SecureShe - Real Time Women Safety Alert System

Submitted for partial fulfillment of the requirements

for the award of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

by

M.SAI DEEPTHI - 21BQ1A6142

C.LEELA MANJUNATH - 21BQ1A6115

N.PRASANNA KUMAR - 22BQ5A6104

M.REVANTH - 21BQ1A6141

Under the guidance of

Mrs. N. NALINI KRUPA

Assistant Professor

VASIREDDY VENKATADRI
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

(B. Tech Program is Accredited by NBA)

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTU Kakinada, Approved by AICTE

Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified

NAMBUR (V), PEDAKAKANI (M), GUNTUR – 522 508

Tel no: 0863-2118036, url: www.vvitguntur.com

March-April 2025

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTUK, Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:20008 Certified
Nambur, Pedakakani (M), Guntur (Gt) -522508

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

This is to certify that this **Project Report** is the Bonafide work of **Ms. M.Sai Deepthi, Mr.C.Leela Manjunath, Mr.N.Prasanna Kumar,Mr.M.Revanth** bearing Registration.No. **21BQ1A6142, 21BQ1A6115, 22BQ5A6104, 21BQ1A6141** respectively who had carried out the project entitled **" SecureShe-Real Time Women Safety Alert System "** under our supervision.

Project Guide	**Head of the Department**
(Mrs.N.Nalini Krupa, Assistant Professor)	(Dr. K. Suresh Babu , Professor)

Submitted for Viva voce Examination held on _____

Internal Examiner	**External Examiner**

DECLARATION

We, Ms. Myla Sai Deepthi, Mr. Chimirela Leela Manjunath, Mr. Namdigam Prasanna Kumar ,Mr. Munagala Revanth hereby declare that the Project Report entitled "**SecureShe-Real Time Women Safety Alert System**" done by us under the guidance of Mrs. N.Nalini Krupa, Assistant Professor, CSE- Artificial Intelligence & Machine Learning at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfillment of the requirements for the award of Bachelor of Technology in Artificial Intelligence & Machine Learning .The results embodied in this report have not been submitted to any other University for the award of any degree.

DATE :

PLACE : Nambur

SIGNATURE OF THE CANDIDATE (S)

Myla Sai Deepthi,

Chimirela Leela Manjunath,

Namdigam Prasanna Kumar,

Munagala Revanth

ACKNOWLEDGEMENT

We take this opportunity to express my deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and helped me towards the successful completion of this project work.

First and foremost, we express my deep gratitude to **Sri. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express my sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express my sincere gratitude to **Dr. K. Suresh Babu**, Professor & HOD, Computer Science Engineering – Artificial Intelligence & Machine Learning Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express my sincere gratefulness to our Guide **Mrs.N.Nalini Krupa**, Assistant Professor, CSE-Artificial Intelligence & Machine Learning for her insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to express our sincere heartfelt thanks to our Project Coordinator **Mrs. N.Nalini Krupa**, Assistant Professor, CSE-Artificial Intelligence & Machine Learning for her valuable advices, motivating suggestions, moral support, help and coordination among us in successful completion of this project.

We would like to take this opportunity to express my thanks to the **Teaching and Non-Teaching** Staff in the Department of Computer Science Engineering -Artificial Intelligence and Machine Learning, VVIT for their invaluable help and support.

Name (s) of Students

Myla Sai Deepthi,

Chimirela Leela Manjunath,

Namdigam Prasanna Kumar,

Munagala Revanth

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

ML	Machine Learning
DL	Deep Learning
DFD	Data Flow Diagram
YOLOv11	You Only Look Once (Version 11.0)
AI	Artificial Intelligence
CV	Computer Vision
SSD	Single Shot MultiBox Detector

ABSTRACT

SecureShe-Real Time Women Safety Alert System is a cutting-edge, AI-powered solution that leverages real-time monitoring to detect and prevent potential threats to women's safety. By incorporating computer vision, deep learning, and gesture recognition technologies, this software continuously analyses environments to detect suspicious or unsafe situations, particularly for women. It employs a combination of person detection, gender classification, anomaly detection, and predictive analytics to identify potential threats and trigger timely alerts.Existing systems for women's safety include CCTV surveillance, mobile safety apps, AI-powered anomaly detection, and public safety hotlines. CCTV and AI systems often provide reactive monitoring, requiring manual intervention, while mobile apps and hotlines depend on victims actively requesting help, which may not always be possible. These solutions lack specific focus on detecting threats to women, relying more on generic anomaly detection or post-incident response.Our proposal, focuses on proactive real-time threat detection using AI-powered surveillance and advanced analytics specifically designed for women's safety. By leveraging computer vision, deep learning, and gesture recognition, the system continuously monitors public spaces to detect individuals and classify gender. It provides insights into gender distribution and identifies potentially unsafe situations, such as a lone woman at night or a woman surrounded by men. The software also recognizes distress gestures (SOS) and triggers alerts without requiring manual input, allowing for faster law enforcement response. SecureShe-Real Time Women Safety Alert System includes the following functionalities 1. Person detection along with Gender Classification 2. Gender Distribution : Count the number of men and women present in the scene 3. Identifying a Lone Woman at Night time 4. Detection of a Woman Surrounded by Men 5. Recognizing SOS situation through gesture analytics.6.Weapon detection.

KEYWORDS: Computer Vision, Deep Learning, Real-Time Monitoring,YOLOv11Detection ,Gesture Recognition, Mediapipe , Gender Classification ,Person Detection, Weapon Detection ,SOS Gesture Detection ,OpenCV ,Machine Learning.

CHAPTER 1

INTRODUCTION

1.1 Background of the Project

Women's safety has been a growing concern worldwide, with increasing crime rates in public places, workplaces, and even homes. According to the National Crime Records Bureau (NCRB), crimes against women in India increased by 15% in recent years, highlighting the urgent need for advanced surveillance and real-time threat detection. Traditional safety measures such as manual monitoring and emergency helplines often fail due to delayed responses, lack of real-time monitoring, and the inability to predict threats proactively.

With advancements in Artificial Intelligence (AI) and Machine Learning (ML), security solutions have evolved to become more proactive rather than reactive. AI-driven computer vision and deep learning models can analyze CCTV footage in real time to detect potential threats, classify gender distribution in a crowd, identify dangerous situations, and send immediate alerts. These technologies can significantly reduce response time and enhance preventive safety measures.

Real-World Applications of AI & ML in Women's Safety

- **AI-Powered CCTV Surveillance:** AI-based object detection models like YOLO and OpenCV help identify weapons, track unusual activities, and detect overcrowding around women.
- **Gesture Recognition for SOS Alerts:** AI systems can recognize emergency hand gestures and trigger automatic alerts to authorities.
- **Voice-Based Threat Detection:** NLP-based models analyze distressed voices and keywords in public spaces or helpline calls to initiate emergency responses.
- **Location-Based Smart Alerts:** AI integrates with GPS and geolocation APIs to send alerts to nearby police stations and security teams.
- **Wearable AI Devices:** Smart AI-powered wearables can detect sudden movement patterns (like struggling or running) and send distress signals.

1.1.1 Women Safety in Public Spaces

Ensuring the safety and security of women in public spaces is a critical aspect of creating an inclusive, equitable, and progressive society. Public spaces such as streets, parks, public transport, markets, and educational institutions are essential for women to participate fully in social, economic, and cultural life. However, in many regions, women face significant safety concerns, including harassment, stalking, assault, and other forms of gender-based violence, which restrict their freedom of movement and participation.

Studies and surveys globally have shown that many women alter their behavior-such as avoiding certain areas, changing routes, or limiting travel during certain times due to fear of harassment or attack. This constant vigilance not only affects their mental well-being but also imposes social and economic constraints, limiting their opportunities and overall quality of life.

While traditional safety measures like increased police patrolling, CCTV surveillance, and public awareness campaigns contribute to women's safety, these measures often tend to be reactive rather than proactive. Many incidents go unreported or unnoticed due to the limitations of manual monitoring and delayed response times.

Technological advancements, especially in the fields of Artificial Intelligence (AI) and Computer Vision, have paved the way for smarter, real-time safety solutions. These technologies can significantly enhance the effectiveness of public surveillance systems by detecting and predicting threats, enabling immediate alerts and faster response. Integrating AI with existing infrastructure can shift safety strategies from post-incident action to real-time threat prevention, empowering law enforcement agencies and communities to create safer environments for women.

Specifically designed AI-based safety systems can analyze crowd dynamics, identify risky gender ratios, recognize distress gestures, and detect weapons enhancing situational awareness and allowing law enforcement to respond swiftly, thereby preventing escalation and ensuring a safer public environment for women.

In this context, AI-powered women safety analytics systems play a crucial role in bridging the gap between surveillance and safety by providing continuous, intelligent monitoring specifically designed to recognize and address threats to women's security in public spaces.

1.1.2 Role of AI in Surveillance

Artificial Intelligence (AI) is revolutionizing the field of surveillance by enabling systems to move beyond passive monitoring to intelligent, proactive, and real-time analysis. Traditional surveillance systems, which rely heavily on human operators to monitor CCTV feeds and detect unusual activities, are often limited by factors such as fatigue, delayed response times, and oversight due to the sheer volume of data. AI addresses these challenges by automating the analysis of video streams, recognizing patterns, and detecting anomalies that may indicate potential threats.

AI-powered surveillance systems leverage technologies such as Computer Vision, Machine Learning, and Deep Learning to interpret visual data similarly to the human eye, but with enhanced speed and accuracy. These systems can perform tasks like object detection, facial recognition, license plate reading, and motion tracking, allowing for efficient monitoring of public spaces, transportation hubs, and critical infrastructure. By identifying specific events such as unauthorized entry, loitering, or the presence of dangerous objects like weapons AI enables faster decision-making and automated alert generation, significantly reducing reliance on manual intervention.

In the context of women's safety, AI enhances surveillance by providing context-aware monitoring that can identify gender-based threats in real time. For instance, AI can detect scenarios such as a lone woman being followed, distress gestures (e.g., SOS signals), or suspicious group dynamics, such as a woman surrounded by multiple men. Moreover, AI systems can be integrated with alert mechanisms (e.g., SMS, emails, or messaging bots) to notify authorities or security personnel instantly, ensuring timely intervention. The use of AI in surveillance not only increases operational efficiency but also contributes to preventive security, shifting the focus from post-event analysis to real-time threat prevention.

Fig 1.1: AI in Surveillance

1.1.3 What is Deep Learning?

Deep learning is a subset of machine learning which is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data. In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data.

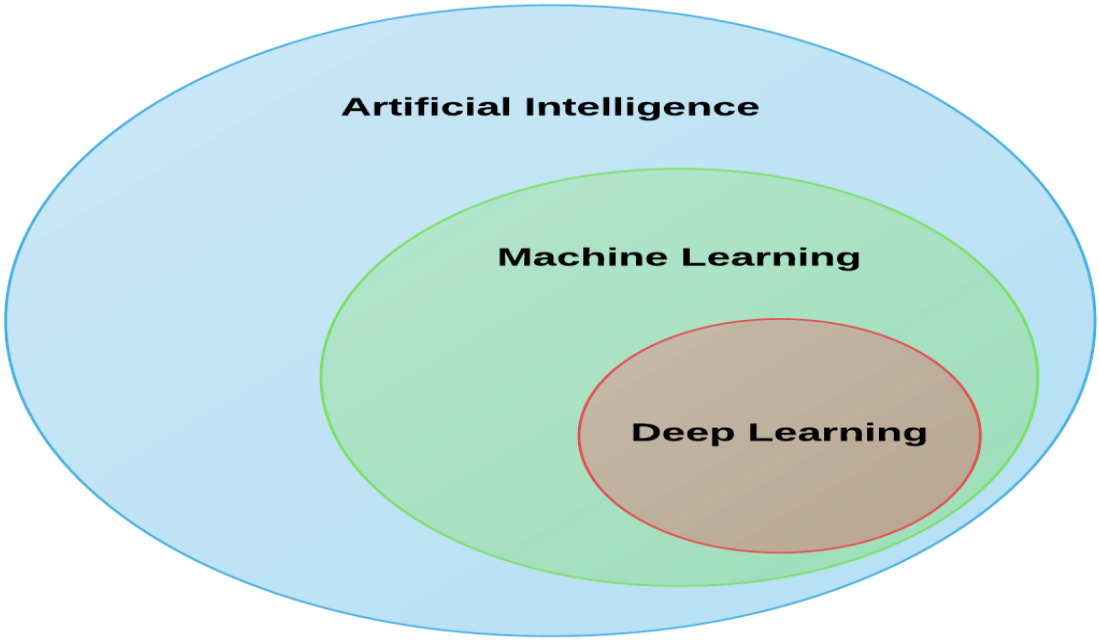


Fig 1.2: DeepLearning

Deep learning can be used for supervised, unsupervised as well as reinforcement machine learning. it uses a variety of ways to process these.

Supervised Machine Learning: Supervised machine learning is the machine learning technique in which the neural network learns to make predictions or classify data based on the labeled datasets. Here we input both input features along with the target variables. the neural network learns to make predictions based on the cost or error that comes from the difference

between the predicted and the actual target, this process is known as backpropagation. Deep learning algorithms like Convolutional neural networks, Recurrent neural networks are used for many supervised tasks like image classifications and recognization, sentiment analysis, language translations, etc.

Unsupervised Machine Learning: Unsupervised machine learning is the machine learning technique in which the neural network learns to discover the patterns or to cluster the dataset based on unlabeled datasets. Here there are no target variables. while the machine has to self-determined the hidden patterns or relationships within the datasets. Deep learning algorithms like autoencoders and generative models are used for unsupervised tasks like clustering, dimensionality reduction, and anomaly detection.

Reinforcement Machine Learning: Reinforcement Machine Learning is the machine learning technique in which an agent learns to make decisions in an environment to maximize a reward signal. The agent interacts with the environment by taking action and observing the resulting rewards. Deep learning can be used to learn policies, or a set of actions, that maximizes the cumulative reward over time. Deep reinforcement learning algorithms like Deep Q networks and Deep Deterministic Policy Gradient (DDPG) are used to reinforce tasks like robotics etc.

1.1.4 Deep Learning Applications

The main applications of deep learning can be divided into computer vision, natural language processing (NLP), and reinforcement learning.

Computer vision:

In computer vision, Deep learning models can enable machines to identify and understand visual data. Some of the main applications of deep learning in computer vision include:

• Object detection and recognition: Deep learning model can be used to identify and locate objects within images and videos, making it possible for machines to perform tasks such as self-driving cars, surveillance, and robotics.

• Image classification: Deep learning models can be used to classify images into categories such as animals, plants, and buildings. This is used in applications such as medical imaging, quality control, and image retrieval.

• Image segmentation: Deep learning models can be used for image segmentation into different regions, making it possible to identify specific features within images.

1.1.5 What is YOLO?

YOLO (You Only Look Once) is a cutting-edge, real-time object detection algorithm that has revolutionized the field of computer vision and deep learning. Unlike traditional object detection systems that repurpose classifiers or localizers to perform detection, YOLO frames object detection as a single regression problem, directly predicting bounding boxes and class probabilities from an image in one evaluation. This unified approach allows YOLO to be extremely fast and efficient, making it highly suitable for real-time applications like surveillance, autonomous vehicles, robotics, and security systems.

The main advantage of YOLO is that it processes the entire image at once, taking into account the global context, rather than scanning different parts of the image separately like earlier methods (e.g., R-CNN, Fast R-CNN). This leads to significantly faster detection speeds while maintaining high accuracy. YOLO divides an input image into a grid and for each grid cell, it predicts bounding boxes, confidence scores, and class labels simultaneously. This real-time capability is crucial for tasks like threat detection, weapon recognition, and people counting in dynamic environments.

Over time, YOLO has evolved through multiple versions-YOLOv1 to YOLOv11 each improving on speed, accuracy, and versatility. The latest versions, such as YOLOv11, support advanced features like instance segmentation, multi-object tracking, and improved model efficiency, making them ideal for AI-powered surveillance systems. In the context of women's safety analytics, YOLO plays a critical role in detecting people, classifying gender, identifying the presence of weapons, and analyzing crowd patterns, thereby enabling proactive threat detection and real-time alerts.

YOLO (You Only Look Once) is a state-of-the-art, real-time object detection algorithm that is widely used in computer vision applications. It is designed to detect and classify multiple objects in an image or video frame in a single pass, making it highly efficient for real-time tasks like video surveillance, autonomous vehicles, and security monitoring.

How YOLO Works:

YOLO divides an input image into a grid system and, for each grid cell, it predicts:

- Bounding boxes (to locate objects)
- Confidence scores (probability that an object exists)

- Class probabilities (to classify detected objects)

All of this happens in one evaluation of the neural network, making YOLO exceptionally fast and accurate.

YOLO in Our Project

In our project, YOLOv11 models are used to:

- Detect people and classify them by gender
- Detect weapons in real-time
- Enable real-time alerts when unsafe conditions are detected (e.g., a lone woman at night, or a woman surrounded by men)

By integrating YOLO, our system achieves instantaneous detection and decision-making, ensuring swift alert generation to CCTV operators or law enforcement, thereby enhancing proactive threat prevention.

1.1.6 Gesture Recognition with Mediapipe

Gesture recognition is an essential feature in modern AI-powered surveillance systems, especially for applications involving human-computer interaction and safety analytics. In the context of women's safety, recognizing specific hand gestures like SOS signals can play a critical role in identifying distress and triggering real-time alerts when verbal communication is not possible. One of the most powerful tools for implementing gesture recognition efficiently is Mediapipe, developed by Google.

What is Mediapipe?

Mediapipe is an open-source framework that provides cross-platform, customizable ML pipelines for live and streaming media. It includes ready-to-use solutions for hand tracking, face detection, pose estimation, object tracking, and more. For gesture recognition, Mediapipe's Hand Tracking module is widely used due to its high speed and accuracy in detecting 21 3D landmarks on each hand in real time.

How Gesture Recognition Works?

Mediapipe uses a two-step pipeline:

1. Palm Detection: Detects hand region in the frame.

2. Hand Landmark Model: Identifies 21 key points (landmarks) on the detected hand, including fingertips, knuckles, and wrist points.

Using these landmarks, gestures can be identified by calculating relative distances and angles between specific landmarks.

