CHAPTER 1

INTRODUCTION

The Efficient Proximity Detection and Safety System represents a pioneering approach to enhancing public safety, focusing on rapid response and advanced criminal identification. By leveraging GPS technology and frameworks, this system is designed to empower individuals during emergencies, offering a reliable safety net that integrates seamlessly into daily life. At the core of this system is a mobile application, carefully crafted with user - eccentric design principles to ensure accessibility for all. With just a tap on the emergency button, users can initiate a distress alert that instantly notifies law enforcement. This immediate communication accelerates the response time of dispatched officers, with real-time updates on their location and estimated arrival providing users with crucial information in high-stress situations.

In addition to its emergency response capabilities, the system incorporates cutting-edge image processing techniques to assist in identifying potential threats. Utilizing Flutter frame works capturing images and also using mobile internal storage images to capture images, facilitating faster and more precise criminal identification. These image processing capabilities not only boost the system's effectiveness but also demonstrate a commitment to proactive safety measures. The rapid advancement of technology has opened up new avenues for enhancing public safety. One such innovation is the development of intelligent EPDS Systems, which aim to provide timely and effective assistance to individuals in distress. This project focuses on developing a robust and efficient EPDS System that leverages cutting-edge technologies to detect, analyze, and respond to emergency situations. In today's fast-paced world, emergencies can strike at any moment. Timely and effective response to these situations is crucial to save lives and minimize damage. Traditional EPDS Systems often rely on manual processes, which can be time-consuming and prone to errors. To address these limitations, we propose an innovative EPDS System that leverages advanced technologies to provide rapid and efficient assistance.

Real-time Location Tracking

Accurate and timely location information is crucial for effective emergency response. This project utilizes GPS technology to pinpoint the user's location in real-time. By integrating GPS with advanced mapping and Geo-fencing techniques, the system can accurately determine the user's location, even in challenging environments. The system can also track the user's movement and generate alerts if they enter predefined danger zones.

Image Recognition using TensorFlow Lite

Image recognition plays a vital role in providing additional context to emergency res-ponders. By analyzing images captured at the scene, the system can identify potential threats, such as hazardous materials or suspicious individuals. TensorFlow Lite, a lightweight machine learning framework, enables efficient image processing and classification on mobile devices. The system can be trained on a large datasets of images to recognize various objects and scenes, improving the accuracy of emergency assessments.

Emergency Signal Detection and Analysis

The system employs advanced signal processing techniques to detect emergency signals, such as screams, loud noises, or specific keywords. By analyzing audio signals, the system can differentiate between genuine emergencies and false alarms. Machine learning algorithms can be used to further refine the detection process, improving accuracy and reducing false positives.

Communication and Notification

A reliable communication infrastructure is essential for effective emergency response. The system utilizes various communication channels, including SMS, and push notifications, to alert emergency services and relevant authorities. Real-time updates on the user's location and the situation on the ground can be transmitted to res-ponders, enabling them to take swift action.

CHAPTER 2

PROBLEM STATEMENT

In today's society, ensuring prompt response and assistance during emergency situations is paramount for public safety. However, existing systems often lack efficient mechanisms for individuals to quickly alert law enforcement authorities when they are in distress. Moreover, the inability to provide real-time location information to res-ponders and the absence of robust tools for suspect identification further hinder effective law enforcement intervention.

1. Unable to Detect user Real-time Location and Tracking Dynamic location.
2. Loss of Evidence
3. Unable to Identify a criminals

The existing systems primarily focus on sending **emergency notifications to the police control room without considering the proximity of the nearest police officers** to the location of the incident. Our system goes beyond by integrating advanced functionalities such as **real-time location tracking and nearest police identification and Criminals Image Recognition**.

while some suspect identification systems allow individuals to capture and send photos of suspects to the police, they are typically standalone solutions not integrated into safety applications.

CHAPTER 3

LITERATURE REVIEW

3.1 Author G. Singh and A. K. Goel, "Face Detection and Recognition System using Digital Image Processing", *2020 2nd International Conference on Innovative Mechanisms for Industry.* *Applications (ICIMIA)*, pp. 348-352, 2020.
LINK :https://ieeexplore.ieee.org/document/9074838

The topic focuses on the development of an interactive map application aimed at facilitating real-time crime reporting. This application is designed to enhance emergency response efforts by providing crucial information to res ponders, such as law enforcement agencies, to better understand the prevailing crime situation in specific areas. By offering real-time updates on criminal activities, the application enables res ponders to make quicker and more informed decisions when addressing incidents.

Advantages:

- Enhanced Public Safety Awareness: Utilizing advanced technologies and data analytic to gather real-time information enables authorities to promptly identify potential risks and ensure proactive measures for public safety.
- Improved Communication and Collaboration: Enhanced communication platforms facilitate seamless information sharing among relevant agencies, fostering swift collaboration and coordination during emergencies for effective response and resource allocation.
- Efficient Emergency Response: Leveraging data-driven insights enables authorities to optimize resource allocation, streamline response efforts, and mitigate risks promptly, ensuring a more efficient and effective emergency response process

Limitations:

- Data Accuracy and Completeness: The reliability of crime data may vary, potentially leading to inaccuracies in reporting. Incomplete or outdated information could hinder the application's effectiveness in providing real-time insights to emergency res ponders.

- Privacy Concerns: Collecting and sharing sensitive crime-related information raises privacy concerns for individuals involved. Safeguarding personal data and ensuring compliance with privacy regulations is paramount to maintain trust and mitigate potential legal ramifications.

3.2 K. B. Obaid et al., "Deep Learning Models Based on Image Classification: A Review", *International Journal of Science Business(ijsab)*, vol. 4, no. 11, pp. 75-81, 2020. LINK:https://www.academia.edu/81078986/Image_Classification_and_Annotation_Using_Deep_ Learning?uc-sb-sw=7555348

This paper focuses on the design and improvement of a Conventional Neural Network (CNN) model with the aim of enhancing the accuracy of face recognition systems. The authors propose modifications and optimizations to the existing CNN architecture specifically tailored for face recognition tasks. By leveraging advancements in deep learning techniques, they seek to address challenges such as variations in lighting, pose, and facial expressions to achieve higher accuracy in identifying individuals from images or video streams

Advantages:

- Improved Accuracy: By fine-tuning the CNN architecture and incorporating enhancements, the accuracy of face recognition systems can be significantly boosted. This leads to more reliable identification of individuals, reducing false positives and negatives.
- Robustness to Variations: The optimized CNN model is likely to be more robust against variations in lighting conditions, facial expressions, and poses. This ensures consistent performance across different scenarios, enhancing the overall reliability of the face recognition system.
- Efficient Feature Extraction: CNNs are adept at automatically learning and extracting relevant features from raw input data. By enhancing the model's architecture, it becomes more efficient at capturing discriminate features from facial images, thereby improving recognition accuracy.

Limitations:

- Data Bias and Generalization: The CNN model's performance may be limited by biases present in the training data, leading to reduced generalization ability across diverse populations.
- Over fitting: Complex CNN architectures and extensive training on limited datasets can lead to over fitting, where the model memorizes specific features of the training data rather than learning generalization patterns.
- Privacy Concerns: Face recognition systems raise significant privacy concerns, particularly regarding the collection, storage, and misuse of biometric data.
- Adversarial Attacks: CNN models for face recognition are vulnerable to adversarial attacks, where imperceptible perturbations to input images can cause miscalculation or unauthorized access.
- Hardware and Computational Resources: The computational requirements of deep learning models, especially complex CNN architectures, may necessitate high-performance hardware and substantial computational resources for training and inference.

3.3. Author: Karur Karthik, Nitin Sharma, Chinmay Dharmatti and Joshua E. Siegel, "A Survey of Path Planning Algorithms for Mobile Robots", *Vehicles*, vol. 3, no. 3, pp. 448-468, 2021.

LINK:https://www.researchgate.net/publication/370821715_A_Comparative_Study_of_Various_ Path_Planning_Algorithms_for_Pick-and-Place_Robots

This paper introduces a dynamic algorithm for path planning utilizing the A* algorithm, while incorporating a distance constraint. Path planning involves finding the optimal route from a starting point to a goal point, considering various factors such as obstacles, terrain, and now, distance constraints. The A* algorithm is a widely used search algorithm in artificial intelligence and robotics for finding the shortest path in a graph or grid, taking into account both the cost to reach a node and an estimated cost to reach the goal.

Advantages:

- Optimal Paths within Distance Limits: The algorithm ensures that the generated paths are not only the shortest but also adhere to specified distance constraints.
- Enhanced Flexibility: By incorporating distance constraints into the A* algorithm, the path planning process becomes more flexible and adaptable to different application scenarios.
- Improved Efficiency: The A* algorithm is known for its efficiency in finding optimal paths, and integrating distance constraints does not significantly degrade its performance.

Limitations:

- Sensitivity to Distance Metric: The effectiveness of the algorithm may depend on the chosen distance metric and how accurately it reflects the actual distance traveled in the environment. Inaccuracies or discrepancies in the distance metric could lead to deviations from expected path lengths and potentially sub-optional route choices.
- Increased Computational Complexity: Incorporating distance constraints into the A* algorithm may increase computational complexity, especially in scenarios with large search spaces or potentially limiting real-time applicability in certain contexts.

3.4 Author: W. Ren, X. Cao, J. Pan, X. Gua, W. Zuo and M.H. Yang, "Image Deblurring via Enchanced Low-Rank Prior", *IEEE Transactions on Image Processing*, vol. 25, no. 7, Jul. 2021.

LINK:https://www.semanticscholar.org/paper/Image-Deblurring-viaEnhanced-LowRank-Prior-Ren-Cao/990c5f2aefab9df89c40025d85013fe28f0a5810

This paper proposes a method for image restoration using optimized Weiner filtering. Image restoration involves the process of improving the quality of an image by reducing noise, blur, or other distortions that may have occurred during image acquisition or transmission. Weiner filtering is a classical technique used for image restoration, aiming to estimate the original, uncorrupted image from the observed degraded version.

