SIH-TDD

Technical Design Document (TDD)

Title: Women Safety Analytics – Technical Design

---

1. Overview:

This document presents the technical design for the Women Safety Analytics system, detailing the architecture, components, data flow, and technology stack. The solution leverages advanced computer vision techniques, machine learning algorithms, and real-time data processing to detect potential threats and ensure women's safety in public spaces.

2. System Architecture:

- **High-Level Architecture:**
  The system is designed using a modular architecture that includes several key components:
  - **Data Collection Layer:** Utilizes surveillance cameras and sensors to capture real-time video feeds.
  - **Processing Layer:** Applies machine learning models for person detection, gender classification, and anomaly detection.
  - **Alert and Notification Layer:** Triggers alerts and notifications to law enforcement based on predefined rules.
  - **Data Storage Layer:** Stores video footage, metadata, and generated alerts for future analysis and reporting.
  - **User Interface Layer:** Provides a dashboard for law enforcement to view live feeds, alerts, and analytical reports.

3. Component Details:

1. **Data Collection:**
   - **Surveillance Cameras and Sensors:**
     - Capture video data in real-time.
     - Minimum resolution: 1080p for clear detection.
     - Equipped with night vision and weather-resistant capabilities to ensure uninterrupted data capture.
2. **Analytics Engine:**
   - **Person Detection and Gender Classification:**
     - **Technology:** OpenCV, Keras, TensorFlow

        - **Model:** Deep learning-based object detection models (e.g., YOLO, SSD) fine-tuned for person detection.
        - **Gender Classification Model:** CNN (Convolutional Neural Network) trained using the VIRAT dataset to classify detected individuals by gender.
    - **Gesture Recognition:**
        - **Technology:** TensorFlow, OpenCV
        - **Model:** Pre-trained models for recognizing specific gestures indicating distress (e.g., waving, defensive postures).
    - **Anomaly Detection:**
        - **Algorithm:** Custom rule-based logic and anomaly detection algorithms to identify suspicious activities, such as a lone woman at night or a woman surrounded by men.
    - **Current Status:** Training of the models is still ongoing to improve accuracy and reduce false positives/negatives.
3. **Alert Generation Module:**
    - **Description:**
        - Generates alerts based on predefined rules, such as detecting a lone woman at night or suspicious gestures.
        - Communicates alerts to law enforcement via SMS, email, or push notifications.
    - **Technology:** Python-based backend service integrated with messaging APIs.
4. **Hotspot Analysis Module:**
    - **Description:**
        - Continuously analyzes past data to identify hotspots where incidents are more likely to occur.
    - **Technology:** Python for data analysis, TensorFlow for predictive modeling.
5. **User Interface:**
    - **Dashboard:**
        - Displays real-time alerts, gender distribution, hotspot analysis, and video feeds.
    - **Technology:** Web-based interface using HTML5, CSS, JavaScript (React.js for the frontend), and Django or Flask for the backend.

4. Data Flow:

- **Real-Time Data Processing Pipeline:**
    1. **Data Ingestion:**
        - Surveillance cameras continuously feed real-time video data to the processing layer.
    2. **Pre-Processing:**
        - Video data is pre-processed (resizing, normalization) using OpenCV before being fed into the detection models.
    3. **Model Inference:**
        - The pre-processed data is analyzed by machine learning models (built using Keras and TensorFlow) for person detection, gender classification, and gesture recognition.
    4. **Alert Generation:**

    - Based on the model outputs and pre-defined rules, the system triggers alerts, which are communicated to law enforcement via the alert module.
5. **Storage and Reporting:**
    - Processed data, alerts, and metadata are stored in a secure database for future analysis, auditing, and reporting.

5. Technology Stack:

- **Computer Vision:**
    - **OpenCV:** For video capture, image pre-processing, and basic computer vision tasks.
- **Machine Learning and Deep Learning:**
    - **Keras with TensorFlow Backend:** For building and training deep learning models (CNNs) for person detection, gender classification, and gesture recognition.
    - **Dataset:** VIRAT (Video and Image Retrieval and Analysis Tool) dataset used for training models; contains annotated videos to detect various activities, gestures, and classify people by gender.
- **Backend and Data Processing:**
    - **Python:** For implementing machine learning algorithms, data analysis, and backend logic.
    - **Django/Flask:** For the backend server handling web requests and integration with the alert module.
- **Frontend:**
    - **React.js:** For building a dynamic and responsive web dashboard.
    - **HTML5/CSS/JavaScript:** For UI/UX design.
- **Database and Storage:**
    - **PostgreSQL/MySQL:** For storing metadata, alert logs, and user data.
    - **Cloud Storage (AWS S3/Azure Blob):** For storing video feeds and large datasets.
- **Alerting and Notifications:**
    - **APIs (e.g., Twilio, SendGrid):** For sending SMS, emails, and push notifications.

6. Security Considerations:

- **Data Encryption:**
    - All data in transit and at rest should be encrypted using AES-256.
- **Access Control:**
    - Role-based access control (RBAC) to restrict access to sensitive data and functionalities.
- **Privacy Compliance:**
    - Ensure compliance with local and international data protection regulations (e.g., GDPR, CCPA).

7. Testing and Validation:

- **Testing Strategy:**

    - **Unit Testing:** For individual components like the detection and classification models.
    - **Integration Testing:** To verify the interaction between different modules (e.g., data ingestion and alert generation).
    - **User Acceptance Testing (UAT):** To ensure the system meets user requirements and is easy to use by law enforcement personnel.

8. Current Status and Next Steps:

- **Current Status:**
    - The training process of the models is still undergoing to enhance accuracy and reduce the rate of false positives/negatives.
    - Initial testing shows promising results, but further fine-tuning and optimization are required.
- **Next Steps:**
    - Complete model training and validation.
    - Integrate all system components.
    - Conduct extensive testing in real-world scenarios.
    - Deploy the system in a pilot environment.
