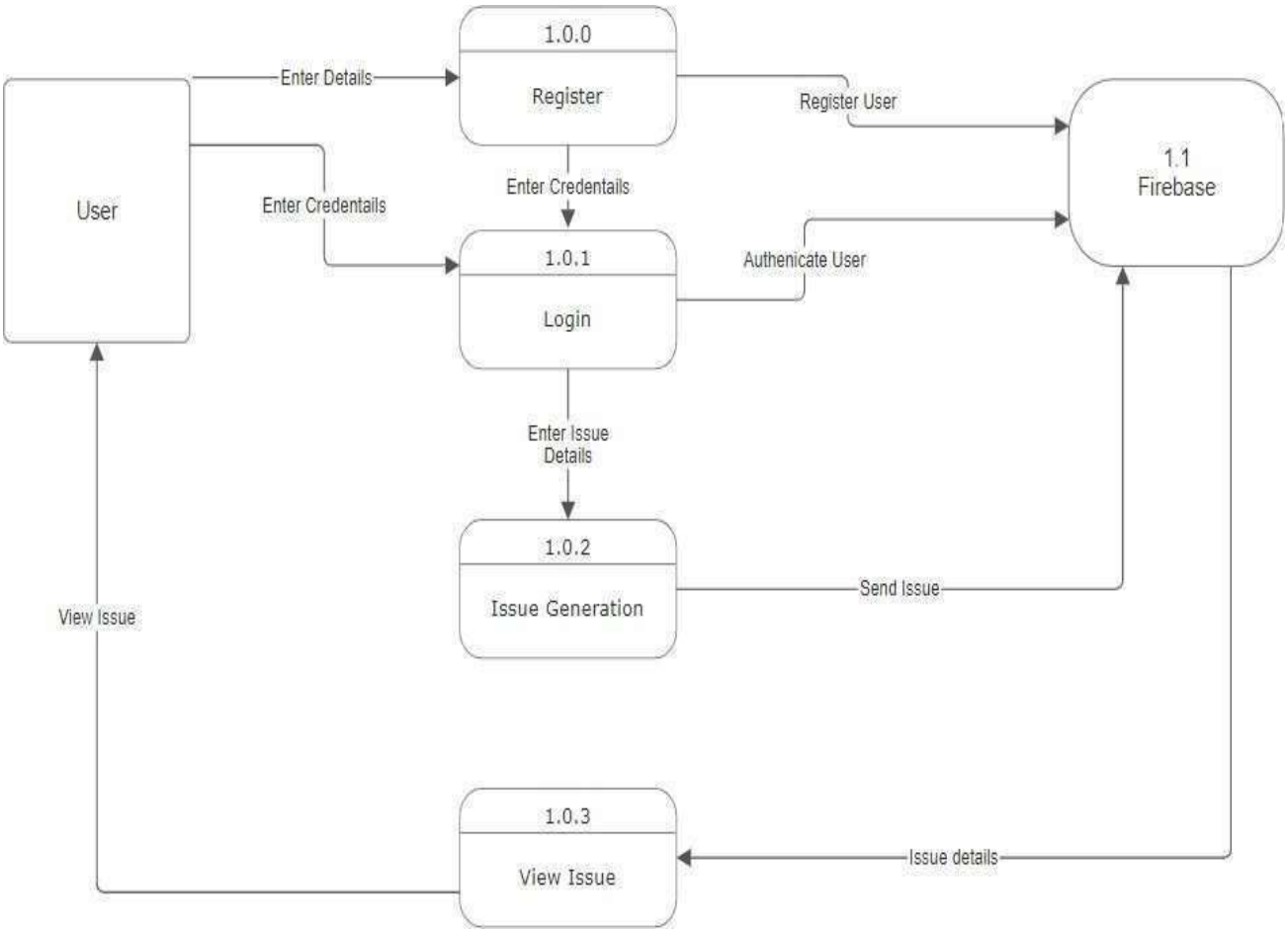



**Fig 3.4.3 Dataflow diagram level 2**

The figures shows the Level-2 Data Flow Diagram of the envirowatch system. This level of the DFD diagrams includes the the internal flow of data in different modules of the system. The directional data flow and processed data can be also seen the diagram.

# CHAPTER 4

# SYSTEM ARCHITECTURE

## 4.1 ARCHITECTURE OVERVIEW

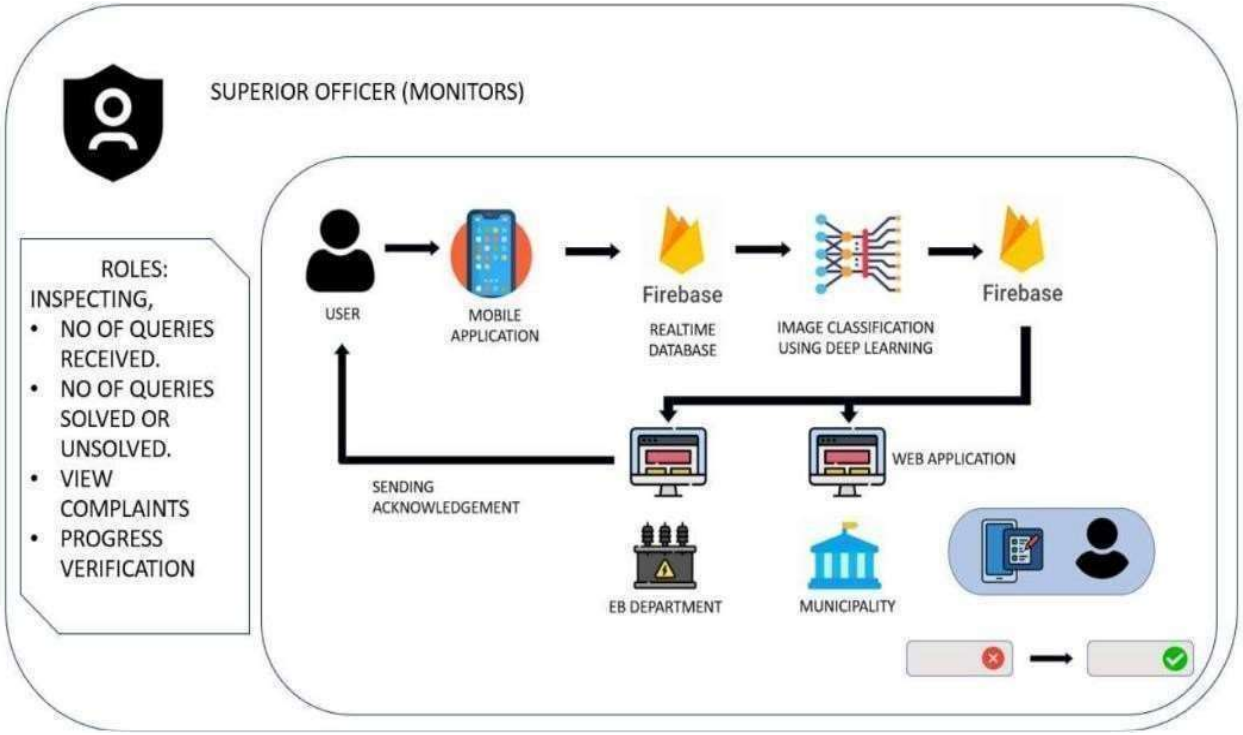



**Fig 4.1 Architecture diagram for Envirowatch application**

The figure 4.1 consists of several interconnected components that work seamlessly to provide users with a personalized and efficient experience. At the core lies the backend server, responsible for processing user requests, managing data, and communicating with external services. The frontend interface, whether web-based or mobile, interacts with the backend to display information and handle user inputs. Here the deep learning algorithm plays a crucial role in classification of problem and responds to corresponding websites.

## 4.2 MODULE DESCRIPTION

The system consists of 4 main modules.

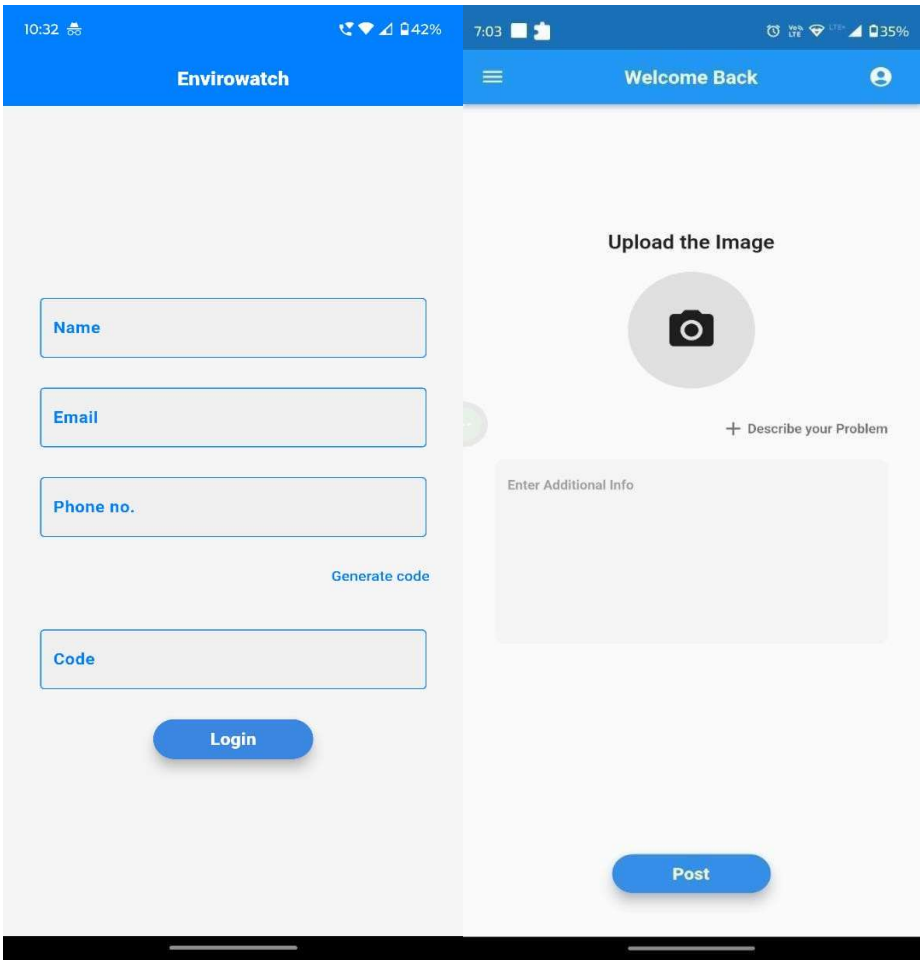
They are

- UI layer
- Firebase Realtime database
- Deep learning model
- Performance

**UI Layer:**

- Here we do have UI for both user and for respective departments. For user UI in mobile application, for department side deals with websites.
- Authentication screens: Sign Up Screen and Sign in Screen.
- Home Screen: Displays recommend trips and its details.
- Image and message acquisition screen: Here we get photo of the problem prevailing for the user, and if needed message can be posted here.
- View complaint screen: Officials in their websites, can view those complaints,with images and message posted.
- View complaint (User) : User can also see his /her complaint posted.

The user interface (UI) in the proposed project plays a crucial role in facilitating a seamless experience for citizens as they report environmental issues and track the progress of their complaints.




**Firebase Realtime Database:**

In the proposed project, Firebase Realtime Database is utilized as a backend service to store and manage the data generated by the mobile application. Here's an explanation of how Firebase Realtime Database is used in the project:

**Data Storage:** Firebase Realtime Database serves as the primary data storage solution for the project. It stores various types of data, including user accounts, complaint reports, complaint statuses, and any other relevant information generated by the application.

**Real-time Updates:** One of the key features of Firebase Realtime Database is its ability to provide real-time updates to connected clients. This means that any changes made to the database are immediately propagated to all connected devices, ensuring that users receive timely updates on the status of their complaints and other relevant information.

