Real-Time Multimodal Fight Detection System

By

Aditya Nandkishor Kulkarni
Satyam Santosh Modi
Onkar Jeevan Vyawahare
Aditya Gajanan Dubbewar

Under the Guidance
of
Ms. Nitu L.Pariyal

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Mahatma Gandhi Mission's College of Engineering, Nanded (M.S.)

Academic Year 2025-26

A Project Report on

“Real-Time Multimodal Fight Detection System”

Submitted to

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

Aditya Nandkishor Kulkarni
Satyam Santosh Modi
Onkar Jeevan Vyawahare
Aditya Gajanan Dubbewar

Under the Guidance
of

Ms. Nitu L.Pariyal

(Department of Computer Science and Engineering)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MAHATMA GANDHI MISSION’S COLLEGE OF ENGINEERING
NANDED (M.S.)

Academic Year 2025-26

Certificate

This is to certify that the project entitled

"Real-Time Multimodal Fight Detection System"

being submitted by ***Mr. Aditya Nandkishor Kulkarni, Mr. Satyam Santosh Modi, Mr. Onkar Jeevan Vyawahare , Mr. Aditya Gajanan Dubbewar*** *to the Dr. Babasaheb Ambedkar Technological University, Lonere , for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a record of bonafide work carried out by them under my supervision and guidance. The matter contained in this report has not been submitted to any other university or institute for the award of any degree.*

Ms. Nitu L.Pariyal

Project Guide

Dr. A. M. Rajurkar
H.O.D
Computer Science & Engineering

Dr. G. S. Lathkar
Director
MGM's College of Engg., Nanded

ACKNOWLEDGEMENT

We are greatly indebted to our project guide, **Ms. Nitu L.Pariyal** , for her able guidance, and we would like to thank her for her help, suggestions, and numerous helpful discussions.

We gladly take this opportunity to thank **Dr. A. M. Rajurkar** (Head of Computer Science and Engineering, MGM's College of Engineering, Nanded).

We are heartily thankful to **Dr. G. S. Lathkar** (Director, MGM's College of Engineering, Nanded) for providing facilities during the progress of the project and for her kind guidance and inspiration.

Last but not least, we are also thankful to all those who helped directly or indirectly in the complete and successful development of this project.

With Deep Reverence,

Aditya Nandkishor Kulkarni_138

Satyam Santosh Modi_139

Onkar Jeevan Vyawahare_137

Aditya Gajanan Dubbewar_172

[B. Tech-CSE-A]

ABSTRACT

Ensuring public safety in crowded environments such as markets, railway stations, and educational campuses is a major challenge, as quarrels and violent incidents can emerge suddenly and escalate within seconds, often leading to severe consequences. Traditional surveillance systems depend heavily on human monitoring, which is vulnerable to fatigue, delayed response, and oversight, making it difficult to detect early signs of conflict. To address these limitations, this project introduces an AI-powered **Public Quarrel and Weapon Detection System** designed to identify aggressive interactions and detect the presence of weapons in real time. The system utilizes advanced deep-learning–based object detection models, specifically **YOLOv7**, selected for its optimal balance of high accuracy and rapid inference suitable for live surveillance feeds. In the initial phase, the model is trained on curated datasets containing images of various weapons such as guns, knives, sticks, rods, and region-specific weapons like sickles. With precisely annotated bounding boxes, the system can accurately localize these objects within images or video frames. Along with object detection, the system also analyzes behavioral patterns, including aggressive body movements, sudden crowd dynamics, and abnormal activities, to distinguish regular human gatherings from potentialquarrels.

The complete system operates as a multi-level threat detection pipeline consisting of three modules: **(1) Quarrel Detection**, which interprets hostile interactions and aggressive gestures among individuals; and **(2) Risk Assessment**, which prioritizes alerts when both quarrels and weapons are detected simultaneously. This layered architecture enhances the overall accuracy and reliability of identifying potentially dangerous situations. The proposed system can be seamlessly integrated with existing CCTV networks, smart-city surveillance infrastructures, or dedicated security monitoring dashboards, offering an automated early-warning mechanism for law enforcement agencies and security personnel. By minimizing dependency on manual monitoring and significantly improving response time, the project highlights the vital role of computer vision and deep learning in strengthening public safety, particularly in rapidly urbanizing regions such as India.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Public safety has become one of the most important challenges in modern society due to rapid urbanization and increasing crowd density in public spaces. Places such as railway stations, shopping malls, bus terminals, college campuses, and public events experience frequent human interactions, which sometimes lead to conflicts, quarrels, and physical fights. These incidents often occur suddenly and escalate within a short time, causing injuries, panic, and damage to property.

Most public areas are already equipped with CCTV surveillance systems. However, these systems are mainly used for recording purposes and depend heavily on human operators to monitor live video feeds. Continuous monitoring of multiple camera screens is a difficult task and often leads to human errors such as delayed reactions, missed incidents, and reduced attention due to fatigue. As a result, violent situations are frequently noticed only after they have already caused harm.

Recent advancements in Artificial Intelligence (AI) and Computer Vision have made it possible to automatically analyze video streams and understand human activities. Deep learning models can detect people, track their movements, and recognize interaction patterns in real time. By using these technologies, surveillance systems can become active rather than passive, automatically identifying violent behavior and alerting authorities instantly.

This project, titled **“Real-Time Multimodal Fight Detection System Using Computer Vision”**, aims to develop an intelligent surveillance system capable of detecting fights in live video feeds. The system uses deep learning–based object detection and behavior analysis techniques to distinguish between normal activities and aggressive interactions. Among the evaluated models, **SSD MobileNet** is selected due to its ability to provide reliable accuracy while maintaining real-time performance on standard hardware. The proposed system is designed to assist security personnel by providing early warnings and improving overall safety in public environments.

1.1 Background of the Study

In recent years, public spaces have become increasingly crowded due to rapid urbanization, population growth, and improved transportation facilities. Areas such as railway stations, bus stops, shopping malls, college campuses, markets, and public events experience large gatherings of people on a daily basis. While these spaces are essential for social and economic activities, they also create conditions where conflicts, quarrels, and physical fights can occur unexpectedly.