Advantages:

- Improved Image Quality: The optimized Weiner filtering algorithm enhances image restoration by effectively reducing noise, blur, or other distortions, resulting in improved image quality and clarity.
- Adaptability to Various Degradation: By optimizing parameters and incorporating adaptive techniques, the method can effectively address a wide range of image degradation types, including noise, blur, and compression artifacts. This adaptability ensures robust performance across diverse image restoration scenarios.
- Automatic Parameter Optimization: The proposed method likely includes techniques for automatically optimizing filter parameters based on the characteristics of the degraded image.

Limitations:

- Sensitivity to Parameter Selection: Despite optimization efforts, the effectiveness of Weiner filtering can still be sensitive to the selection of parameters such as the signal-to-noise ratio (SNR) and blur kernel.
- Limited Performance in Complex Scenes: In complex scenes with high levels of noise, blur, or occlusions, Weiner filtering may struggle to achieve satisfactory restoration results.
- Assumption of Stationary Noise: Weiner filtering assumes that the noise in the degraded image is stationary and follows a known statistical distribution.

3.5 Title: Face Detection and Recognition System using Digital Image Processing
Authors: Singh and A. K. Goel, Date 07 March 2020
Link: https://ieeexplore.ieee.org/document/7473901

This paper presents a system for face detection and recognition using digital image processing techniques. Face detection involves identifying and locating human faces within an image or video frame, while face recognition focuses on identifying individuals by comparing their facial features with a database of known faces.The authors likely describe a comprehensive system that integrates various image processing algorithms and techniques to achieve accurate and efficient face detection and recognition. Key components of the system may include:

Advantages:

- Enhanced Security: The system provides a reliable method for identifying individuals, enhancing security in applications such as access control, surveillance, and authentication.
- Automation: Automation of face detection and recognition tasks reduces the need for manual intervention, leading to increased efficiency and productivity in various domains.
- Versatility: The system can be adapted for use in diverse environments and applications, including law enforcement, border control, attendance tracking, and personalized marketing.
- Accuracy: Digital image processing techniques enable accurate detection and recognition of faces, even in challenging conditions such as varying lighting, facial expressions, and occlusions.
- Real-time Performance: Efficient algorithms and optimization techniques facilitate real-time processing of images or video streams, allowing for rapid detection and recognition of faces in dynamic environments.

Limitations:

- Performance in Challenging Conditions: The system may struggle to accurately detect and recognize faces in challenging conditions such as low light, varying poses, occlusions, or poor image quality.
- Privacy Concerns: Deployment of face detection and recognition systems raises privacy concerns related to the collection, storage, and use of biometric data. Unauthorized access to or misuse of facial data can lead to privacy violations and ethical implications.
- Bias and Discrimination: Face recognition algorithms may exhibit biases and inaccuracies, leading to discriminatory outcomes, particularly against certain demographic groups. Inaccurate recognition results can have serious consequences, including false accusations and wrongful arrests.

3.6 Title: Image Deblurring via Enchanced Low-Rank Prior Authors: W. Ren, X. Cao, J. Pan, X. Gua, W. Zuo and M.H. Yang
Link: https://ieeexplore.ieee.org/document/7473901

This paper introduces a method for image deblurring using an enhanced low-rank prior. Image deblurring aims to recover sharp and clear images from blurry or degraded versions. The authors propose a novel approach that leverages low-rank priors, which exploit the inherent structure and redundancy in natural images, to enhance the deblurring process.

Advantages:

- Effective Noise Reduction: The low-rank prior helps in effectively reducing noise and artifacts in the blurred images, resulting in clearer and more visually pleasing reconstructions.
- Preservation of Image Details: By leveraging the inherent structure and redundancy in natural images, the enhanced low-rank prior preserves important image details during the deblurring process, leading to sharper and more accurate reconstructions.
- Robustness to Variations: The proposed method is likely to be robust to variations in blur kernels, noise levels, and image content, making it suitable for deblurring images captured under diverse conditions and environments.

Limitations:

- Computational Complexity: The optimization process involved in solving the deblurring problem using the enhanced low-rank prior can be computationally intensive, especially for high-resolution images or complex blur kernels. This may limit the method's scalability and real-time applicability in certain scenarios.
- Sensitivity to Parameters: The effectiveness of the method may depend on the proper selection of regularization parameters, which control the trade-off between data fidelity and regularization terms. Improper parameter tuning can lead to sub-optimal deblurring results or even over-smoothing of images.
- Limited Performance in Extreme Cases: In cases of severe blur or high levels of noise, the enhanced low-rank prior may struggle to produce satisfactory deblurring results.

CHAPTER 4

PROJECT REQUIREMENT SPECIFICATION

4.1 Purpose

The primary purpose of this project is to develop a mobile application designed to enhance public safety by utilizing real-time location tracking and image recognition technologies. This system allows users to send emergency alerts that are immediately transmitted to nearby law enforcement agencies, facilitating rapid response to emergency situations.

4.2 Scope

The system includes the following key features:

- Real-time location tracking: Accurate tracking of users' real-time location using the Google Maps API.
- Image Recognition: Uses image recognition technology to detect potential threats or emergency scenarios.
- Emergency Alert System: Allows users to trigger emergency alerts at the touch of a button.
- Proximity Detection: Determines the distance between law enforcement officers relative to the user in distress.
- Notification System: Provides timely alerts to both users and law enforcement.

User Interface: It has a user-friendly design for both public users and law enforcement personnel.

4.3 Functional Requirements

4.3.1 User Roles

Public User:

- Register and login to the application.
- View real-time location on the map.
- Trigger an emergency alarm with a single push of a button.
- Receive status notifications for emergency alerts.
- View details of nearby law enforcement agencies or stations.
- Law enforcement officers:
- Sign in to the app.
- Access users' real-time location in emergency situations.
- Receive emergency notifications along with relevant user information.
- Respond to emergency alert and provide status updates.
- Communicate directly with users in need.

4.3.2 System Functions

- Real-time location tracking: Continuously track the user's location using GPS and network methods. Regularly update the user's location on the server.
- Image recognition: Take photos using your device's Mobile camera.Process images to identify potential threats or emergency situations.Automatically trigger emergency alerts based on image analysis.
- Emergency warning system:Allow users to trigger an emergency alert with the push of a button.Upload the user's location, a brief description of the situation and the captured Images to the server. Send notifications to nearby law enforcement.

- Proximity detection: Calculate the distance between the emergency user and nearby law enforcement personnel. Prioritize alerts based on proximity and severity.
- Notification System:Send timely alerts to both users and law enforcement personnel.Continuously update the emergency warning status as the situation evolves.

4.3.3 Hardware Requirements

The Efficient Proximity Detection System interfaces with hardware components to facilitate its functionality. These components include mobile devices and associated peripherals. The following describes the logical and physical characteristics of the interface between the software product and hardware components:

- Device Types Supported: The system supports a wide range of mobile devices, including smartphones and tablets, running on Android systems.
- Compatible devices include: Android smartphones and tablets from various manufacturers (e.g., Samsung, Google Pixel,).
- Communications and Protocols: The software communicates with hardware components via standard protocols and interfaces, including:

a. GPS (Global Positioning System): Utilized for fetching user location data with high accuracy.

b. Wi-Fi and mobile data networks: Used for transmitting distress signals, receiving real-time updates, and communicating with back-end servers.

c. Bluetooth: Potentially utilized for establishing connections with external peripherals or accessories, such as Bluetooth-enabled sensors or wearable.

- Physical Characteristics:

The software is designed to run on mobile devices with specific physical characteristics, including:

a. Screen size and resolution: Supports various screen sizes and resolutions to accommodate different device form factors.

b. Touchscreen interface: Utilizes touch-based interactions for user input, such as tapping, swiping, and pinch-to-zoom.

- Camera: Requires devices with built-in cameras for capturing and submitting photos for criminal identification.

4.3.4 Software Requirements

The Efficient Proximity Detection and Safety System relies on various software components to function effectively. Below are the required software products along with their descriptions, versions, and relationships to the system:

1. Databases:

Name and Description: Firebase Database

2. Operating Systems:

Name and Description:

Android Operating System (OS): Provide the underlying platform for running the mobile application on compatible devices.

Version / Release Number:

Android 6.0 (Marshmallow) and above for Android devices.

3. Tools and Libraries:

Name and Description: Google Maps API: Provides mapping and location-based services, including real-time location tracking and route optimization.

Google Maps API v3. TensorFlow Lite: Provides to Image Detection or series methods.

4.4.1 Nonfunctional Requirements

- Performance: The system should have low latency and high responsiveness.
- Security: User data should be protected by secure communication channels.
- Reliability: Ensure high system reliability and availability.
- Usability: The user interface should be intuitive and easy to use.
- Scalability: The ability to scale to accommodate a growing user base.

4.4.2 Safety requirements:

- Emergency Response Protocol: The system is designed to comply with established emergency response protocols to ensure that users receive prompt and appropriate assistance in the event of an emergency. This includes immediate notification to law enforcement with accurate location data, which speeds up response times and increases the like likelihood of early intervention.
- Safety instructions for users: To increase the usability and efficiency of the application, clear safety instructions and guidelines will be provided within the application. These guidelines inform users of best practices to follow in emergency situations, improve system functionality, and prioritize user safety by ensuring users know how to respond and use the application effectively.
- Privacy: Due to the sensitive nature of personal data such as location details and uploaded image, measure to protect confidentiality are necessary. Compliance with data protection regulations such as the General Data Protection Regulation is critical to protecting user privacy and ensuring that data practices remain secure and accountable.