**Structured Data:** Firebase Realtime Database uses a JSON data structure, which allows for flexible and efficient data organization. Data is stored in a hierarchical manner, making it easy to retrieve and manipulate data as needed within the application.

**Integration with Flutter:** Firebase Realtime Database integrates seamlessly with Flutter, the framework used for developing the mobile application. This integration allows developers to leverage Firebase's features and capabilities directly within the Flutter environment, simplifying the development process and enhancing the overall user experience.

**Authentication and Security:** Firebase Realtime Database offers built-in authentication and security features, allowing developers to control access to the database based on user roles and permissions. This ensures that sensitive data is protected and only accessible to authorized users.

**Scalability and Performance:** Firebase Realtime Database is designed to be highly scalable and performant, capable of handling a large volume of concurrent connections and data transactions. This scalability ensures that the application can accommodate growing user bases and handle spikes in traffic without sacrificing performance.

**Offline Support:** Firebase Realtime Database provides offline support, allowing users to access and interact with the application even when they are offline. Any changes made to the database while offline are synchronized with the server once an internet connection is restored, ensuring data consistency across devices.

Overall, Firebase Realtime Database serves as a reliable and efficient backend solution for the project, providing developers with the tools and capabilities needed to build a robust and scalable mobile application for reporting and managing environmental issues within communities.

**Deep learning model:**

In the proposed project, a custom deep learning model is developed and trained to classify images of environmental issues captured by users through the mobile application. Here's an explanation of the deep learning model used in the project:

**Architecture Design:** The custom deep learning model is designed specifically to address the task of classifying environmental issues depicted in images. The architecture of the model is carefully crafted to effectively capture relevant features from the input images and make accurate predictions about the type of environmental problem depicted in each image.

**Convolutional Layers:** Convolutional layers play a crucial role in the model's ability to extract spatial features from input images. These layers use convolutional filters to perform operations such as edge detection, texture recognition, and feature extraction, allowing the model to identify patterns and structures within the images.

**Pooling Layers:** Pooling layers are used to reduce the spatial dimensions of the feature maps produced by the convolutional layers while retaining important information. Common pooling operations include max pooling and average pooling, which help to downsample the feature maps and improve computational efficiency.

**Fully Connected Layers:** Fully connected layers are typically used towards the end of the model architecture to perform classification based on the extracted features. These layers connect every neuron in one layer to every neuron in the next layer, allowing the model to learn complex relationships between the features and make predictions about the class labels.

**Activation Functions:** Activation functions are applied to the outputs of neurons in the model to introduce non-linearity and enable the model to learn complex patterns and relationships within the data. Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh functions

**Performance:**

In the proposed project, the performance module is a crucial component that evaluates the effectiveness and efficiency of the entire system in addressing environmental issues. Here's an explanation of the performance module and its key components:

The performance module is designed for continuous monitoring and improvement of the system over time. By regularly evaluating performance metrics and soliciting user feedback, the system can identify areas for optimization and enhancement, ensuring that it remains effective and responsive in addressing environmental issues within communities. Overall, the performance module serves as a critical tool for assessing and optimizing the effectiveness, efficiency, and user satisfaction of the entire system for managing environmental issues. By incorporating a comprehensive range of performance metrics and feedback mechanisms, the module enables ongoing refinement and improvement of the system to better serve the needs of users and communities.

# CHAPTER 5
# SYSTEM IMPLMENTATION

**Main.dart:**

```
import'package:cba/authenicate/authencation_ins.
dart';
import 'package:cba/screens/starting_page.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:get/get.dart'; import 'firebase_options.dart'; void main() async {
),
const SizedBox(
height: 20, ),
const Padding(
padding: EdgeInsets.all(10),
child: Text(
"Tribute your Contribution to make a World
Better", textAlign: TextAlign.center,
style: TextStyle(
fontWeight: FontWeight.w700,
fontSize: 20,
), ), ),
const Icon(
Icons.camera,
size: 40,
color: Colors.grey,
) ], ), ), ),
bottomNavigationBar: SizedBox(
```

```
height:        100,
child: Padding(
padding: const EdgeInsets.only(bottom: 30.0),
child: Center(
child: SizedBox( height: 40, width: 150,
child:  ElevatedButton(
onPressed:  ()  {  Navi-
gator.push(      context,
MaterialPageRoute(
builder: (context) => const RegisterPage()));
},
style: ElevatedButton.styleFrom(
elevation:  10,
shape:  RoundedRectangleBorder(  borderRadi-
us:  BorderRadius.circular(20)),  backgroundCol-
or: const Color(0xff358FEA)), child: const Text(
"Get Started",
style: TextStyle(
fontWeight: FontWeight.w800,
fontSize:  17,
color: Colors.white),
)), ), )),  ), ); } }
```

**Success_page.dart**

```
import 'package:cba/screens/dashboard.dart';
import 'package:flutter/material.dart';
class  SuccessPage  extends  StatelessWidget  {
 const SuccessPage({super.key});
```

```
@override
Widget build(BuildContext context)
{ return Scaffold(
backgroundColor: const Color.fromARGB(255, 245, 245, 245),
body:    Center(
child: Padding(
padding:  const  EdgeInsets.only(left:  35,  right:  35),
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children:          [
const SizedBox(
height: 80,
),
ClipRRect(
borderRadius:
BorderRadius.circular(15), child:
Image.asset(                    "as-
sets/images/success.jpg",      fit:
BoxFit.cover,
height: 320,
width: double.infinity,
), ),
const SizedBox(
height: 20,
),
const Padding(
padding: EdgeInsets.all(10),
child:          Text(
"Thanks for Your
```

```
Contribution.",textAlign:
TextAlign.center, style:
TextStyle(
fontWeight: FontWeight.w700,
fontSize: 24,
), ), ),
const Icon(
Icons.thumb_up_alt,
size: 40,
color: Colors.grey,
) ], ), ), ),
bottomNavigationBar: SizedBox(
height: 100,
child: Padding(
padding: const EdgeInsets.only(bottom: 30, right: 30),
child: Row(
mainAxisAlignment: MainAxisAlignment.end,
children: [
SizedBox(
height: 40,
width: 150,
child: ElevatedButton( on-
Pressed: () { Naviga-
tor.pushAndRemoveUntil(
context, Materi-
alPageRoute(
builder: (context) => const Dash-
board()), (route) => false);
},
```

```
style: ElevatedButton.styleFrom(
elevation: 10,
shape: RoundedRectangleBorder( bor-
derRadius: BorderRadius.circular(20)),
backgroundColor: Colors.white),
child: const
Text( "Exit",
style: TextStyle(
fontWeight: FontWeight.w800,
fontSize: 17,
color: Color(0xff358FEA)),
)), ), ], )), ), ); } }
```

**Sub_test.py**

```
import firebase_admin
from firebase_admin import
db,credentials import time
import requests
import os
import matplotlib
from matplotlib import pyplot as plt
import tensorflow as tf
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import datetime
from datetime import datetime
import warnings warn-
ings.filterwarnings("ignore"
```

```
)
from flask import Flask, render_template, request,make_response,jsonify,redirect,url_for model = load_model("models/best_weights_2step_28loss.h5")
app = Flask(_name_)
cred = credentials.Certificate("static/credential.json") firebase_admin.initialize_app(cred, {"databaseURL": "https://access-car-boot-default-rtdb.firebaseio.com/"})
dd=[]
dd_m=[]
"""def
   get_data_from_firebase(): ref
   = db.reference('/') data
   = ref.get()["data"]
   return data"""
def download_img_from_frbase(img_url, local_file):
   response = requests.get(img_url)
    if response.status_code == 200:
      with open(local_file, 'wb') as
      file:
         file.write(response.content)
      print(f"Image downloaded successfully and saved as {local_file}")
      img = cv2.imread("static/testing.jpg")
      resize = tf.image.resize(img, (312,
      312)) #
      plt.imshow(resize.numpy().astype(int))
      # plt.show()
      yhat = model.predict(np.expand_dims(resize / 255,
      0)) lst = []
```

```
lst.append(yhat[0][0])

lst.append(yhat[0][1])
lst.append(yhat[0][2])

print(lst.index(max(lst)))

clss1 = {0: "damaged_roads", 1: "flood", 2: "others"}
print(clss1[lst.index(max(lst))])
first                                =
clss1[lst.index(max(lst))] #  0
damaged_roads
# 1 flood
# 2 others
if (lst.index(max(lst)) == 2):
    model_2 = load_model("models/best_weights_new_0.44loss.h5")

    yhat = model_2.predict(np.expand_dims(resize / 255, 0))
    lstt = []
    lstt.append(yhat[0][0])

    lstt.append(yhat[0][1])
    lstt.append(yhat[0][2])

    print(lstt.index(max(lstt)))

    clss2 = {0: "garbage", 1: "others", 2: "eb"}
    print(clss2[lstt.index(max(lstt))])
    second                                =
    clss2[lstt.index(max(lstt))] # 0

    garbage
    # 1 others

    # 2 eb
if (first != 'others'):

    out = first
```

```
        else:
            out = second
        return out
    else:
        print(f"Failed to download image. Status code: {response.status_code}")
@app.route("/")
def home():
    return     render_template("index.html")
@app.route('/eb')
def first_page():
    return render_template('sub_eb.html')
@app.route('/second_page')
def second_page():
    return render_template('sub_test.html')
@app.route("/eb_complaints", methods=['GET','POST'])
def complaints():
    ref    =    db.reference('/')
    data = ref.get()["data"]
    #data=get_data_from_firebase()
    img_url=data["imageUrl"]
    img_dir = "static/testing.jpg"
    res=download_img_from_frbase(img_url,img_dir)
    headings=['QUERIES','DATE','LATITUDE','LONGITUDE']
    if(res=='eb'):
d=[data['imageUrl'],data['queries'],str(datetime.now().date()),data['latitude'],data['longitude'],data
['name'],data['email'],data['phone number']]
        if not dd or dd[len(dd)-1][0]!=d[0]:
            dd.append(d)
```

```
    print(dd)
  return dd
@app.route("/view_complaint")
def view_complaint():
  data1 =
  request.args.get('data1') data2
  = request.args.get('data2')
  data3 =
  request.args.get('data3') data4
  = request.args.get('data4')
  data5 =
  request.args.get('data5') data6
  = request.args.get('data6')
  data7 =
  request.args.get('data7') data8
  = request.args.get('data8')
  # Redirect the user to the detailed view page
  return redirect(url_for('detailed_view', data1=data1,
data2=data2,data3=data3,data4=data4,data5=data5,data6=data6,data7=data7,data8=data8))
# Route for the detailed view
page
@app.route('/detailed_view')
def detailed_view():
  data1 =
  request.args.get('data1') data2
  = request.args.get('data2')
```

```
    data3 =
    request.args.get('data3')  data4
    = request.args.get('data4')
    data5 =
    request.args.get('data5')  data6
    =  request.args.get('data6')
    data7 =
    request.args.get('data7')  data8
    = request.args.get('data8')
    # Render the HTML template for the detailed view page and pass data to the tem-
    plate return render_template('view.html', data1=data1,
data2=data2,data3=data3,data4=data4,data5=data5,data6=data6,data7=data7,data8=data8)
@app.route('/transfer_data', methods=['POST'])
def transfer_data():
    # Retrieve data from the form
    data6 = re-
    quest.form.get('data6') data7
    = request.form.get('data7')
    # Redirect to another page with data in URL parameters re-
    turn redirect(url_for('map_page', data6=data6, data7=data7))
@app.route('/map_page'
) def map_page():
    # Retrieve data from URL parameters
    data6 = request.args.get('data6') da-
    ta7 = request.args.get('data7')
    # Render another HTML page and pass the data to it
```

```
    return  render_template('map.html',    data6=data6,    data7=data7)
@app.route("/municipality"
) def mp():
    return render_template("mp.html")
    @app.route("/mp_complaintss") def
    mp_complaint():
    return render_template("mp_complaints.html")
@app.route("/mp_comp", methods=['GET','POST'])
def      m_complaints():
    ref = db.reference('/')
    data  =  ref.get()["data"]  #da-
    ta=get_data_from_firebase()
    img_url=data["imageUrl"]
    img_dir = "static/testing.jpg"
    res=download_img_from_frbase(img_url,img_dir)
    headings=['QUERIES','DATE','LATITUDE','LONGITUDE']
    if(res=='garbage' or res=='flood'):
d_m=[data['imageUrl'],data['queries'],str(datetime.now().date()),data['latitude'],data['longitude'],d
ata['name'],data['email'],data['phone   number']]
        if      not      dd_m      or      dd_m[len(dd)-1][0]!=d_m[0]:
            dd_m.append(d_m)
        print(dd_m)
    return                              dd_m
@app.route("/mp_view_complaint")
def mp_view_complaint():
    data1                                =
    request.args.get('data1')     data2
    =          request.args.get('data2')
```

```
data3 =
request.args.get('data3') data4
= request.args.get('data4')
data5 = request.args.get('data5')
data6 =
request.args.get('data6') data7
= request.args.get('data7')
data8 = request.args.get('data8')
# Redirect the user to the detailed view page
return redirect(url_for('mp_detailed_view', data1=data1,
dta2=data2,data3=data3,data4=data4,data5=data5,data6=data6,data7=data7,data8=data8))
# Route for the detailed view page
@app.route('/mp_detailed_vie
w') def mp_detailed_view():
data1 =
request.args.get('data1') data2
= request.args.get('data2')
data3 =
request.args.get('data3') data4
= request.args.get('data4')
data5 =
request.args.get('data5') data6
= request.args.get('data6')
data7 =
request.args.get('data7') data8
= request.args.get('data8')
# Render the HTML template for the detailed view page and pass data to the tem-
plate return render_template('mp_view.html', data1=data1,
```

```
data2=data2,data3=data3,data4=data4,data5=data5,data6=data6,data7=data7,data8=data8)
@app.route('/mp_transfer_data', methods=['POST'])
def mp_transfer_data():
# Retrieve data from the form data6 = request.form.get('data6
data7= request.form.get('data7')
# Redirect to another page with data in URL parameters
return redirect(url_for('mp_map_page', data6=data6, data7=data7))
@app.route('/mp_map_page')
def mp_map_page():
# Retrieve data from URL parameters
data6 = request.args.get('data6') data7 =
request.args.get('data7')
# Render another HTML page and pass the data to it
return render_template('mp_map.html', data6=data6, data7=data7)
if_name_____=='_main_':
app.run(debug=True)
```