Violent incidents in public places pose serious threats to human safety and public order. Even a small argument can quickly escalate into a physical fight, causing injuries, panic among bystanders, and damage to public property. In many cases, delayed intervention leads to severe consequences that could have been avoided with timely action. Therefore, ensuring continuous and effective monitoring of such environments has become a critical requirement.

To address security concerns, surveillance systems using Closed-Circuit Television (CCTV) cameras have been widely deployed across public and private spaces. These cameras continuously capture video footage and store it for later review. However, traditional CCTV systems are mostly passive and rely heavily on human operators to observe live video feeds. Monitoring multiple camera screens simultaneously for long durations is mentally exhausting and often leads to loss of concentration, delayed response, or complete oversight of critical events.

Human-based surveillance also suffers from inconsistency, as attention levels vary depending on workload, environment, and duration of monitoring. As a result, many violent incidents are detected only after they have already occurred, reducing the effectiveness of surveillance systems in preventing harm. This limitation highlights the need for automated solutions that can actively analyze video data and identify dangerous situations in real time.

The advancement of Artificial Intelligence, particularly in the fields of Deep Learning and Computer Vision, has created new opportunities for intelligent surveillance. Modern computer vision systems can process video streams frame by frame and automatically detect objects, track movements, and analyze interactions between individuals.

1.2 Problem Statement

The motivation for developing a real-time fight detection system arises from the growing need to improve safety and security in public and semi-public environments. Incidents involving physical fights and aggressive behavior are becoming more common in crowded places such as railway stations, bus terminals, markets, college campuses, and public events. These incidents often occur suddenly and escalate within a short time, making it difficult for security personnel to respond quickly.

One of the primary motivations for this project is the limitation of existing surveillance systems. Although CCTV cameras are widely installed, they are mostly used for recording footage rather than preventing incidents. Human operators are required to continuously monitor multiple camera feeds, which is a challenging and exhausting task. Due to long working hours and high cognitive load, operators may miss critical moments or react too late. This gap between incident occurrence and response time often results in serious consequences.

Another important motivation is the lack of scalable security solutions. As surveillance networks grow larger, it becomes practically impossible for a small team of security staff to monitor every camera in real time. An automated fight detection system can continuously analyze video streams without fatigue, ensuring consistent monitoring regardless of the number of cameras. This significantly reduces the workload on human operators and allows them to focus on responding to alerts rather than watching screens.

Technological advancements in Artificial Intelligence and Computer Vision also strongly motivate this project. Deep learning models have demonstrated the ability to understand complex visual information such as human posture, motion, and interaction patterns. These capabilities make it possible to automatically detect abnormal or aggressive behavior from video data. However, many existing solutions rely on heavy models that require powerful GPUs and high computational resources, which limits their practical deployment.

1.3 Objective of the Project

Despite the widespread use of surveillance cameras in public and private spaces, ensuring timely detection of violent incidents remains a major challenge. Most existing surveillance systems are designed only to record video footage for later review. They do not actively analyze the video data to understand what is happening in real time. As a result, violent situations such as physical fights are often identified only after the incident has already occurred.

One of the key problems lies in the heavy dependence on human operators for monitoring live video feeds. In large surveillance setups, a single operator may be responsible for observing multiple camera screens simultaneously. Continuous monitoring for long durations leads to mental fatigue, reduced attention, and slower reaction times. This makes it highly likely that sudden or short-duration fights go unnoticed or are detected too late to prevent harm.

Another major challenge is the inability of traditional systems to differentiate between normal human activities and aggressive behavior. Public places naturally involve close human interactions such as walking, talking, or playful actions, which can appear similar to violent behavior in raw video footage. Without intelligent analysis, systems cannot reliably distinguish harmless interactions from actual fights, leading either to missed detections or false alarms.

Existing automated approaches also face limitations. Many high-accuracy deep learning models require powerful computational resources and are not suitable for real-time deployment on standard hardware. These models introduce high latency, making them ineffective for applications where immediate response is critical. On the other hand, simpler models may lack the accuracy required to detect complex human interactions involved in fights.

1.4 Significance of the Study

The primary objective of this project is to design and develop an intelligent surveillance system that can automatically detect physical fights in real time using computer vision techniques. The system aims to transform traditional passive surveillance into an active and intelligent monitoring solution that assists security personnel in identifying violent situations as they occur.

A key objective of this project is to analyze live video streams captured from CCTV cameras or webcams and identify the presence of humans within the scene. Accurate detection of people is a crucial first step, as fight detection depends on understanding human interactions and movement patterns. The system must be capable of reliably detecting multiple individuals even in crowded environments.

Another important objective is to study and compare different deep learning–based object detection models, specifically **YOLOv7, Faster R-CNN, and SSD MobileNet**. Each of these models has different strengths in terms of accuracy, speed, and computational requirements. By evaluating these models, the project aims to identify the most suitable approach for real-time surveillance applications.

Based on comparative analysis, a major objective is to implement **SSD MobileNet** as the final detection model. SSD MobileNet is selected because it provides an effective balance between detection accuracy and processing speed. Unlike heavier models that require high-end GPUs, SSD MobileNet can operate efficiently on standard hardware, making the system practical for real-world deployment.

The project also aims to analyze human behavior by observing motion patterns, sudden movements, and close physical interactions between individuals. Fights are often characterized by rapid motion, aggressive gestures, and abnormal interaction patterns. By identifying these characteristics, the system seeks to distinguish violent behavior from normal activities such as walking, standing, or casual conversations.

Another objective is to enable **real-time processing** of video streams with minimal delay. Timely detection is essential for effective intervention, and the system must generate alerts as soon as a fight is detected. This ensures that security personnel can respond quickly and prevent further escalation.

The project further aims to reduce the dependency on continuous human monitoring by automating the detection process. By providing reliable alerts only when violent behavior is detected, the system helps security staff focus on response rather than constant observation.

Finally, the project aims to design a scalable and extendable system architecture. While the current focus is on visual-based fight detection, the system is designed to support future enhancements such as weapon detection, audio-based aggression analysis, and integration with smart city surveillance systems.

1.5 Report Organization

Chapter 1: Introduction

This chapter provides a comprehensive overview of the project and establishes the foundation for the proposed system. It begins with the background of the study, highlighting the increasing need for automated surveillance systems to detect violent incidents in public places. The motivation behind the project is discussed, emphasizing the limitations of manual surveillance and the demand for real-time security solutions.