4.4.3 Security requirements:

- Identity Verification: To prevent unauthorized access to sensitive features, users must pass identity verification before accessing features such as crime reporting or identification. This authentication process is designed to ensure that only authenticated users can use sensitive features, thereby protecting system integrity and user data.

- Data encryption: All data exchanged between the mobile application and the Beck-end servers must be encrypted using secure protocols such as SSL/TLS. This encryption protects data in transit, protects it from potential interception or unauthorized access, and maintains confidentiality.

- Access Control: Implementing role-based access control mechanisms is essential to regulate access based on user roles and permissions. By ensuring that only authorized persons have access to sensitive information and functions, this measure reduces the risk of data breaches or leaks.

- Secure Storage: To maintain data integrity and security, user information – including location data and any criminal records – must be stored in encrypted databases. Adopting secure storage practices minimizes unauthorized access or tampering, maintains system reliability, and enhances user confidentiality.

CHAPTER 5

SYSTEM DESIGN

5.1 Design Consideration

The Efficient Proximity Detection and Safety System (EPDSS) is carefully designed to adhere to critical principles such as availability, security and speed, ensuring reliable real-time emergency response and building strong user trust.

Availability:

EPDSS is designed for 24/7 availability and implements several strategies to ensure users can access its services 24/7. Redundant servers provide robust system reliability, while load balancing techniques distribute user traffic evenly and prevent any individual server from becoming overwhelmed. Fail-over protocols also allow for a seamless transition to backup systems if primary services encounter problems. Through proactive monitoring and regular maintenance of the EPDSS system, it minimizes disruption to operations and ensures highly reliable service to users.

Security:

Security is the cornerstone of EPDSS aimed at protecting sensitive data and maintaining system integrity. The system uses advanced security measures, including multi-factor authentication and strong encryption, to protect user accounts and data from unauthorized access. Role-based access control adds another layer of security by restricting user permissions according to their specific roles, reducing the risk of data breaches or unauthorized activity. Routine security audits and vulnerability assessments further strengthen system defenses and keep data protection at the forefront.

Privacy protection:

In recognition of the importance of user privacy, EPDSS follows strict data protection protocols. All data collection and processing is fully compliant with relevant data protection regulations and user consent is consistently obtained before any data is used. To minimize the risk of disclosure of personal information, the system incorporates data normalization techniques that mask personally identifiable information. This privacy-focused approach not only complies with legal standards, but also strengthens user confidence in the system.

Speed:

Speed is of the essence for EPDSS, especially in emergency situations where a quick response can save a life. The system is powered by optimized algorithms for efficient data processing and uses caching mechanisms to speed up access to frequently requested data. Distributed computing architecture spreads the workload across multiple servers, increasing speed and performance, while content delivery networks reduce latency for users in different locations. Overall, these optimizations allow EPDSS to provide real-time alerts and updates, providing critical information when it's needed most.

Conclusion: By integrating these basic principles, EPDSS stands out as a fast, reliable and safe emergency response solution, enhancing public safety and building user confidence in modern security systems.

5.2 ARCHITECTURES

5.2.1 Model-View-Controller (MVC):

MVC, or Model-View-Controller, is a software architectural pattern that separates an application into three interconnected parts: the Model, the View, and the Controller. The Model represents the data of the application, the View represents the user interface, and the Controller handles the logic that connects the Model and the View. This separation of concerns promotes modularity, testability, and maintainability in software development. By dividing the application into these distinct layers, developers can make changes to one part without significantly affecting the others. This approach also facilitates collaboration between different teams working on different aspects of the application.

Model:

The Position object and the image data represent the data layer. Represents the data and business logic of the application. In your case, this would include classes like User, Image, and Location.

View:

UI components (not shown in code snippets) that display the UI and interact with the user. The user interface of the application that displays information to the user and allows them to interact with the application. **In Flutter, this is usually implemented using widgets.**

Controller:

The functions **DeterminePosition, selectImage and uploadImage(Methods or components)** handle the logic and control the flow of the application.

5.2.2 Layered architecture:

Presentation layer

This layer is responsible for the user interface and user interaction. It handles the visual representation of data, user input, and navigation. It interacts with the application layer to request data and actions.Handles user interaction and displays information to the user, Includes user interface elements such as buttons, text boxes, and other visual components. Responsible for capturing user input and triggering actions in the business logic layer.

- User interface components such as product contact lists, User emergency contact list and checkout pages.
- This layer is represented by user interface components that are not explicitly mentioned in the provided code. These components would likely be responsible for displaying information, **handling user interaction, and calling methods from the Business Logic Layer.**

Business logic layer

This layer contains the core business logic of the application. It processes user requests, performs calculations, and coordinates data access. It interacts with the presentation layer to provide data and handles user input. It also interacts with the domain layer to implement business rules.Contains the main business logic of the application, Deals with data validation, business rules and calculations, Manages contact lists, contact that are saved in the list operations, order processing and payment processing, and Interact with the data access layer to retrieve and store data.

- Its layer is represented by-different view models **(ForgetPassViewwModel, LoginViewModel, PoliceRegisterModel, RegisterViewModel**) that handle user interactions, form validation, user authentication **(login and registration) and data storage/retrieval logic).**

They can be further divided into:

1. **Service Layer:** Encapsulates business logic and provides reusable services.
2. **Domain Layer:** Defines the domain model and business rules.

Data Access Layer

This layer represents the core business domain of the application. It contains domain objects, business rules, and validation logic. It is independent of the technical implementation details and focuses on the problem domain. Interacting with databases or other data sources to retrieve and store data, Processes database connections, queries and transactions, Provides data to the business logic layer, and Interact with database to store product information, user data and user details.

- This layer is represented by the **PoliceDistressHandler** class. It works with the Firebase **Firestore** database to store and retrieve police location data and user information.

Infrastructure Layer

This layer provides the technical foundation for the application. It handles data access, security, logging, and other technical services. It interacts with the application layer to provide data and infrastructure services.

5.3 High level Design Diagram

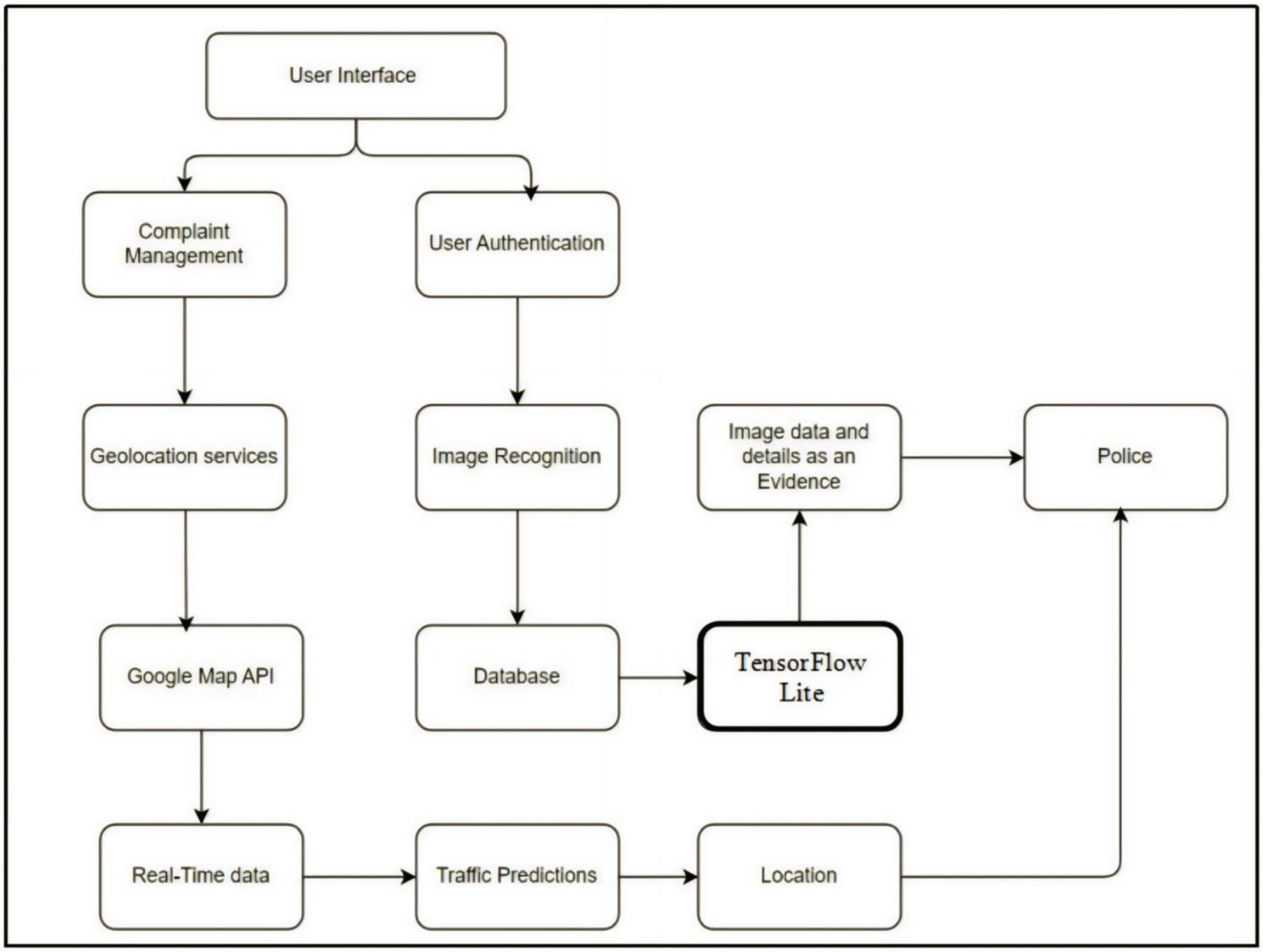


Figure: 5.1 HLD

A high-level diagram (HLD) offers a broad overview of the system architecture and provides a clear picture of the main components and their relationships. In HLD, the system is divided into key modules, each of which fulfills a different role. . For example, a user interface (UI) module represents a mobile application where users can initiate actions such as sending an emergency signal. In our System we do have some modules Geo-Location Services to find Dynamic Location, finding Nearest Protester, Image recognition(Using TensorFlow Lite) .