**Dashboard.dart**

```
import 'dart:convert';
import 'dart:io';
import 'package:cba/screens/complaint_dashboard.dart';
import 'package:cba/screens/complaint_status.dart';
import 'package:cba/screens/register_page.dart';
import 'package:cba/screens/success_page.dart';
import 'package:cba/utils/edit_data.dart';
import 'package:cba/utils/shared_preference_functions.dart';
import 'package:cba/widgets/snackbar_widget.dart';
import 'package:firebase_database/firebase_database.dart';
```

```
import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:geolocator/geolocator.dart';
import 'package:image_picker/image_picker.dart';
import 'package:permission_handler/permission_handler.dart';
import 'package:shared_preferences/shared_preferences.dart';
class Dashboard extends StatefulWidget {
  const Dashboard({super.key});
  @override
  State<Dashboard> createState() => _DashboardState();
}
class _DashboardState extends State<Dashboard> {
  final dataref = FirebaseDatabase.instance.reference();
  String imageUrl = '';
  String lat = "";
  String long = "";
  String? _phone;
  String? _name;
  String? _email;
  String complaintId = "";
  bool tap = false;
  bool uploaded = false;
  bool uploading = false;
  TextEditingController additionalInfoController = TextEditingController();
  @override
  void initState() {
    super.initState();
    fetchData();
  }
  Future<void> fetchData() async {
    String? p = await loadData('phoneNo');
    String? n = await loadData('name');
    String? e = await loadData('email');
```

```
  _phone = p;
  _name = n;
  _email = e;
}
@override
Widget build(BuildContext context) {
  Future.delayed(const Duration(seconds: 2)).then((value) {
    setState(() {});
  });
  return Scaffold(
    appBar: AppBar(
      title: const Text(
        "Envirowatch Complaint",
        style: TextStyle(
          fontWeight: FontWeight.w900,
          fontSize: 19,
          color: Color.fromARGB(255, 255, 255, 255),
        ), ),
      backgroundColor: const Color.fromARGB(255, 0, 128, 254),
      centerTitle: true,
      actions: [
        GestureDetector(
          onTap: () {
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => const ComplaintStatus()));
          },
          child: const Padding(
            padding: EdgeInsets.only(right: 25),
            child: Icon(
              Icons.info_outline_rounded,
              color: Colors.white,
            ),
```

```
    ), ), ], ),
drawer: Drawer(
  backgroundColor: const Color.fromARGB(255, 28, 141, 255),
  child: SafeArea(
    child: Column(
      children: [
        Center(
          child: Container(
            height: 80,
            width: 80,
            decoration: BoxDecoration(
                color: Colors.grey.shade200,
                borderRadius: BorderRadius.circular(100)),
            child: const Center(
              child: Icon(
                Icons.person,
                size: 45,
              ), ), ), ),
        Padding(
          padding: const EdgeInsets.only(top: 8, bottom: 15),
          child: Center(
            child: Text(
              _phone ?? " ",
              style: const TextStyle(
                  fontWeight: FontWeight.w700, fontSize: 20),
            ), ), ),
        ListTile(
          // tileColor: Colors.white,
          tileColor: Colors.white,
          leading: const Icon(
            Icons.email_outlined,
            color: Colors.red,
          ),
          onTap: () {
```

```
          Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) => ComplaintDashboard(
                    phoneNo: _phone ?? " ",
                )), ); },
          title: const Text(
            "Inbox",
            style: TextStyle(
              fontWeight: FontWeight.w500,
              fontSize: 18,
            ),),),
        ListTile(
          onTap: () {
            clearData();
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => const RegisterPage()));
          },
          leading: const Icon(
            Icons.logout,
            color: Colors.yellow,
          ),
          title: const Text(
            "Logout",
            style: TextStyle(
              fontWeight: FontWeight.w500,
              fontSize: 18,
              color: Colors.black,
            ), ), ) ], ), ), ),
  body: Padding(
    padding: const EdgeInsets.all(30),
    child: SingleChildScrollView(
```

```
child: Column(
  mainAxisAlignment: MainAxisAlignment.start,
  crossAxisAlignment: CrossAxisAlignment.center,
  children: [
    const SizedBox(
      height: 100,
    ),
    const Center(
      child: Text(
        "Upload the Image",
        style: TextStyle(
          fontWeight: FontWeight.w600,
          fontSize: 20,
        ), ), ),
    const SizedBox(
      height: 20,
    ),
    uploading == true && uploaded == false
        ? const Padding(
            padding: EdgeInsets.all(40),
            child: SizedBox(
                height: 40,
                width: 40,
                child: CircularProgressIndicator()),
          )
        : uploading == true && uploaded == true
            ? const Icon(
                Icons.photo_size_select_actual_outlined,
                size: 120,
                color: Colors.grey,
              )
            : Container(
                height: 120,
                width: 120,
```

```
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(100),
  color: Colors.grey.shade300,
),
child: IconButton(
    onPressed: () async {
      var status =
          await Permission.location.request();
      if (status.isGranted) {
        Position position =
            await Geolocator.getCurrentPosition(
          desiredAccuracy: LocationAccuracy.high,
        );
        setState(() {
          lat = position.latitude.toString();
          long = position.longitude.toString();
        });
        ImagePicker imagePicker = ImagePicker();
        XFile? file = await imagePicker.pickImage(
            source: ImageSource.camera);
        // print('${file?.path}');
        setState(() {
          uploading = true;
        });
        if (file == null) {
          return;
        }
        String uniqueFileName = DateTime.now()
            .millisecondsSinceEpoch
            .toString();
        complaintId =
            (uniqueFileName + (_phone ?? " "));
        Reference referenceRoot =
            FirebaseStorage.instance.ref();
```

```
                Reference referenceDirImages =
                    referenceRoot.child('images');
                Reference referenceImageToUpload =
                    referenceDirImages.child(uniqueFileName);
                try {
                  await referenceImageToUpload.putFile(
                      File(file.path),
                      SettableMetadata(
                        contentType: "image/jpeg",
                      ));
                  imageUrl = await referenceImageToUpload
                      .getDownloadURL();
                  setState(() {
                    uploaded = true;
                  });
                } catch (error) {
                  if (kDebugMode) {
                    print(error.toString());
                  } }
                } else {
                  if (kDebugMode) {
                    print("no");
                  }} },
              icon: const Icon(
                Icons.camera_alt,
                size: 50,
              )), ),
uploading == true && uploaded == false
    ? const Padding(
        padding: EdgeInsets.only(top: 10),
        child: Text(
          "Uploading Image.....",
          style: TextStyle(
              fontWeight: FontWeight.w500,
```

```
            color: Colors.orange,
            fontSize: 13),
        ),
      )
    : uploaded == true && uploading == true
        ? const Padding(
            padding: EdgeInsets.only(top: 10),
            child: Text(
              "Uploded Successfully!!",
              style: TextStyle(
                  fontWeight: FontWeight.w500,
                  color: Colors.green,
                  fontSize: 13),
            ),
          )
        : Container(),
const SizedBox(
  height: 30,
),
GestureDetector(
  onTap: () {
    toggle();
  },
  child: const Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      Icon(
        Icons.add,
        size: 23,
        color: Colors.black54,
      ),
      SizedBox(
        width: 2,
      ),
```

```
      Text(
        "Describe your Problem",
        style: TextStyle(
            fontWeight: FontWeight.w600,
            fontSize: 16,
            color: Colors.black54),
      ), ] ,), ),
const SizedBox(
  height: 20,
),
tap == true
    ? TextField(
        keyboardType: TextInputType.multiline,
        controller: additionalInfoController,
        maxLines: 10,
        style: const TextStyle(
            color: Colors.black,
            fontWeight: FontWeight.w600,
            fontSize: 13),
        decoration: InputDecoration(
          //  fillColor: "#FAFAFA".toColor(),
          fillColor: Colors.grey.shade300,
          filled: true,
          hintText: "Enter Additional Info",
          hintStyle: const TextStyle(
            fontSize: 13,
            color: Colors.grey,
            fontWeight: FontWeight.w600,
          ),
          border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
              borderSide: BorderSide.none),
        ),    )
    : Container(),
```

```
        ],    ),  ),  ),
    bottomNavigationBar: SizedBox(
      height: 100,
      child: Padding(
          padding: const EdgeInsets.only(bottom: 30.0),
          child: Center(
            child: SizedBox(
              height: 40,
              width: 150,
              child: ElevatedButton(
                  onPressed: () async {
                    insert(imageUrl);
                    showSnackBar(context, "Issue sent!!!!");
                    Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (context) => const SuccessPage()));
                  },
                  style: ElevatedButton.styleFrom(
                      elevation: 10,
                      shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(20)),
                      backgroundColor: const Color(0xff358FEA)),
                  child: const Text(
                    "Post",
                    style: TextStyle(
                        fontWeight: FontWeight.w800,
                        fontSize: 17,
                        color: Colors.white),
                  )),),)), ),);}
  void insert(String imageUrl) {
    String date = DateTime.now().toString();
    dataref.child("data").set({
      "imageUrl": imageUrl,
```

```
            "latitude": lat.toString(),
            "longitude": long.toString(),
            "queries": additionalInfoController.text,
            "name": _name,
            "phone number": _phone,
            "email": _email,
            "complaintID": complaintId,
            "date": date,
          });
          var data = {
            "imageUrl": imageUrl,
            "latitude": lat.toString(),
            "longitude": long.toString(),
            "queries": additionalInfoController.text,
            "name": _name ?? "",
    "phone number": _phone ?? "",
    "email": _email ?? "",
    "complaintID": complaintId,
    "status": "issue sent",
    "date": date,
    "department": "",
  };
  dataref.child("issue").child(_phone!).child(complaintId).set(data);
  saveIssueData(imageUrl);
  addIssue(data);
}
toggle() {
  setState(() {
    tap = !tap;
  });   }
 void saveIssueData(String imageUrl) async {
  Map<String, String?> data = {
    "imageUrl": imageUrl,
    "latitude": lat.toString(),
```

```
    "longitude": long.toString(),
    "queries": additionalInfoController.text,
    "name": _name,
    "phone number": _phone,
    "email": _email,
    "complaintID": complaintId,
   };
   SharedPreferences pref = await SharedPreferences.getInstance();
   pref.setString(
    complaintId,
    jsonEncode(data),
   );}}
```