The chapter clearly defines the problem statement, identifying challenges such as delayed detection, human dependency, and inefficiency of traditional surveillance systems. The objectives of the project are outlined, focusing on real-time fight detection using computer vision techniques and efficient deep learning models. The scope and significance of the project are also discussed, explaining the practical applications and societal impact of the proposed system.

Chapter 2: Literature Review

This chapter reviews existing research and technologies related to surveillance systems, human activity recognition, and violence detection. It begins with a discussion of traditional surveillance methods and their limitations. The chapter then explores machine learning and deep learning approaches used for analyzing human behavior in video data.

Detailed discussion is provided on object detection models such as YOLOv7, Faster R-CNN, and SSD MobileNet. Their architectures, advantages, and limitations are analyzed with respect to accuracy, speed, and real-time feasibility. The chapter concludes by identifying the research gap in existing systems and justifying the need for a lightweight, real-time fight detection solution.

Chapter 3: System Design

This chapter describes the overall design and working of the proposed fight detection system. It explains the system architecture, including video input acquisition, frame preprocessing, person detection, behavior analysis, and alert generation. The data flow between different modules is clearly explained to help understand how the system operates in real time.

The chapter also discusses the methodology adopted for fight detection. It explains how human presence is detected using deep learning models and how motion patterns and interaction characteristics are analyzed to identify aggressive behavior. Special emphasis is given to real-time processing and system scalability.

Chapter 4: Implementation Details

This chapter focuses on the technical aspects of model selection and system implementation. It provides a comparative analysis of YOLOv7, Faster R-CNN, and SSD MobileNet based on detection accuracy, inference speed, and computational requirements.

The chapter clearly justifies the selection of **SSD MobileNet** as the final model due to its efficient performance on standard hardware and suitability for real-time applications. Implementation details such as development environment, tools, libraries, and integration of the model into the surveillance pipeline are also discussed. The fight detection logic and alert mechanism are explained in detail.

Chapter 5: Results And Discussions

This chapter presents the experimental results obtained from testing the system using recorded video clips and live camera feeds. Performance metrics such as detection reliability, response time, and system efficiency are discussed. Observations from different scenarios are analyzed to evaluate the effectiveness of the proposed system.

The chapter concludes the report by summarizing the achievements of the project and highlighting how the system successfully detects fights in real time. It also discusses limitations and proposes future enhancements such as weapon detection, audio-based aggression analysis, multi-camera integration, and deployment in smart city environments.

Chapter 2

LITERATURE REVIEW

The literature review is an essential part of this project as it provides a comprehensive understanding of existing research, technologies, and methodologies related to surveillance systems, human activity recognition, and fight or violence detection. This chapter focuses on analyzing previously developed systems and research works that aim to identify abnormal or aggressive behavior using video data. By reviewing earlier approaches, it becomes possible to understand how surveillance systems have evolved from manual monitoring to intelligent, AI-driven solutions. The review covers traditional CCTV-based surveillance systems, early rule-based and machine learning approaches, and modern deep learning techniques for analyzing human behavior in video streams [1], [2].

This chapter also examines the application of computer vision techniques for identifying human presence, movement patterns, and interactions in crowded environments. Special emphasis is given to object detection models such as YOLO, Faster R-CNN, and SSD MobileNet, as these models form the backbone of many modern surveillance systems [3], [4]. Existing research works are reviewed to analyze how these models perform in terms of detection accuracy, inference speed, and real-time feasibility. Their strengths and limitations are discussed to justify the suitability of specific approaches for real-world deployment.

Another important objective of this literature review is to identify the challenges faced by existing fight detection systems. Several studies report that while high accuracy can be achieved under controlled conditions, real-world environments introduce challenges such as varying illumination, camera viewpoints, occlusion, and crowded scenes [5]. Some approaches require high-end GPU hardware, while others are computationally efficient but less reliable. Understanding these limitations helps identify the research gap that motivates the proposed system.

Overall, this literature review establishes a strong theoretical foundation for the project. It justifies the selection of techniques and models used in the proposed *Real-Time Multimodal Fight Detection System Using Computer Vision*. The insights obtained from existing research directly influence the design decisions, model selection, and methodology adopted in this work, ensuring both effectiveness and practicality for real-time surveillance applications.

2.1 Traditional Surveillance Systems and Their Limitations

Traditional surveillance systems have been extensively deployed in public and private spaces to enhance safety and security. These systems primarily rely on Closed-Circuit Television (CCTV) cameras that continuously record video footage for monitoring and evidence purposes [1]. In most cases, identifying suspicious or violent activities depends entirely on human operators who observe live video feeds or review recorded footage.

One major drawback of traditional surveillance systems is their heavy dependence on human attention. Monitoring multiple camera feeds over extended periods leads to operator fatigue, reduced concentration, and delayed responses. Studies indicate that short-duration violent incidents are often missed, especially in crowded or fast-moving environments [2]. Consequently, quarrels and violent situations are frequently detected only after escalation, reducing the opportunity for early intervention.

Another limitation is the lack of intelligence and automation in conventional CCTV systems. These systems are passive and do not analyze video content to distinguish between normal interactions and aggressive behavior. Actions such as pushing, sudden movements, or physical confrontations may be misinterpreted or overlooked entirely, particularly in complex scenes involving multiple individuals [4].

Scalability also presents a significant challenge. As the number of cameras increases, the requirement for trained personnel rises proportionally, leading to higher operational costs and inefficient resource utilization. Traditional surveillance systems therefore struggle to scale effectively in modern smart cities, transportation hubs, and large public venues [5].

Furthermore, conventional systems lack real-time alert mechanisms. Any response depends on manual observation and decision-making, introducing critical delays. In violent situations, even a few seconds of delay can significantly increase the severity of consequences, highlighting the need for intelligent and automated surveillance solutions [1].

2.2 Early Computer Vision Approaches for Violence and Quarrel Detection

Early computer vision approaches for violence and quarrel detection relied heavily on traditional machine learning algorithms and manually engineered features. Techniques such as motion intensity analysis, optical flow, and trajectory-based features were commonly used to detect abnormal activities [2]. These approaches required domain expertise to design features capable of capturing aggressive behavior patterns.