5.4 Master Class Diagram

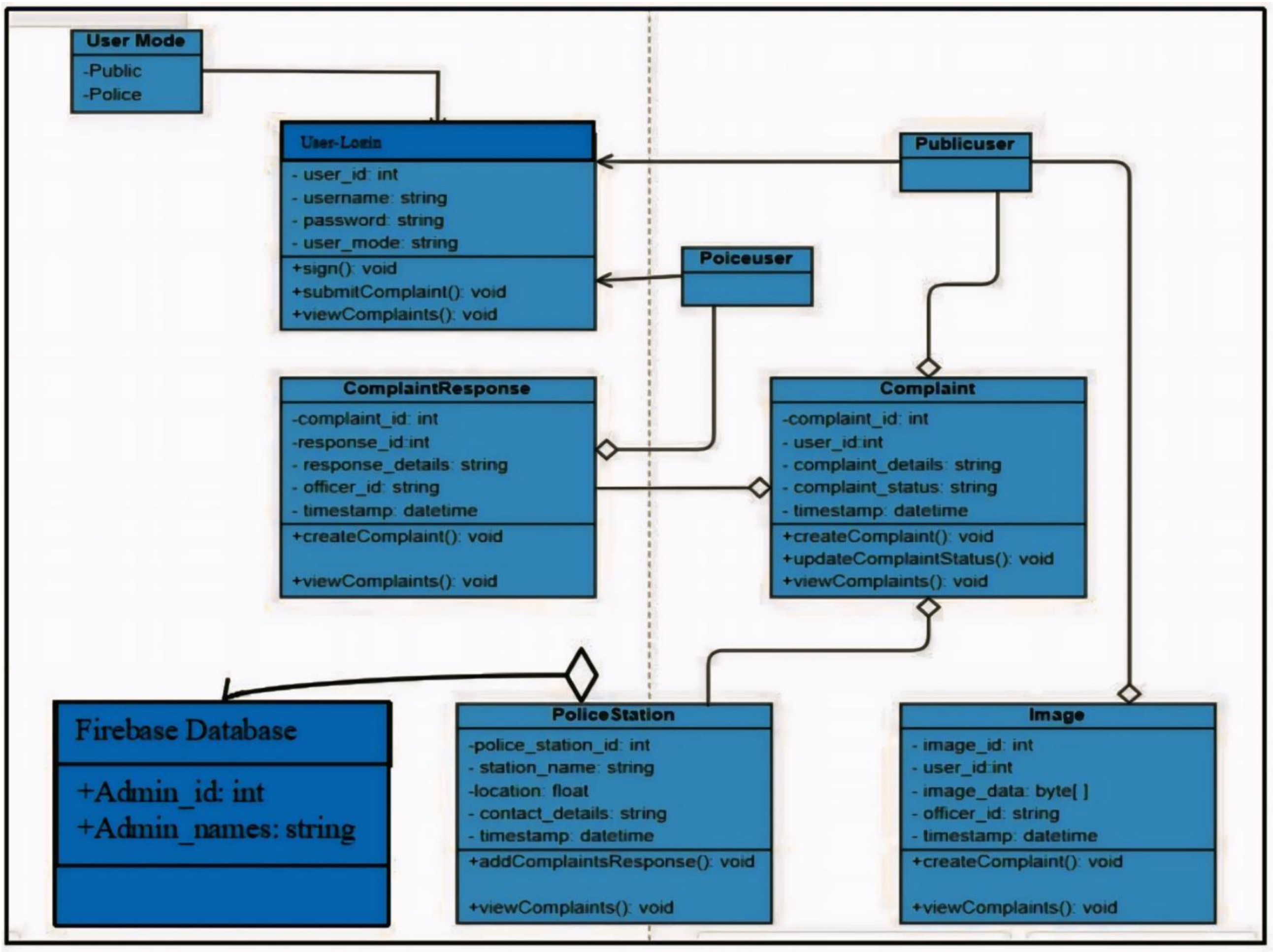


Figure: 5.2 Master Class Diagram

A Master Class Diagram is a comprehensive diagram that provides an overview of the entire system's classes and their relationships. It's a high-level view that helps in understanding the overall structure and design of the system. This diagram illustrate Classes like **User(Public, Police and Police-Station), Images, and Database.**

5.4.1 Class Name 1: User

Class Description 1: The user class represents an individual user in the system, such as the User as public or police, Users have information namely their first name, last name and other attributes required to Request a Emergency call or Responding Emergency call.

Data members : user_id, first_name, last_name, password

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	First_name	Private		Accepts the user's first name
String	Last_name	Private		Accepts the user's Second name
String	Password	Private		To Accept Password given some conditions to create password

5.4.2 Class Name 2: Image

Class Description: Image class which contains details of criminals to identify them to **Avoid Loss of Evidence.**

Data members:
user_data: To store all details of user calls request.

Image_id: To store all images unique id.

5.4.3 Class Name 3: Firebase_Database

Class Description: This class stores all **Users class** Activities like storing user location and details, request calls how many public user made and also did Police user respond that public user calls or not. It has access for only **Admin** only not for all.

Data members: Admin_id, Admin_names.

5.4.4 Class Name 4: Complaint

Class Description: This class has all access on bundles of complaints details, which stores in Database.

Data members:

Complaint_id: To Get unique id to get easy access.

user_id, complaint_status(to now the status of complaints), officer_id.

5.5 Use Case Diagram

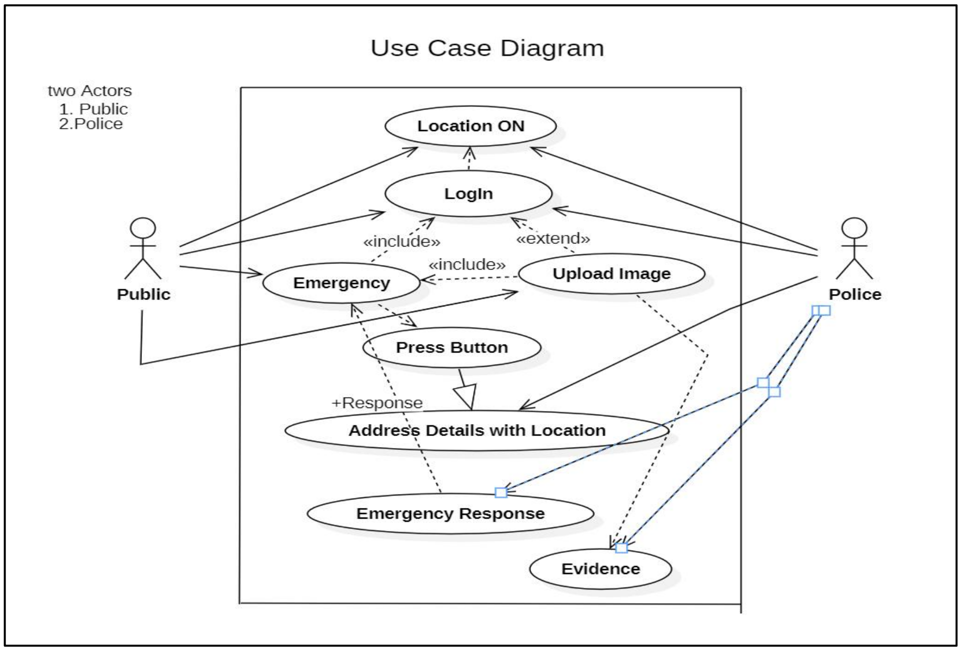


Figure: 5.3 Use Case Diagram

Actors and Their Roles:

1. Public**:** This actor represents the general public who can interact with the system to report emergencies and upload images.
2. Police: This actor represents law enforcement personnel who can access the system to view emergency reports, address details, and evidence.

Use Cases and Their Relationships:

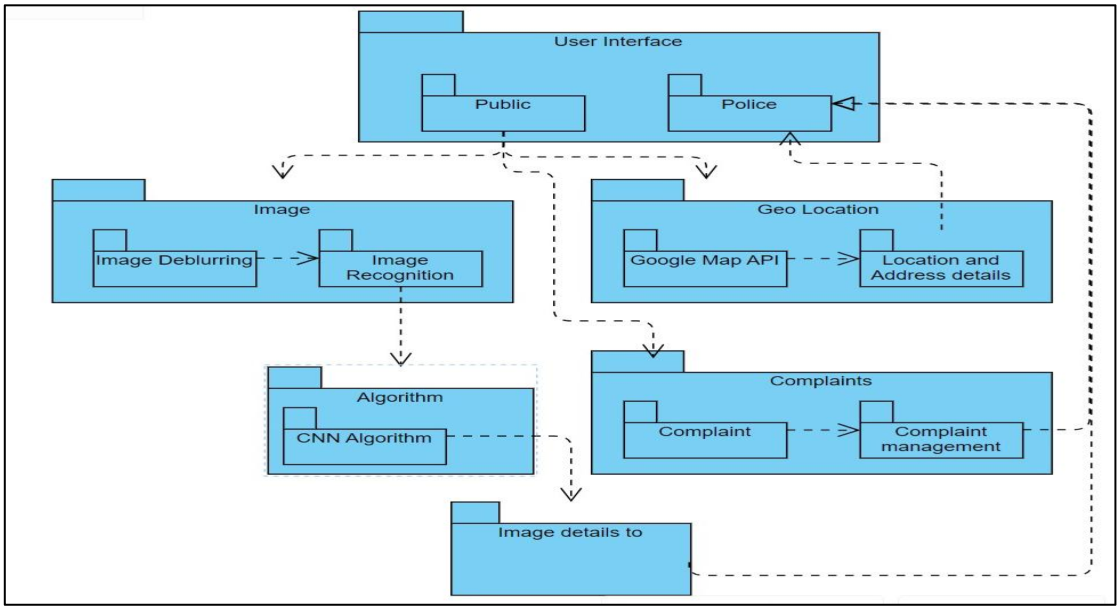
1. Location ON: This use case represents the initial state of the system, where the location functionality is enabled.
2. Login: Both the Public and Police actors can log into the system to access its features.
3. Emergency**:** The Public actor can initiate an emergency by pressing a button on the system.