**EV.HTML & CSS**

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>
  <link rel="icon" href="img/ewg.png">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <script src="https://kit.fontawesome.com/7248d46176.js" crossorigin="anonymous"></script>
  <script src="https://kit.fontawesome.com/41a4fc5a13.js" crossorigin="anonymous"></script>
</head>
<title>Enivrowatch AI</title>
<style>
body {
```

```
  font-family: Arial, Helvetica, sans-serif;
  margin: 0;
  color: #000000;
  text-align: center;
  width: auto;
 }
.header{
   color: white;
   /* background-image: url('img/im.jpg');  */
   background-repeat: no-repeat;
   background-attachment: fixed;
   background-size: 100% 100%;
   width: 100%;
}
/* Increase the font size of the heading */
.header h1 {
  font-size: 36px;
}
@media screen and (max-width: 100%) {
  .topnav a:not(:first-child), .dropdown .dropbtn {
    display: none;
  }
  .topnav a.icon {
    float: left;
  display: block;
  }
}
@media screen and (max-width: 100%) {
  .header {
    width: 100%;
    display: flex;
  }
}
@media screen and (max-width: 100%) {
```

```
  .topnav.responsive {position: relative;}
  .topnav.responsive .icon {
    position: absolute;
    right: 0;
    top: 0;
  }
  .topnav.responsive a {
    float: none;
    display: block;
    text-align: left;
  }
  .topnav.responsive .dropdown {float: none;}
  .topnav.responsive .dropdown-content {position: relative;}
  .topnav.responsive .dropdown .dropbtn {
    display: block;
    width: 100%;
    text-align: left;
  }
}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}
/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}
```

```
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
.active {
  background-color: #717171;
}
/* Fading animation */
.fade {
  animation-name: fade;
  animation-duration: 1.5s;
}
@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}
/* On smaller screens, decrease text size */
 @media only screen and (max-width: 300px) {
  .text {font-size: 11px}
 }
.navbar-nav{
    text-align: center;
    margin-left: 42%;
}
.footer {
   position: fixed;
   left: 0;
   bottom: 0;
```

```
  width: 100%;
  background-color: #3284f7;
  color: white;
  text-align: center;
  height: 50px;
   /* background-image: url('img.jpg'); */
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
}
</style>
</head>
<body>
 <div class="header" style="width: 100%; margin-bottom:0; color: white; background-color: #7aadf5; ">
   <table align="center">
   <tr>
     <td width="10%"><img src="img/ev_logo2.png" height="142" width="142"></td>
     <td width="80%">
       <center>
         <h1><b>Enivrowatch AI</b></h1>
         <h2><b>Tamil Nadu</b></h2>
       </center>
     </td>
     <td width="10%"><img src="img/govt.png" height="145" width="142"></td>
   </tr>
  </table>
</div>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <!-- navbar -->
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
```

```
    <li class="nav-item">
      <a class="nav-link" href="https://envirowatch-ai.github.io/Enivrowatch/ev/ev"><b>Home</b></a>
    </li>
     <li class="nav-item">
      <a class="nav-link" href="#"><b>About us</b></a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#"><b>Compliant</b></a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="https://envirowatch-ai.github.io/Enivrowatch/Evlogin/index.html"><b>Login</b></a>
    </li>
   </ul>
  </div>
</nav><br><br>
<!-- img slider -->
<div class="demo6">
                    <ul>
                            <li><p style = "color:blue;  text-align:center; margin-top:-3px; font-size:20px;">Energy / Electricity Saved is Energy Produced </p></li>
                            <li><p style = "color:blue;  text-align:center; margin-top:-3px; font-size:20px;" >Energy conservation, a key to sustainable development </p></li>
                            <li><p style = "color:blue;  text-align:center; margin-top:-3px; font-size:20px;" >Energy is in limited supply, Use it wisely</p></li>
                            <li><p style = "color:blue;  text-align:center; margin-top:-3px; font-size:20px;" >Turn off lights and equipments when not in use</p></li>
                            <li><p style = "color:blue;  text-align:center; font-size:20px; margin-top:-3px;">Save Energy for benefit of self and Nation</p></li>
                            <li><p style = "color:blue;  text-align:center; margin-top:-3px; font-size:20px;" >Minimize load shedding by switching off  lights and appliances when not in use</p></li>
          </ul>
          </div>
<div class="mySlides fade">
<img src="img/BB-TNEB.jpg" style="width:80%">
```