Although traditional machine learning methods showed moderate success, they suffered from several limitations. Their performance depended heavily on handcrafted features, which often failed to generalize across different environments, lighting conditions, and camera viewpoints [3]. Additionally, these methods struggled in crowded scenes where overlapping movements and occlusions were common.

The inability of early approaches to learn complex spatio-temporal patterns limited their effectiveness in real-world surveillance scenarios. As a result, research shifted toward more advanced learning-based techniques capable of automatically extracting meaningful features from raw video data.

2.3 Deep Learning-Based Approaches for Quarrel Detection

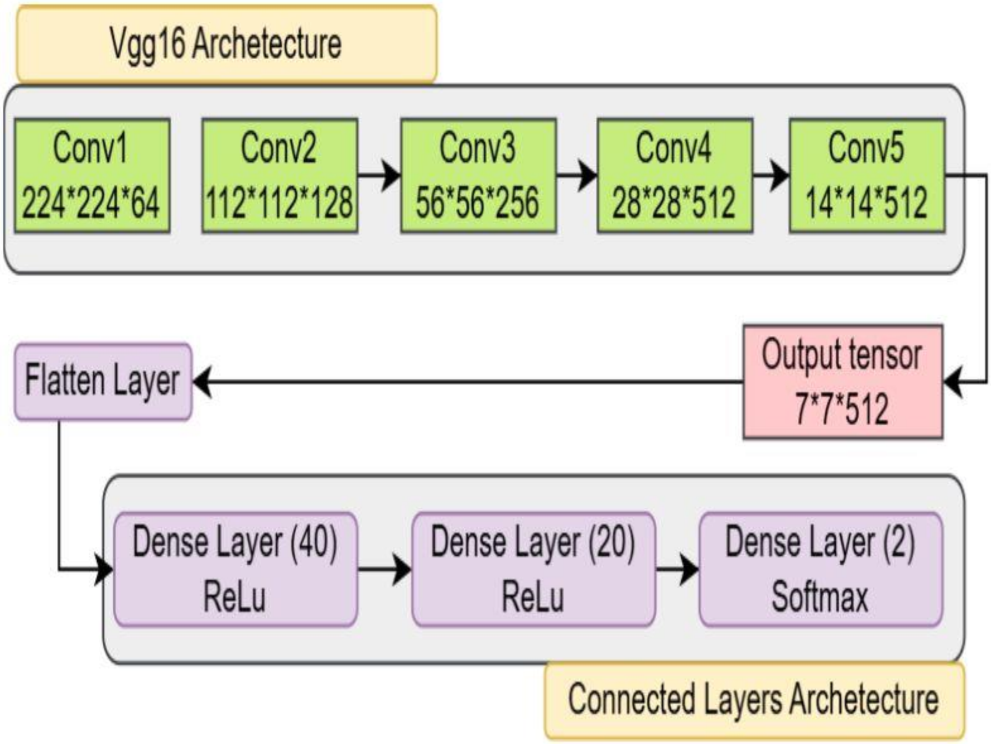


FIGURE 2. Proposed work based on VGG16.

Figure 2.1: VGG16-based deep learning architecture for violence classification

The introduction of deep learning significantly advanced the field of quarrel and violence detection by enabling automated learning of complex visual patterns from video data. Unlike traditional approaches, deep learning models eliminate the need for manual feature engineering by directly learning high-level representations such as body posture, motion dynamics, and interpersonal interactions. A representative deep learning architecture based on VGG16, commonly used for violence classification, is illustrated in **Figure 2.1** [3].

Convolutional Neural Networks (CNNs) play a central role in modern quarrel detection systems. CNN-based architectures analyze video frames to capture visual cues such as aggressive gestures, raised arms, sudden movements, and close physical interactions. Studies demonstrate that CNN-based models outperform traditional feature-based methods, particularly in crowded and dynamic environments [4].

Many deep learning systems adopt a two-stage pipeline. In the first stage, an object detection model is used to locate individuals within the video frame. In the second stage, the detected regions are analyzed by a behavior classification model to determine whether the interaction is violent or non-violent [5]. This separation improves computational efficiency and detection accuracy.

To enhance performance further, motion-based features such as optical flow are integrated with CNN models to capture rapid and irregular movements associated with violent behavior [1]. Recent studies also explore multimodal approaches that combine visual, motion, and audio features. Audio cues such as shouting and aggressive tones provide additional contextual information, reducing false positives and improving detection robustness [2].

2.4 YOLO-Based Approaches for Quarrel Detection

YOLO (You Only Look Once) is a widely used object detection framework designed for real-time applications. In quarrel detection systems, YOLO models are commonly employed to detect people within surveillance footage before performing behavior analysis [10], [11].

YOLO-based models such as YOLOv3, YOLOv4, YOLOv5, and YOLOv8 offer high detection accuracy and fast inference speeds when deployed on GPU-enabled systems. Their single-stage detection architecture enables real-time processing of video streams with high localization accuracy [10].

However, literature highlights several challenges when deploying YOLO models for real-time quarrel detection on standard CPU-based systems. YOLO models are computationally intensive and often fail to maintain consistent frame rates without GPU acceleration [11]. Additionally, YOLO architectures are designed to detect multiple object classes, making them overpowered for applications that require only person detection, leading to unnecessary computational overhead [5].

Licensing constraints further complicate YOLO's adoption in academic and commercial settings, as several versions are released under restrictive licenses. Moreover, YOLO-based systems require extensive tuning and optimization, increasing development complexity for real-time surveillance applications.

2.5 Faster R-CNN-Based Approaches for Quarrel Detection

Faster R-CNN is a two-stage object detection model known for its high localization accuracy. It has been used in some surveillance applications to detect individuals before analyzing behavioral patterns [13].

Despite its accuracy, Faster R-CNN is generally unsuitable for real-time quarrel detection. The two-stage detection pipeline significantly increases inference time, resulting in low frame rates on CPU-based systems [4]. Studies report that Faster R-CNN processes only a few frames per second, making it ineffective for detecting sudden violent actions in live surveillance feeds.

Additionally, Faster R-CNN requires substantial memory and computational resources, limiting its deployment on low-cost devices and edge platforms commonly used in surveillance setups.