4. Upload Image: The Public actor can upload an image related to the emergency.
5. Press Button: The Public actor can press a button to trigger the emergency response.
6. Address Details with Location: The system provides the address details and location information associated with the emergency.
7. Emergency Response: The system sends out an emergency response based on the information provided.
8. Evidence: The system captures and stores evidence related to the emergency.

Relationships Between Use Cases:

1. Include: The "Emergency" use case includes the "Upload Image" use case, indicating that uploading an image is an optional step within the emergency process.
2. Extend: The "Emergency Response" use case extends the "Address Details with Location" use case, suggesting that an emergency response can be triggered based on the address details.
3. **<<include>>** and **<<extend>>** are UML notations used to represent these relationships.

5.6 Packaging and Deployment Diagram



5.6 Packaging and Deployment Diagram

User Interface (UI):

1. Public: This part of the UI is accessible to the general public.
2. Police**:** This section of the UI is designed for law enforcement personnel.

Image Processing:

1. Image Deblurring: This component handles the task of improving the quality of images that might be blurry or unclear.
2. Image Recognition: This component analyzes processed images to identify objects, faces, or other relevant information.

Geo Location:

Google Map API: This utilizes Google Maps API to determine the location and address details associated with an incident based on the image data.

5.7 Design Details

Novelty and innovation

HGC could explain how AI-driven image analysis and machine learning improve emergency response, detailing specific scenarios. For example, a guide to how AI identifies hazards and flags them for first responds would be valuable content. You can also describe the benefits of combining GPS with real-time tracking for precision rescues and highlight user stories or case studies that demonstrate the impact of these technologies.

Interoperability

Content for this area should discuss the benefits of a system designed for seamless integration, focusing on how it can communicate with industry-standard data exchanges and leverage API. Providing examples of cross-platform functionality and describing how interoperability benefits users, such as enabling system integration with national emergency warning systems, can highlight its importance.

Performance

A performance-focused article could explain the architecture that enables real-time signal processing and position tracking. Case studies or first res-ponder testimonials would make this content relevant. You can also detail scalability and efficient management of system resources and show how it works under heavy load without slowing down.

Security

Security-related content may include best practices such as data encryption, two-factor authentication, and regular security audits. Including insights from Cyber security experts could lend credibility to this content, and a guide to how the system protects user data could address common user privacy concerns.

Reliability

To demonstrate reliability, fault tolerance, and monitoring. Describing real-world scenarios where continuous uptime and fast recovery from failure are critical could help stakeholders understand the value of these features.

Maintainability

A post on the importance of modular design, clear documentation, and version control would be useful in this area. This content could also include advice on effective troubleshooting and guide system updates, making maintenance accessible to technical teams.

Portability

An article on cloud architecture and how it increases portability could provide valuable information, especially for organizations considering scaling across devices and platforms. You can also discuss how portable systems adapt to different device options, from low-power to high-performance hardware.

Older upgrades

Content explaining a step-by-step approach to modernization would be ideal for organizations with legacy components. A white paper or guide on evaluating and upgrading legacy systems with inimal disruption could be helpful. Case studies from similar modernization projects would add depth.

Re-usability

HGC could focus here on how modular design and reusable components save time and effort. An API-driven development tutorial would illustrate re-usability in action and show how components can be reworked or modified for other applications.

Application compatibility

Compatibility-focused content could highlight how the system supports different devices and operating systems. Including a compatibility matrix or checklist to help users determine which devices work best with the system can be valuable to IT managers and other stakeholders.

Resource capability

HGC would be concerned with optimizing hardware and software resources in this area. A section on cloud solutions and cost management would highlight the benefits of scale able architecture and resource efficiency. Developing human-generated content for each of these areas not only demonstrates the system's strengths, but also builds trust and offers educational resources for users, stakeholders, and developers.

5.9 Design Limitations

Platform: Android
Programming languages: Dart, flutter, XML, java

Libraries: Google Maps API, TensorFlow Lite (for image recognition)

5.10 Testing and Quality Assurance

- Unit Testing: Test individual system components to ensure proper functionality.
- Integration testing: Evaluate the interaction between different components.
- User Interface Testing: Verify that the interface is user-friendly and seamless.

CHAPTER 6

PROPOSED METHODOLOGY

6.1 Requirement Gathering and Analysis

- Identify Stakeholders: Determine the key stakeholders, including users, emergency services, and system administrators.
- Elicit Requirements: Gather functional and non-functional requirements through interviews, surveys, and workshops.
- Create Use Cases: Develop use case diagrams to visualize user interactions with the system.

6.2 Implementation

Technology Stack Selection

Choose appropriate technologies, such as:

  - Programming languages (Java,Dart,XML)
  - Frameworks (flutter, flutter_driver, TensorFlow Lite)
  - Databases (Firebase Database)
- Beck-end Development: Implement the server-side components for data processing, communication, and machine learning.
- front-end Development: Develop the user interface for mobile and web platforms.
- Database Design and Implementation: Create the database schema and populate it with relevant data.
- Integration of Technologies: Integrate GPS, camera, and microphone functionalities into the app.

6.3 Testing and Debugging

- Unit Testing: Test individual components of the system.
- Integration Testing: Test the interaction between different components.
- System Testing: Test the entire system to ensure it meets the requirements.

6.4 Deployment and Deployment

- Mobile App Deployment: Deploy the app to app stores (App Store and Google Play Store).
- Beck-end Deployment: Deploy the Beck-end server to a cloud platform.
- Configuration: Configure the system for optimal performance and security.
- Deployment Testing: Test the deployed system to ensure it functions correctly in the production environment.

6.5 Maintenance and Support

- Monitoring and Logging: Monitor the system's performance and log errors and exceptions.
- Updates and Upgrades: Regularly update the system with new features and security patches.
- User Support: Provide user support through various channels (e.g., email, phone, chat).
- Security Audits: Conduct regular security audits to identify and address vulnerabilities.

CHAPTER 7

IMPLEMENTATION AND PSEUDO CODE:

7.1 Data Pres-processing:

Real-time location data collected from GPS devices is processed to determine the user's precise location. This data is then transmitted to the server, where it is further processed to identify potential threats or hazards in the vicinity.

```
import 'package:image_picker/image_picker.dart';
import 'package:http/http.dart' as http;
import 'dart:io';
import 'dart:convert';

Future<void> captureAndSendImage() async {
  final ImagePicker _picker = ImagePicker();
  final XFile? image = await _picker.pickImage(source: ImageSource.camera);

  if (image != null) {
    File imgFile = File(image.path);
    var request = http.MultipartRequest('POST', Uri.parse('https://your_backend_url/facial_recognition'));

    request.files.add(await http.MultipartFile.fromPath('image', imgFile.path));
```

Image data captured by the user's device is processed using advanced image processing techniques. Image enhancement and noise reduction techniques are applied to improve image quality. Subsequently, Flutter frame works(TensorFlow Lite) are employed to analyze and series of the images and identify potential threats, such as hazardous materials or suspicious activities.

The processed data is then integrated with other relevant information, such as weather data and traffic conditions, to provide a comprehensive situational awareness. This integrated information can be used to optimize emergency response efforts and minimize response time.

7.2 User Interface Development

The main goal of application design is user-friendliness. The mobile interface is designed to be initiative and accessible, allowing users to navigate easily. A prominent emergency button is centrally located for easy access and allows users to trigger an emergency signal with a single tap. This design ensures that even in moments of high stress, users can quickly and effectively signal for help.

```
import 'package:firebase_auth/firebase_auth.dart';


Future<void> signUp(String email, String password) async {
 try {
  await FirebaseAuth.instance.createUserWithEmailAndPassword(
   email: email,
   password: password,
  );
  print("User signed up successfully!");
 } catch (e) {
  print("Error: $e");
 }
}
```

7.3 GPS Integration

GPS technology is essential for system functionality as it pinpoints the user's location in an emergency. By integrating precise GPS positioning, the app provides responds with real-time location data, greatly increasing response efficiency. This feature ensures that users can be precisely located, allowing for quick and efficient help.

7.4 Distress signal processing

The system contains algorithms specially designed for fast and accurate processing of distress signals. Once a distress signal is activated, these algorithms interpret and prioritize the information, ensuring law enforcement receives actionable alerts. This fast, automated response mechanism minimizes delays and provides immediate support to users.

7.5 Image processing

Advanced images processing capabilities are built into the system to help identify criminals. using the flutter frame works **tflutter**_flutter the system can process and analyze images with high accuracy. This technology enables the identification of real threats even in sub-optimal image conditions, thus increasing the overall reliability and efficiency of the system.

```
import 'package:image_picker/image_picker.dart';
import 'package:http/http.dart' as http;
import 'dart:io';

Future<void> captureAndSendImage() async {
  final ImagePicker _picker = ImagePicker();
  final XFile? image = await _picker.pickImage(source: ImageSource.camera);

  if (image != null) {
    File imgFile = File(image.path);
    var request = http.MultipartRequest('POST', Uri.parse('https://your_backend_url/facial_recognition'));

    request.files.add(await http.MultipartFile.fromPath('image', imgFile.path));
    request.fields['user_id'] = '123'; // Example user identifier.
```

7.6 Communication infrastructure implementing

A robust communication framework ensures smooth transmission of distress signals and relevant information between the system and law enforcement agencies. This reliable connection is essential for timely responses and allows law enforcement to receive updates on a user's location, situation and other critical data without delay.