```
</div>
<!-- <div style="text-align:center">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div><br><br>-->
<!-- footer -->
<div class="footer">
 <h5 style="text-align: center; margin-top: 1px; padding: 12px; font-size: 15px;"><b>Copyright © 2024  Developed By Enivrowatch.</b></h5>
</div>
<!-- <script src="ev.js"></script> -->
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js" integrity="sha384-IDwe1+LCz02ROU9k972gdyvl+AESN10+x7tBKgc9I5HFtuNz0wWnPclzo6p9vxnk" crossorigin="anonymous"></script>
</body>
</html>
```

# CHAPTER 6

# SYSTEM TESTING

## 6.1 TEST CASES AND REPORTS

| TEST CASE ID | TESTCASE/ACTION TO BE PERFORMED | EXPECTED RESULT | ACTUAL RESULT | PASS/ FAIL |
|---|---|---|---|---|
| 1. | Opening Envirowatch Application | Login page appears | Login page appears | Pass |
| 2. | App version check and Update | Update successful | Update successful | Pass |
| 3. | Enter required sign up Details | Registered successfully | Registered successfully | Pass |
| 4. | Enter login details | Home page appears | Home page appears | Pass |
| 5. | View Issue list | Display list of Issue raised by the user | Display list of Issue raised by the user | Pass |
| 6. | Post image | Image successfully uploaded | Image successfully uploaded | Pass |
| 7. | Enter message | Message Uploaded | Message uploaded | Pass |
| 8. | DL model classification | Classified to wright class | Classified to wright class | Pass |
| 9. | Update the complaint to respective departments in live. | Updated | updated | Pass |
| 10. | Response given by de-partment side. | Reponse posted | Response posted | Pass |

| | | | | |
|---|---|---|---|---|
| 11. | View complaint | Complaint Viewed | Complaint viewed | Pass |
| 12. | Get complaint status | Status received | Status received | Pass |
| 13. | Get image from the user to the DL Model | Image received | Image received with in 10 seconds | Pass |
| 14. | Logout from the application | Logout and remove Meta Data | Logout and Clear Meta Data | Pass |

## 6.2 RESULTS AND DISCUSSION:

In the results and discussion section, we analyze the performance of our system in addressing environmental issues. Our custom deep learning model demonstrated robust classification capabilities, accurately identifying various problems such as flood water stagnation, damaged roads, electrical pole damage, and garbage accumulation. The real-time communication features provided users with immediate updates on their complaints' status, enhancing transparency and accountability. User feedback highlighted the system's efficiency and ease of use, underscoring its effectiveness in empowering citizens to actively participate in environmental stewardship. Ongoing monitoring and optimization ensure continuous improvement and responsiveness to community needs.

# CHAPTER 7

# CONCLUSION

In conclusion, our project represents a significant step forward in addressing environmental issues within communities through the integration of technology and citizen engagement. By leveraging a custom deep learning model, real-time communication features, and user-friendly interfaces, we have developed a robust system that empowers citizens to report and address environmental concerns effectively. The results demonstrate the system's ability to accurately classify various environmental problems and provide timely updates to users, enhancing transparency and accountability in environmental management. Moving forward, we remain committed to refining and expanding the system to better serve the needs of communities and promote sustainable environmental stewardship.

## 7.1 FUTURE ENHANCEMENTS:

Enhanced Image Classification: Explore advanced deep learning techniques, such as transfer learning or ensemble models, to further improve the accuracy and robustness of image classification for identifying environmental issues.

**Geolocation Integration:** Incorporate geolocation features into the mobile application to automatically tag the location of reported issues. This additional information can help government departments prioritize and allocate resources more effectively.

**Crowdsourced Data Validation:** Implement mechanisms for crowdsourced data validation, allowing users to verify and confirm reported issues. This can enhance the reliability of reported data and reduce the burden on government agencies for manual validation.

**Predictive Analytics:** Develop predictive models to forecast potential environmental issues based on historical data and environmental factors. This proactive approach can enable preemptive interventions and mitigate the impact of recurring issues.

**Community Engagement Initiatives:** Introduce gamification elements or community challenges to incentivize and motivate users to actively participate in reporting and addressing environmental issues. This can foster a sense of community ownership and responsibility towards environmental conservation.

**Integration with IoT Devices:** Integrate with Internet of Things (IoT) devices, such as sensors or cameras, to augment data collection capabilities and provide real-time monitoring of environmental conditions in key areas.

**Multilingual Support:** Expand language support within the mobile application to accommodate users from diverse linguistic backgrounds. Providing multilingual support can enhance accessibility and inclusivity, facilitating broader community engagement.

**Social Media Integration:** Enable seamless integration with social media platforms to allow users to share their environmental concerns and actions with their social networks. This can amplify the reach and impact of the project and encourage broader community involvement.

**Data Visualization Tools:** Develop interactive data visualization tools and dashboards to present aggregated data on environmental issues in an easy-to-understand format. This can help stakeholders identify trends, patterns, and areas requiring attention more effectively.

Partnerships with NGOs and Businesses: Forge partnerships with non-governmental organizations (NGOs), businesses, and other stakeholders to collaborate on environmental initiatives and leverage additional resources and expertise for sustainable environmental management.

# CHAPTER 8

# APPENDICES

## 8.1 SAMPLE SCREENSHOTS

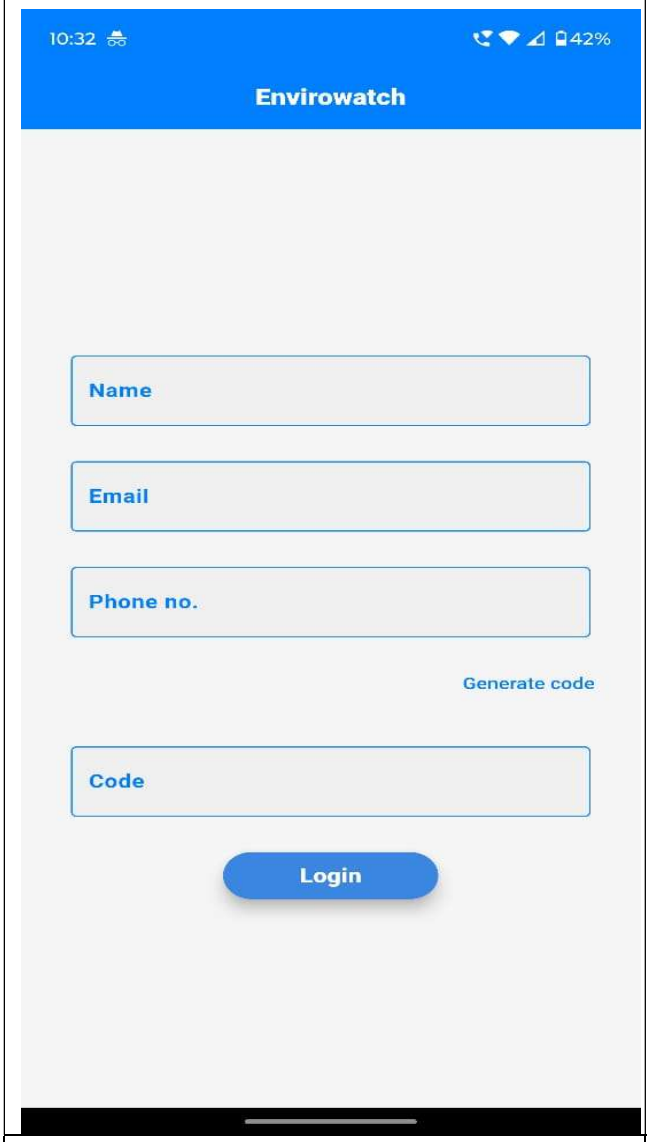



Fig 8.1 Sign In Screen

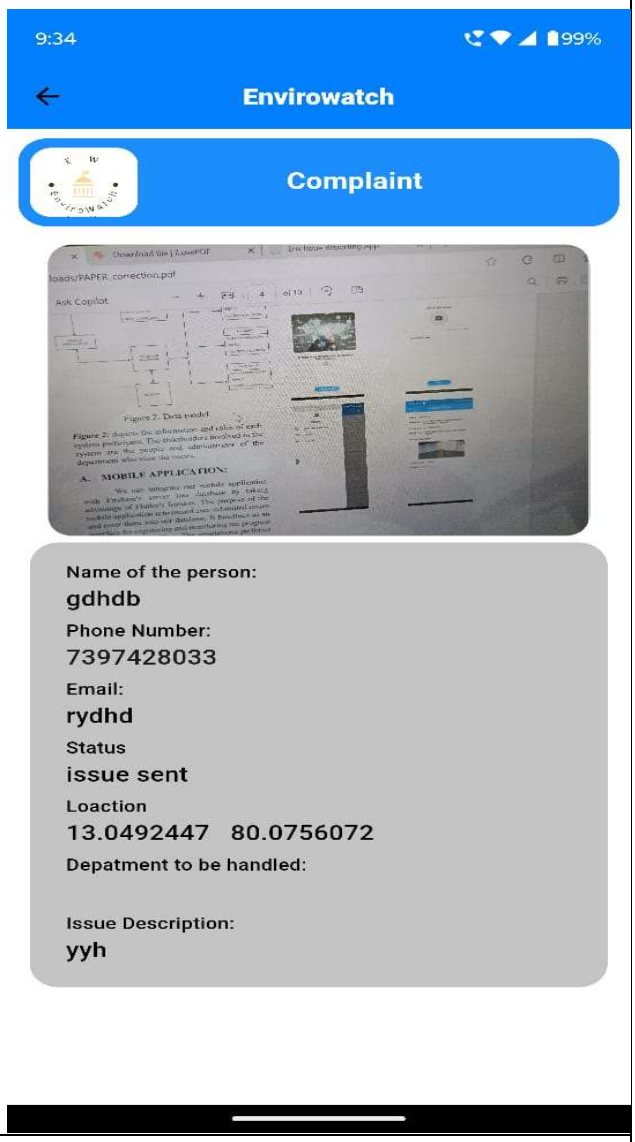



Fig 8.2 Issue Detail

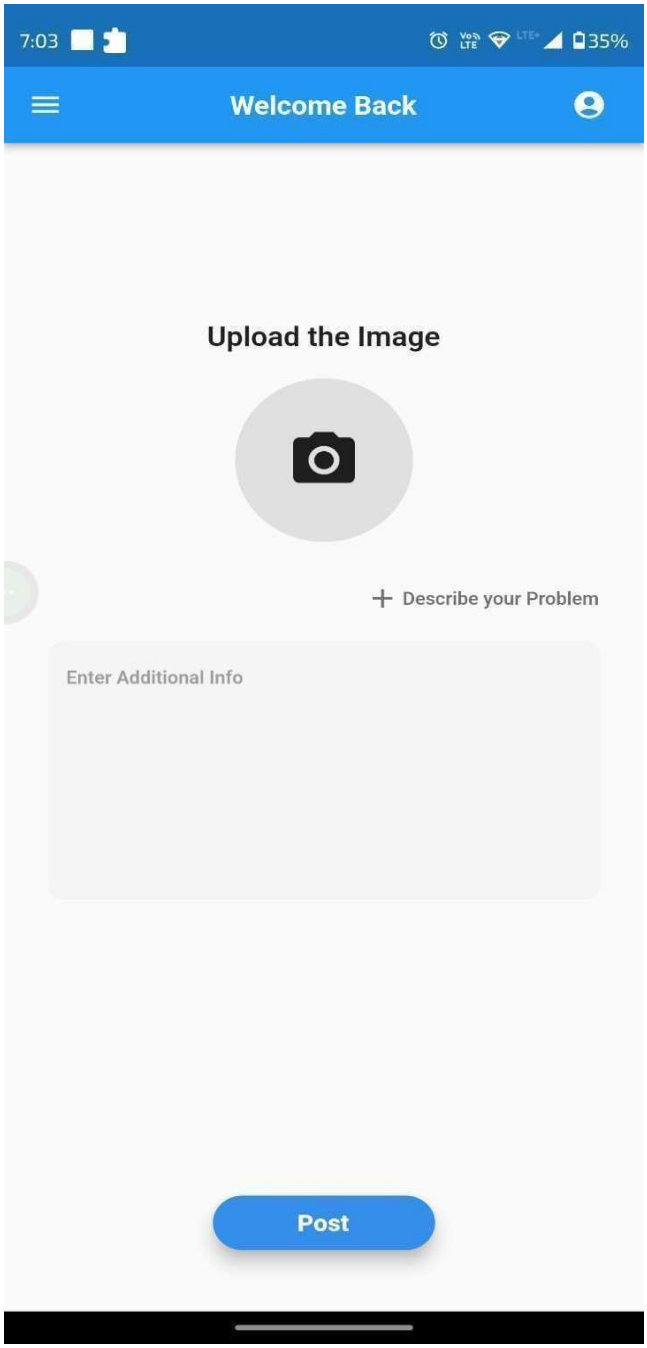



Fig 8.3 Image and message posting screen

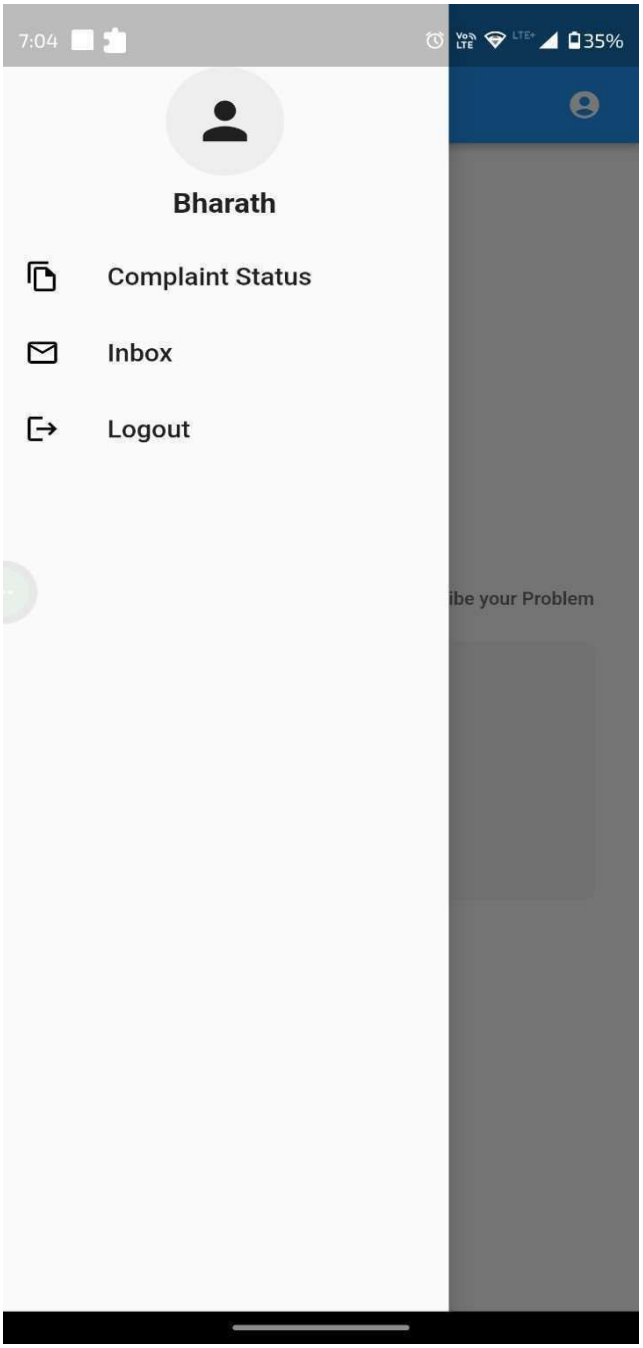



Fig 8.4 Details Screen



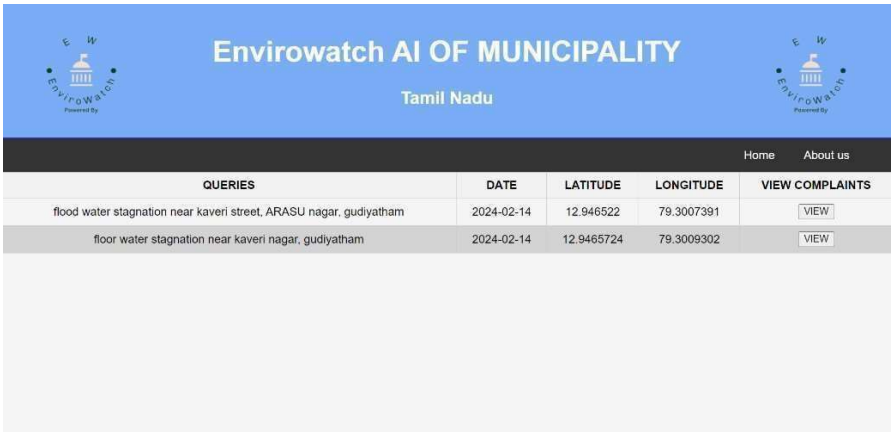



**Fig 8.5 Data set and website**

## 8.2 SDG GOALS:

- SDG 8 - **Decent Work and Economic Growth**:

  By promoting tourism and facilitating bookings for accommodations, your application contributes to economic growth by supporting businesses in the hospitality sector, creating employment opportunities, and stimulating local economies in tourist destinations.

- SDG 9 - **Industry, Innovation, and Infrastructure**:

  Your application leverages advanced technologies and robust data integration to provide a seamless platform for travel planning, accommodation booking, and destination exploration, thereby contributing to the development of innovative infrastructure within the tourism industry.

- SDG 11 - **Sustainable Cities and Communities:**

  By providing transportation recommendations and promoting responsible tourism practices, your application encourages sustainable mobility and helps alleviate the environmental impact of tourism on urban areas, contributing to the development of sustainable cities and communities.

- SDG 12 - **Responsible Consumption and Production:**

  By addressing environmental issues promptly and efficiently, your project promotes responsible consumption and production practices. By streamlining the process of reporting and addressing issues, resources can be allocated more effectively, leading to reduced waste and improved resource management.

- SDG 15 **- Life on Land**:

  Your project directly contributes to preserving terrestrial ecosystems and biodiversity by addressing issues such as garbage accumulation and infrastructure damage. By maintaining healthy and sustainable environments, you help support diverse ecosystems and habitats on land.

## 8.3 Paper Publication

# ENVIROWATCH – MOBILIZING COMMUNITIES TO SOLVE VARIOUS ENVIRONMENTAL ISSUES

BHARATH.P [3], BAALA.B [4], MARIA ROBIN ANDREW [5], DR. P.J. SATHISH KUMAR [1],

DR. N.PUGAZHENDI [2].

FACULTY [1,2], UG SCHOLAR [3,4,5]

COMPUTER SCIENCE AND ENGINEERING, PANIMALAR ENGINEERING COLLEGE

barath.p2002@gmail.com, balamani432003@gmail.com, mariarobin1717@gmail.com, sathishjraman@gmail.com, pughazendi@gmail.com.

***Abstract*— Understanding India's flood issues involves more than just knowing the immediate cause, effects, and damage; it also entails realizing the intricate interactions between environmental, social, and economic factors that lead to this recurring crisis. Disaster recovery is a challenging task that involves managing and prioritizing rescue efforts in addition to locating the impacted areas. The goal of this study is to limit the amount of resources needed for recovery in the aftermath of a disaster with the assistance of the people. The solution we propose will gather the bare minimum of information from the impacted individuals regarding the challenges and problems brought on by the tragedy using a mobile application. The visual data is classified based on the problem using deep learning techniques which are then sent to the relevant departments to monitor the issue using our interface. We developed the deep learning model to utilize neural network technology for data classification and is trained on relevant data sets corresponding to different problems which include water stagnation, damaged electric pole, damaged roads and waste garbage accumulation. Our technology shows its ability to crowd source flood related issues from the community in tragedy and acts as a go-between for the affected people and the relevant authorities to accelerate the necessary work.**

***Keywords*— CNN, Environmental issues, Disaster recovery**

## I. INTRODUCTION

India's monsoons, a double-edged sword, bring both life-giving rain and destructive floods. Floods in India are a recurrent natural disaster with profound impacts on the country's socio-economic fabric. The densely populated regions of India, coupled with poor infrastructure in many areas, exacerbate the devastation caused by floods.