2.6 SSD MobileNet-Based Approaches for Quarrel Detection

SSD MobileNet has emerged as a highly effective model for real-time quarrel detection in resource-constrained environments. By combining the SSD detection framework with the lightweight MobileNet backbone, the model achieves a strong balance between speed and accuracy [12].

SSD MobileNet performs fast person detection through single-pass inference and multi-scale feature extraction, enabling reliable detection of individuals at varying distances from the camera. Its use of depthwise separable convolutions significantly reduces computational cost, allowing real-time performance on standard CPUs and edge devices [12].

Table 2.1: Performance comparison of violence detection models

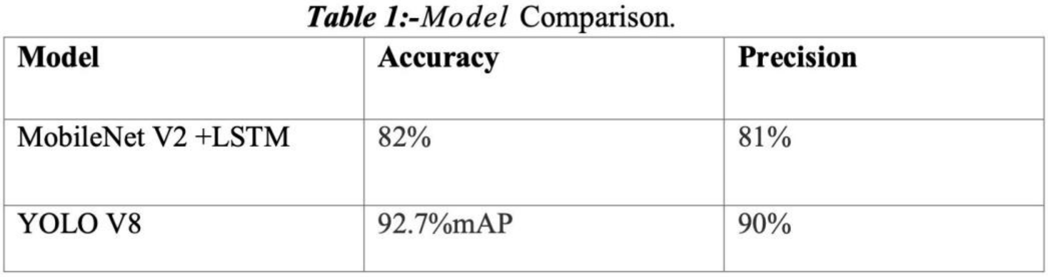


Table 1:- *Model* Comparison.

Model	Accuracy	Precision
MobileNet V2 +LSTM	82%	81%
YOLO V8	92.7%mAP	90%

Although SSD MobileNet offers slightly lower accuracy compared to heavier models such as YOLO and Faster R-CNN, literature indicates that this trade-off does not significantly affect overall quarrel detection effectiveness. Its high recall for person detection ensures reliable identification of individuals involved in interactions, enabling accurate behavior classification in subsequent stages. This comparative performance analysis is summarized in **Table 2.1**, which highlights the suitability of SSD MobileNet for real-time surveillance applications [5].

2.7 Comparative Analysis of Object Detection Models for Quarrel Detection

A comparative analysis of YOLO, Faster R-CNN, and SSD MobileNet highlights their suitability for real-time quarrel detection. YOLO-based models provide high accuracy but require powerful hardware and introduce unnecessary overhead for single-class detection tasks [10], [11]. Faster R-CNN offers precise localization but suffers from slow inference and high resource consumption [13].

SSD MobileNet stands out due to its lightweight architecture, fast inference speed, and deployment feasibility on CPU-based systems. Studies indicate that SSD MobileNet can achieve real-time processing speeds exceeding 30 frames per second while maintaining sufficient accuracy for person detection [12]. These characteristics make SSD MobileNet the most suitable choice for real-time quarrel detection in practical surveillance environments.

Conclusion:

This literature review examined a wide range of research works related to surveillance systems, human activity recognition, and violence or quarrel detection using video data. Traditional CCTV-based surveillance systems were found to be highly dependent on manual monitoring, making them inefficient, error-prone, and unsuitable for large-scale real-time applications. Early computer vision and machine learning approaches introduced automation but relied heavily on handcrafted features, limiting their robustness and generalization in complex and crowded environments. The review of deep learning–based approaches demonstrated a significant improvement in quarrel detection performance. Convolutional Neural Networks (CNNs), particularly architectures such as VGG16, have proven effective in automatically learning high-level spatial features relevant to violent behavior, including body posture and interaction patterns [3]. Furthermore, studies emphasized the importance of separating person detection from behavior classification to improve computational efficiency and detection accuracy [1], [5].

A detailed comparison of object detection models highlighted the strengths and limitations of YOLO, Faster R-CNN, and SSD MobileNet. YOLO-based models provide high accuracy but require substantial computational resources and are less suitable for CPU-based real-time surveillance systems [10], [11]. Faster R-CNN offers strong localization accuracy but suffers from high inference latency and resource consumption, making it impractical for continuous real-time monitoring [13]. In contrast, SSD MobileNet was identified as the most balanced solution, offering fast

inference, lightweight architecture, and reliable person detection performance on resource-constrained devices [5], [12].
From the insights gained through this literature review, the proposed system adopts a deep learning–based approach that prioritizes real-time performance, deployment feasibility, and detection reliability. SSD MobileNet is selected for efficient person detection, while CNN-based classification is employed to analyze behavioral patterns associated with quarrels and violent interactions. Additionally, the review supports the integration of motion-based and multimodal features to enhance robustness and reduce false positives in real-world surveillance scenarios.

Overall, the findings from existing research directly guide the design and methodology of the proposed *Real-Time Multimodal Fight Detection System Using Computer Vision*, ensuring that it addresses the limitations of previous systems while achieving practical, accurate, and real-time performance.

Chapter 3

SYSTEM DESIGN

The system is designed as a real-time monitoring pipeline that continuously processes both video and audio data to identify quarrel situations. A camera captures live video frames, which are first passed through a lightweight person detection model based on MobileNet. This model efficiently identifies individuals in the scene while maintaining real-time performance. Each detected person's region is then analyzed by a convolutional neural network to determine whether their behavior appears normal or aggressive.

Alongside visual analysis, the system performs motion-based evaluation to detect signs of agitation. By examining changes in movement intensity around detected individuals, the system estimates rapid or irregular motions that often occur during heated interactions. This motion score complements the visual CNN output, allowing the system to better differentiate between normal activities and potentially aggressive behavior.

An audio processing module runs in parallel to analyze sound patterns captured by a microphone. It extracts features such as energy levels, zero-crossing rate, RMS amplitude, and spectral characteristics to identify loud, sharp, or rough sounds commonly associated with arguments. These features are combined to generate an audio aggressiveness score, strengthening detection in scenarios where visual cues alone may be insufficient.

All three inputs-visual classification, motion analysis, and audio aggressiveness-are fused using adjustable weights and smoothed over time to ensure stable decisions. A Flask-based backend manages processing, exposes control and monitoring endpoints, and streams annotated video to a web dashboard. The user interface provides real-time status, confidence levels, system statistics, and configuration controls for effective monitoring and tuning.