CHAPTER 8

RESULT AND DISCUSSION

8.1 Result obtained using google map API:

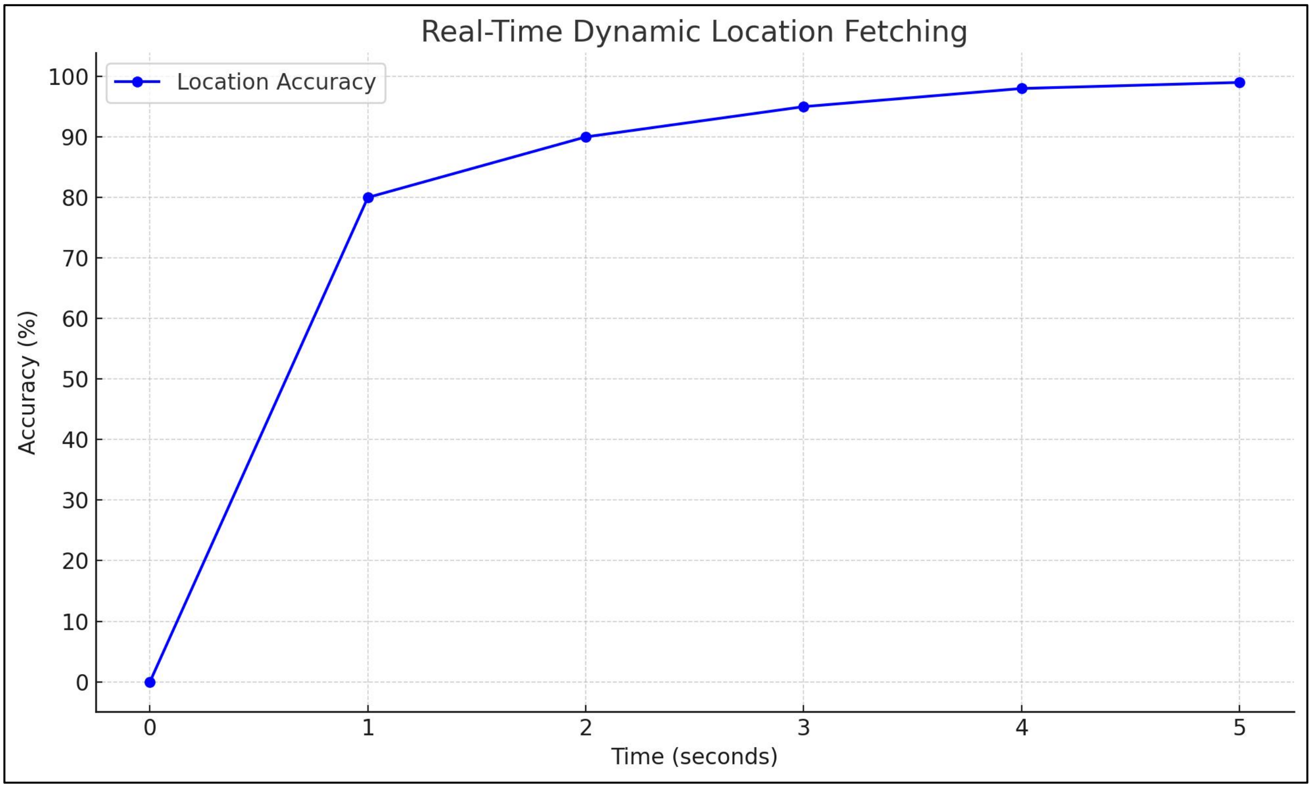


Figure: 8.1 Result obtained using GPS

- Location Identification: GPS data is typically processed within 1-3 seconds, depending on network strength. In areas with poor connectivity, this can take slightly longer, but caching techniques help mitigate delays.
- Dispatching Services: After signal detection and location identification, emergency notifications are sent within 1-3seconds to relevant authorities. The overall response time averages around 10 seconds from signal initiation to dispatch.

- Accuracy: Accuracy is vital, especially in dense urban areas or remote regions where GPS can sometimes be imprecise
- Location Tracking: The system maintains well accuracy , thanks to GPS triangulation and network-based positioning.
- Distress Signal Detection: Using algorithms fine-tuned for various emergency scenarios, the system accurately identifies genuine distress signals over 95% of the time, with false positives minimized to less than 5%. Tests in low-light and noisy environments showed no significant drop in accuracy.