The impacts of these inundations are multifaceted, ranging from loss of life and injury to extensive damage to homes, infrastructure, and agriculture. Displacement of communities is a common consequence, leading to humanitarian crises and strained resources for relief efforts. Moreover, floods disrupt transportation networks, hinder access to essential services, and impede economic activities, further exacerbating the challenges faced by affected populations. 2023 saw another deluge, particularly ravaging Assam [4], Bihar [3], Sikkim [1], Chennai [2] and Kerala. Millions were displaced, villages vanished underwater, and fertile fields transformed into wastelands, threatening livelihoods and food security. Floods caused an average annual loss of $100 billion worldwide in terms of the economy [5]. The population and economic worth of material assets in flood-prone areas are expected to rise, which will likely be amplified by other changes in baseline conditions such as climate, drainage basin features, river modifications, etc. These numbers are predicted to rise [6].

With a history of recurring floods across various regions, India has developed an extensive framework for disaster management, including response, relief, and recovery efforts. However, the sheer scale and complexity of flood disasters often strain these mechanisms, necessitating innovative approaches to facilitate effective recovery. The process of flood disaster recovery in India is

multifaceted, encompassing various stages from immediate post-disaster response to long-term rehabilitation and reconstruction. Immediate response efforts focus on saving lives, providing emergency relief, and ensuring the basic needs of affected populations are met. This is followed by the assessment of damages and needs, which informs the planning and implementation of recovery strategies.

The Indian government faces a myriad of challenges in short-term disaster recovery, ranging from logistical hurdles to systemic issues. Disasters strain infrastructure and resources, hindering swift mobilization of personnel and supplies, especially in remote areas with damaged access routes. Coordination complexities among government agencies, NGOs, and international organizations lead to delays and inefficiencies in relief distribution. Systemic challenges include financial limitations, bureaucratic hurdles, and concerns over corruption, slowing aid distribution and reconstruction efforts. Climate change intensifies disasters, while social vulnerabilities exacerbate challenges in reaching marginalized groups. Additional obstacles include data and communication gaps, neglect of survivors' mental health, and difficulty in restoring livelihoods. Addressing these multifaceted challenges demands comprehensive strategies integrating short-term relief with long-term resilience-building, necessitating coordinated efforts across sectors and stakeholders.

Pinpointing flood-stricken regions in India can be a harrowing task, cloaked in a fog of logistical and technological hurdles. The sheer vastness of the affected areas, often spanning multiple states, presents a geographical challenge. The lack of real-time data collection systems further compounds the issue, leaving authorities scrambling for accurate assessments of the situation on the ground. Locating flood-affected areas in India is not just a matter of distance; it's a race against time and a battle against nature's fury, demanding innovative solutions and unwavering resilience.

Prioritizing the recovery of the flood-damaged area first can cause problems during the recovery process, and some of the impacted areas might not be located. Allowing people to report complaints to the system, which categorizes them and forwards them to the proper department for resolution. The system serves as an intermediary for people and the department responsible for addressing the problems that the people is facing after the flood tragedy. With understanding of the areas affected by water damage, this can speed up the rescue and recovery efforts. Flooding resulted in damage to roads, electrical poles, waste accumulation, and water stagnation which hinder the pace of the rescue. With the knowledge of these issues, which the public can assist in reporting, could help to accelerate the rescue effort.

## II. RELATED WORK

A small number of strategies have been implemented to address the issue of the mentioned problem which our project focuses on. A portion of them are in a state of doom, while others have been put into action. Some of them are being in a gloom and some of them has been implemented. Previous solutions have been shown to be able to predict some types of problems based on a predetermined set of situations, such as flood and hurricane predictions. Processing the specified problem's kind is one approach to solving it. One such instance is categorizing the nature of the issue, such as flood stagnation. And our project is built around the categorization of the issues that people raise to solve.

In order to discover an effective solution within the parameters of our project, certain parties have devised their own methodologies. An example of a comparable government strategy is the Namma Salai app, which was created by the Tamil Nadu government [7]. Namma Salai, aptly named "Our Road," transforms smartphones into tools for tackling road woes. You spot a pothole, uneven surface, or another road problem. This smartphone app bridges the gap between public concern and government action, aiming to improve road conditions across the state. Simply open the app, locate your spot using GPS, and snap a picture. Briefly describe the issue and submit the report. The app functions as a public forum. See reported issues near you, upvote concerns gaining traction, and even add comments or photos for better context. This collective awareness helps prioritize critical repairs. Upon repair, your feedback holds authorities accountable. Because of its fundamental features, communities are able to demand improved roads collectively, making everyone's trip safer and more comfortable.

A Video Surveillance System for realtime Flood Detection and a Mobile App for Flood Alert using AI Video Classification[8]. This is an early prediction an alert mechanism which sends an alert via the application. Other approaches include a

Snow and Ice Cover Classification based on the complexity of snow cover on road pavement based on image-meteorology-temperature fusion[9]. This system runs on the combination between different existing AI model and provides precised results. This system is mainly used for Statistical Weather Data to determine the density of snow cover on road pavements particularly during winter.

Every research and system development leads to our project development. Our project development relies on the people's interaction of the problems they face and reporting them to us through an application. The application reports a compliant of the problems to the system. The app we developed uses the same stratgories like the Namma salai app. The project uses the deep learning model developed by us. The deep learning model classifies the issue based on the visual data provided by people. The classification of the classified data helps the different departments to deal with the given issues effectively.

Few of the approaches has been in place in order to solve the problem in shadow and some of them has been in use. Some of them are being in a gloom and some of them has been implemented. The solution proposed may find predict a problem like flood prediction, hurricane prediction and other solutions may predict type of the problem from a given set of scenarios. Some solution include processing of the type of the problem mentioned. One such example include classfying the type of the cause of the problem such as flood stagnation. Our project is based upon the classification of the type of the problem raised by the people.

Some parties have developed approaches to find the efficient solution related to scope of our project. One similar approach taken by the government is the Namma Salai[7] application developed by the government of Tamil Nadu. Namma Salai, aptly named "Our Road," transforms smartphones into tools for tackling road woes. You spot a pothole, uneven surface, or another road problem. This smartphone app bridges the gap between public concern and government action, aiming to improve road conditions across the state. Simply open the app, locate your spot using GPS, and snap a picture. Briefly describe the issue and submit the report. The app functions as a public forum. See reported issues near you, upvote concerns gaining traction, and even add comments or photos for better context. This collective awareness helps prioritize critical repairs. Upon repair, your feedback holds authorities accountable. Its core functionality empowers communities to collectively demand better roads, leading to smoother and safer commutes for all.

A video surveillance system was developed for realtime flood detection and mobile app for flood alert using AI video classification[8]. This is a early prediction an warning mechanism which sends alert via the app. Other approaches include a snow and ice cover classfication based upon the complexity of the coverage of the snow[9]. This system run on the combination of different existing AI models and gives precised results accordingly. This system Is mainly used for statistical meteorological data to find the density of the snow cover on the road pavements especially during winter.

Each and every researches and systems lead to the development of our project. Our project develop uses the people's interaction of the issues faced and report them to us. An application is used to report a compliant about the issues to our system. The application we developed uses similar stratgories as such as the Namma Salai app. This project uses a deep learning model which was develop by us. This model classify the issue based upon the visual data given by the people. The system uses these classified data to help the various department to handle the given issues efficiently.

## III. SYSTEM MODEL DESCRIPTION

*Figure 1*. depicts the overall system data flow overview. It displays the data flow within the system that the user has started. An application made for mobile platforms was created using the Flutter mobile development framework, which is cross-platform. Through the use of this mobile application, the user's data regarding the issue of people is gathered. After that, the system's database receives this data. We take advantage of Firebase, a serverless cloud software as a service platform offered by Google, and its real-time database features [10]. Next, our trained deep learning model receives the Firebase data and uses it to classify it. Based on the category of the issue set used to train the model, the data is categorized. After classifying the data according to the issue, the model forwards the information to the appropriate department. The system's backend work has been handled with the Flask backend Python framework. A web interface integrated with the Flask framework is used to see the data. The updated classified data from the Firebase database is shown on the web interface.