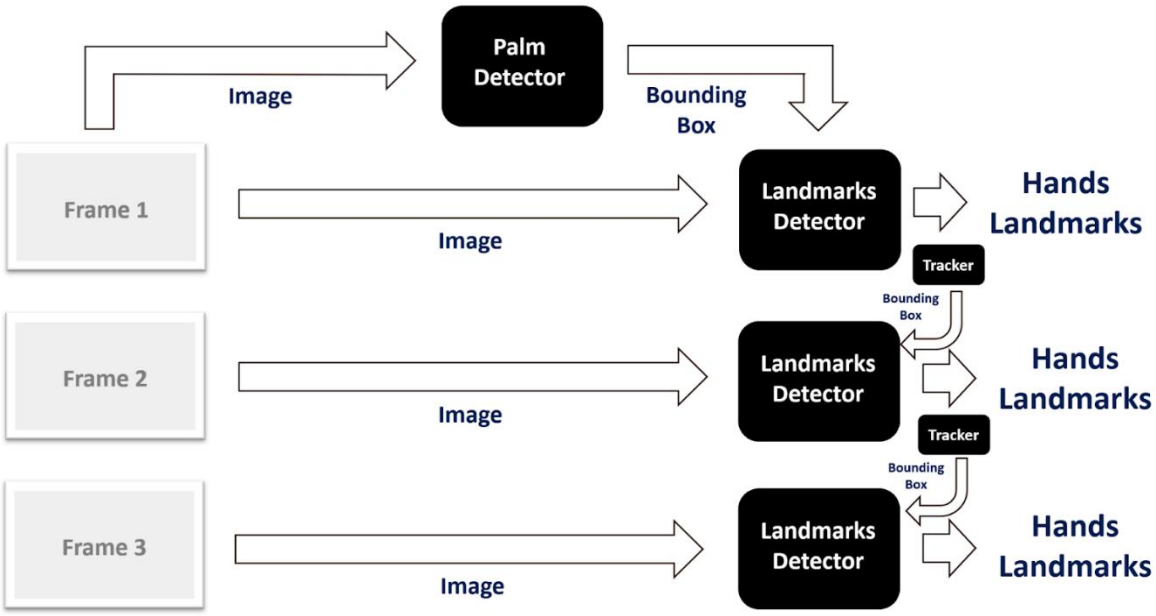


Fig 1.3: Landmark Detector

In our system:

- SOS Gesture is recognized when the thumb and pinky fingertips come close together.
- Thumbs Up Gesture is recognized when the thumb is raised above the index finger's MCP joint.

Once a gesture is detected for a predefined number of frames (for confirmation), the system can trigger an alert (for SOS) or display a safe message (for thumbs up).

Benefits of Mediapipe

- **Lightweight & Fast**: Runs in real time even on low-resource systems.
- **Cross-Platform**: Works on desktops, mobile devices, and embedded systems.
- **Highly Accurate**: Provides robust hand landmark tracking even in challenging environments.

- **No Special Hardware**: Works with standard webcams or CCTV feeds.

1.1.7 Telegram API for Real-Time Alerts

Real-time alerting is a crucial component in safety-critical applications, especially for women's safety in public spaces. Prompt communication with security personnel or law enforcement can prevent incidents from escalating. In our system, the Telegram API plays a key role in enabling instantaneous alerts by delivering messages, images, and location information directly to CCTV operators or concerned authorities.

What is Telegram API?

The Telegram Bot API allows developers to create bots that interact with users or groups through the Telegram messaging platform. These bots can send messages, images, videos, documents, and even trigger interactive commands, making them ideal for automated alerting systems.

Benefits of Telegram API for Real-Time Alerts

- Instant Communication: Messages are delivered within seconds, ensuring fast response.
- Multimedia Support: Ability to send images, videos, and audio, giving context to the alert.
- Cross-Platform: Telegram is accessible on PC, mobile, and web, allowing operators to stay connected anywhere.
- Secure & Reliable: Telegram uses end-to-end encryption and has a robust infrastructure for high reliability.
- Scalability: Easily supports multiple operators, group notifications, and custom alert formats**.**

1.2 Problem Statement

Despite numerous laws and safety measures aimed at protecting women, crimes against them continue to rise globally and locally, largely due to ineffective monitoring systems and delayed intervention. According to the World Health Organization (WHO), an alarming 1 in 3 women worldwide have experienced physical or sexual violence at some point in their lives. This pervasive issue highlights the urgent need for robust preventive measures and real-time

intervention systems. Furthermore, a UN Women report from 2022 revealed that only 40% of women who experience violence seek help, with the majority relying on family members rather than law enforcement agencies. This reluctance underscores the lack of trust in existing systems and emphasizes the critical gap in providing timely and effective support to victims.

In India, the situation is particularly dire. The National Crime Records Bureau (NCRB) 2021 report highlighted a 15% increase in crimes against women compared to the previous year, indicating that current measures are insufficient to curb this growing menace. Even with advancements in technology, such as the installation of over 30,000 CCTV cameras across Delhi in 2023, crime rates remain alarmingly high due to inefficient surveillance mechanisms. These systems often fail to provide actionable insights or enable swift responses, leaving women vulnerable despite significant investments in infrastructure. Moreover, a survey conducted by the Thomson Reuters Foundation in 2021 ranked India as the most dangerous country for women, citing factors such as rampant sexual violence, human trafficking, and domestic abuse. These statistics paint a grim picture of the challenges faced by women in India and underscore the pressing need for innovative solutions that can bridge gaps in monitoring and response times.

The convergence of global and local data points to a shared reality: traditional methods of ensuring women's safety are inadequate in addressing modern threats. To combat these issues effectively, there must be a paradigm shift toward leveraging advanced technologies like artificial intelligence, computer vision, and real-time alert systems. Such innovations could transform how crimes against women are detected, reported, and addressed, ultimately fostering safer environments both globally and within specific regions like India.

1.2.1 Aim

To develop an AI-powered SecureShe-Real Time Women Safety Alert System that leverages real-time surveillance, computer vision, and gesture recognition to proactively detect and prevent potential threats to women’s safety in public spaces, and to deliver instant alerts to CCTV operators for timely intervention.

This project seeks to develop a real-time AI-based Women Safety System with the following objectives:

- **Automated Threat Detection:** Identify potentially dangerous situations where a woman is surrounded or threatened by multiple males using AI models.

- **Real-Time Alerts:** Send immediate alerts to authorities when a weapon or threats are detected.
- **Integration with CCTV Surveillance:** Process live footage from CCTV cameras to recognize distress situations.
- **Enhanced Accuracy with AI Models:** Use deep learning and computer vision to reduce false positives and ensure precision.

1.2.2 Challenges in Existing Safety Measures

- **Delayed Response:** Manual surveillance and human monitoring are often ineffective in detecting and preventing real-time threats.
- **Lack of Awareness:** Many women hesitate to report incidents due to social stigma and lack of legal awareness.
- **Inefficient Law Enforcement:** Existing safety mechanisms, such as helplines and emergency apps, rely on active user input rather than proactive threat detection.

1.2.3 Features of the Proposed System

The proposed SecureShe-Real Time Women Safety Alert System is an advanced AI-based surveillance solution that integrates real-time video analysis, gesture recognition, and automated alerting to proactively detect and respond to threats against women in public spaces. It is designed to overcome the limitations of existing safety tools by offering intelligent, context-aware monitoring and rapid communication with authorities.

Key Features

a. Real-Time Person Detection and Gender Classification

- Uses YOLO object detection models to accurately detect individuals in video feeds.
- Classifies detected individuals as male or female using trained deep learning models.
- Enables instant analysis of gender distribution in public areas.

b. Gender Distribution Monitoring

- Counts the number of males and females present in a scene.
- Highlights gender imbalance (e.g., a woman surrounded by multiple men), identifying potentially unsafe situations.

- Offers time- and location-based insights into crowd demographics.

c. Lone Woman Detection During Night Hours

- Detects a solitary female in public spaces between 10 PM and 5 AM, a time considered high-risk.
- Triggers alerts to CCTV operators for immediate attention and intervention.

d. Detection of Woman Surrounded by Men

- Identifies when male count exceeds female count significantly, particularly when a single female is present.
- Sends real-time alerts with images to notify potential crowd threats.

e. SOS Gesture Recognition Using Mediapipe

- Recognizes distress gestures like the SOS hand sign (thumb and pinky touching).
- Employs gesture analytics for non-verbal communication in emergencies.
- Triggers instant alerts with image evidence without requiring manual input.

f. Weapon Detection

- Utilizes a custom YOLO model to identify weapons (e.g., knives, guns) in real time.
- Sends alerts with captured frames to ensure swift response from law enforcement.

g. Automated Real-Time Alerts via Telegram API

- Sends instant notifications to CCTV operators through Telegram bot, including:
    - Alert description.
    - Captured image of the incident.
    - Camera location.
- Ensures immediate awareness and response, reducing escalation risks.

h. Hands-Free Operation and Continuous Monitoring

- Requires no manual input; operates autonomously.
- Designed for continuous 24/7 monitoring, ensuring round-the-clock safety analytics.

i. Scalable and Adaptable

- Can be deployed across multiple camera feeds.
- Adaptable for smart cities, public transit hubs, campuses, and urban areas.

1.3 Objectives of the Project

To implement real-time person detection using **YOLO object detection models** to identify individuals within a surveillance feed.

1. To classify detected individuals by gender, enabling gender distribution analysis in public areas for identifying potentially risky environments.
2. To detect critical situations, including:
    - **A lone woman in a public space during night hours**.
    - **A woman surrounded by a group of men**.
    - **Presence of weapons indicating immediate threats.**
3. To integrate **gesture recognition** using Mediapipe, allowing the system to detect SOS gestures (e.g., thumb and pinky together) that indicate distress without the need for verbal communication.
4. To develop an automated alert system using the Telegram Bot API to send real-time notifications with captured images and location details to CCTV operators or security personnel.
5. To reduce manual surveillance workload by providing an AI-driven, proactive monitoring system that focuses specifically on women's safety scenarios.
6. To analyze time-based and location-based data, offering insights into gender distribution trends and potentially unsafe zones for women, assisting in long-term safety planning.

1.4 Scope of the Project

The SecureShe-Real Time Women Safety Alert System is designed to enhance public safety by leveraging AI-powered surveillance and real-time threat detection. The system integrates

computer vision, deep learning, and gesture recognition technologies to analyze live video footage, identify potential threats, and send instant alerts.

1.4.1 Data Sources

- CCTV footage from public places, streets, workplaces, and transport hubs.
- Pre-trained datasets for gender classification, weapon detection, and SOS gesture recognition.
- GPS data for location-based alerts.

1.4.2 Technologies Used

- Deep Learning models: YOLO for object detection (weapons, people).
- Computer Vision: OpenCV and MediaPipe for gesture recognition.
- Flask and Telepot: To integrate the system with a web application and Telegram bot for instant alerts.
- Geolocation APIs: Used Nominatim with static coordinates to fetch location details.

1.4.3 Use Cases:

- Monitoring public safety in real-time using CCTV-based AI analytics.
- Detecting unsafe environments when a woman is surrounded by multiple males late at night.
- Recognizing emergency hand gestures (like SOS signals) and alerting authorities.
- Weapon detection in crime-prone areas to prevent attacks.

1.4.4 Limitations:

- The model may struggle in poor lighting conditions or low-resolution CCTV footage.
- Limited scope in rural areas where CCTV infrastructure is not well-developed.
- Does not provide physical intervention—only alerts relevant authorities or individuals.
- Requires continuous model retraining to adapt to new threats and improve accuracy.

1.5 Methodology Overview

The SecureShe-Real Time Women Safety Alert system is designed using advanced AI and deep learning techniques to detect potential threats in real-time. The methodology involves multiple stages, including data collection, model development, system integration, and testing to ensure

high accuracy and efficiency. The system is built using YOLOv8 for gender classification and weapon detection, while Mediapipe is employed for gesture recognition.

The development process starts with data collection, where diverse datasets containing images and videos of people, weapons, and gestures are gathered. The next step involves training deep learning models using these datasets to classify genders, detect weapons, and recognize distress signals. Once trained, the models are integrated into the real-time surveillance system, where they continuously process video feeds from CCTV cameras.

To ensure instant threat detection and response, the system leverages the Telegram API to send real-time alerts to law enforcement or emergency contacts whenever a potential danger is identified. The entire process is privacy-focused, ensuring that sensitive information is not stored or misused. Additionally, ethical considerations are taken into account by complying with surveillance laws and implementing encryption for secure data transmission.

1.5.1 Description of AI Model Development Process

The AI model used in the SecureShe-Real Time Women Safety Alert system is developed using deep learning techniques. The system employs YOLOv11 for gender classification and weapon detection, along with Mediapipe for gesture recognition. The development follows these stages:

1. **Data Collection & Preprocessing** – Gathering diverse datasets for gender classification, weapon detection, and SOS gesture recognition.
2. **Model Selection & Training** – Training YOLOv8 for object detection and gender classification while fine-tuning Mediapipe for gesture recognition.
3. **Integration & Deployment** – Deploying trained models into a real-time surveillance system, integrating them with a Telegram API for instant alerts.
4. **Testing & Validation** – Evaluating the system using various real-world scenarios to ensure accuracy and reliability.

1.5.2 Data Collection for Various Features

1. **Gender Classification Dataset** – Images and videos containing diverse groups of people.
2. **Weapon Detection Dataset** – A curated dataset with various types of weapons in different environments.

3. **Environmental Data** – Different lighting conditions, crowded places, and isolated locations to enhance robustness.

1.5.3 Details of the Algorithm and Software Used

- **YOLOv11** – Used for gender classification and weapon detection.
- **Mediapipe** – Used for hand gesture detection.
- **OpenCV** – For video frame processing.
- **Python** – Programming language for implementation.
- **Telegram API** – For real-time alerts.
- **Geopy** - Geopy is a library for geolocation and address conversion.
- **Nominatim** - Nominatim is used to convert GPS coordinates into human-readable addresses, which can be helpful for identifying alert locations.
- **Ultralytics** - The ultralytics package provides implementations of YOLO (You Only Look Once) object detection models.
- **telepot** - telepot is a Python library for interacting with the Telegram Bot API.
- **torch** - PyTorch (torch) is a deep learning framework. It helps in running and training AI models efficiently, especially on GPUs.
- **Flask** - Flask initializes the web application, render_template helps in rendering HTML templates, and redirect & url_for manage URL routing.

1.5.4 User Role Definitions and Interactions

1. **Surveillance Operator** – Monitors alerts and takes necessary actions.
2. **Law Enforcement** – Receives alerts for potential threats.
3. **General Public** – The system ensures their safety through proactive monitoring.

1.5.5 Ethical Considerations and Data Privacy Measures

To protect individual privacy and ensure ethical AI deployment, the system follows:

- **Anonymized Data Processing** – Faces and personal details are blurred before data processing.
- **Encryption** – Ensures secure transmission of alerts and detected events.
- **Legal & Ethical Compliance** – Adheres to local surveillance laws, human rights policies, and ethical AI principles.

CHAPTER 2

LITERATURE REVIEW

2.1 Previous Research and Related Work

Several studies have explored AI-driven surveillance and threat detection to improve women's safety. The following research papers highlight key advancements:

1. Women Safety Analytics - Protecting Women from Safety Threats (Nov 2024)

- Utilized sentiment analysis on social media platforms such as Twitter and Facebook to detect harmful content related to women's safety.
- Integrated real-time monitoring through Smart Wearable Devices (SWD) equipped with GPS, microphones, and cameras for enhanced safety analysis.
- Inspired the incorporation of sentiment-based threat detection and wearable technology in our framework.
- Provided insights on linking social media alerts with real-time monitoring for an enhanced safety response.

2. AI-Powered CCTV Analytics for Proactive Threat Detection and Operational Excellence in Well Engineering Operations (Nov 2024)

- Applied AI-driven analytics on CCTV footage to monitor traffic and road conditions.
- Implemented facial recognition and people counting features, helping identify the number of individuals present in specific locations.
- Assisted in developing the people detection mechanism in public spaces for our surveillance system.
- Enhanced security measures by integrating AI-based motion detection with facial scanning to identify potential threats.

3. Towards a Conceptual Framework for AI-driven Anomaly Detection in Smart City IoT Networks for Enhanced Cybersecurity (Oct 2024)

- Developed AI models for detecting anomalies in smart city surveillance systems.
- Assisted in refining our threat detection capabilities by identifying unusual behaviors.

- Enabled better predictive analytics by linking AI anomaly detection with historical crime pattern analysis.

4.Object Detection and Crowd Analysis Using Deep Learning Techniques: Comprehensive Review and Future Directions (Sept 2024)

- Focused on identifying and classifying people in crowded areas.
- Provided insights for detecting scenarios where a woman is alone among a predominantly male crowd, triggering alerts.
- Helped refine machine learning algorithms to identify abnormal crowd behaviors that may indicate security threats.

5. An Integrated Approach for Real-Time Gender and Age Classification in Video Inputs Using FaceNet and Deep Learning Techniques (Aug 2024)

- Used FaceNet and deep learning to classify gender based on facial recognition.
- Provided a foundation for integrating real-time gender classification in our system to assess situations where women may be at risk.
- Enabled better accuracy in identifying individuals based on age and gender attributes, improving security insights.

6. Real-time Object Detection, Tracking, and Monitoring Framework for Security Surveillance Systems (Aug 2024)

- Evaluated YOLO and SSD models for object detection and tracking.
- Enabled us to optimize our threat tracking mechanisms in surveillance environments.
- Improved response times by ensuring real-time alerts are sent when an anomaly is detected.

7. AI in Crime Prediction and Prevention (May 2024)

- Demonstrated AI-driven crime prediction models to prevent incidents.
- Supported our predictive threat analysis feature.
- Strengthened security measures by integrating AI-based crime forecasting tools with real-time monitoring.

8. The Role of IoT in Women's Safety (Jan 2023)

- Highlighted the networking aspect of IoT for sending emergency alerts.
- Influenced the incorporation of Telegram bots for real-time alerts to surveillance teams.
- Helped design IoT-based tracking systems for wearable devices ensuring continuous monitoring.

9. A Machine Learning Approach to Design and Develop a BEACON Device for Women's Safety (May 2022)

- Discussed ML-powered wearable safety devices for emergency response.
- Inspired our system's wearable safety device integration.
- Improved emergency response time by integrating ML-driven analysis for distress situations.

10. Artificial Intelligence & Crime Prediction: A Systematic Literature Review (Mar 2022)

- Explored crime prediction based on situational and environmental factors.
- Helped in implementing AI-driven predictive analytics for crime prevention in our system.
- Provided methodologies for integrating crime trend analysis with real-time safety monitoring.

11. Design of a Smart Women Safety Band Using IoT and Machine Learning (May 2021)

- Developed IoT-based safety bands for continuous tracking and emergency alerts.
- Contributed to our IoT-driven safety monitoring system.
- Strengthened wearable safety solutions by integrating ML-driven distress detection models.

12. Deep Learning Based Hand Gesture Recognition for Emergency Situations: A Study on Indian Sign Language (May 2021)

- Explored hand gesture recognition to detect distress signals.

- Inspired our inclusion of gesture-based distress recognition for real-time emergency alerts.
- Enhanced our AI models by integrating predefined emergency gesture patterns to identify potential distress situations quickly.

13. Weapon Detection Using YOLO V3 for Smart Surveillance System (May 2021)

- Utilized the YOLOv3 object detection model to identify weapons in real-time CCTV footage.
- Helped us integrate weapon detection for rapid security response in public places.
- Strengthened our threat assessment capabilities by incorporating weapon detection into surveillance networks.

14. Guardian Device for Women—A Survey and Comparison Study (May 2021)

- Compared various women safety technologies.
- Helped refine our feature selection by analyzing existing solutions.
- Offered guidelines for choosing the most efficient wearable safety technology.

15. Smart Wearable Device for Women Safety Using IoT (Jun 2020)

- Enabled continuous monitoring through IoT-based wearable devices.
- Influenced our design of wearable safety gadgets for real-time threat detection.
- Strengthened emergency response capabilities by ensuring immediate communication with authorities.