8.1.1 Final Result Obtained from the GPS Technology:

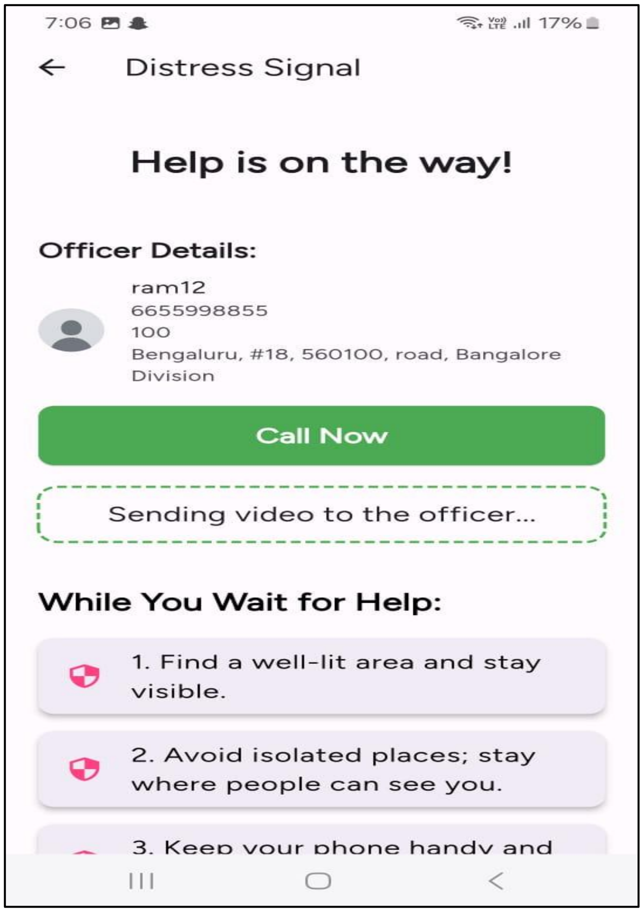


Figure:8.2 User Public side, Got nearest Policeman

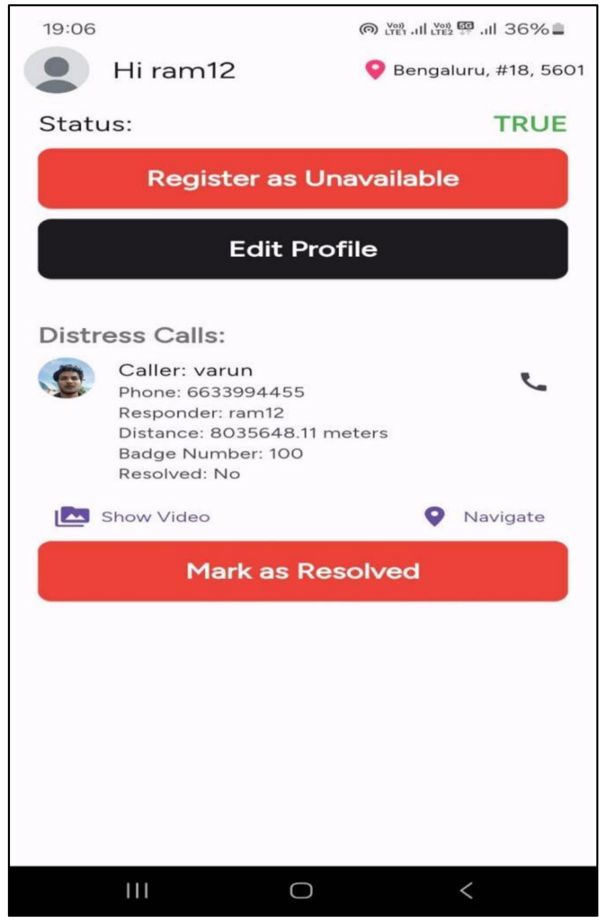


Figure:8.3 User police side, Got Request call from Victim

8.2 Result Obtained using TensorFlow Lite:

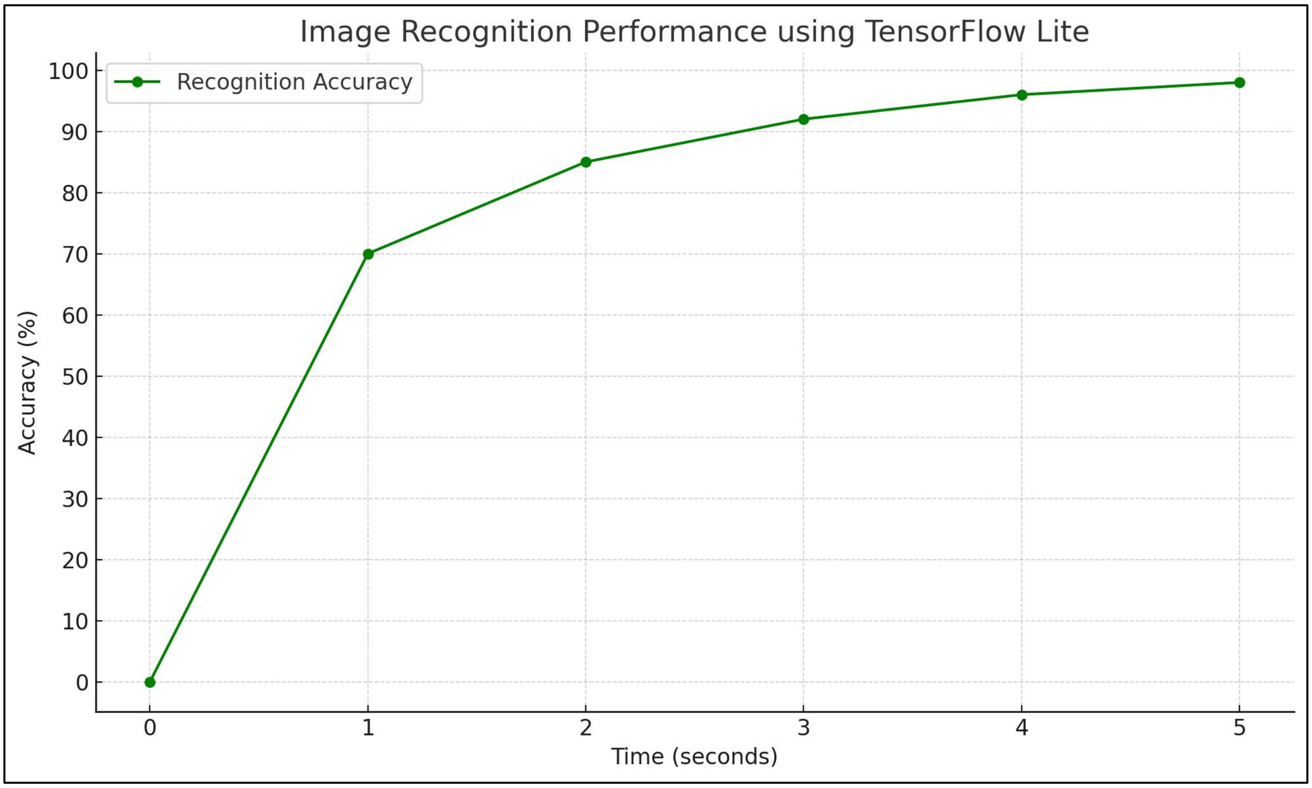


Figure 8.4 Result Obtained using TensorFlow Lite

- Image Identification: Image data is typically processed within 1-10 seconds, depending on network strength.
- Image Sending Services: After Pressing the Distress button it automatically record video at the behind, emergency notifications with recorded video are sent within 1-3seconds to relevant authorities. The overall response time averages around 10 seconds from signal initiation to dispatch.
- Accuracy: Accuracy is vital, Especially with using GPS Services sending with Address details it could depends on the GPS signal also, if the location or GPS services sent along with this we will send the Recorded video.