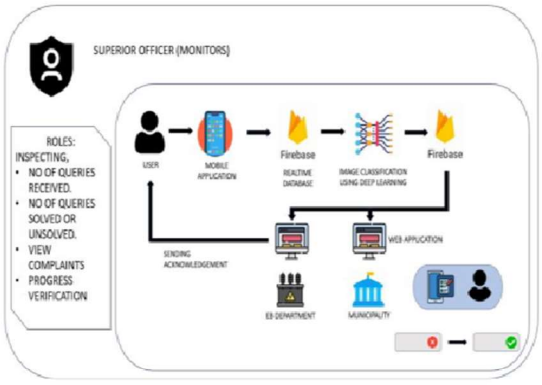



Figure 1. Work flow

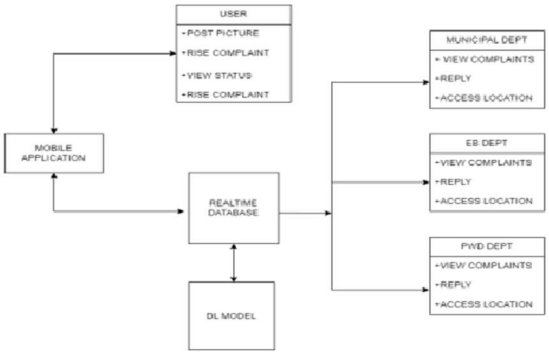


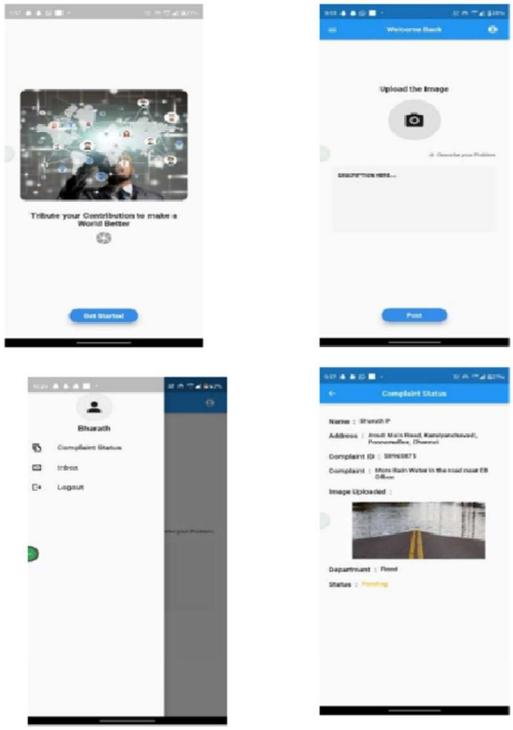

Figure 2. Data model

***Figure 2:*** depicts the information and roles of each system participant. The stakeholders involved in the system are the people and administrator of the department who view the issues.

## A. MOBILE APPLICATION:

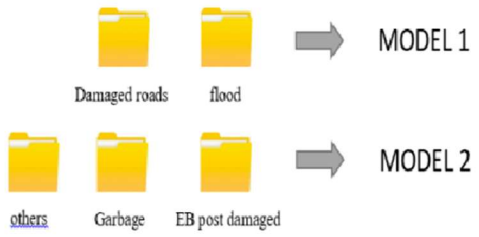
The mobile application has been developed using Flutter Framework. Flutter is an open-source UI software development kit created by Google, designed to build natively compiled applications for mobile, web, and desktop from a single codebase. One of Flutter's key features is its use of the Dart programming language, which offers a simple yet powerful syntax and provides robust tools for building modern applications. We can integrate our mobile application with Firebase's server less database by taking advantage of Flutter's features.

The purpose of the mobile application is to record user-submitted issues and enter them into our database. It functions as an interface for registering and monitoring the progress of the reported problem. The smartphone performs the functions shown in Figure 2. The user provides the mobile device with the issue's specifics as input. The user's name, phone number, and a description of the issue are included in the issue information. Using the smartphone application, the user snaps a photo of the problem. The smartphone application takes a picture and simultaneously records the coordinates of the shot place. These data are delivered to the system's database so that Firebase can store them for classification. The Firebase authentication mechanism of the mobile application is used to verify the user's phone number [10]. The phone number is verified in order to thwart spam and other fraudulent information. The image that is taken by the mobile application is utilized to determine the nature of the issue and locate the appropriate department. Following this procedure, further data are updated, including complaint ID, problem type, and department to be handled. An interface to view the concerns raised and their status is also provided via the mobile application.



## B. DATASET DESCRIPTION:

As we use technique called ensemble learning, two classes of images given to model1, and the other three classes of images given to model 2.



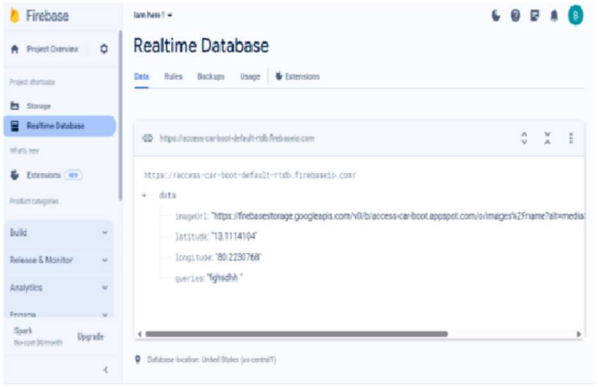

| CLASSES | DATASET SIZE |
|---|---|
| DAMAGED ROADS | 135 |
| FLOOD | 135 |
| GARBAGE | 148 |
| EB POST DAMAGED | 148 |
| OTHERS | 148 |

## SAMPLE DATA:


FIGURE 3

## C. DATABASE DESCRIPTION:

Firebase l e is a cloud-hosted evelopers to time across . It offers offline support, automatic synchronization, and seamless integration with Firebase Authentication and other Firebase services. Data is stored as JSON and can be accessed directly from clients or via server-side code. Firebase Real time Database is commonly used for building real-time applications such as chat apps, collaborative tools, and live data dashboards.




Fields in database, are

- image url
- Latitude
- Longitude
- problem description
- Name and mobile number of the person who raised the complaint.

And these data were queried from the realtime database via Flask. And by using

**Fetchlist()** and **setInterval()** functions in java script, the data was retrieved in real time and reported to responsible entities.

## MODEL ARCHITECTURE:

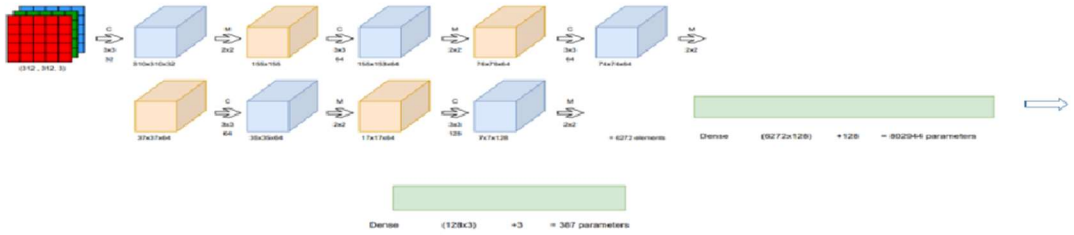


FIGURE .4

## TYPES OF OPERATIONS PERFORMED IN CONSTRUCTED CNN:

### CONVOLUTION:

Convolution is a fundamental operation in the realm of convolutional neural networks (CNNs) primarily employed for feature extraction, notably in image processing tasks. This operation revolves around the application of a filter, often referred to as a kernel, onto an input image or feature map. As the filter traverses the input, it performs element-wise multiplication between its weights and corresponding input pixels, followed by summation.

((W - F + 2P) / S) + 1)

- W is the input volume size (312 in this case)
- F is the receptive field size (3 in this case)
- P is the padding size (0 in this case, as it's 'valid')
- S is the stride (1 in this case)

### MAX POOLING:

Max pooling stands as a pivotal down sampling technique frequently utilized in CNNs to diminish the spatial dimensions of feature maps while preserving salient information. This operation partitions the input feature map into non-overlapping rectangular regions and selects the maximum value from each region to represent it in the output.

$$\text{output\_height} = \frac{\text{input\_height} - \text{pool\_size}}{\text{strides}} + 1$$

$$\text{output\_width} = \frac{\text{input\_width} - \text{pool\_size}}{\text{strides}} + 1$$

### FILTERS:

Filters, also known as kernels, are small spatially-located matrices that are convolved (slid) across input data (such as images) to perform operations like feature detection. In the context of image processing, filters are typically small grids of numbers (often 3x3 or 5x5) representing weights.

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

FIGURE 5.VERTICAL FILTER

## ACTIVATION FUNCTIONS USED:

- **RELU**

  The ReLU (Rectified Linear Unit) activation function is a simple mathematical function used in artificial neural networks. It is defined as follows:

  **f(x)=max(0,x)**

  In simpler terms, if the input x is greater than zero, the output is equal to the input. If the input is less than or equal to zero, the output is zero.

- **SOFTMAX**

  The softmax activation function is commonly used in neural networks, particularly in the output layer of classification tasks. It takes a vector of arbitrary real-valued scores as input and converts them into probabilities that sum up to 1. The function is defined as follows:

  $$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

  In simpler terms, the softmax function exponentiates each element of the input vector and then normalizes these values by dividing each exponentiated value by the sum of all exponentiated values in the input vector.

## FLOW IN ARCHITECTURE:

Initially after resizing the image in pre-processing stage, the image is fed to the model. Eventually the image in RGB format, (312 x 312 x 3).

### Input Layer (Conv2D):

The model starts with a convolutional layer (Conv2D). The layer has 32 filters/kernels, each with a size of 3x3. The activation function used is ReLU, which introduces non-linearity into the model.The input shape is (312, 312, 3), indicating an image with a height and width of 312 pixels and 3 color channels (RGB).

= ((312 – 3) /1) +1

= 309+1 = 310

➔(310 X 310 X 32)

### MaxPooling2D Layer:

After the first convolutional layer, a max-pooling layer (MaxPooling2D) with a pool size of 2x2 is added. Max pooling reduces the spatial dimensions of the feature maps by selecting the maximum value from each 2x2 region. This helps in reducing computational complexity and retaining important features while downsampling the feature maps.

= ((310 – 2 ) / 2 ) + 1

➔(155 X 155 X 32)

### Second Convolutional Layer (Conv2D):

Another convolutional layer is added with 64 filters of size 3x3. The activation function used is ReLU.

= ((155 -3) /1)+1

➔(153 X 153 X64)

### Second MaxPooling2D Layer:

Similar to the first max-pooling layer, a max-pooling layer with a pool size of 2x2 is added after the second convolutional layer.

= ((153-2) /2)+1

➔(76 X 76 X 64)

### Third Convolutional Layer (Conv2D):

Another convolutional layer is added with 64 filters of size 3x3. The activation function used is ReLU.

= ((76-3)/1)+1

➔(74 X 74 X 64)

### Third MaxPooling2D Layer:

Similar to previous max-pooling layers, a max-pooling layer with a pool size of 2x2 is added after the third convolutional layer.

= ((74-2)/ 2)+1

➔(37 X 37 X 64)

### Fourth Convolutional Layer (Conv2D):

Another convolutional layer is added with 64 filters of size 3x3. The activation function used is ReLU.

= ((37-3)/1)+1

➔(35 x 35 x 64)

### Fourth MaxPooling2D Layer:

Similar to previous max-pooling layers, a max-pooling layer with a pool size of 2x2 is added after the fourth convolutional layer.

= ((35-2)/2) +1

➔(!7 X 17 X 64)

### Fifth Convolutional Layer (Conv2D):

The final convolutional layer is added with 128 filters of size 3x3. The activation function used is ReLU.

= ((17 -3)/1)+1

➔(15 X 15 X128)

### Fifth MaxPooling2D Layer:

Similar to previous max-pooling layers, a max-pooling layer with a pool size of 2x2 is added after the fifth convolutional layer.

=((15-2)/2)+1

➔(7 X 7 X 128)

### Flatten Layer:

After the convolutional layers, the feature maps are flattened into a 1D vector using the Flatten layer. This prepares the data to be fed into the fully connected layers by converting the 3D feature maps into a 1D vector.

= ( 7* 7* 128)

= 6272 elements.

### Dense (Fully Connected) Layers:

Two dense layers (Dense) follow the flattened layer. The first dense layer consists of 128 neurons with ReLU activation. The second dense layer consists of 3 neurons, representing the output classes in this case.

### DENSE LAYER 1:

(6272 X 128 ) +128 = 802944 parameters.

### DENSE LAYER 2:

(128 X 3) +3 = 387 parameters.

The activation function used in the second dense layer is softmax, which converts the raw output scores into probabilities, indicating the likelihood of each class. In summary, this model architecture consists of alternating convolutional and max-pooling layers for feature extraction, followed by fully connected layers for classification. The convolutional layers detect and learn features from the input images, while max-pooling layers down sample the feature maps. Finally, the fully connected layers perform classification based on the learned features.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 310, 310, 32)      896

max_pooling2d (MaxPooling2   (None, 155, 155, 32)      0
D)

conv2d_1 (Conv2D)            (None, 153, 153, 64)      18496

max_pooling2d_1 (MaxPoolin   (None, 76, 76, 64)        0
g2D)

conv2d_2 (Conv2D)            (None, 74, 74, 64)        36928

max_pooling2d_2 (MaxPoolin   (None, 37, 37, 64)        0
g2D)

conv2d_3 (Conv2D)            (None, 35, 35, 64)        36928

max_pooling2d_3 (MaxPoolin   (None, 17, 17, 64)        0
g2D)

conv2d_4 (Conv2D)            (None, 15, 15, 128)       73856

max_pooling2d_4 (MaxPoolin   (None, 7, 7, 128)         0
g2D)

flatten (Flatten)            (None, 6272)              0

dense (Dense)                (None, 128)               802944

dense_1 (Dense)              (None, 3)                 387