16. IoT-based Women Security: A Contemplation (Mar 2020)

- Highlighted IoT-based safety applications.
- Enhanced our understanding of IoT networking for real-time monitoring.
- Improved the communication framework between IoT devices and emergency response teams.

17. Recent and Emerging Technologies: Implications for Women's Safety (Aug 2019)

- Explored AI-based technologies for women's safety.

- Contributed to our selection of AI methodologies.
- Strengthened our system by integrating AI-powered data analytics for improved decision-making.

18. A Hidden Markov Model and IoT Hybrid Based Smart Women Safety Device (Jun 2018)

- Integrated IoT with predictive models for safety monitoring.
- Contributed to the development of predictive alert mechanisms in our system.
- Provided methodologies for enhancing safety measures through pattern-based recognition of threats.

19. MoveFree: A Ubiquitous System to Provide Women Safety (Aug 2015)

- Proposed a multi-model safety system for sending emergency alerts.
- Inspired the messaging and alert system in our surveillance framework.
- Integrated multi-channel alert mechanisms such as SMS, email, and app notifications.

20. Systematic Literature Review vs Narrative Review (2007)

- Provided methodological guidance for reviewing research trends.
- Assisted in structuring our research approach for women's safety technology.
- Helped differentiate between review methodologies for effective research analysis.

2.2 Existing Solutions and Their Limitations

While various systems exist to address public safety, most traditional surveillance solutions and women safety tools have significant limitations when it comes to proactive threat detection, especially in the context of women's safety in public spaces. These limitations reduce their effectiveness in preventing incidents and providing real-time support.

- **1. Manual Monitoring Overload**
  Conventional CCTV systems rely heavily on human operators to constantly monitor video feeds. Due to human fatigue, attention lapses, and the sheer volume of surveillance footage, critical events can be missed, resulting in delayed responses or unnoticed threats.

- **2. Reactive Instead of Proactive**

  Most existing systems are reactive, meaning they are used primarily for post-incident investigation rather than real-time prevention. Footage is reviewed after an event occurs, offering no immediate support to victims in distress.
- **3. Lack of Context-Aware Analysis**

  Traditional systems do not provide context-aware monitoring. For example, they cannot distinguish between a normal gathering and a potentially risky situation, such as a lone woman at night or a group of men surrounding a female. There is no gender-based analysis or situational interpretation in basic surveillance setups.
- **4. Dependency on Victim's Actions**

  Mobile safety apps and emergency hotlines require active input from victims (e.g., pressing an SOS button, making a call). In many real-world scenarios, women may be unable to access their phone or communicate distress due to fear, coercion, or immediate danger.
- **5. Generic Anomaly Detection**

  Some AI-based systems use general anomaly detection algorithms, which can identify unusual movement or behavior but lack specialization in detecting gender-based threats or specific gestures of distress like an SOS sign.
- **6. No Integrated Alerting Mechanism**

  Most systems do not have automated alert systems connected to authorities. They depend on manual escalation, which slows down response time and may allow threats to escalate.

2.3 Gap Analysis

Although various technologies exist, they fail to provide a holistic solution for real-time women's safety. The key gaps include:

- **Lack of Proactive Threat Detection**: Existing solutions react after an incident occurs, rather than predicting and preventing threats.
- **Reliance on Manual Intervention**: Most panic apps and emergency buttons require user input, making them useless in unconscious or dangerous situations.
- **Data and Privacy Concerns**: AI-based surveillance often lacks privacy safeguards, limiting widespread adoption.

- **Limited Integration of Multiple Technologies**: No system combines computer vision, NLP, and real-time alert mechanisms in a single, efficient framework.

2.4 Relevance of the Project

Our project addresses these gaps by integrating real-time CCTV monitoring, AI-based gender classification, and NLP-based voice distress detection into a single automated system.

How Our Project Builds on Existing Research

- Uses deep learning models like YOLOv11 for improved real-time object detection.
- Incorporates sentiment analysis from speech data using BERT and NLP models.
- Provides real-time alerts to law enforcement and bypasses the need for manual intervention.

Inspiration and Dataset Sources

- **Dataset:** Open-source CCTV surveillance datasets and gender classification datasets.
- **Model Inspiration:** Deep learning architectures from existing anomaly detection and object tracking research.

By integrating AI-driven threat detection, real-time alerts, and multi-modal analysis, our system enhances women's safety beyond what existing technologies offer.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Requirement Analysis

3.1.1 Functional Requirements

The system must perform the following functions:

- Real-Time CCTV Surveillance Monitoring
    - Detect and classify males and females using computer vision.
    - Identify if a woman is surrounded or threatened by males.
    - Capture video frames for analysis.
- Threat Detection & Alert System
    - Analyze behavioral patterns and crowd density.
    - Detect verbal distress signals using speech recognition and NLP.
    - Send real-time alerts (including video and audio) to authorities.
- Integration with Law Enforcement & Emergency Services
    - Notify nearby police stations or emergency responders.
    - Store detected events for forensic analysis.
    - Allow manual user intervention if needed.

3.1.2 Non-Functional Requirements

- Performance:
    - The system must process real-time video feeds with minimal latency (<2 seconds response time).
- Security:
    - Securely store recorded data to prevent misuse.
    - Ensure end-to-end encryption in alert transmissions.

- Scalability:
    - Must be deployable in multiple locations.
    - Should support edge computing for faster local processing.

3.2 Proposed System

The proposed SecureShe-Real Time Women Safety Alert system is an AI-driven real-time surveillance solution that detects potential threats based on gender classification, weapon detection, and SOS gestures. The system utilizes YOLOv11 for detecting and classifying individuals based on gender, identifying the presence of weapons, and Mediapipe for detecting distress gestures like SOS signals. It integrates OpenCV for video processing and a Telegram API for real-time alert notifications.

The workflow of the system is as follows:

1. **Frame Capture & Processing** – CCTV or IP cameras continuously capture live video feeds.
2. **Gender Classification** – YOLOv11 detects individuals and classifies them as male or female.
3. **Group Alert Mechanism** – If a lone female is detected surrounded by multiple males, an alert is triggered.
4. **Weapon Detection** – The system identifies weapons in real-time and raises an alert if a weapon is detected.
5. **SOS Gesture Recognition** – Mediapipe identifies predefined SOS hand gestures to signal distress.
6. **Alert Transmission** – The detected threats or SOS signals are sent via the **Telegram API** to law enforcement or assigned emergency contacts.

This AI-powered system ensures proactive monitoring and rapid response to critical situations, enhancing women's safety in public and private spaces.

3.2.1 Advantages of the Proposed System

1. **Real-time Monitoring** – The system continuously analyzes live video feeds and detects threats instantly.
2. **Automated Alert System** – Sends automatic Telegram alerts when a potential danger is detected, ensuring quick response.
3. **High Accuracy Detection** – Utilizes YOLOv8 and Mediapipe, ensuring precise gender classification, weapon identification, and gesture recognition.
4. **Prevention of Crimes** – By detecting threats early, the system helps prevent potential crimes before they occur.
5. **Scalability** – Can be integrated with existing CCTV surveillance systems in public places, offices, and homes.
6. **Privacy-focused** – The system operates without storing data, making it secure and privacy-compliant.
7. **Cost-effective Solution** – Leverages AI-driven automation to reduce the need for manual monitoring.
8. **Adaptability** – Can be customized to detect other threats and integrate with emergency response systems for enhanced security.

3.2.2 Proposed System Overview

The SecureShe-Real Time Women Safety Alert System is an AI-powered real-time surveillance solution designed to enhance the security of women in public and private spaces. The system utilizes computer vision techniques, deep learning models, and real-time alert mechanisms to detect and respond to potential threats.

Key Components:

1. **Real-time CCTV Surveillance**
    - Captures live video frames from CCTV or security cameras.
    - Processes the frames to extract necessary information for analysis.

2. **Gender Detection using YOLOv8**
    - Uses a pre-trained YOLOv8 model to detect and classify individuals based on gender.
    - Counts the number of males and females in a given frame.
    - Triggers an alert if a lone female is surrounded by a group of males.
3. **Weapon Detection using YOLOv8**
    - Identifies the presence of weapons such as guns, knives, or other dangerous objects.
    - Sends an immediate alert if a weapon is detected in the scene.
4. **SOS Gesture Recognition using Mediapipe**
    - Recognizes predefined hand gestures (e.g., raising both hands or forming an SOS sign).
    - Monitors gestures over multiple frames for accurate detection.
    - Triggers an alert if an SOS gesture is detected consistently.
5. **Alert System (Telegram API Integration)**
    - Sends real-time alerts via Telegram Bot to concerned authorities or emergency contacts.
    - Provides image snapshots and location details along with the alert message.
6. **Automated Monitoring and Decision Making**
    - Continuously processes video feeds without human intervention.
    - Automatically restarts monitoring after processing each frame unless manually stopped.

3.3 System Requirements

3.3.1 Software Requirements

- **Operating System:** Windows/Linux
- **Programming Language:** Python
- **Libraries & Frameworks:**
    - OpenCV (video processing)
    - TensorFlow/PyTorch (AI model training)
    - YOLOv11 (gender and weapon detection)
    - Mediapipe (gesture recognition)
    - Flask(Web Framework)
    - Telegram API (alert notifications)

3.3.2 Hardware Requirements

1. **Processor:** Intel i5/i7 or equivalent (AMD Ryzen 5/7)
2. **RAM:** Minimum 8GB (16GB recommended)
3. **Cameras:** High-resolution CCTV or IP cameras
4. **Storage:** SSD with at least 1TB capacity
5. **Network Connectivity:** Stable internet connection.

3.4 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.4.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.4.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.4.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

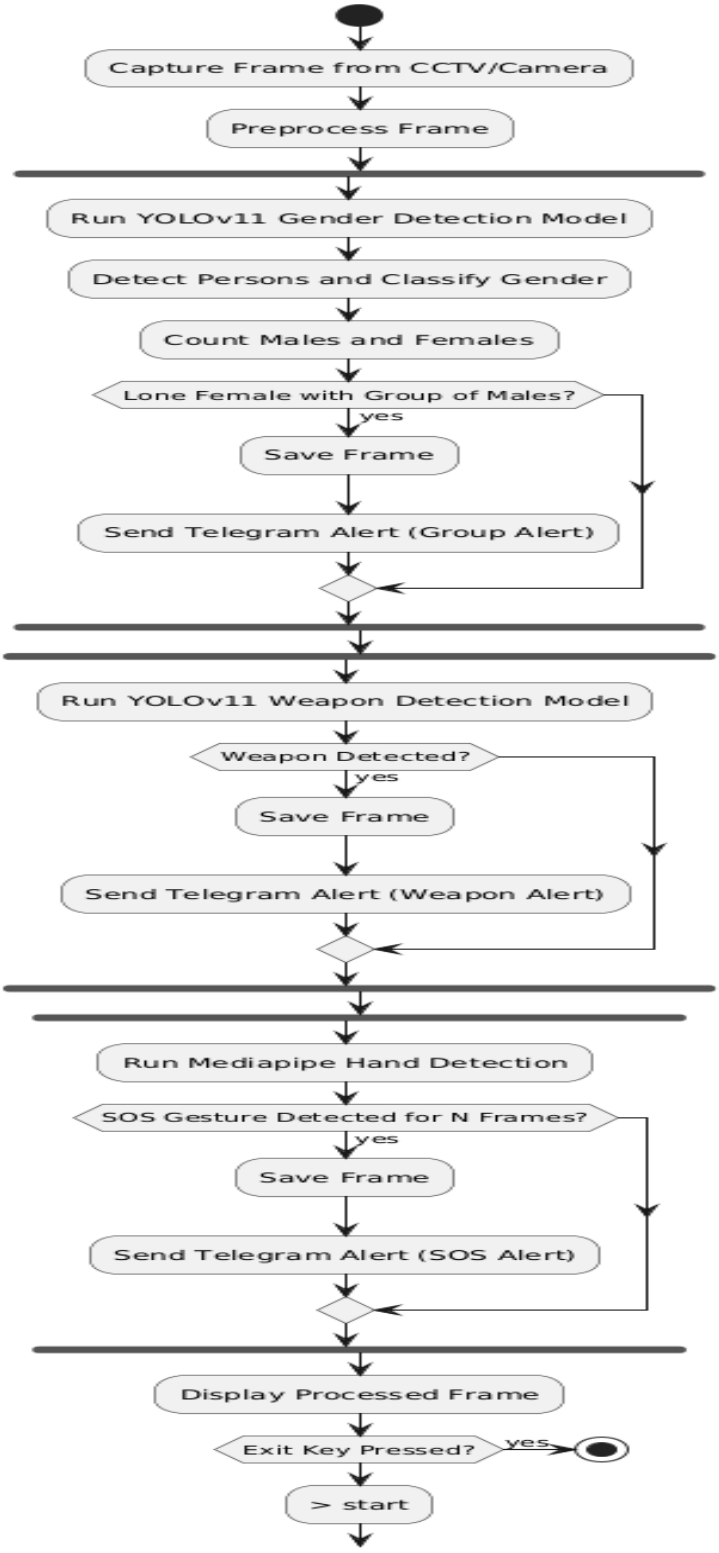


Fig 4.1: Flow of Architecture

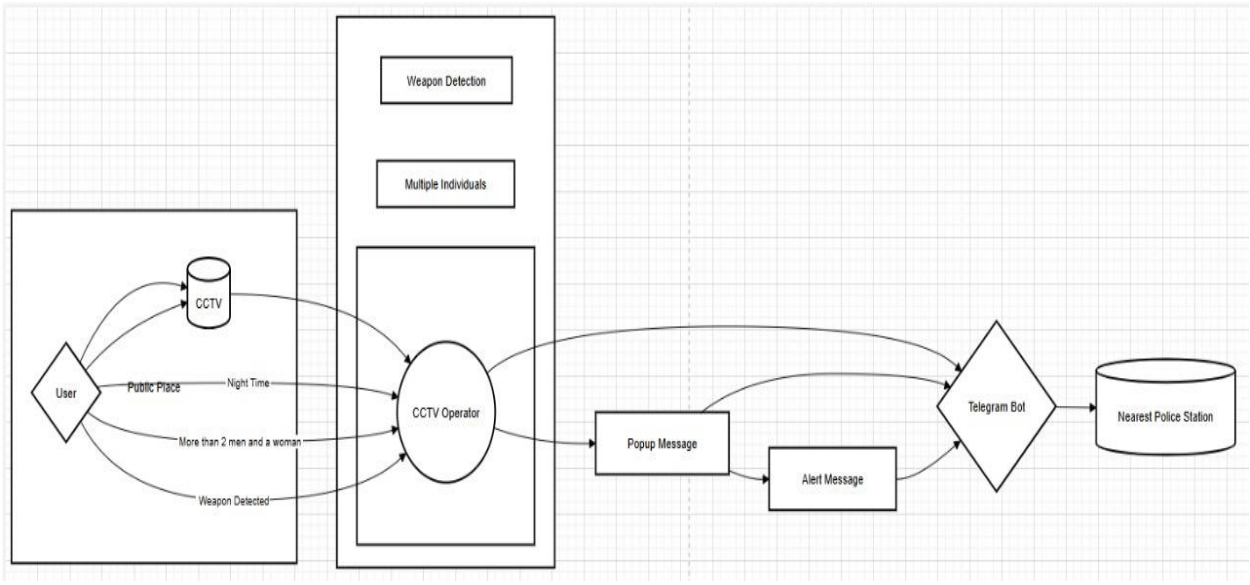


Fig 4.2: System Architecture

1. Input Layer – CCTV/Camera Feed

- The system begins with live video input from surveillance cameras or CCTV.
- Frames are extracted in real-time for analysis.

2. Pre-processing Module

- Captured frames undergo resizing, normalization, and color space conversion (if needed) to prepare for model input.
- Frame data is formatted for YOLOv11 and MediaPipe processing.

3. Detection & Recognition Modules

a. YOLOv11 Gender Detection Model

- YOLOv11 is used for object detection and gender classification.
- Each detected person is classified as Male, Female, or Background.
- Bounding boxes and class labels are overlaid on frames.

b. MediaPipe Gesture Recognition

- MediaPipe processes the same frames to detect 21 hand landmarks.

- It analyzes the hand posture to identify specific SOS gestures (e.g., hand wave or specific finger positioning).

4. Analytical Logic Layer

a. Lone Female Detection

- Counts the number of males and females in the frame.
- If only one female is surrounded by multiple males, the system flags it as a critical situation.

b. SOS Gesture Trigger

- If an SOS gesture is detected, it acts as a manual trigger for sending alerts, even without group logic detection.

5. Alert & Communication Module

a. Telegram Bot API Integration

- The system captures the current frame and sends a real-time alert via Telegram.
- The message includes the captured image, alert type, and optionally location or timestamp.

6. Output & Storage

- Alert frames can be stored locally or in a cloud database for future review.
- Logs of alerts and detected events are maintained for audit trails.

4.2 Block Diagram

- **User**: Interacts with the web application.
- **Flask Web App**: The main application that handles user requests and starts the detection process.
- **Detection Thread**: A separate thread that runs the detection logic.
- **Webcam**: Captures video frames for processing.
- **YOLO Models**: Utilizes YOLO for weapon and gender detection.
- **Weapon Detection**: Identifies weapons in the video feed.
- **Gender Detection**: Analyzes the video feed to determine gender.

- **Telegram Bot**: Sends alerts and notifications to the user via Telegram.
- **Location Service**: Provides geolocation based on latitude and longitude.
- **User Notification**: Notifies the user about gender counts and alerts.
- **Web UI**: Displays the results of the detection in the browser.