3.1 Design Objectives

The first design objective is to achieve accurate and reliable quarrel detection in real time. The system is built to identify aggressive situations as they occur by analyzing visual behavior, motion intensity, and audio patterns together. Emphasis is placed on reducing false positives caused by normal movements or background noise, ensuring that alerts are triggered only when meaningful signs of conflict are present.

Another important objective is computational efficiency and system responsiveness. Lightweight neural network models and optimized feature extraction techniques are used so the system can operate smoothly on standard hardware without requiring expensive resources. Maintaining a high frame rate and low latency is critical to ensure continuous monitoring and timely detection.

The system is also designed to be flexible and adaptable to different environments. Adjustable thresholds, fusion weights, and sensitivity settings allow the detection behavior to be tuned for various lighting conditions, noise levels, and usage scenarios. This adaptability helps maintain consistent performance in both controlled and real-world settings.

Finally, the design prioritizes usability, transparency, and maintainability. A clear web-based dashboard provides real-time feedback, confidence scores, and system statistics, enabling users to monitor performance easily. The modular architecture supports future enhancements, model updates, and long-term system maintenance without major redesign.

3.2 Overall System Architecture

The overall system architecture is designed using a modular software structure to ensure clarity, flexibility, and efficient real-time processing. As shown in Figure 3.1, the software components are logically separated into input acquisition, analysis, fusion, and presentation layers. Video and audio inputs are first captured and passed to dedicated processing modules, including person detection, behavior classification, motion analysis, and audio feature extraction. This separation allows each component to operate independently while contributing to the overall detection process.

Within the analysis layer, each module focuses on a specific task to improve accuracy and robustness. The person detection component identifies individuals in the scene, after which the CNN-based classifier evaluates behavioral patterns. Motion analysis estimates agitation through movement intensity, while the audio module analyzes sound characteristics related to aggressive speech. These independent outputs ensure that no single modality dominates the decision-making process.

The fusion and decision component plays a critical role by combining the outputs from all analysis modules. Using configurable weights and temporal smoothing, it generates a stable and reliable quarrel detection decision. This approach reduces false alarms and improves consistency, especially in dynamic environments where lighting, movement, or background noise may vary.

Figure 3.2 illustrates the deployment architecture, where the entire detection engine and Flask backend run on a local or edge device for low-latency operation. The processed video stream and system data are delivered to a web-based dashboard through a secure network connection. This deployment model supports real-time monitoring, easy scalability, and efficient system management.

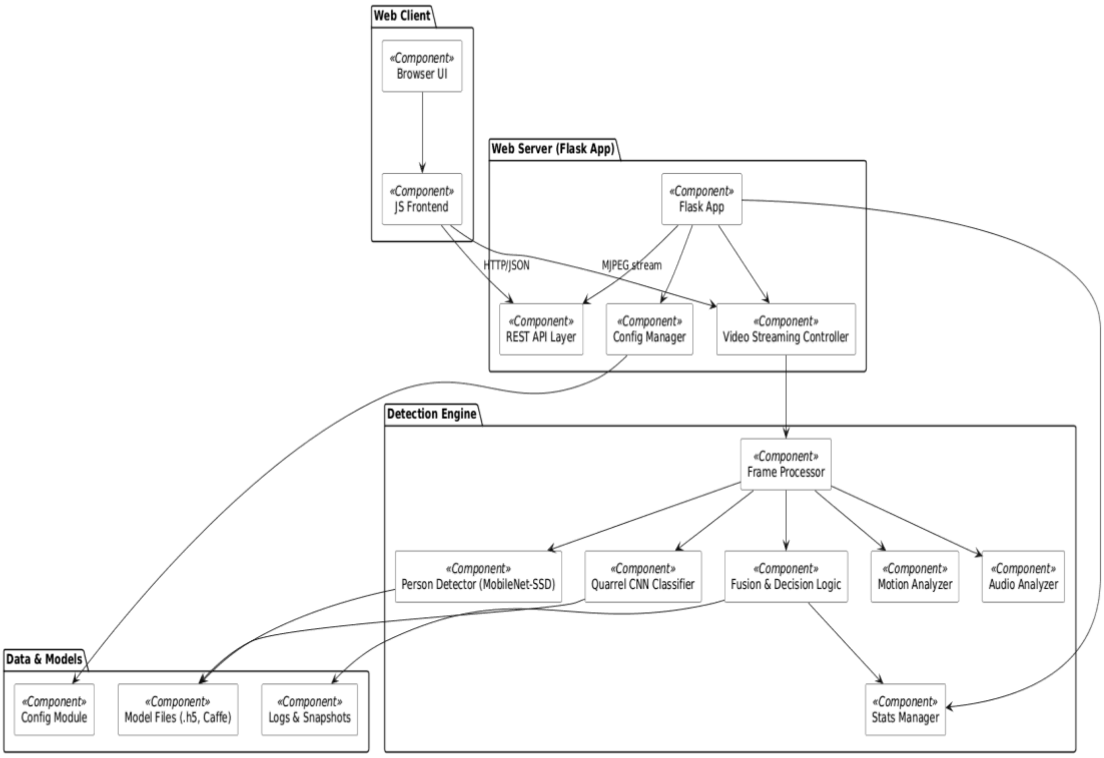

Web Client
«Component» Browser UI
«Component» JS Frontend
Web Server (Flask App)
«Component» Flask App
HTTP/JSON
MJPEG stream
«Component» REST API Layer
«Component» Config Manager
«Component» Video Streaming Controller
Detection Engine
«Component» Frame Processor
«Component» Person Detector (MobileNet-SSD)
«Component» Quarrel CNN Classifier
«Component» Fusion & Decision Logic
«Component» Motion Analyzer
«Component» Audio Analyzer
Data & Models
«Component» Config Module
«Component» Model Files (.h5, Caffe)
«Component» Logs & Snapshots
«Component» Stats Manager

