W-Secure: Women Safety App

Abstract

In today's world, women's safety is a critical issue that requires innovative technological solutions to ensure protection. This project presents a Women Safety App, **W-Secure**, that utilizes **video recording** and **mobile application-based SOS features** to monitor and analyze a woman's surroundings in real-time.

The project employs **AI-based video analytics (computer vision)** to detect potential threats, such as **harassment, violence, or distress signals**, by analyzing **body language, facial expressions, and unusual activities** like **men surrounding women suspiciously**.

Upon detecting a threat, the app will automatically:

- Alert **emergency contacts** added by the user.
- Notify **nearby mobile application users** with an ALERT.
- **Dial the Women Helpline** along with location details.

Additionally, the system integrates **AI algorithms** (computer vision and deep learning) to **improve threat detection accuracy over time**. By utilizing **real-time surveillance, AI-based threat detection, mobile SOS features, and automated emergency response**, this app aims to **create a safer environment for women**.

Methodology

1. Android App Development

Objective: Build a security-focused Android app with discreet emergency features.

Key Features & Implementation:

- **Location Sharing:**
  - Use **GeoLocator** for real-time GPS tracking.
- **Helpline Calling:**
  - Pre-set emergency numbers (e.g., **100, 1091**).
  - **One-tap calling** without confirmation pop-ups.
- **Stealth Video Recording & Storage:**
  - Use Android's **MediaRecorder API** to **record video without opening the camera UI**.
  - Upload recordings directly to **cloud storage (Firebase Storage)** to prevent tampering.
- **Safe Spaces Marking:**
  - Utilize **Stadia Maps API** to mark verified safe places (e.g., **police stations, hospitals, malls**).
- **SOS Button with Live Location:**

    - Implement a **persistent floating SOS button**.
- **Phone Shutdown Prevention:**
    - Use **DeviceAdminReceiver and DevicePolicyManager** to restrict power-off options.
    - Implement a **background service** to restart the app in case of forced closure.

2. AI Models for CCTV-Based Threat Detection

Objective: Detect potential threats by analyzing movement patterns and gestures.

AI Models & Techniques:

1. **Human Detection & Tracking (Multi-Object Tracking):**
    - Use **YOLOv8 or Mask R-CNN** for real-time person detection.
    - Implement **DeepSORT** for tracking multiple individuals.
2. **Pattern Recognition (Women in Distress Detection):**
    - Identify anomalies like **a single woman being followed by multiple men**.
    - Use a **threshold system** (e.g., **3+ men following for X seconds** triggers an alert).
3. **Gesture & Behavioral Analysis:**
    - Use **OpenPose or MediaPipe** to analyze **body posture and movements**.
    - Train an **LSTM-based CNN model** to detect:
        - **Timid or defensive gestures** (e.g., arms crossed, looking back frequently).
        - **Sudden movements** like running or erratic walking.
    - Compare detected behavior with **predefined distress patterns**.

3. Backend & Cloud Infrastructure

- **Cloud Storage:** Firebase for video storage.
- **Database:** Firebase Firestore for user & emergency data.

4. Development Phases

Phase 1: Research & Planning

- **Understanding User Requirements:**
    - Define key features like **SOS alerts, live location tracking, Safe Spaces, and video recording**.
- **Identifying AI Use Cases:**
    - **Crowd Analysis:** Tracking **the number of men following a woman**.
    - **Gesture Recognition:** Recognizing **distress gestures (timid, running, etc.)**.
- **Finding & Collecting Datasets:**

    - **Crowd Analysis:** Datasets like **CrowdHuman, MOT17** for tracking individuals.
    - **Gesture Recognition:** Datasets like **Jester, NTU RGB+D** for identifying distress-related gestures.

Phase 2: AI Model Development

- **Choosing the Best AI Model:**
    - **Mask R-CNN** for detecting and segmenting individuals.
    - **YOLOv8 or EfficientDet** for fast person detection and tracking.
    - **LSTM/RNN-based gesture recognition** for distress detection.
- **Training & Fine-Tuning the Model:**
    - Train AI models on datasets using **PyTorch/TensorFlow**.
    - Fine-tune the models on **real-world surveillance videos**.
- **Testing AI Models:**
    - Validate models with **test datasets**.
    - Optimize **model accuracy and reduce false positives**.

Phase 3: Mobile Application Development

- **Setting Up the Flutter Project:**
    - Configure **Firebase** for authentication, real-time database, and storage.
    - Implement **Geolocator** for live location tracking.
- **Building Core Features:**
    - **Safe Spaces Mapping** – Show Safe Spaces using **Google Maps**.
    - **SOS Alert System** – Implement an **SOS button** to send live location to emergency contacts.
    - **Stealth Video Recording** – Capture and upload video **without opening the camera UI**.
    - **Blocking Phone Shutdown** – Prevent forced shutdown in distress situations.

Phase 4: AI Integration in CCTV Systems

- **Deploying Models on Edge Devices:**
    - Convert AI models to **TFLite/ONNX** for **real-time CCTV processing**.
    - Deploy on edge devices like **NVIDIA Jetson or Raspberry Pi**.
- **Testing AI in Real-World Scenarios:**
    - Evaluate AI performance on **CCTV footage**.
    - Improve **detection speed and accuracy**.

Phase 5: System Testing & Improvement

- **Real-World Testing:**
    - Validate the app in **real-life distress scenarios**.
    - Test AI models on **live surveillance feeds**.
- **Performance Optimization:**
    - Reduce **app latency** and **AI processing time**.
    - Enhance **UI/UX** for better usability.
- **Feedback & Iteration:**
    - Gather **user feedback** and refine security features.

    - Improve **AI accuracy and reduce false alerts**.

Phase 6: Deployment & Maintenance

- **Launch the App on Google Play Store:**
    - Ensure compliance with **security and privacy policies**.
- **AI Model Deployment on CCTV Networks:**
    - Install models in **public places** for **distress detection**.
- **Continuous Monitoring & Updates:**
    - Improve AI models with **new datasets**.
    - Update the app with **new security features**.

Team:

Mayank Sharma E22CSEU0839

Moksh Mehndiratta E22CSEU1173

Shivansh Chauhan E22CSEU0814

Rakesh Sharma E22CSEU0823