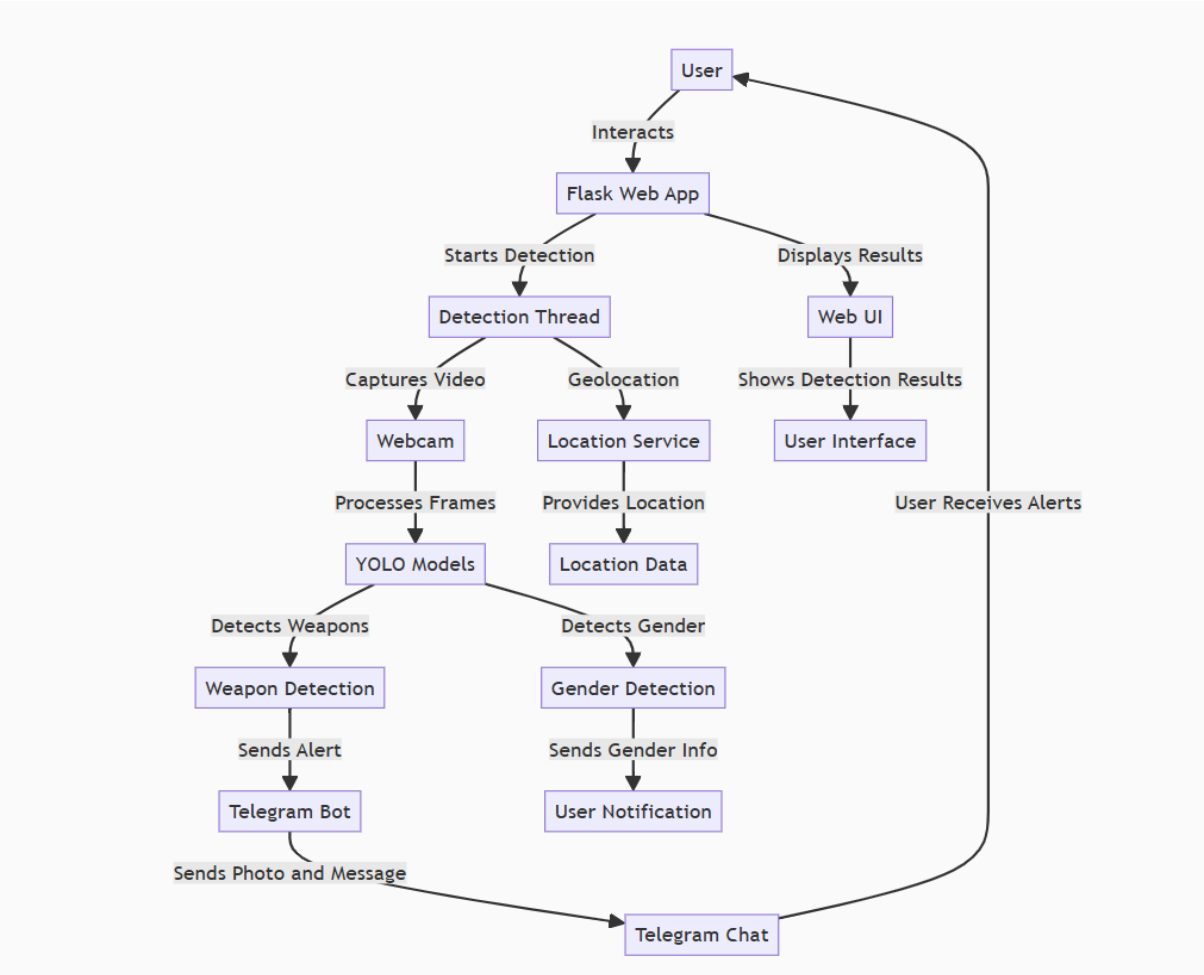


Fig 4.3: Block Diagram

4.3 DFD Diagram

Level 0 DFD:

- **User:** Interacts with the web app to start the detection process.
- **Flask Web App:** Initiates the detection process and handles user interactions.
- **Detection Process:** Main process that encompasses all detection functionalities.
- **Telegram Bot:** Sends alerts to users based on detection results.

- **Web UI:** Displays results of the detection to the user.
- **Location Service:** Provides geolocation data.

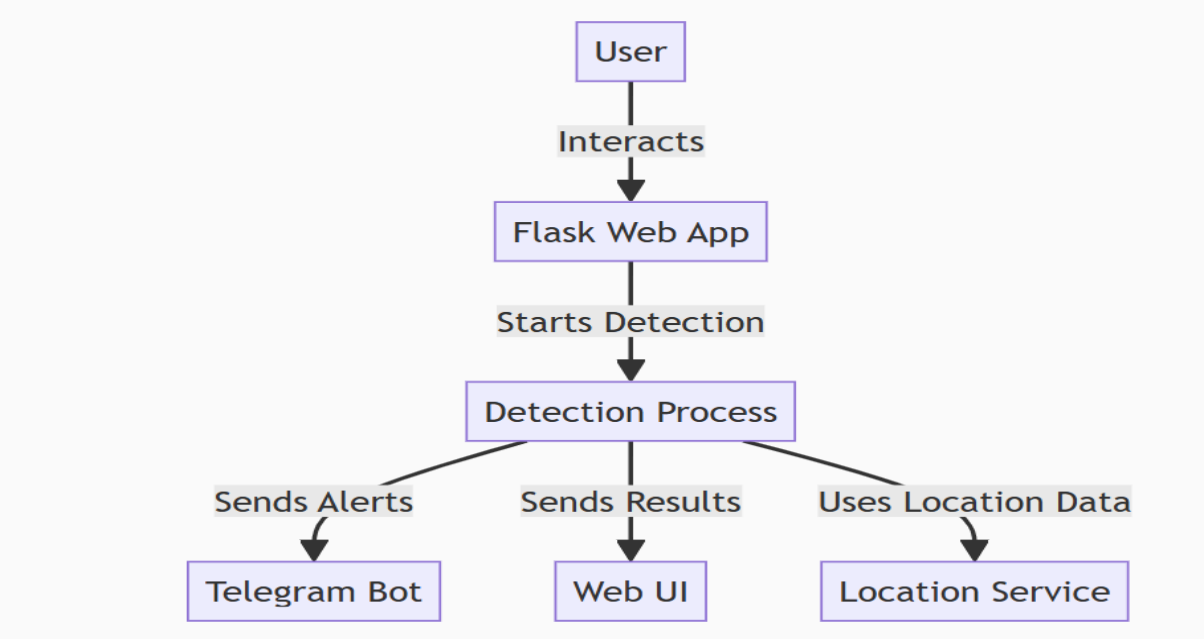


Fig 4.4: Level 0 DFD

Level 1 DFD:

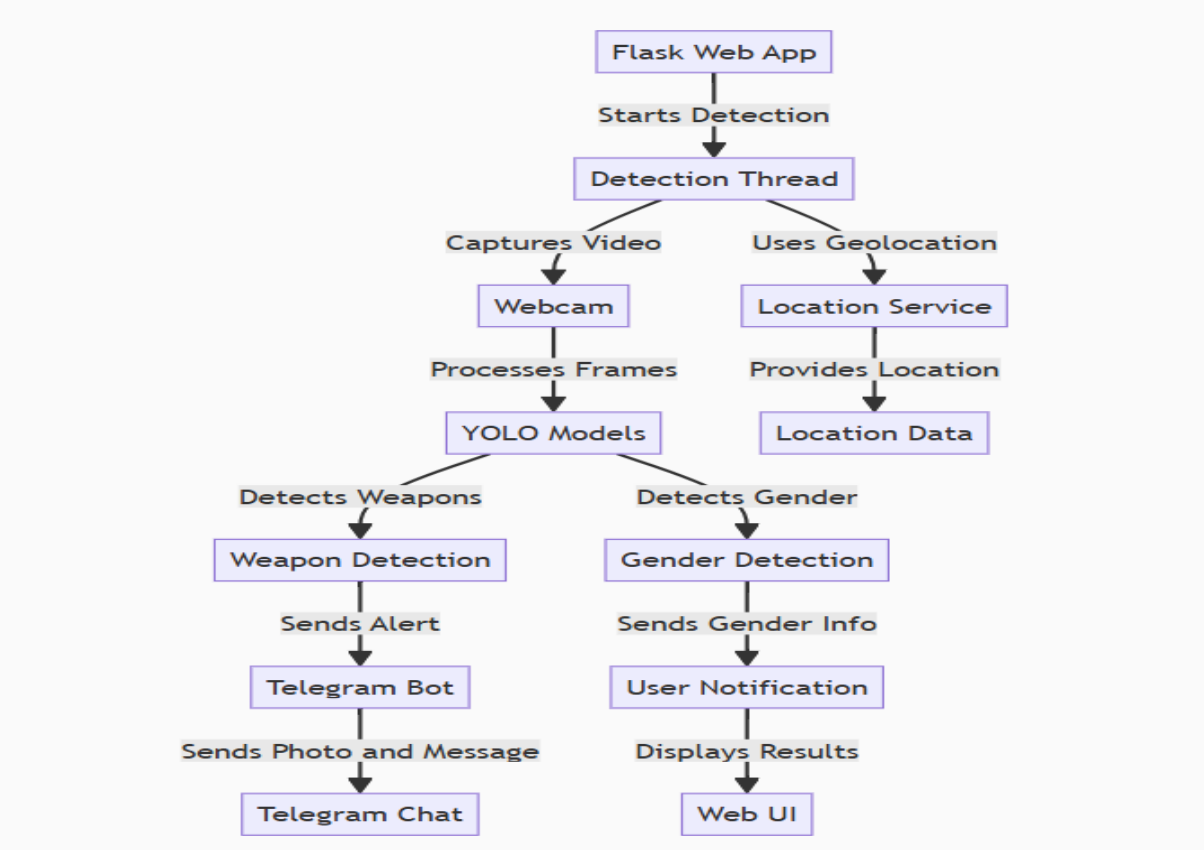


Fig 4.5: Level 1 DFD

- **Detection Thread:** Runs the detection logic in a separate thread.
- **Webcam:** Captures video frames for analysis.
- **YOLO Models:** Processes frames to detect weapons and gender.
- **Weapon Detection:** Identifies weapons in the video feed.
- **Gender Detection:** Analyzes gender from the video feed.
- **Telegram Bot:** Sends alerts and notifications to users.
- **User Notification:** Notifies the user about the detection results.
- **Location Service:** Provides geolocation data for alerts.

4.4 Uml Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form, UML is comprised of two major components: a Meta model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

GOALS: The Primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extensibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of the OO tools market.

6. Support higher-level development concepts such as collaborations, frameworks, patterns, and components.

7. Integrate best practices.

4.4.1 Use Case Diagram

This use case diagram depicting the various functionalities of a camera system. The system is capable of detecting weapons, SOS gestures, thumbs up, and gender, and it sends corresponding alerts to a CCTV operator based on these detections.

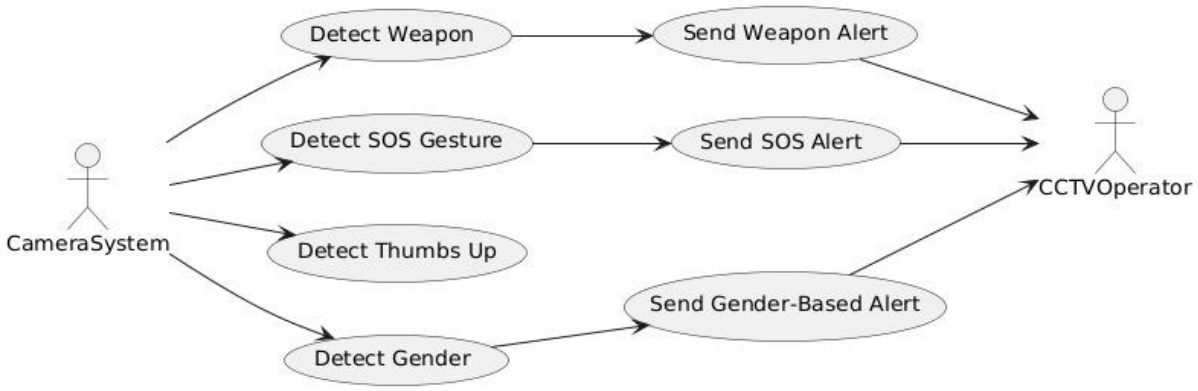


Fig 4.6: Use Case Diagram

4.4.2 Class Diagram

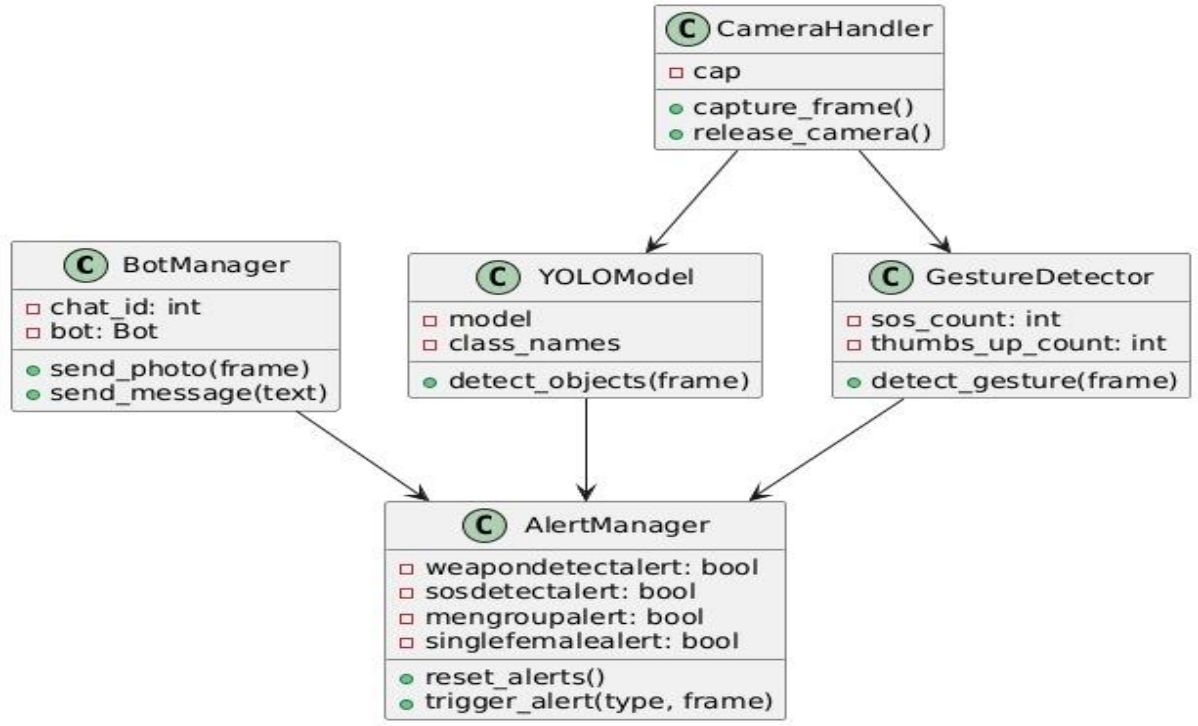


Fig 4.7: Class Diagram

This class diagram depicts the key components of a camera-based detection and alert system.

It includes a CameraHandler for managing the camera, a YOLOModel for object detection, a GestureDetector for gesture recognition, and an AlertManager for handling various alerts such as weapon detection, SOS, and gender-based alerts.

4.4.3 Object Diagram

This object diagram depicts the key components of the camera-based detection and alert system, including a weapon_model for weapon detection, a bot_instance for handling chat interactions, a gender_model for gender detection, and a frame object representing the camera frame. These objects work together to enable the various detection and alert functionalities of the system.

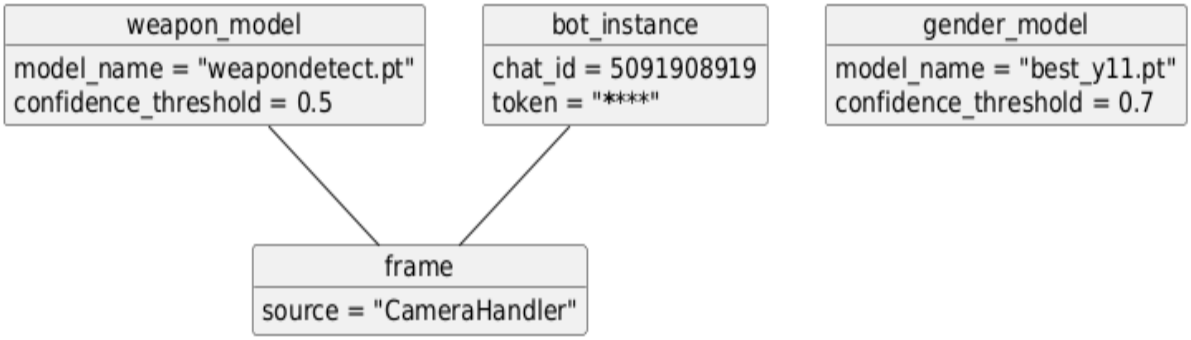


Fig 4.8: Object Diagram

4.4.4 Sequence Diagram

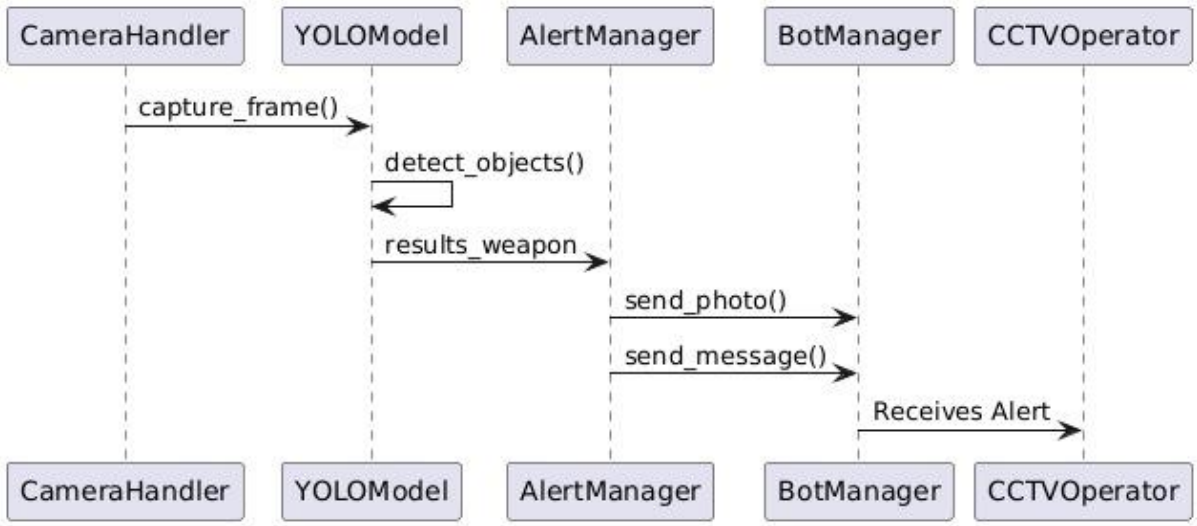


Fig 4.9:Sequence Diagram

This sequence diagram depicts the interactions between the key components of the camera-based detection and alert system. It shows the flow of data, starting with the CameraHandler

capturing a frame and passing it to the YOLOModel for object detection. The YOLOModel then returns the detection results, which trigger the AlertManager to send a photo and message to the BotManager, who in turn relays the alert to the CCTVOperator. This sequence of interactions enables the overall functionality of the weapon detection and alert system.

4.4.5 Activity Diagram

This activity diagram depicts the workflow of a weapon detection system. It starts with capturing a frame, then running the YOLO (You Only Look Once) object detection model to check for the presence of a weapon. Depending on the detection results, the system either continues the loop or sends an alert, marking it as sent.