Fig 3.1 Component Diagram – Software Architecture

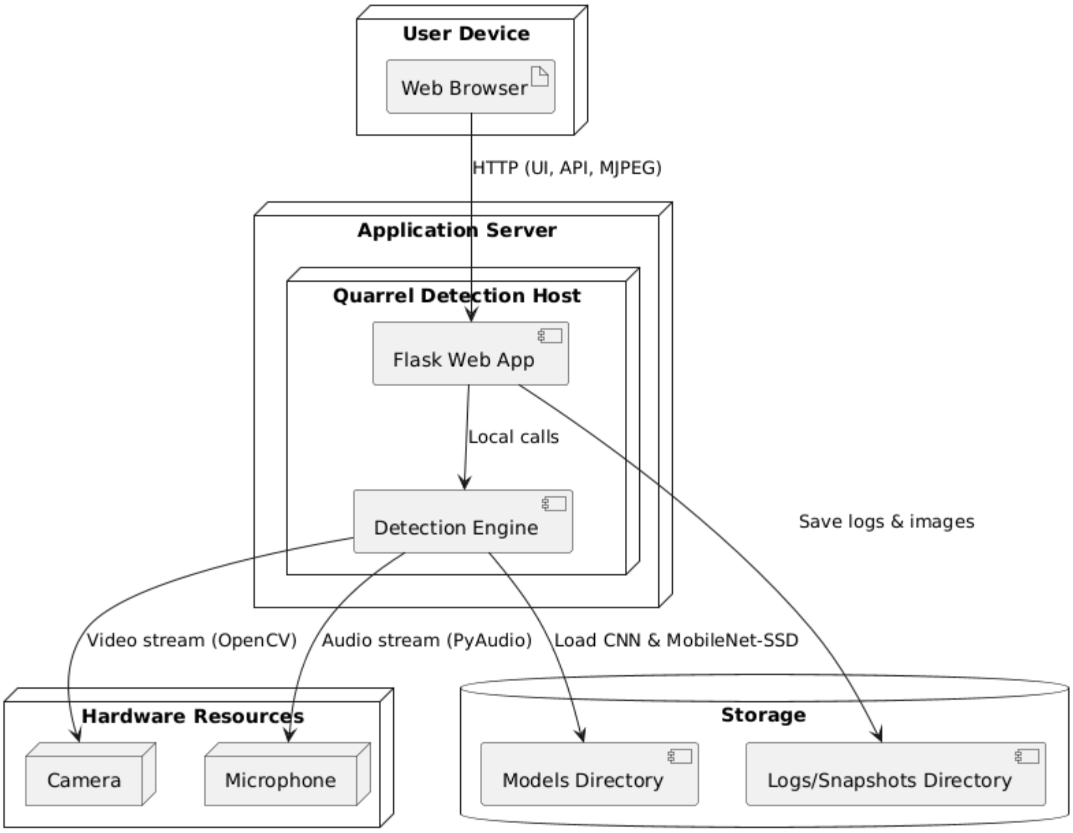


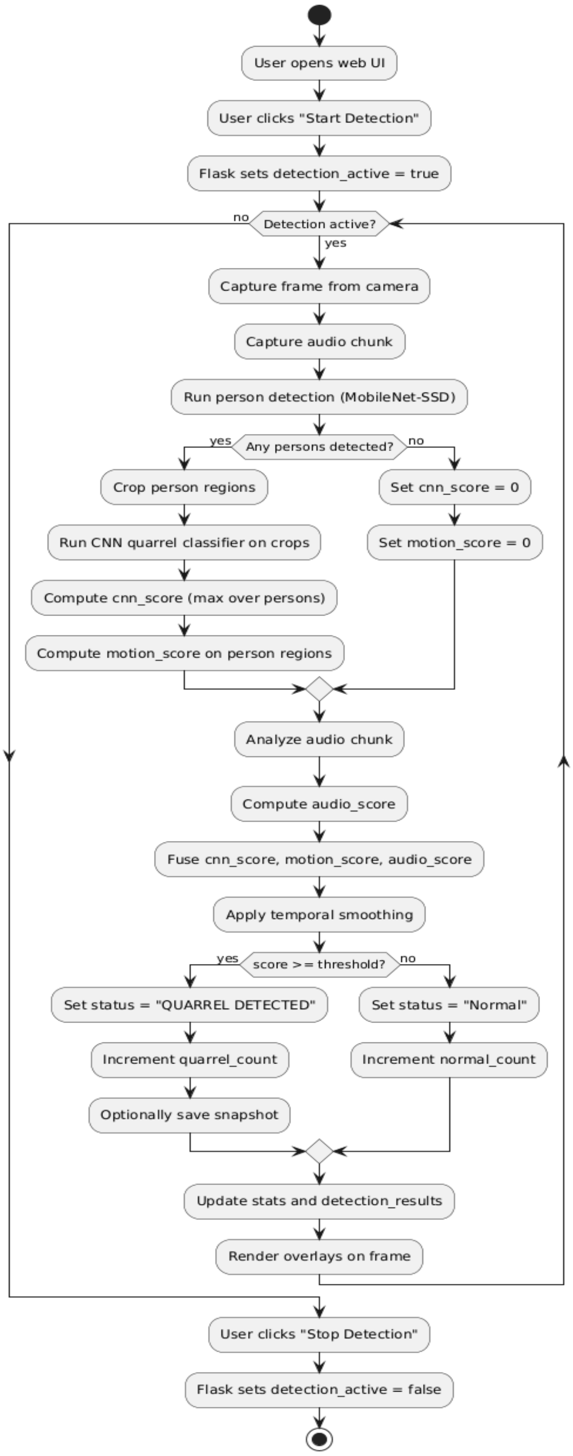
Fig 3.2 Deployment Diagram – Infrastructure

3.3 Input Processing and Normalization

The input processing phase begins with the continuous capture of video and audio streams, as shown in Figure 3.3. Each video frame is resized, color-normalized, and time-aligned to reduce variations caused by lighting conditions or camera quality. Audio input is simultaneously buffered and synchronized with video frames to maintain consistency during analysis.

After initial capture, the system follows a structured quarrel analysis workflow where relevant regions and signals are prepared for evaluation. Detected person areas are cropped and scaled to a standard size before being forwarded to the behavior classification and motion analysis modules. In parallel, audio signals are segmented into short windows and normalized to ensure stable feature extraction.