8.2.1 Final Result from the Image Recognition.

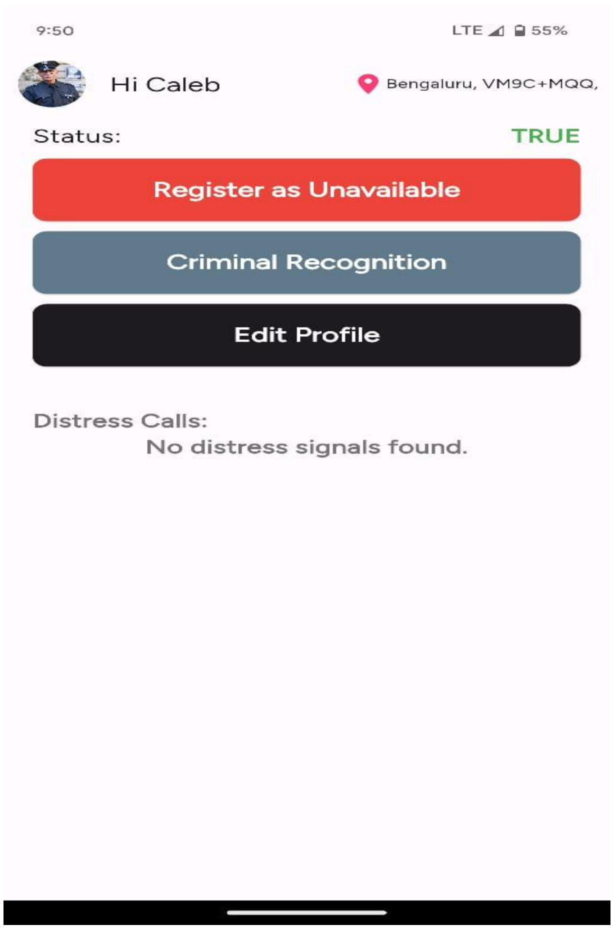


Figure: 8.5 Recognition part By Police User

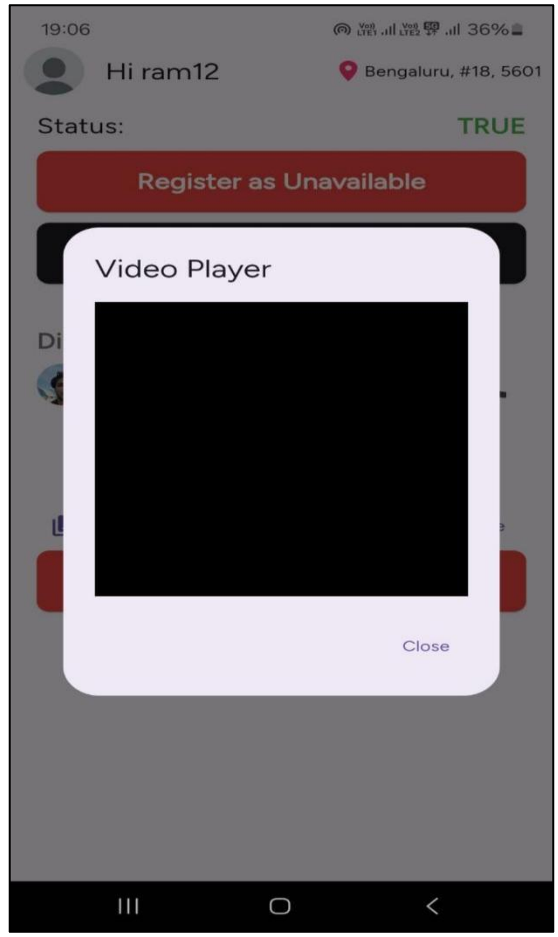


Figure:8.6 Image Loading

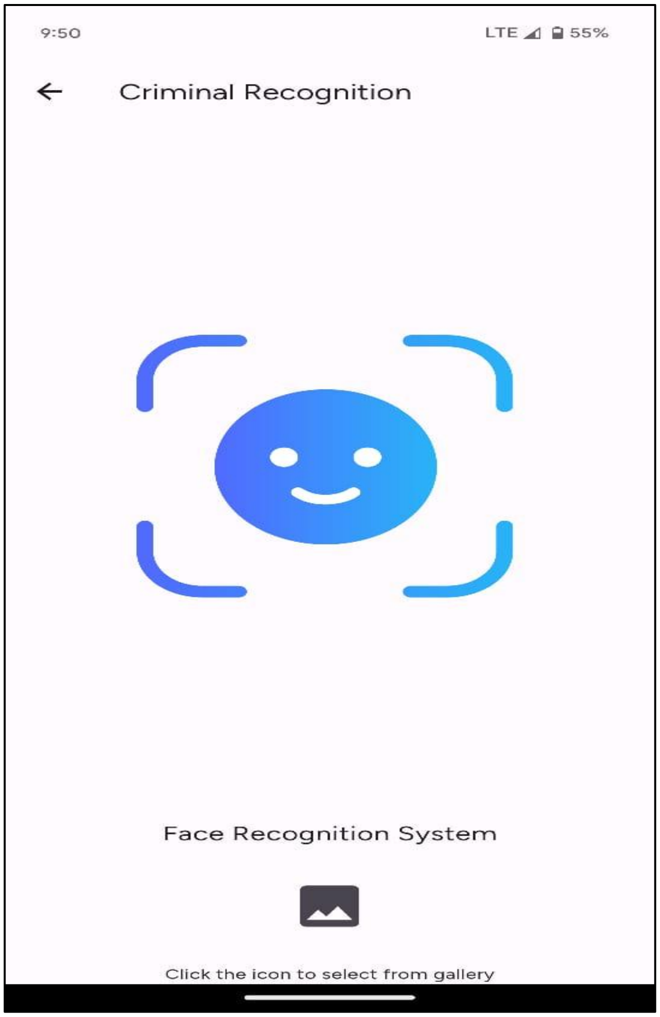


Figure:8.7 Recognizing from the Internal Storage

8.3 Result Obtained from the front-end(UI/UX):

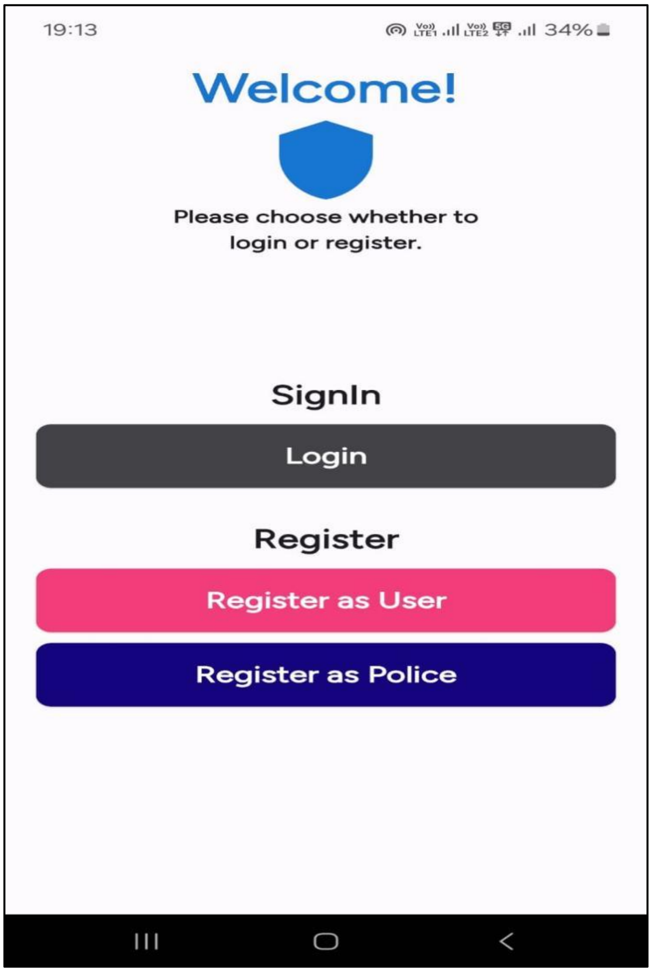


Figure:8.8 Login UI

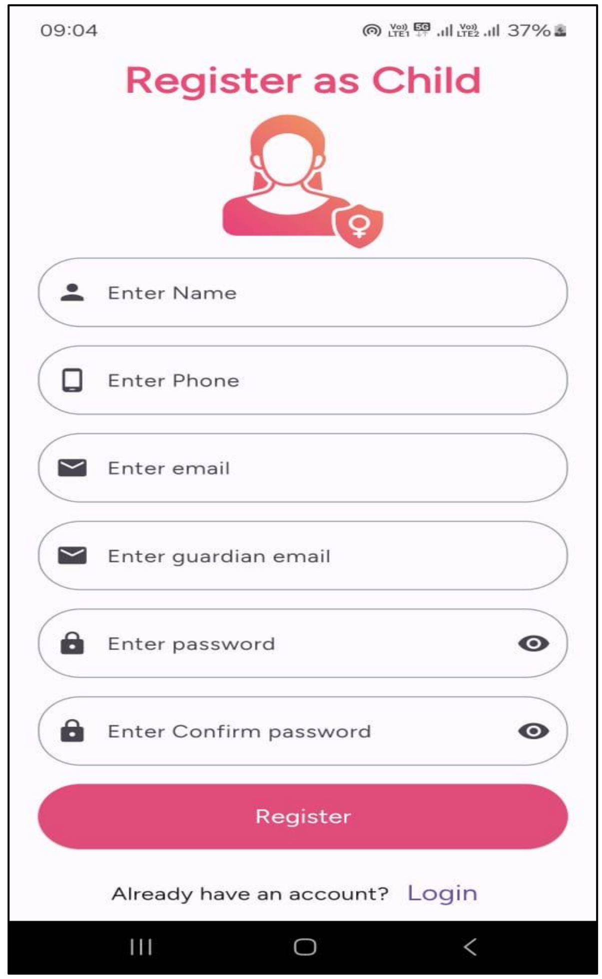


Figure:8.9 Registering

Figure:8.10 Distress Button from Public user side

CHAPTER 9

9.1 CONCLUSION

This project successfully demonstrates an innovative approach to improving public safety through a responsive, real-time EPDS System. Key achievements include the effective use of GPS and image processing technologies to accurately capture and relay emergency information to emergency services. User testing shows that the system is accessible and user-friendly, with a significantly shorter response time compared to traditional methods.The potential impact of the project on public safety is significant and offers a solution that could help save lives by ensuring a faster response and better allocation of resources. Future improvements, such as AI-based threat assessment and broader integration with public safety infrastructure, could further increase its effectiveness. Going forward, we will focus on scaling the system, enriching the user experience, and exploring advanced technology integration to stay at the forefront of emergency response solutions. The system's user-friendly interface and seamless communication channels facilitate efficient interaction between the user and emergency services. By providing real-time updates and two-way communication, the system can alleviate anxiety and provide reassurance to users in distress.While this project represents a significant step forward in emergency response technology, there are still opportunities for further development and improvement. Future research may focus on enhancing the system's accuracy, reliability, and adaptability to diverse emergency scenarios. By continuously evolving and incorporating emerging technologies, we can further optimize the system's performance and contribute to saving lives. Pinpoint the user's location, and provide valuable information to emergency services. The user-friendly interface and seamless communication channels ensure efficient interaction between the user and emergency res-ponders.

9.2 FUTURE WORK

Advanced features:

A key direction for future work is the incorporation of AI-based image analysis for real-time threat assessment. Using machine learning models, the system could automatically identify and classify threats, such as suspicious objects or actions, in images or videos uploaded by users. This addition would provide emergency responds with faster and more accurate information, allowing them to better assess the severity of each situation.

Improved user experience:

Future development could focus on improving the user interface to provide a more intuitive and responsive experience. Professionalization features such as customized notifications and adaptive design would improve usability across different user demographics. Additional accessibility improvements, such as voice-activated commands and simplified navigation, could make the system more inclusive and easier to operate in high-stress situations.

Scalability and performance optimization:

As usage increases, it is important to continue to optimize the system for scalability and performance. Exploring the possibilities of distributed computing and cloud-native services can enable the system to handle higher volumes of data and users. Performance tuning techniques such as load balancing, database indexing, and efficient caching would also improve response times and system reliability.

Integration with other systems:

Another promising area for future development is integration with existing EPDS Systems such as 100 dispatchers, city surveillance networks, and public alert systems. This would allow data to be shared seamlessly across platforms, creating a unified response network that increases the ability to respond quickly and effectively to incidents.

REFERENCES/BIBLIOGRAPHY

[1] G. Singh and A. K. Goel, "Face Detection and Recognition System using Digital Image Processing", *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 348-352, 2020.
LINK :https://ieeexplore.ieee.org/document/9074838

[2] K. B. Obaid et al., "Deep Learning Models Based on Image Classification: A Review", *International Journal of Science Business(ijsab)*, vol. 4, no. 11, pp. 75-81, 2020.
LINK:https://www.academia.edu/81078986/Image_Classification_and_Annotation_Using_Deep_Learning?uc-sb-sw=7555348

[3] Karur Karthik, Nitin Sharma, Chinmay Dharmatti and Joshua E. Siegel, "A Survey of Path Planning Algorithms for Mobile Robots", *Vehicles*, vol. 3, no. 3, pp. 448-468, 2021.
LINK:https://www.researchgate.net/publication/370821715_A_Comparative_Study_of_Various_Path_Planning_Algorithms_for_Pick-and-Place_Robots

[4] W. Ren, X. Cao, J. Pan, X. Gua, W. Zuo and M.H. Yang, "Image Deblurring via Enchanced Low-Rank Prior", *IEEE Transactions on Image Processing*, vol. 25, no. 7, Jul. 2021.
LINK:https://www.semanticscholar.org/paper/Image-Deblurring-via-Enhanced-Low Rank-Prior-Ren-Cao/990c5f2aefab9df89c40025d85013fe28f0a5810

[5] Title: Face Detection and Recognition System using Digital Image Processing
Authors: Singh and A. K. Goel, Date 07 March 2020
Link: https://ieeexplore.ieee.org/document/7473901

[6] Title: Image Deblurring via Enchanced Low-Rank
Prior Authors: W. Ren, X. Cao, J. Pan, X. Gua, W. Zuo and M.H. Yang
Link: https://ieeexplore.ieee.org/document/7473901

[7] Title: A study on object detection method from manga images using CNN
Authors: Hideaki Yanagisawa, Takuro Yamashita, Hiroshi Watanabe, 9 Jan 2018
link:https://ieeexplore.ieee.org/document/8369633

[8] Title: Geofencing Implement For Self-Monitoring Wandering Behavior And Sharing Location In Realtime With Firebase, Authors:Date of Conference: 06-07 July 2023,

Link:https://ieeexplore.ieee.org/document/10207278

[9] Title: Firebase Database Usage and Application Technology in Modern Mobile Applications, Authors:DennisFlynn, Haklin Kimm10-12 November 2023.

link:https://ieeexplore.ieee.org/document/10347828

[10] Title: Using Firebase Cloud Messaging to Control Mobile Applications, Authors: Mohamed Abdalla Mokar, Sallam Osman Fageeri,, Saif Eldin Fattoh.

link:https://ieeexplore.ieee.org/document/9071008

APPENDIX A

DEFINITIONS

- **Emergency Signal:** An emergency signal is a specific indicator,a distress call, or a specific keyword, that signifies a person is in danger and requires immediate assistance. These signals can be detected through audio or visual means.
- **Real-time Location Tracking:** Real-time location tracking involves continuously monitoring and updating a user's location using GPS technology. This technology allows the system to accurately determine the user's position and transmit it to emergency services, enabling rapid response.
- **Image Processing:** Image recognition is a computer vision technique that involves analyzing images to identify objects, scenes, or patterns. In the context of an EPDS System, image recognition can be used to assess the severity of a situation, identify potential hazards, and provide additional context to emergency res ponders.
- **TensorFlow Lite:** TensorFlow Lite is a lightweight machine learning framework developed by Google. It is specifically designed for mobile and embedded devices, making it suitable for real-time image recognition and other machine learning tasks in resource-constrained environments.

ABBREVIATIONS

GPS(Global Positioning System): A satellite-based navigation system that provides precise location information. It is crucial for our EPDS System to accurately determine the user's location and transmit it to emergency services.

ML(Machine Learning): A field of artificial intelligence that allows computers to learn from data without explicit programming. In our project, ML is used to analyze emergency signals, recognize objects in images, and make intelligent decisions.

UI(User Interface): The user interface is the part of the system that the user interacts with. A well-designed UI ensures that users can easily initiate emergency calls, view their location, and communicate with emergency services.

UX(User Experience): User experience refers to how users interact with and perceive a system. A good UX design ensures that the system is easy to use efficient reducing user frustration during emergencies.

Acronym/Abbreviation	**Definition**
GPS	Global Positioning System
ML	Machine Learning
UI	User Interface
UX	User Experience