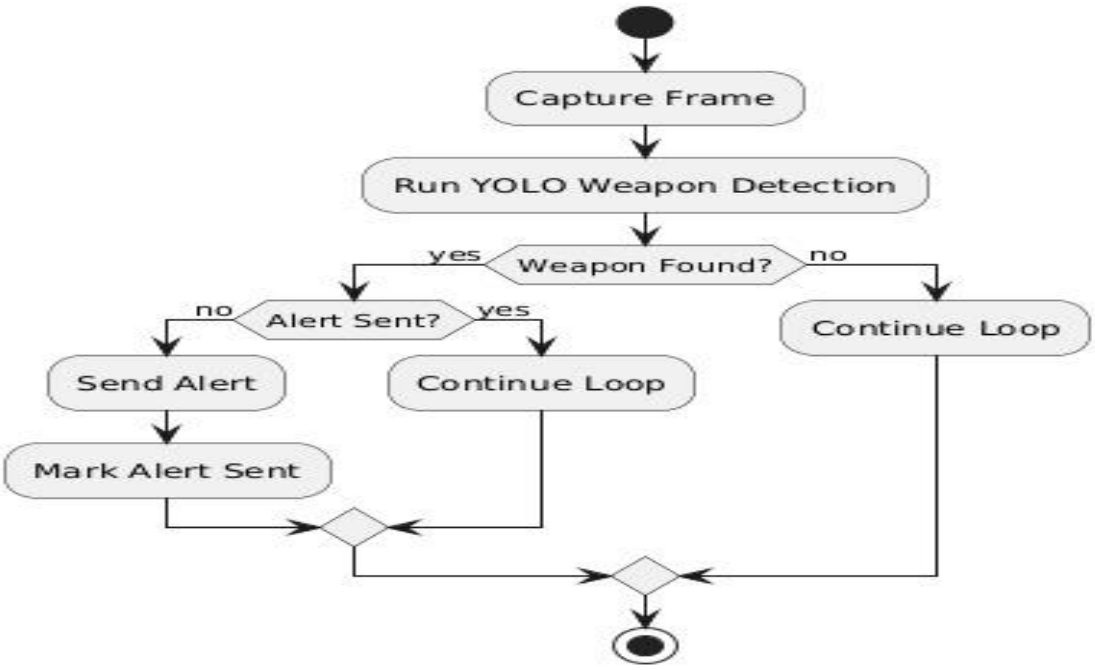


Fig 4.10: Activity Diagram

4.4.6State Chart Diagram

This state chart diagram depicts the flow of a weapon detection and alert system. It starts with the "Idle" state, where a frame is captured and the system enters the "Detecting" state. If an SOS gesture is detected, an alert is sent, and the system enters the "SendingAlert" state. The diagram shows the various components involved in this process, such as the CameraHandler, YOLOModel, AlertManager, BotManager, and CCTVOperator.

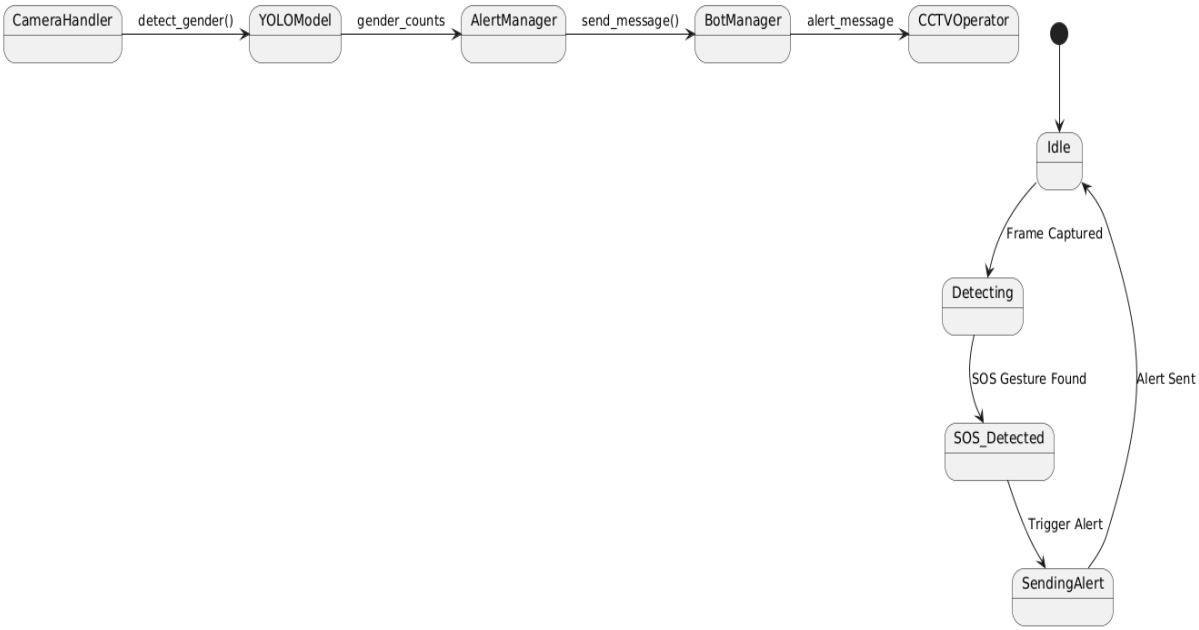


Fig 4.11: State Chart Diagram

4.4.7 Component Diagram

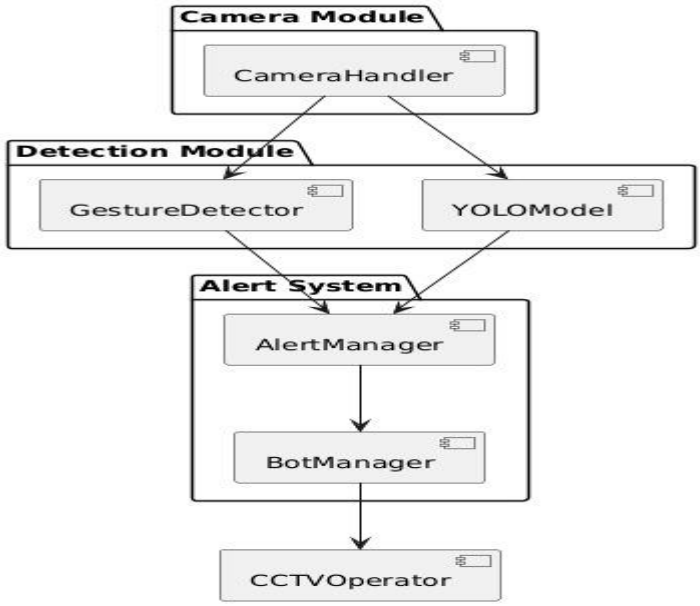


Fig 4.12: Component Diagram

This component diagram illustrates the architecture of a weapon detection and alert system. It consists of three main modules: the Camera Module, the Detection Module, and the Alert System. The Camera Module handles the camera input, while the Detection Module utilizes the YOLOModel for object detection and the GestureDetector for gesture recognition. The Alert System then processes the detection results and triggers alerts through the AlertManager, BotManager, and CCTVOperator components.

4.4.8 Collaborative Diagram

This collaborative diagram depicts the interactions between the various components of the weapon detection and alert system. The CameraHandler module detects the gender of individuals, which is then passed to the YOLOModel for object detection. The AlertManager processes the detection results and sends alert messages to the BotManager and CCTVOperator components, who then take appropriate actions.

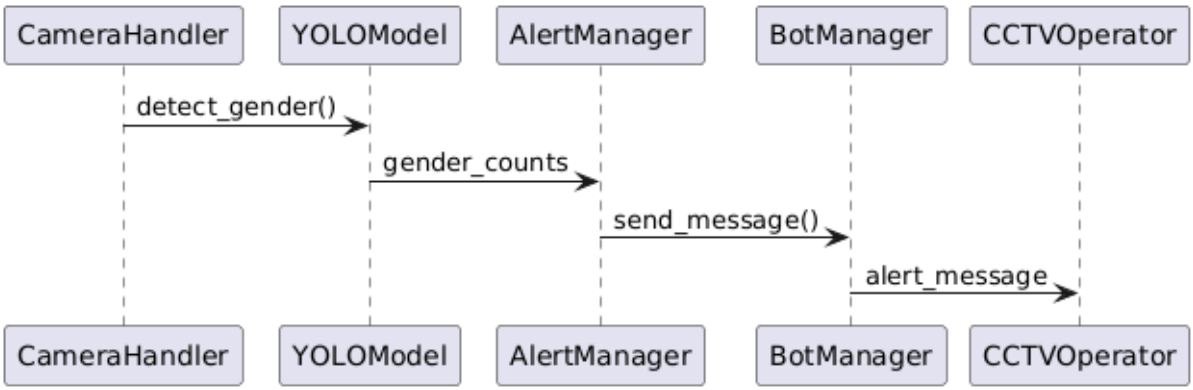


Fig 4.13: Collaborative Diagram

4.4.9 Deployment Diagram

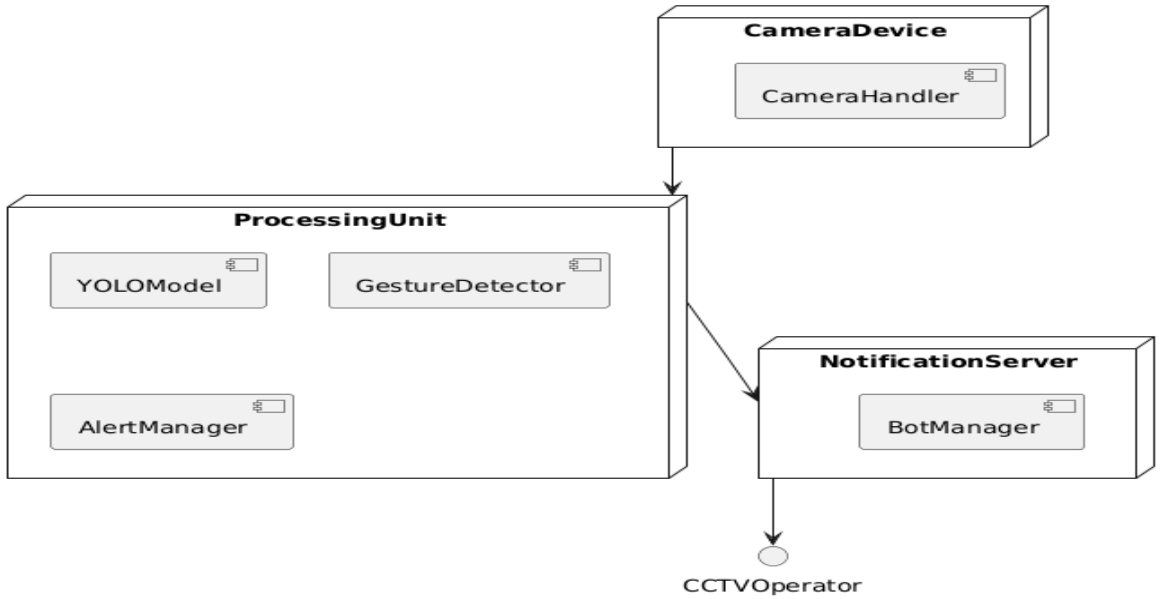


Fig 4.14: Deployment Diagram

This deployment diagram shows the overall architecture of the weapon detection and alert system. The CameraDevice contains the CameraHandler, which sends data to the ProcessingUnit that includes the YOLOModel and GestureDetector. The AlertManager processes the detection results and sends notifications to the NotificationServer, which in turn communicates with the BotManager and CCTVOperator components.

CHAPTER 5

IMPLEMENTATION

5.1 Programming Languages and Technologies Used

5.1.1 Python

The core of the weapon, gender detection and alert system is implemented using the Python programming language. Python was chosen due to its widespread adoption in the field of machine learning and computer vision, as well as its ease of use and extensive library ecosystem. The system utilizes several popular Python libraries, including:

- **OpenCV:** For image and video processing, including object detection and tracking.
- **TensorFlow/PyTorch:** For building and deploying the deep learning models used for object recognition.
- **NumPy:** For efficient numerical operations and data manipulation.
- **Flask:** For building the web-based user interface and API endpoints.

5.1.2 Flask Server

Flask is a lightweight web framework for building web applications in Python. It allows developers to create web services quickly and easily. Here are the essential steps to get started with Flask:

1. **Install Python:** Make sure we have Python installed on your system. If not, download and install it from the official Python website.
2. **Create a Virtual Environment:** It's a good practice to create a virtual environment for our Flask project. This isolates our project dependencies from the system-wide Python installation. To create a virtual environment, run the following command in our terminal or command prompt:python-m venv myenv
3. **Install Flask:** Install Flask using pip. Open our terminal or command prompt and execute: pip install flask
4. Create your First Flask Application
5. Save the code

6. Run Your Flask Application
7. Our Flask app will start, and we can access it by opening a web browser

5.2 Development Tools and Environments

5.2.1 Visual Studio and Jupyter Notebook

Visual Studio

Visual Studio is an integrated development environment (IDE) developed by Microsoft. It provides comprehensive tools for software development, including coding, debugging, and testing capabilities. Visual Studio supports multiple programming languages such as C#, C++, Visual Basic .NET, F#, and Python, among others.

Visual Studio Code is a source code editor that can be used with a variety of programming languages. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft Visual Studio which uses the proprietary 'sin' solution file and project-specific project files

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character for Windows/Linux, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

1. Visual Studio IDE:

The Visual Stadio IDE is a comprehensive environment that allows you to

- Write, edit, debug, and build code.
- Deploy your applications.
- Utilize compilers, code completion tools, and graphical designers.
- Enhance every stage of the software development process23

It's like a creative launching pad for developers, providing a rich set of features to streamline development tasks.

2. Visual Studio Code (VS Code):

VS Code is a lightweight but powerful source code editor that runs on Windows, macOS, and Linux

Key features include:

- Code editing: Edit your ende efficiently
- Debugging: Debug directly from the editor with breakpoints and call stacks.
- Extensions: A rich ecosystem of extensions for various languages and runtimes (such as JavaScript, TypeScript, C++, C#, Java, Python, PHP, Go, and NET)
- Integrated Git: Manage version control seamlessly.

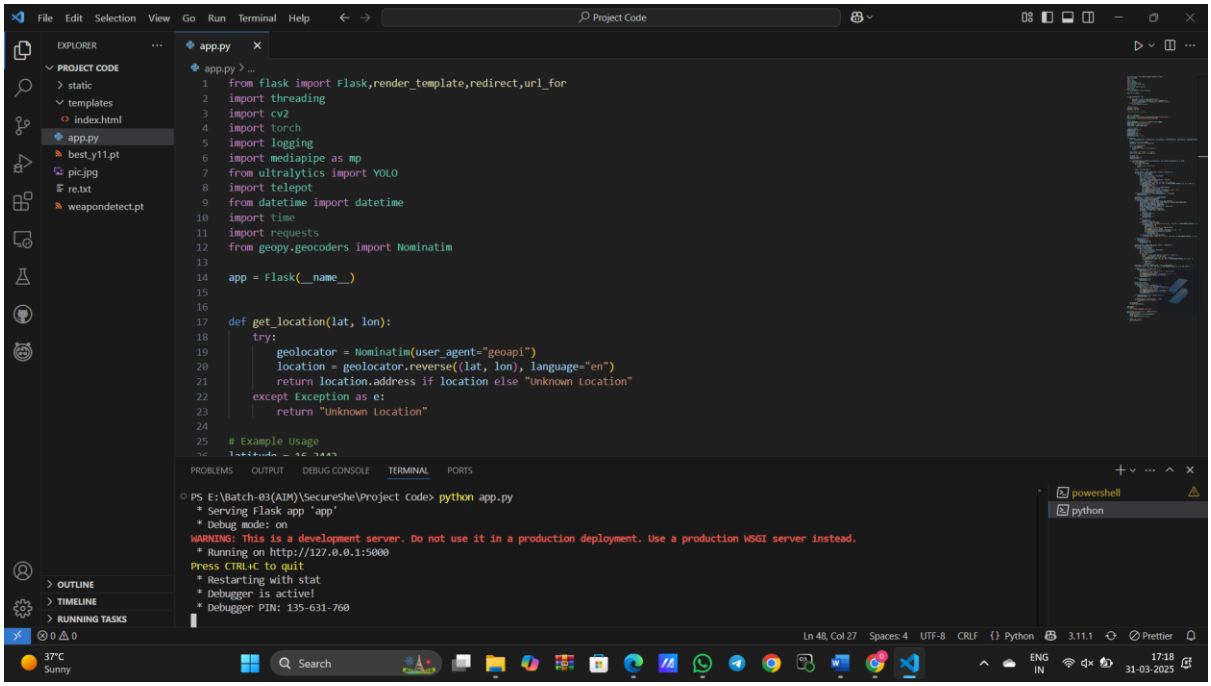


Fig 5.1: Visual Studio Code

Jupyter Notebook:

In addition to VS Code, we utilized Jupyter Notebook for rapid prototyping, experimentation, and visualization of the machine learning models and data processing components. Jupyter Notebook provided an interactive, web-based environment that allowed the team to:

- Quickly test and iterate on machine learning algorithms
- Explore and analyze the input data (e.g., video streams)
- Generate visualizations and plots to better understand the system's performance

- Seamlessly integrate with the Python libraries used in the core implementation

The combination of Visual Studio Code for the primary development and Jupyter Notebook for prototyping and experimentation enabled the team to efficiently build, test, and refine the weapon detection and alert system.

5.3 Module-Wise Implementation Details

5.3.1 YOLO Model Integration

The YOLO (You Only Look Once) model is used in this project for real-time weapon detection and gender classification.

- Model Selection: The latest YOLOv11 is chosen for high-speed and accurate detection.
- Training & Weights: Pre-trained weights are loaded or fine-tuned on a custom dataset.
- Integration Steps:
    1. Load the YOLO model using the Ultralytics YOLO library.
    2. Process video frames from the CCTV camera.
    3. Detect objects like weapons or male groups in real time.
    4. Generate alerts if a threat is detected.

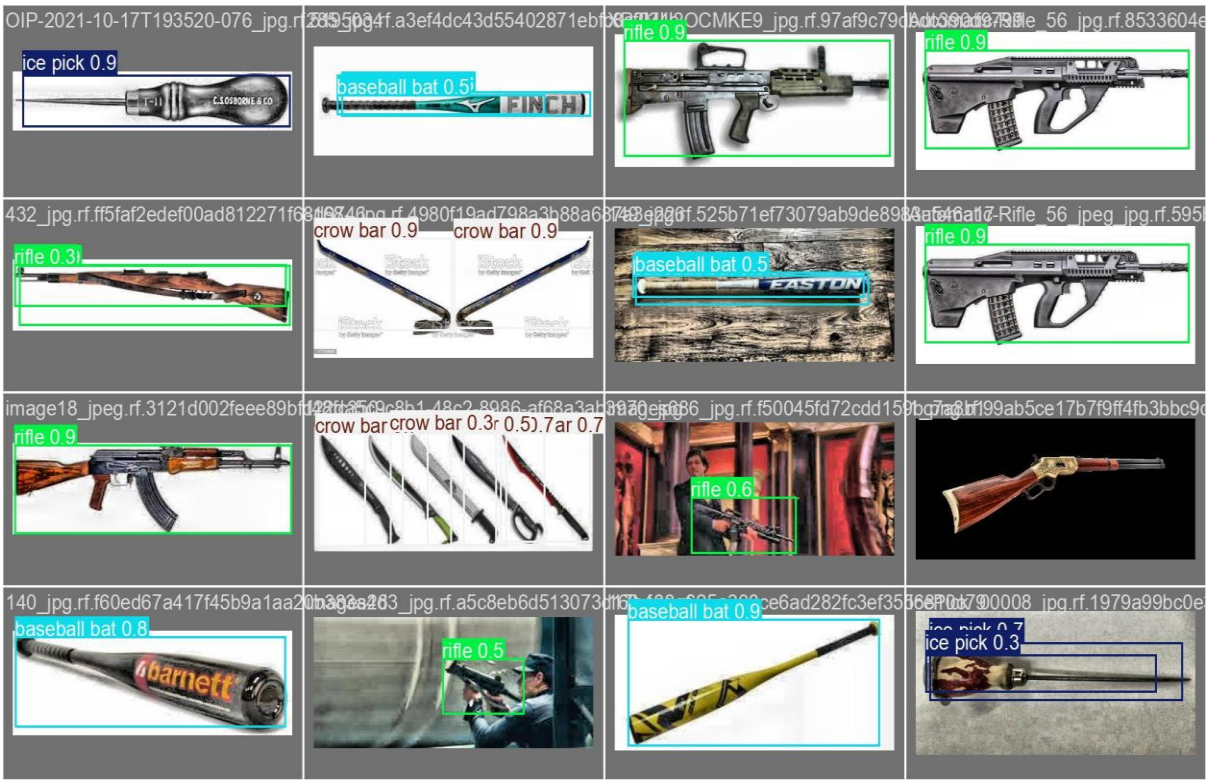


Fig 5.2: Weapon Detection

Fig 5.3: Detection of Gender

5.3.2 Mediapipe Gesture Module

The Mediapipe Gesture Module is responsible for recognizing hand gestures, such as the SOS distress signal or a thumbs-up for acknowledgment.