=================================================================
Total params: 970435 (3.70 MB)
Trainable params: 970435 (3.70 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

THE FINAL SOFTMAX ACTIVATION FUNCTION CLASSIFIES THE GIVEN IMAGE BETWEEN THESE CLASSES:

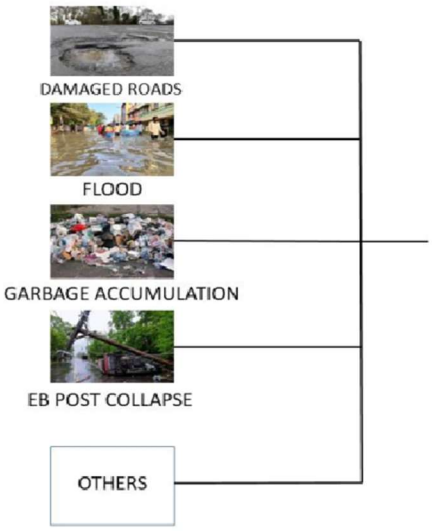



FIGURE 6

## APPLICATION SCREEN SHOTS:


FIG.7: EB DEPARTMENT WEBSITE


FIG.8: VIEWING LIVE UPDATES


FIG.9: MUNICIPALITY WEBSITE


FIG. 10: VIEWING LIVE UPDATES

## Conclusion:

The study employed ensemble learning to classify data into five distinct classes, with the first model focusing on two classes and the second model addressing three classes. The utilized Convolutional Neural Network (CNN) architecture featured convolutional layers alternating with max-pooling layers, culminating in fully connected layers for classification purposes. Following training, the model was seamlessly integrated into a web application for deployment.

In summary, the ensemble learning strategy, coupled with the CNN architecture, exhibited notable efficacy in tackling multi-class classification challenges. Through the synergistic utilization of

multiple models, the ensemble approach demonstrated improved accuracy and resilience in classification tasks. The successful deployment of the model within a web application context underscores its practical applicability and scalability. Future research avenues may explore additional optimizations in model architecture, data preprocessing methodologies, and ensemble strategies to further enhance classification performance in real-world scenarios.

## REFERENCES:


[1] India Today, 6th October 2023 - Sikkim flash flood: Disaster relief team fails to land, death toll rises to 30 - https://www.indiatoday.in/india/story/sikkim-flash-floods-update-dead-missing-search-operation-army-jawan-weather-forecast-2445272-2023-10-06

[2] The Hindu, 20th November 2020 - Days after rain, water stagnation continues to plague many areas of Chennai - https://www.thehindu.com/news/cities/chennai/days-after-rain-water-stagnation-continues-to-plague-many-areas-of-chennai/article33208101.ece

[3] The Times Of India, 22nd july 2023 - Flood-affected areas of Haridwar to be declared disaster-hit - https://timesofindia.indiatimes.com/city/dehradun/flood-affected-areas-of-haridwar-to-be-declared-disaster-hit-says-dhami/articleshow/102027373.cms

[4] Aljareeza, 18 September 2023 - 'Nowhere else to go': In India's Assam, frequent floods hit river islanders - https://www.aljazeera.com/gallery/2023/9/18/photos-nowhere-else-to-go-in-indias-assam-frequent-floods-hit-river-islanders

[5] Desai, B., Maskrey, A., Peduzzi, P., De Bono, A., Christian, H., and United Nations Office for Disaster Risk Reduction (UNISDR), 2015. Making Development Sustainable: The Future of Disaster Risk Management, Global Assessment Report on Disaster Risk Reduction. Genève, Suisse : United Nations Office for Disaster Risk Reduction (UNISDR).

[6] Kettner, A.J., Cohen, S., Overeem, I., Fekete, B.M., Brakenridge, G.R., and Syvitski, J.P.M., 2018. Estimating Change in Flooding for the 21st Century Under a Conservative RCP Forcing, In: Global Flood Hazard: Applications in Modeling, Mapping, and Forecasting, eds: G.J.-P. Schumann P.D. Bates H. Apel, and G. T. Aronica. Geophysical Monograph Series, Chapter 9, doi:10.1002/9781119217886.ch9, Wiley, 2018.

[7] Namma Salai App website by the government of Tamil Nadu, India to improve the conditions of the problems in the state highway - https://frs.tnhw.pmis.cloud

[8] K. Priya Menon, L. Kala, International Conference on Computing Methodologies and Communication (ICCMC), Video surveillance system for realtime flood detection and mobile app for flood alert, DOI: 10.1109/ICCMC.2017.8282518, IEEE, 2018

[9] Sen Yang, Chengwei Lei, VII IEEE SENSORS JOURNAL, Research on the classification method of complex snow and ice cover on highway pavement based on image-meteorology-temperature fusion, DOI 10.1109/JSEN.2023.3336667, IEEE, 2023

[10] Pankaj Chougale, Vaibhav Yadav, Dr. Anil Gaikwad, 2021, International Research Journal of Modernization in Engineering Technology and Science, Firebase - overview and usage, Volume:03/Issue:12, 2021

## 8.4 Plagiarism

ORIGINALITY REPORT

| 16% | 7% | 7% | 12% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to University of Western Ontario<br>Student Paper | 2% |
| 2 | Submitted to CITY College, Affiliated Institute of the University of Sheffield<br>Student Paper | 1% |
| 3 | Submitted to universititeknologimara<br>Student Paper | 1% |
| 4 | Submitted to University of Huddersfield<br>Student Paper | 1% |
| 5 | A.J. Kettner, G. J.-P. Schumann, G.R. Brakenridge. "Applying Remote Sensing to Support Flood Risk Assessment and Relief Agencies: A Global to Local Approach", IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020<br>Publication | 1% |
| 6 | Submitted to SP Jain School of Global Management<br>Student Paper | 1% |

7 ieeexplore.ieee.org
Internet Source

8 Submitted to INTI Universal Holdings SDM BHD
Student Paper

9 www.mdpi.com
Internet Source

10 ijircce.com
Internet Source

11 Submitted to University of Technology, Sydney
Student Paper

12 Submitted to Berlin School of Business and Innovation
Student Paper

13 Submitted to CSU, San Jose State University
Student Paper

14 Submitted to Sunway Education Group
Student Paper

15 thesai.org
Internet Source

16 Submitted to Adtalem Global Education
Student Paper

17 Sen Yang, Chengwei Lei. "Research on the

ice cover on highway pavement based on image-meteorology-temperature fusion", IEEE Sensors Journal, 2023
Publication

Publication

18 www.ijraset.com
Internet Source

19 Submitted to Higher Education Commission Pakistan
Student Paper

20 Submitted to Liverpool John Moores University
Student Paper

21 Submitted to University of Lancaster
Student Paper

22 www.daltoncole.com
Internet Source

23 Submitted to University of Birmingham
Student Paper

24 www.tojned.net
Internet Source

25 Guanjin Qu, Naichuan Cui, Huaming Wu, Ruidong Li, Yuemin Ding. "ChainFL: A Simulation Platform for Joint Federated Learning and Blockchain in Edge/Cloud Computing Environments", IEEE

| | | |
|---|---|---|
| 26 | discovery.researcher.life<br>Internet Source | <1% |
| 27 | Submitted to British University in Egypt<br>Student Paper | <1% |
| 28 | Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur<br>Student Paper | <1% |
| 29 | Submitted to October University for Modern Sciences and Arts (MSA)<br>Student Paper | <1% |
| 30 | Rim Trabelsi, Khaled Nouri, Imen Ammari. "Enhancing Traffic Sign Recognition Through Daubechies Discret Wavelet Transform and Convolutional Neural Networks", 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET), 2023<br>Publication | <1% |
| 31 | Submitted to University of Hertfordshire<br>Student Paper | <1% |
| 32 | www.atlantis-press.com<br>Internet Source | <1% |

# PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

## PECTEAM 2K24

ICOnIC

This is to certify that ~~Dr./Mr./Ms~~... BHARATH: P ..... of ..... CSE ..... from PANIMALAR ENGINEERING COLLEGE ..... has presented paper entitled ENVIRO WATCH-AI MOBILIZING COMMUNITIES TO SOLVE VARIOUS ENVIRONMENTAL ISSUES ..... in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

| FACULTY CO-ORDINATOR | CO - CONVENER | CO - PATRON |
|---|---|---|
| | Prof. S. VIMALA,M.TECH.,(Ph.D) | Dr.K.MANI,M.E.,Ph.D., |

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY

# PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

## PECTEAM 2K24

ICOnIC

This is to certify that ~~Dr./Mr./Ms~~.....B...BABLA.....of .....CSE.....from PANIMALAR ENGINEERING .....COLLEGE.....has presented paper entitled..ENVIRO..WATCH-.MOBILIZING..... .COMMUNITIES.....TO.....SOLVE.....VARIOUS....ENVIRONMENTAL.....ISSUES..... in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

R. Josephine
FACULTY CO-ORDINATOR

CO - CONVENER
Prof. S. VIMALA,M.TECH.,(Ph.D)

K. Mani
CO - PATRON
Dr.K.MANI,M.E.,Ph.D.,

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY

# PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

## PECTEAM 2K24

This is to certify that ~~Dr./~~Mr./~~Ms.~~ MARIA ROBIN ANDREW of CSE from PANIMALAR ENGINEERING COLLEGE has presented paper entitled ENVIRO WATCH - AI MOBILIZING COMMUNITIES TO SOLVE VARIOUS ENVIRONMENTAL ISSUES in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

| FACULTY CO-ORDINATOR | CO - CONVENER | CO - PATRON |
|---|---|---|
| | Prof. S. VIMALA,M.TECH.,(Ph.D) | Dr.K.MANI,M.E.,Ph.D., |

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY

# CHAPTER 9

## REFERENCES


- The Hindu, 20th November 2020 - Days after rain, water stagnation continues to plague many areas of Chennai - https://www.thehindu.com/news/cities/chennai/days-after-rain-water-stagnation-continues-to-plaguemany-areas-of-chennai/article33208101.ece
- The Times Of India, 22nd july 2023 - Floodaffected areas of Haridwar to be declared disasterhit - https://timesofindia.indiatimes.com/city/dehradun/flood-affected-areas-of-haridwar-to-be-declareddisaster-hit-saysdhami/articleshow/102027373.cms
- Aljareeza, 18 September 2023 - 'Nowhere else to go': In India's Assam, frequent floods hit river islanders - https://www.aljazeera.com/gallery/2023/9/18/photos-nowhere-else-to-go-in-indias-assam-frequentfloods-hit-river-islanders
- Desai, B., Maskrey, A., Peduzzi, P., De Bono, A., Christian, H., and United Nations Office for Disaster Risk Reduction (UNISDR), 2015. Making Development Sustainable: The Future of Disaster Risk Management, Global Assessment Report on Disaster Risk Reduction. Genève, Suisse : United Nations Office for Disaster Risk Reduction (UNISDR).
- Kettner, A.J., Cohen, S., Overeem, I., Fekete, B.M., Brakenridge, G.R., and Syvitski, J.P.M., 2018. Estimating Change in Flooding for the 21st Century Under a Conservative RCP Forcing, In: Global Flood Hazard: Applications in Modeling, Mapping, and Forecasting, eds: G.J.-P. Schumann P.D. Bates H. Apel, and G. T. Aronica. Geophysical Monograph Series, Chapter 9, doi:10.1002/9781119217886.ch9, Wiley, 2018.
- Namma Salai App website by the government of Tamil Nadu, India to improve the conditions of the problems in the state highway - https://frs.tnhw.pmis.cloud.
- K. Priya Menon, L. Kala, International Conference on Computing Methodologies and Communication (ICCMC), Video surveillance system for realtime flood detection and mobile app for flood alert, DOI: 10.1109/ICCMC.2017.8282518, IEEE, 2018.
- Sen Yang, Chengwei Lei, VII IEEE SENSORS JOURNAL, Research on the classification method of complex snow and ice cover on highway pavement based on image-meteorology-temperature fusion, DOI 10.1109/JSEN.2023.3336667, IEEE, 2023.
- Pankaj Chougale, Vaibhav Yadav, Dr. Anil Gaikwad, 2021, International Research Journal of Modernization in Engineering Technology and Science, Firebase - overview and usage, Volume:03/Issue:12,