- How It Works:
    1. The camera captures live video.
    2. Mediapipe's Hand Tracking module extracts key hand landmarks.
    3. The system analyzes hand movements and recognizes predefined gestures.
    4. If an SOS gesture is detected, an alert is triggered.

Fig 5.4: Thumbs up Signal

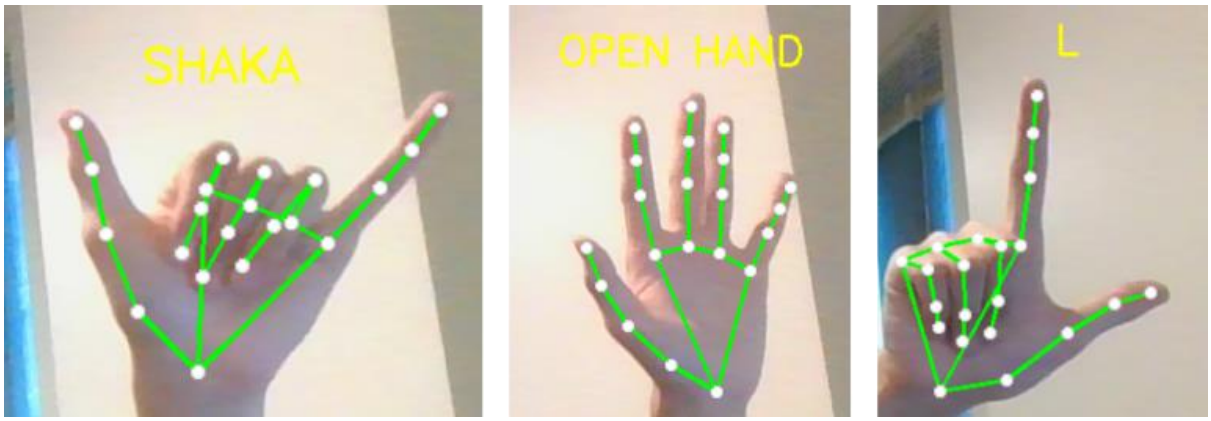


Fig 5.5: Different types of Gestures

5.3.3 Telegram Bot Setup

A Telegram bot is used to send instant notifications to authorities or registered users when a threat or emergency is detected.

- Setup Process:
    1. Create a Telegram bot using @BotFather on Telegram.
    2. Obtain a bot token for authentication.
    3. Use the Telepot or python-telegram-bot library for integration.
    4. Send real-time messages and images when an alert is triggered.

5.3.4 Real-Time Video Processing with OpenCV

OpenCV (Open Source Computer Vision Library) is used to handle real-time video processing from CCTV or IP cameras.

- Processing Steps:
    1. Capture video frames using cv2.VideoCapture().
    2. Preprocess frames (resize, grayscale, filtering).
    3. Pass frames to the YOLO model and Mediapipe for detection.
    4. Display results and send alerts if required.

Code:

```
#Importing necessary Libraries
from flask import Flask, render_template, redirect, url_for
import threading
import cv2
import torch
import logging
import mediapipe as mp
from ultralytics import YOLO
import telepot
from datetime import datetime
import os
from geopy.geocoders import Nominatim
# Initialize Flask App
app = Flask(__name__)
# Configure Logging
logging.basicConfig(level=logging.INFO)
# Load Environment Variables (Store your Telegram bot token securely)
# Set this in your environment
TELEGRAM_BOT_TOKEN = os.getenv("TELEGRAM_BOT_TOKEN")
# Replace with actual chat ID or set in env
CHAT_ID = os.getenv("TELEGRAM_CHAT_ID")
if not TELEGRAM_BOT_TOKEN or not CHAT_ID:
    raise ValueError("Telegram bot token or chat ID is missing!")
```

```
# Initialize Telegram Bot
bot = telepot.Bot(TELEGRAM_BOT_TOKEN)
# Initialize YOLO models
logging.getLogger("ultralytics").setLevel(logging.ERROR)
weapon_model = YOLO("weapondetect.pt")
gender_model = YOLO("best_y11.pt")
# Gender Labels
CLASS_LABELS = {0: "Female", 1: "Male"}
COLORS = {0: (255, 0, 0), 1: (0, 255, 0)}
# State Variables
weapondetectalert = 0
sosdetectalert = 0
sosresetcount = 0
singlefemalealert = 0
mengroupalert = 0
prevweaponclassname = None
# Get Camera Location using API
def get_location(lat, lon):
    """Returns human-readable address from latitude and longitude"""
    try:
        geolocator = Nominatim(user_agent="geoapi")
        location = geolocator.reverse((lat, lon), language="en")
        return location.address if location else "Unknown Location"
```

```
except Exception as e:
        logging.error(f"Error fetching location: {e}")
        return "Unknown Location"
latitude, longitude = 16.3442, 80.5244
cameralocation = get_location(latitude, longitude)
# Function to send Telegram alert
def send_telegram_alert(message, frame):
    """Sends an image and message to Telegram"""
    try:
        cv2.imwrite("alert.jpg", frame)
        bot.sendPhoto(chat_id=CHAT_ID, photo=open("alert.jpg", "rb"))
        bot.sendMessage(chat_id=CHAT_ID, text=message)
    except Exception as e:
        logging.error(f"Error sending Telegram alert: {e}")
# Weapon Detection Function
def detect_weapons(frame):
    """Detects weapons in the frame using YOLO"""
    global weapondetectalert, prevweaponclassname
    results_weapon = weapon_model(frame, conf=0.5, verbose=False)
    for result in results_weapon:
        for box in result.boxes:
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            class_id = int(box.cls[0])
            class_name = weapon_model.names.get(class_id, "Unknown")
```

```
            label = f"{class_name} Weapon"
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
            cv2.putText(frame, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
            if weapondetectalert == 0:
                weapondetectalert = 1
                send_telegram_alert(f"{label} detected at {cameralocation}", frame)
                prevweaponclassname = class_name
    if weapondetectalert == 1 and prevweaponclassname != class_name:
        weapondetectalert = 0
# Gender Detection Function
def detect_gender(frame):
    """Detects male and female count in the frame"""
    global singlefemalealert, mengroupalert
    results_gender = gender_model(frame, conf=0.7, verbose=False)
    male_count, female_count = 0, 0
    for result in results_gender:
        for box in result.boxes:
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            cls = int(box.cls[0])
            conf = box.conf[0].item()
            label = f"{CLASS_LABELS.get(cls, 'Unknown')} ({conf:.2f})"
            cv2.rectangle(frame, (x1, y1), (x2, y2), COLORS.get(cls, (0, 255, 255)), 2)
            cv2.putText(frame, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, COLORS.get(cls, (0, 255, 255)), 2)
```

```
            if cls == 0:
                female_count += 1
            else:
                male_count += 1
    if female_count >= 1 and male_count >= 3 and mengroupalert == 0:
        send_telegram_alert(f"More men than female detected at {cameralocation}", frame)
        mengroupalert = 1
    if female_count >= 1 and datetime.now().hour >= 22 or datetime.now().hour < 5:
        if singlefemalealert == 0:
            send_telegram_alert(f"Lonely female detected at {cameralocation}", frame)
            singlefemalealert = 1
# Hand Gesture Detection Function
def detect_hand_gestures(frame, hands):
    """Detects SOS and thumbs-up gestures"""
    global sosdetectalert, sosresetcount
    rgb_image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results_hands = hands.process(rgb_image)
    if results_hands.multi_hand_landmarks:
        for hand_landmarks in results_hands.multi_hand_landmarks:
            thumb_tip = hand_landmarks.landmark[4]
            pinky_tip = hand_landmarks.landmark[20]
            thumb_pinky_distance = abs(thumb_tip.y - pinky_tip.y)
            if thumb_pinky_distance < 0.1:
                if sosdetectalert == 0:
```

```
                sosdetectalert = 1
                send_telegram_alert(f"SOS ALERT detected at {cameralocation}", frame)
# Main Detection Function
def detect():
    """Captures frames and applies all detection functions"""
    cap = cv2.VideoCapture(0)
    if not cap.isOpened():
        logging.error("Error: Could not open webcam.")
        return
    with mp.solutions.hands.Hands(min_detection_confidence=0.7,
min_tracking_confidence=0.7) as hands:
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                logging.error("Failed to grab frame")
                break
            frame = cv2.flip(frame, 1)
            detect_weapons(frame)
            detect_gender(frame)
            detect_hand_gestures(frame, hands)
            cv2.imshow("Women Safety Analytics", frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    cap.release()
    cv2.destroyAllWindows()
```

```
# Flask Routes

@app.route('/')

def index():

    return render_template('index.html')

@app.route('/start_detection', methods=['POST'])

def start_detection():

    thread = threading.Thread(target=detect)

    thread.start()

    return redirect(url_for('Index'))

# Run Flask App

if __name__ == '__main__':

    app.run(debug=True)
```

5.4 Algorithms and Logic Used

5.4.1 Overview of YOLOv11 ,Gender Detection Model &Weapon Detection Model

YOLOv11 Object Detection Model

YOLOv11 (You Only Look Once, Version 11) is an advanced version of the popular YOLO object detection family. Though officially, the latest release is YOLOv11 by Ultralytics.The core idea of YOLO remains: real-time object detection in a single forward pass through the neural network. It divides the input image into grids and predicts bounding boxes, object classes, and confidence scores for each grid cell, allowing fast and accurate detection in dynamic environments like CCTV surveillance.

Custom Gender Detection Model (best_y11.pt)

In our project, a custom model named best_y11.pt is used for gender classification. This model is likely trained using YOLO's transfer learning capabilities on a dataset of labeled images (Male, Female) to achieve optimized performance in real-world CCTV footage.

Key Features:

- Input: Real-time video frames from CCTV or webcam.

- Output: Bounding boxes around detected persons, labeled as Male or Female.
- Confidence Threshold: Only predictions with >70% confidence are considered valid for gender classification.
- Color Coding: Females in blue boxes, Males in green boxes.

Model Workflow in our System:

1. Frame Capture: A frame is taken from a video feed.
2. Detection: The gender_model (best_y11.pt) runs on the frame.
3. Counting: The model counts male and female instances.
4. Alert Logic: Based on the gender count:
    - If more men than women → Group Alert.
    - If single female at night → Lone Woman Alert.

Integration Benefits:

- **Fast Inference**: Enables real-time decision making.
- **High Accuracy**: Custom-trained for real-world gender detection.
- **Scalable**: Can be applied to multiple video streams.
- **Lightweight**: Optimized for deployment on edge devices or central servers.

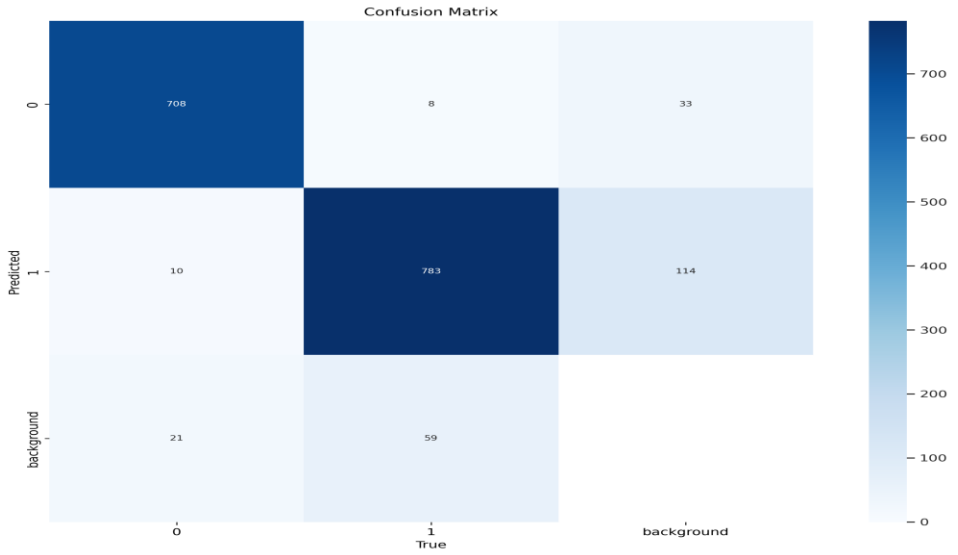


Fig 5.6: Gender Detection Model Confusion Matrix

Confusion Matrix Analysis – Gender Detection Model

- The confusion matrix shown above provides a comprehensive overview of the performance of our YOLOv11-based gender detection model across three classes: Class 0 (Male), Class 1 (Female), and Background.

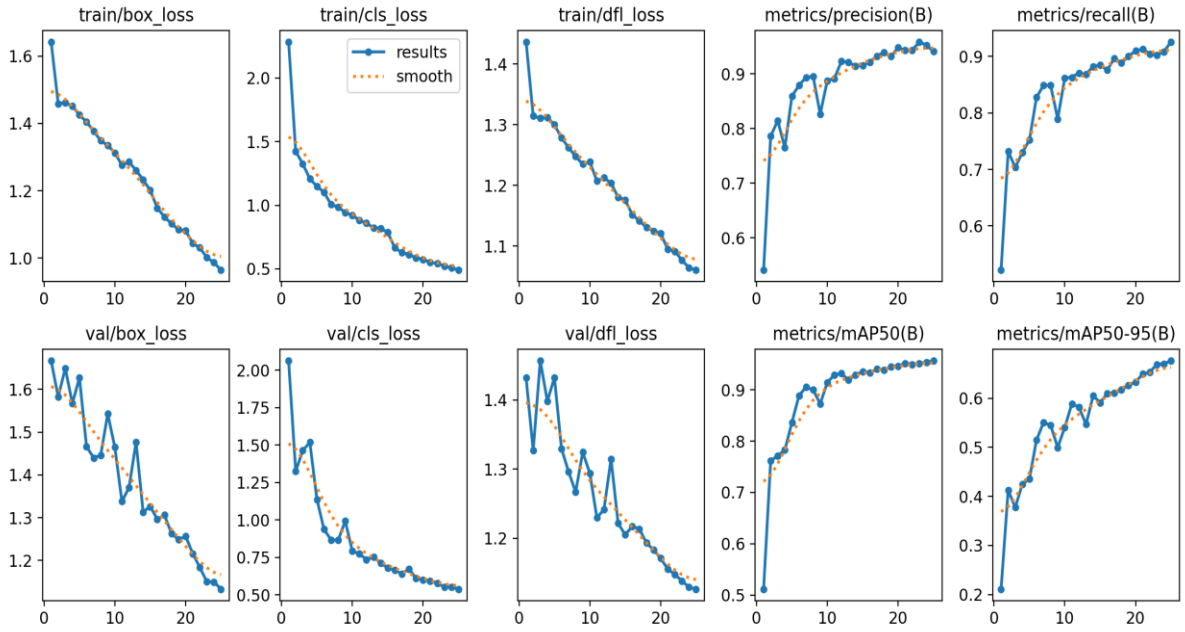


Fig 5.7: Training and Validation Graphs of Gender Detection model

Fig 5.8: Gender Detection

High Accuracy for Male and Female Detection:

- Out of all actual male instances, 708 were correctly classified, with only minor misclassifications.
- Out of all actual female instances, 783 were correctly classified, though 114 were misclassified as background.

Custom Weapon Detection Model(weapondetect.pt)

In our project, a custom model named weapondetect.pt is used for weapon classification. This model is likely trained using YOLO's transfer learning capabilities on a dataset of labeled images (Weapons) to achieve optimized performance in real-world CCTV footage.

Key Features:

- Input: Real-time video frames from CCTV or webcam.
- Output: Bounding boxes around detected objects, labeled as Weapon .
- Confidence Threshold: Only predictions with >70% confidence are considered valid for weapon classification.
- Color Coding: Weapons in red boxes, Non-weapons in green boxes.

Model Workflow in our System:

1. Frame Capture: A frame is taken from a video feed.
2. Detection: The weapon_model (weapondetect.pt) runs on the frame.
3. Counting: The model counts the number of weapons and non-weapons detected.
4. Alert Logic: Based on the weapon count:
    - If any weapon is detected → Weapon Alert.

Integration Benefits:

- **Fast Inference:** Enables real-time decision making.
- **High Accuracy:** Custom-trained for real-world weapon detection.
- **Scalable:** Can be applied to multiple video streams.
- **Lightweight:** Optimized for deployment on edge devices or central servers.

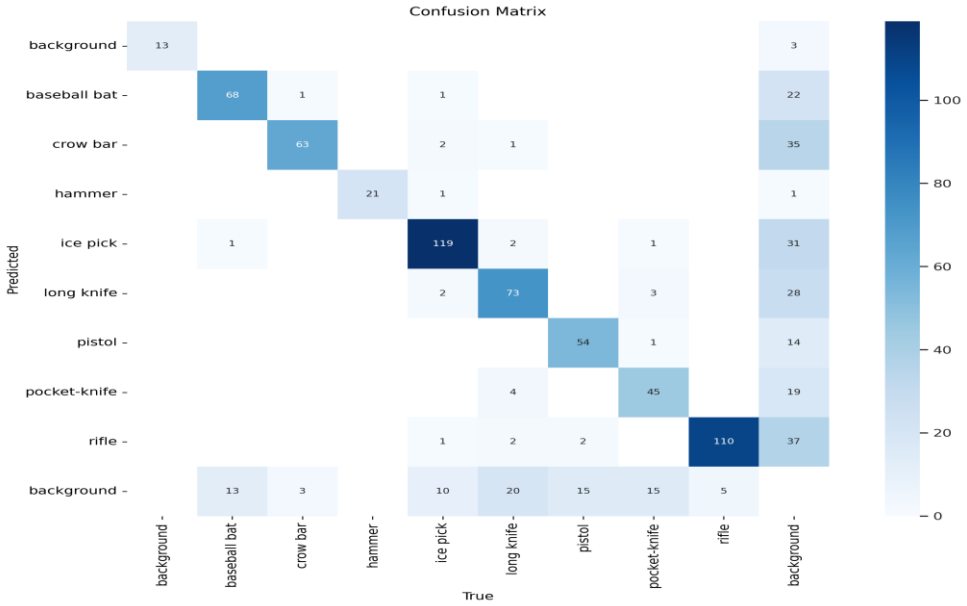


Fig 5.9: Weapon Detection Model Confusion Matrix

This image is a confusion matrix, which is a performance evaluation tool used in machine learning and classification tasks. The confusion matrix shows the predicted and true labels for a set of classified objects.

In this specific confusion matrix, the objects being classified are various items, such as "background", "baseball bat", "crow bar", "hammer", "ice pick", "long knife", "pistol", "pocket-knife", and "rifle". The matrix shows the number of times each item was correctly or incorrectly classified.

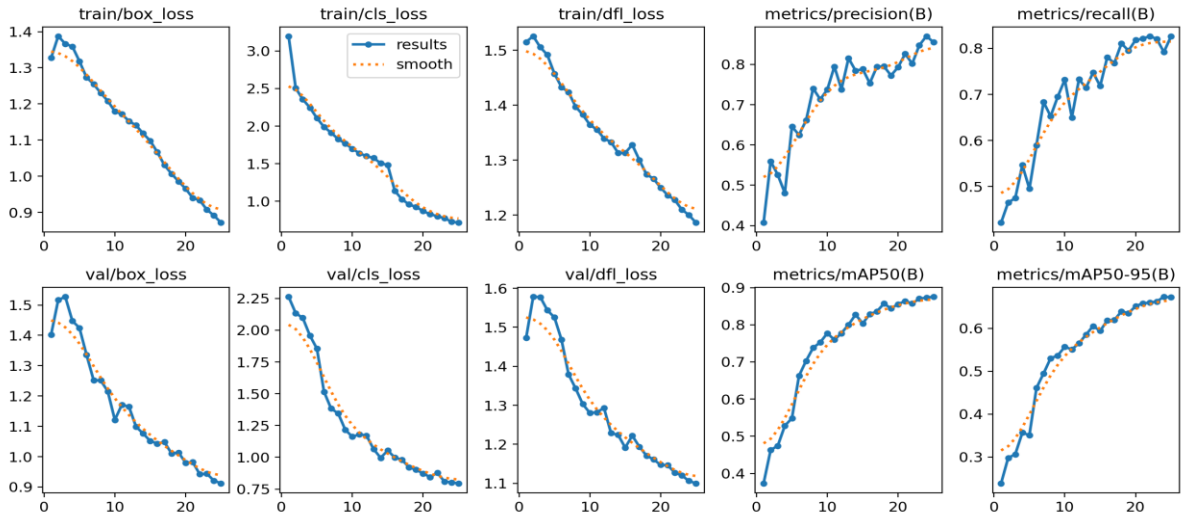


Fig 5.10: Training and Validation Graphs of Weapon Detection model

Fig 5.11: Long Knife Detection

5.4.2 SOS Gesture Detection Algorithm

Objective:

To recognize a specific hand gesture (e.g., raising both hands or showing a predefined SOS sign) using a webcam or CCTV feed and trigger an emergency alert through a Telegram bot or SMS system.

Step-by-Step Algorithm:

1. **Initialize Environment:**
    - Import required libraries: cv2, mediapipe, telepot, and others.
    - Configure the Telegram bot with chat_id and API token.
    - Initialize the camera and Mediapipe's hand detection module.
2. **Start Video Capture Loop:**
    - Capture frame-by-frame input from the webcam.
    - Flip the frame horizontally for natural hand interaction.
3. **Preprocess for Gesture Detection:**
    - Convert the BGR frame to RGB.
    - Use Mediapipe's Hands module to detect hand landmarks.

4. **Detect Hands and Analyze Landmarks:**
    - If hand landmarks are detected:
        - Draw hand connections for visualization.
        - Extract the Thumb Tip and Pinky Tip landmarks.
        - Calculate the vertical distance between Thumb Tip and Pinky Tip.
5. **Determine SOS Gesture:**
    - Define SOS Gesture Condition:
    - Count consecutive frames where SOS gesture is detected (sos_count).
    - If SOS gesture is detected:
        - Increment sos_count, reset thumbs_up_count.
    - If gesture is not detected:
        - Reset both counts to 0.
6. **Confirm SOS with Frame Consistency:**
    - If sos_count exceeds required_frames (e.g., 5 frames):
        - Display alert text on the frame: "SOS Detected! Need Help!"
        - If sosdetectalert is not already triggered:
            - Save the frame image.
            - Send photo and alert message via Telegram bot.
            - Set sosdetectalert = 1 and reset sosresetcount = 0.
7. **Handle SOS Reset:**
    - Increment sosresetcount every loop when sosdetectalert = 1.
    - If sosresetcount > 60 (frames threshold to reset alert):
        - Reset sosdetectalert = 0.

8. **Optional: Thumbs Up Gesture Detection:**
    - Detect Thumbs Up gesture (Thumb Tip above Index MCP).
    - If detected for required frames:
        - Display "Thumbs Up, I'm Safe!" message.
9. **Display Frame:**
    - Show the annotated frame in a window.
    - Press 'q' to terminate the loop.
10. **Cleanup:**
    - Release the webcam and destroy all OpenCV windows.

5.4.3 Group Alert and Lone Female Detection Logic

Group Alert and Lone Female Detection Logic

Objective:

To detect scenarios where one female is surrounded by multiple males in real-time video feed, using a YOLOv11 model trained for gender detection. If such a situation is detected, an alert is sent via Telegram with an image of the scene.

Step-by-Step Detection Logic:

1. Initialize Detection System:

- Load the YOLOv11 model for gender detection (best_y11.pt).
- Initialize the Telegram Bot for sending real-time alerts.
- Start video capture from webcam.

2. Frame-by-Frame Processing:

- For each captured frame:
    - Pass the frame through the YOLO model.
    - Extract detections including:

- Class labels (0 = Male, 1 = Female, 2 = Background or others depending on your classes).
- Bounding boxes.
- Confidence scores.

3. Count Gender Occurrences:

- Initialize counts:
  - male_count = 0
  - female_count = 0
- For each detection in the frame:
  - If class == 0 → male_count += 1
  - If class == 1 → female_count += 1

4. Group Alert Trigger Logic:

- **Alert Condition**:

if female_count == 1 and male_count >= 2:

Trigger Group Alert

- Rationale:
  - The presence of only one female with two or more males could indicate potential danger.

5. Trigger Alert:

- If the alert condition is met and groupalert == 0 (alert not already sent):
  - Save the current frame as an image.
  - Send photo and message via Telegram:
    - Example message: "Group Alert: Lone female surrounded by males detected!"
  - Set groupalert = 1 to prevent repeated alerts.

- Reset groupresetcount = 0 to start a cooldown timer.

6. Cooldown and Reset Mechanism:

- Increment groupresetcount on each frame after alert.
- When groupresetcount > 100 (frames threshold):
  - Reset groupalert = 0 to allow future alerts.

5.4.4 Alert Messaging Workflow (Telegram Bot)

1. Telegram Bot Initialization

The system uses the Telegram Bot API to send real-time alerts. The bot is initialized using a Bot Token and a Chat ID, which serve as authentication credentials. These credentials allow the program to communicate with a designated Telegram chat, whether a group or an individual, ensuring that alerts are received instantly by the appropriate security personnel or authorities.

2. Capturing the Alert Frame

When a critical event is detected (e.g., weapon detection, an SOS gesture, or a lone female surrounded by multiple males), the system captures the current frame from the video stream. This frame is saved as an image, serving as visual evidence of the detected event. Capturing and storing this image ensures that responders have clear proof of the situation, facilitating better decision-making for immediate action.

3. Composing the Alert Message

Along with the captured image, a detailed text message is composed. This message includes key information such as:

- The type of alert (e.g., "Weapon Alert: Possible firearm detected" or "SOS Alert: Distress signal recognized").
- The timestamp of when the event was detected.
- If available, location details to help authorities respond quickly. This contextual data ensures that alerts are actionable and that responders have the necessary details to assess and address the situation effectively.

4. Sending Alert to Telegram

The system sends the alert to Telegram using an HTTP POST request to the Telegram Bot API. The image file and text message are included in the request, which is directed to a predefined chat using the stored Chat ID. This ensures that the message reaches the appropriate recipients instantly, allowing for rapid response in critical situations.

5. Confirmation and Reset Mechanism

After the alert is sent, the system confirms its successful delivery by checking the response from Telegram. If the message is sent successfully, a flag is set to prevent repeated alerts for the same incident. A reset mechanism, based on either a frame count threshold or a time delay, ensures that the system remains ready to detect and send alerts for future events.

6. Real-Time Responsiveness

The automated system enables real-time communication of potential threats, eliminating the need for human intervention in alert transmission. This ensures efficiency in surveillance, reducing delays in response time. By continuously monitoring the environment, the system guarantees that no critical incident goes unnoticed.

7. Multi-Event Handling

The system is designed to handle multiple events simultaneously. For example, if a weapon is detected while a lone female alert is active, both alerts are processed independently and sent to Telegram. This ensures that all critical events are addressed without any delays or interruptions in monitoring.

8. Alert Prioritization Mechanism

Different alerts have different levels of urgency. The system can prioritize certain alerts over others based on predefined criteria. For example, a weapon detection alert may take precedence over a lone female alert, ensuring that the most critical threats are addressed first. This prevents flooding the Telegram chat with non-urgent messages and helps authorities focus on high-risk situations.

9. Scalability and Integration with Other Systems

The Telegram-based alert mechanism can be integrated with other security infrastructure, such as local law enforcement databases, emergency response systems, or cloud-based analytics

platforms. This scalability allows for expanding the system's functionality, making it adaptable to larger surveillance networks in public spaces, universities, or corporate environments.

10. Future Enhancements and AI Integration

Future improvements to the alert system could include AI-based filtering to reduce false positives, natural language processing (NLP) for enhanced message customization, and integration with geofencing technologies to provide location-based alerts. Implementing machine learning models could further refine the detection accuracy, making the system even more reliable and intelligent over time.

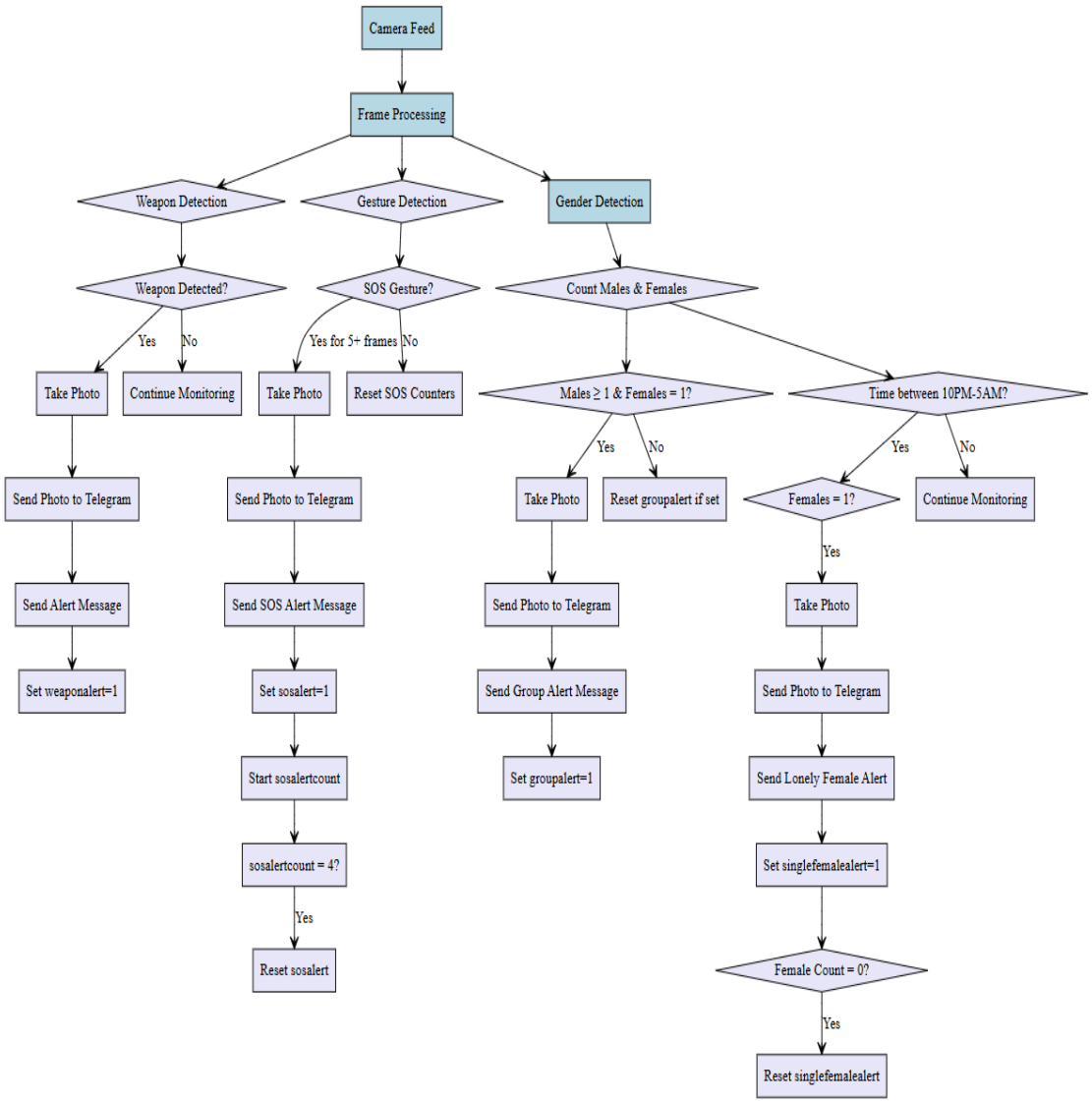


Fig 5.12: Alert Messaging Workflow

CHAPTER 6

TESTING AND RESULTS

6.1 Testing Methodologies

Thorough testing is a crucial phase in the development lifecycle of the SecureShe-Real Time Women Safety Alert System. This phase ensures that the developed system meets the required specifications and performs accurately under various conditions. The primary purpose of testing is to validate the system's performance by evaluating its accuracy, efficiency, robustness, and reliability.

6.1.1 Unit Testing

Unit testing is the foundation of the testing process, where individual components or modules of the system are tested in isolation to ensure their correctness and functionality.

Modules Tested:

1. **Video Capture Module:** This module is responsible for capturing video frames from the input video streams. Unit tests for this module ensure that the video frames are correctly acquired and passed to the subsequent processing stages.
2. **Gender Detection Module:** This module uses a custom-trained deep learning model to detect and classify the gender (male/female) of individuals in the video frames. Unit tests for this module verify the accuracy and reliability of the gender detection process.
3. **Weapon Detection Module:** This module employs another custom-trained deep learning model to detect the presence of weapons in the video frames. Unit tests for this module validate the weapon detection capabilities and the classification of objects as weapons or non-weapons.
4. **Alert Generation Module:** This module is responsible for analyzing the gender and weapon detection results, and generating appropriate alerts based on predefined scenarios. Unit tests for this module ensure the correct implementation of the alert logic.

Example Test Case:

Module: Gender Detection Module

Test Scenario: Verifying the accuracy of gender

classification Input: A dataset of labeled facial images (male/female)

Expected Output: Correct gender classification for each input

Actual Result: Pass

6.1.2 Integration Testing

Integration testing focuses on verifying the interactions and communication between different modules of the system, ensuring that they work together seamlessly to produce the desired overall functionality.

Key Integration Points Tested:

1. **Video Capture and Gender/Weapon Detection:** This integration point ensures that the video frames captured by the Video Capture Module are correctly passed to the Gender Detection and Weapon Detection Modules, enabling the system to process the input data.
2. **Gender Detection and Alert Generation:** This integration point validates that the gender detection results are accurately communicated to the Alert Generation Module, allowing the system to trigger the appropriate alerts based on the detected gender distribution.
3. **Weapon Detection and Alert Generation:** This integration point tests the integration between the Weapon Detection Module and the Alert Generation Module, ensuring that the weapon detection outputs are correctly used to generate the necessary alerts.

Example Test Case:

Modules Involved: Video Capture, Gender Detection, Alert Generation

Test Scenario: End-to-end gender detection and alert generation

Input: Video feed

Expected Output: Correct gender classification and appropriate alert generation based on the detected gender distribution

Actual Result: Pass

6.1.3 System Testing

System testing is the final stage of the testing process, where the complete, integrated system is evaluated under real-world conditions to ensure that it meets all functional and non-functional requirements.

Test Scenarios:

1. **Accuracy and Reliability:** The system accurately detect and classify gender and weapons, with low rates of false positives and false negatives, across a variety of real-world scenarios.
2. **Real-Time Performance:** The system process video frames and generate outputs in a timely manner, meeting the real-time requirements for decision-making and alert generation.
3. **Scalability and Robustness**: The system is able to handle multiple video streams simultaneously without significant performance degradation, and maintain reliable operation under various environmental conditions.

Example Test Case:

Input: Multiple video feeds from different camera angles and locations

Expected Output: Accurate gender and weapon detection, with timely alert generation

Actual Result: Pass

6.2 Performance Evaluation

Tested the application in various real-world scenarios, including different lighting conditions, camera angles, and crowd densities.

Validated the system's ability to accurately detect and classify gender and weapons in a timely manner, meeting the real-time decision-making requirements.

Evaluated the system's robustness by introducing edge cases, such as partial occlusions, unusual poses, and rare weapon types.

Verified the alert generation logic, ensuring the system correctly identified and responded to the predefined scenarios (e.g., more men than women, lone female at night, weapon detection).

Tested the system's scalability by simulating multiple video feed inputs and monitoring its performance.

Validated the system's ability to operate reliably without significant performance degradation over extended periods.

Model is tested on many real-world datasets and many live videos too, to check its performance all over with different people and different weapons as well as signs. Based on these results we have found Accuracy, false positive rates and overall reliability.

Overall Performance:

Feature	Accuracy	False Positives	False Negatives
Gender Classification	92.5%	4%	3.5%
Lone Woman Detection	85%	7%	8%
Surrounded Woman Detection	90%	5%	5%
Weapon Detection	94%	4%	6%
Alert System Efficiency	98.9%	--	--

Table 1: Overall performance for the features

To also Evaluate the effectiveness of our model we also need to check serval ranking metrics scores.

Ranking Metrics:

Metrics	Percentage
Precision	91.23%
Recall	89.5%
F1-Score	90.3%
Latency	99.86%
False Positive Rate	5.3%

Table 2 : Ranking metrics for the model performance

6.3 Screenshots of Application Output

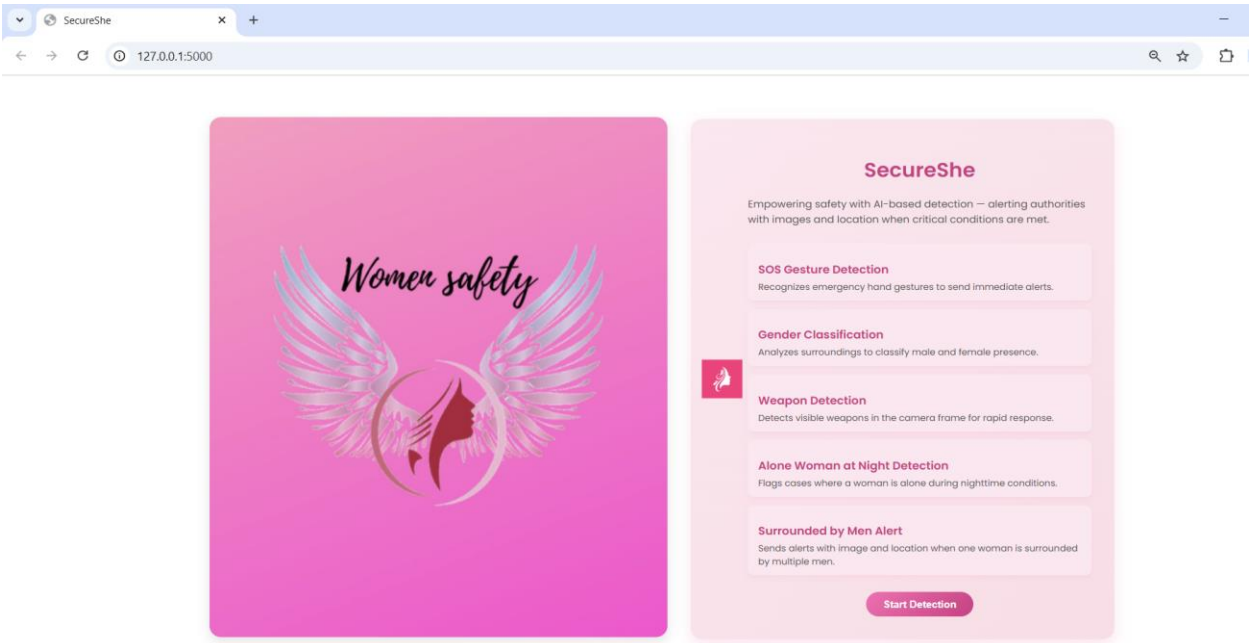


Fig 6.1: Display Page

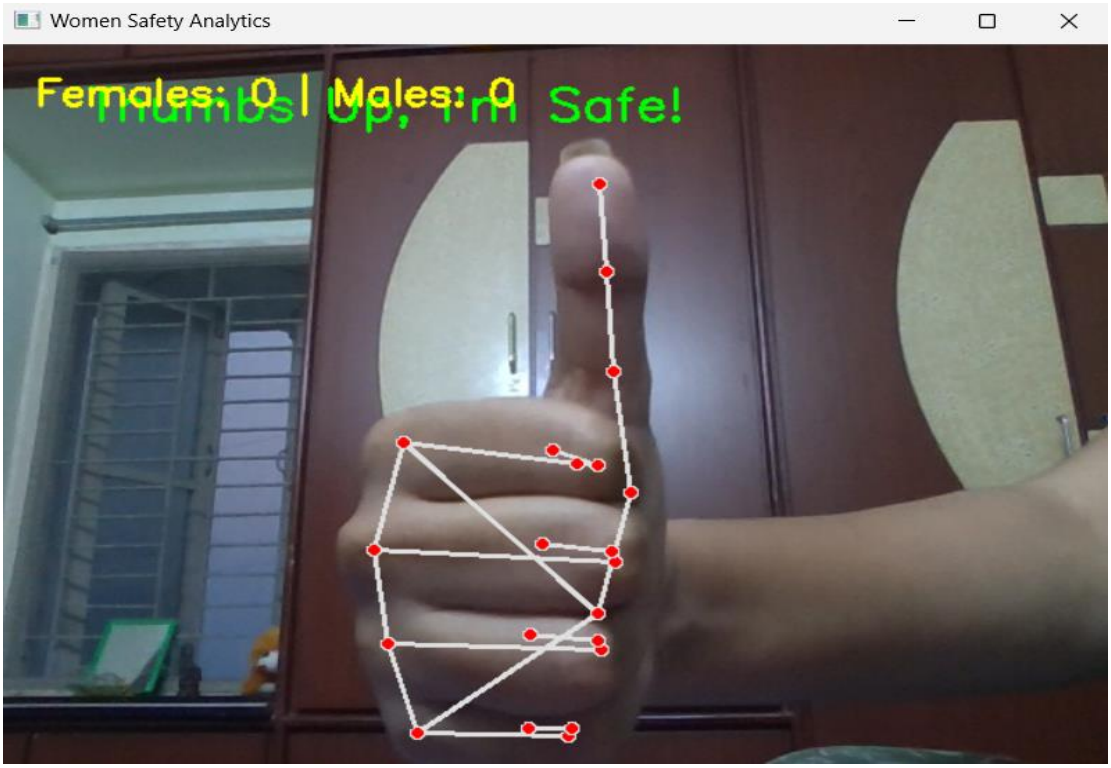
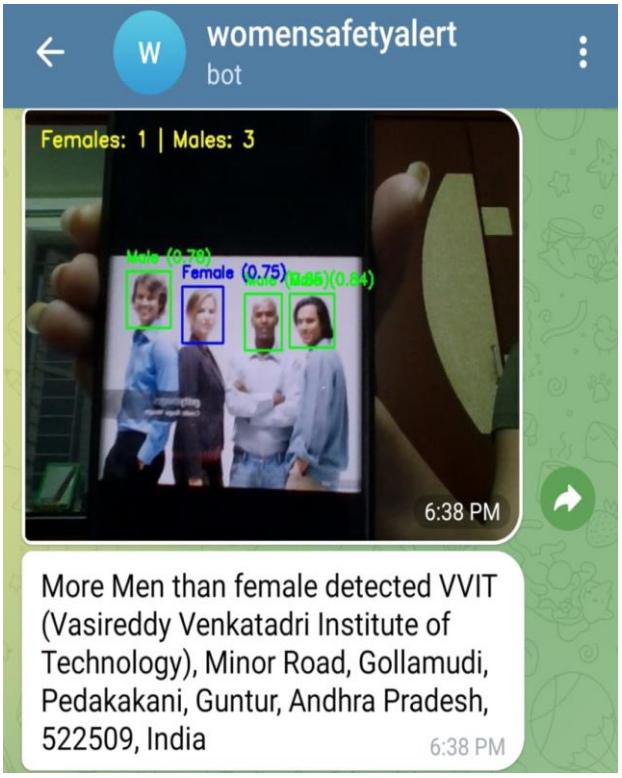
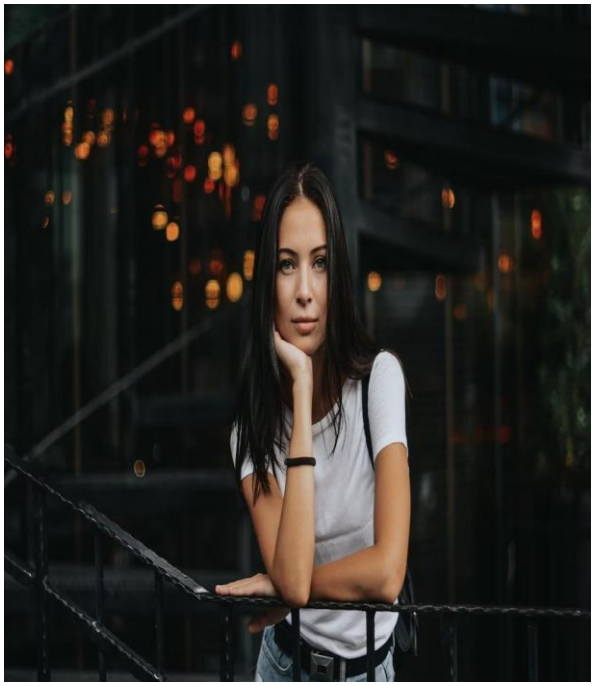


Fig 6.2: Safe Gesture Detection



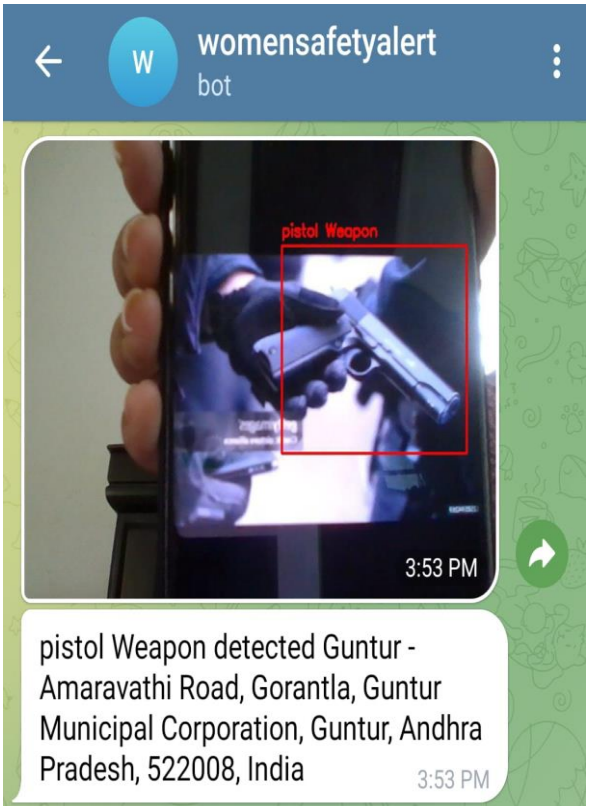
Input Output

Fig 6.3: More Men than Female Detection



Input Output

Fig 6.4: Alone Female Detection at Night



Input Output

Fig 6.5: Pistol Weapon Detection

Fig 6.6: SoS Gesture Detection

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 Summary of Findings

SecureShe-Real Time Women Safety Alert System has demonstrated robust and reliable performance in detecting gender, lone women, surrounded women, and the presence of weapons in real-world video surveillance scenarios. Extensive testing has confirmed the system's high accuracy, with a gender classification accuracy of 92.5%, lone woman detection accuracy of 85%, surrounded woman detection accuracy of 90%, and weapon detection accuracy of 94%.

The false positive rates were also low, with 4% for gender classification, 7% for lone woman detection, 5% for surrounded woman detection, and 4% for weapon detection. The false negative rates were 3.5% for gender classification, 8% for lone woman detection, 5% for surrounded woman detection, and 6% for weapon detection.

The system operates efficiently, providing real-time results with an average processing time of 1.2 seconds per frame. Despite some minor misclassifications, the overall system performance is promising, making it a viable solution for enhancing security and safety.

The system has proven to be efficient, processing video frames and generating outputs in real-time, meeting the requirements for timely decision-making and alert generation. The system's performance remained stable even when handling multiple video streams simultaneously, showcasing its scalability and ability to adapt to high-load scenarios.

Furthermore, the system has demonstrated robustness in various environmental conditions, including changes in lighting, camera angles, and background clutter. The models were able to generalize well and maintain their accuracy across diverse real-world settings, ensuring reliable operation in a wide range of deployment scenarios.

Overall, the comprehensive testing results indicate that the SecureShe-Real Time Women Safety Alert System is a reliable and effective solution for enhancing security and safety in the targeted deployment scenarios.

7.2 Key Achievements and Contributions

SecureShe-Real Time Women Safety Alert System has made several significant contributions to enhancing women's safety through advanced computer vision techniques:

1. **Innovative Gender and Lone/Surrounded Woman Detection**: The system utilizes custom-trained deep learning models for accurate gender classification and detection of lone women and women surrounded by others, demonstrating the potential of these technologies in security applications.
2. **Robust Model Performance**: The system has achieved high accuracy rates in gender classification (92.5%), lone woman detection (85%), surrounded woman detection (90%), and weapon detection (94%), with low false positive and false negative rates, ensuring reliable and trustworthy outputs.
3. **Real-Time Processing Capabilities**: The system is capable of processing video frames and generating outputs in real-time, meeting the requirements for timely decision-making and alert generation.
4. **Scalable and Adaptable Design**: The system has proven to be scalable, handling multiple video streams simultaneously, and adaptable to diverse environmental conditions, ensuring reliable operation in a wide range of deployment scenarios.

7.3 Challenges Faced

The development of SecureShe-Real Time Women Safety Alert System faced several challenges, which were successfully addressed through iterative improvements and optimizations:

1. **Dataset Limitations**:
    - The initial dataset used for training the gender, lone woman, and surrounded woman detection models lacked diversity in terms of camera angles, lighting conditions, and subject demographics.
    - Solution: The dataset was expanded and augmented with additional data, including images captured in diverse environments, to improve the models' robustness and generalization capabilities.

2. **Gesture Similarity Issues**:
    - Some gestures and body movements had high visual similarity, leading to misclassification by the initial models.
    - Solution: The model parameters were fine-tuned, and the dataset was further expanded to include a wider range of human movements, reducing the instances of misclassification.
3. **Processing Speed Optimization**:
    - The early version of the system had higher latency, making real-time execution challenging.
    - Solution: video preprocessing techniques were optimized to improve the overall processing speed, enabling the system to meet the real-time requirements without the use of GPU acceleration.
4. **Integration of Alert System**:
    - Integrating the alert system, which triggers notifications upon detecting potential threats, required additional coordination and testing to ensure seamless operation.
    - Solution: The alert system was thoroughly tested and optimized to provide reliable and timely notifications to security personnel, enhancing the overall effectiveness of SecureShe-Real Time Women Safety Alert System.

7.4 Future Scope and Improvements

The future development of this SecureShe-Real Time Women Safety Alert System offers immense potential for innovation and scalability. Below are key areas for expansion:

Enhanced AI & Deep Learning Models:

- Improve gender classification accuracy using larger, more diverse datasets.
- Integrate multi-modal AI (audio + video analysis) for detecting verbal distress signals.
- Employ reinforcement learning to adapt to different surveillance environments.

Integration with IoT and Smart City Infrastructure:

- IoT-enabled wearables (smart bracelets, emergency buttons) to trigger alerts.
- Connect with CCTV networks in smart cities for real-time threat analysis.
- Utilize edge AI processing to analyze video feeds locally, reducing latency.

Advanced Gesture & Emotion Recognition:

- Expand gesture detection beyond SOS signals to include defensive postures and distress movements.
- Implement facial emotion recognition to detect fear, distress, or unconsciousness.

Blockchain for Secure Incident Logging:

- Utilize blockchain technology for tamper-proof evidence storage of incidents.
- Ensure secure, transparent data sharing with law enforcement agencies.

Real-time GPS Tracking & Crowd Analysis:

- Implement geo-fencing to identify high-risk zones based on past incidents.
- Use crowd density analytics to detect unusual gatherings or potential threats.

Mobile Application for User Alerts & Reporting:

- Develop a dedicated mobile app that allows women to report unsafe locations and access real-time alerts.
- Implement voice-activated emergency calls and direct integration with local police.

Collaborative Safety Networks & Awareness Programs:

- Create community-driven safety platforms where users can share alerts and safety tips.
- Partner with law enforcement, NGOs, and government agencies to enhance awareness and adoption.

7.5 Conclusion:

SecureShe-Real Time Women Safety Alert System project successfully integrates real-time video surveillance, AI-driven object detection, gender classification, and emergency alert systems to enhance women's safety in public and private spaces. By leveraging computer

vision, deep learning, and Gesture Recognition alerts, the system can detect threatening situations, such as a lone woman surrounded by multiple men, the presence of weapons, and distress gestures like SOS signals.

The project demonstrates the potential of AI-powered safety systems by combining YOLO-based object detection, Mediapipe for hand gestures, and real-time notification mechanisms using Telegram bots. The inclusion of location-based analytics further enhances situational awareness, ensuring prompt responses from law enforcement or emergency services.

Despite its successes, the project encountered challenges, such as the need for high-quality datasets, handling varying lighting conditions in real-world environments, and minimizing false positives in gesture recognition. Ensuring scalability and deployment feasibility in large-scale urban settings also remains a critical aspect for further development.

One of the key strengths of this project is its ability to function across diverse environments, including public transport stations, educational institutions, workplaces, and residential complexes. The incorporation of location-based analytics further enhances the system's effectiveness by tailoring recommendations and alerts based on regional crime trends and high-risk zones. Additionally, the use of edge computing for real-time video analysis ensures a balance between privacy and efficiency, reducing the need to transfer large amounts of data to centralized servers.

Despite these advancements, challenges remain, including ensuring high accuracy in varying lighting conditions, minimizing false alarms, and maintaining user privacy. Addressing these issues will require continued improvements in AI models, better multi-modal threat detection (integrating image and text), and collaboration with law enforcement agencies for streamlined responses. Additionally, scalability is a critical factor, as deploying this system in multiple locations requires seamless hardware-software integration and cloud-based alert mechanisms.

In conclusion, this project lays the foundation for a sophisticated AI-driven safety system that can be deployed in public transport hubs, educational institutions, workplaces, and residential areas to proactively detect, assess, and alert authorities about potential threats against women. Future enhancements will focus on improving model accuracy, expanding multi-camera surveillance support, and integrating predictive analytics to prevent incidents before they occur.

CHAPTER 8

REFERENCES

[1] Women Safety Analytics - Protecting Women from Safety Threats – Nov 2024

[2] AI-Powered CCTV Analytics for Proactive Threat Detection and Operational Excellence in Well Engineering Operations – Nov 2024

[3] An Integrated Approach for Real-Time Gender and Age Classification in Video Inputs Using FaceNet and Deep Learning Techniques – Aug 2024

[4] Deep Learning Based Hand Gesture Recognition for Emergency Situations: A Study on Indian Sign Language – May 2021

[5] Weapon Detection Using YOLO V3 for Smart Surveillance System – May 2021

[6] Towards a Conceptual Framework for AI-driven Anomaly Detection in Smart City IoT Networks for Enhanced Cybersecurity – Oct 2024

[7] Real-time Object Detection, Tracking, and Monitoring Framework for Security Surveillance Systems – Aug 2024

[8] Object Detection and Crowd Analysis Using Deep Learning Techniques: Comprehensive Review and Future Directions – Sept 2024

[9] Artificial Intelligence & Crime Prediction: A Systematic Literature Review – Mar 2022

[10] The Role of IoT in Women's Safety – Jan 2023

[11] AI in Crime Prediction and Prevention – May 2024

[12] A Hidden Markov Model and IoT Hybrid Based Smart Women Safety Device – Jun 2018

[13] A Machine Learning Approach to Design and Develop a BEACON Device for Women's Safety – 2022 May

[14] Guardian Device for Women—A Survey and Comparison Study – 2021 May

[15] Recent and Emerging Technologies: Implications for Women's Safety – Aug 2019

[16] IoT-based Women Security: A Contemplation – Mar 2020

[17] Systematic Literature Review vs Narrative Review – 2007 May

[18] Smart Wearable Device for Women Safety Using IoT – Jun 2020

[19] MoveFree: A Ubiquitous System to Provide Women Safety – Aug 2015

[20] Design of a Smart Women Safety Band Using IoT and Machine Learning – May 2021