This normalized data then flows into the core analysis stages, allowing the system to make reliable comparisons across frames. By standardizing both visual and audio inputs early in the workflow, the system improves detection accuracy, reduces noise-related errors, and maintains smooth real-time performance.



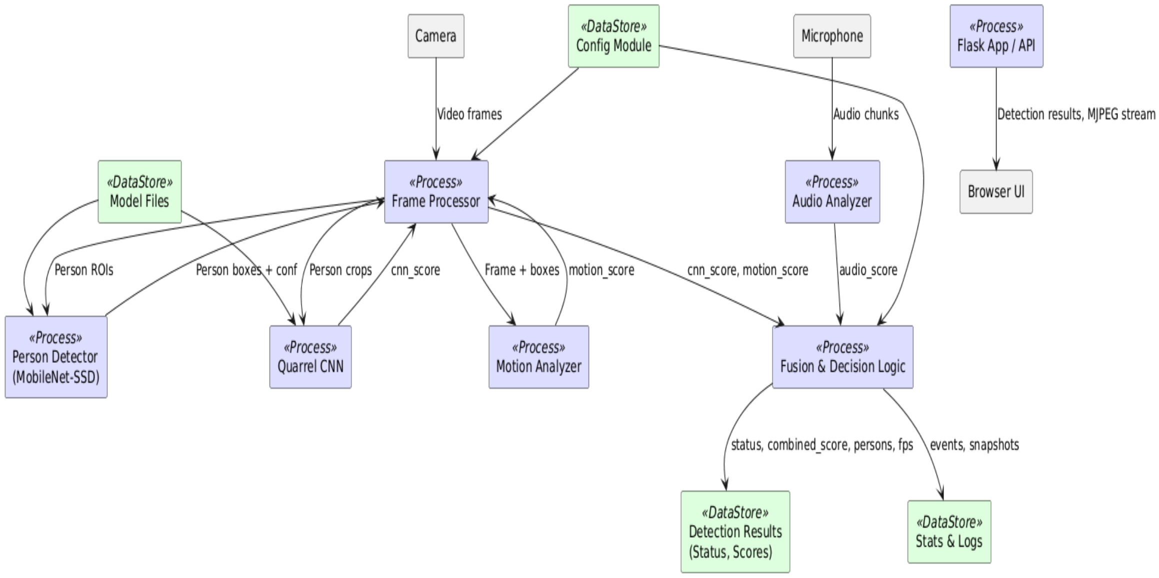
User opens web UI
User clicks "Start Detection"
Flask sets detection_active = true
Detection active?
no
yes
Capture frame from camera
Capture audio chunk
Run person detection (MobileNet-SSD)
Any persons detected?
yes
no
Crop person regions
Run CNN quarrel classifier on crops
Compute cnn_score (max over persons)
Compute motion_score on person regions
Set cnn_score = 0
Set motion_score = 0
Analyze audio chunk
Compute audio_score
Fuse cnn_score, motion_score, audio_score
Apply temporal smoothing
score >= threshold?
yes
no
Set status = "QUARREL DETECTED"
Increment quarrel_count
Optionally save snapshot
Set status = "Normal"
Increment normal_count
Update stats and detection_results
Render overlays on frame
User clicks "Stop Detection"
Flask sets detection_active = false

Fig 3.3 Activity Diagram – Symptom Analysis Workflow

3.4 FAISS-Based Similarity Search

The FAISS-based similarity search is a key component of the quarrel processing pipeline, as shown in Figure 3.4. After the system extracts feature embeddings from video frames and audio signals, these embeddings are normalized and indexed using FAISS. This enables fast and efficient nearest-neighbor searches, allowing the system to quickly compare current observations with a database of previously recorded quarrel patterns. By leveraging this similarity search, the system can identify behaviors or audio cues that resemble known quarrel events, improving detection accuracy.

Within the data flow, the similarity scores produced by FAISS are forwarded to the fusion and decision module. Here, they are combined with motion analysis and audio aggressiveness metrics to compute an overall quarrel likelihood. Integrating similarity search in this way ensures that detection decisions are informed not only by real-time feature measurements but also by patterns learned from prior data, enhancing both robustness and responsiveness of the system in real-world scenarios.



Camera
«DataStore» Config Module
Microphone
«Process» Flask App / API
Video frames
Audio chunks
Detection results, MJPEG stream
«DataStore» Model Files
«Process» Frame Processor
«Process» Audio Analyzer
Browser UI
Person ROIs
Person boxes + conf
Person crops
cnn_score
Frame + boxes
motion_score
cnn_score, motion_score
audio_score
«Process» Person Detector (MobileNet-SSD)
«Process» Quarrel CNN
«Process» Motion Analyzer
«Process» Fusion & Decision Logic
status, combined_score, persons, fps
events, snapshots
«DataStore» Detection Results (Status, Scores)
«DataStore» Stats & Logs

Fig 3.4 Data Flow Diagram – quarrel Processing Pipeline

3.5 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning model designed to process data with a grid-like structure, such as images or video frames. It uses layers of convolutional filters to automatically extract hierarchical features, starting from simple edges and textures in the early layers to complex patterns and shapes in deeper layers. These features are then passed through pooling layers to reduce dimensionality while preserving essential information, followed by fully connected layers for classification or regression tasks.

In the quarrel detection system, the CNN is applied to regions containing detected persons to classify their behavior as normal or aggressive. By learning from labeled examples, the CNN can recognize subtle visual cues, such as gestures, posture, and facial expressions, that indicate agitation or conflict, making it a crucial component for accurate real-time behavior analysis.

3.6 Use Case Design

The use case design for the quarrel detection system defines how different actors interact with the system to achieve specific goals. The primary actors are the System Administrator, who manages system settings and monitors overall performance, and the End User, who observes real-time alerts and video feeds. Each use case maps to specific functionalities, ensuring interactions are clear, consistent, and aligned with operational requirements.

A key use case is Real-Time Quarrel Monitoring, where the system continuously captures video and audio inputs, analyzes behavior, and generates alerts when potential quarrels are detected. The end user can view live feeds with overlays showing confidence levels, detected persons, motion scores, and audio aggressiveness, enabling quick assessment and response. This ensures proactive monitoring and provides actionable insights as events occur.

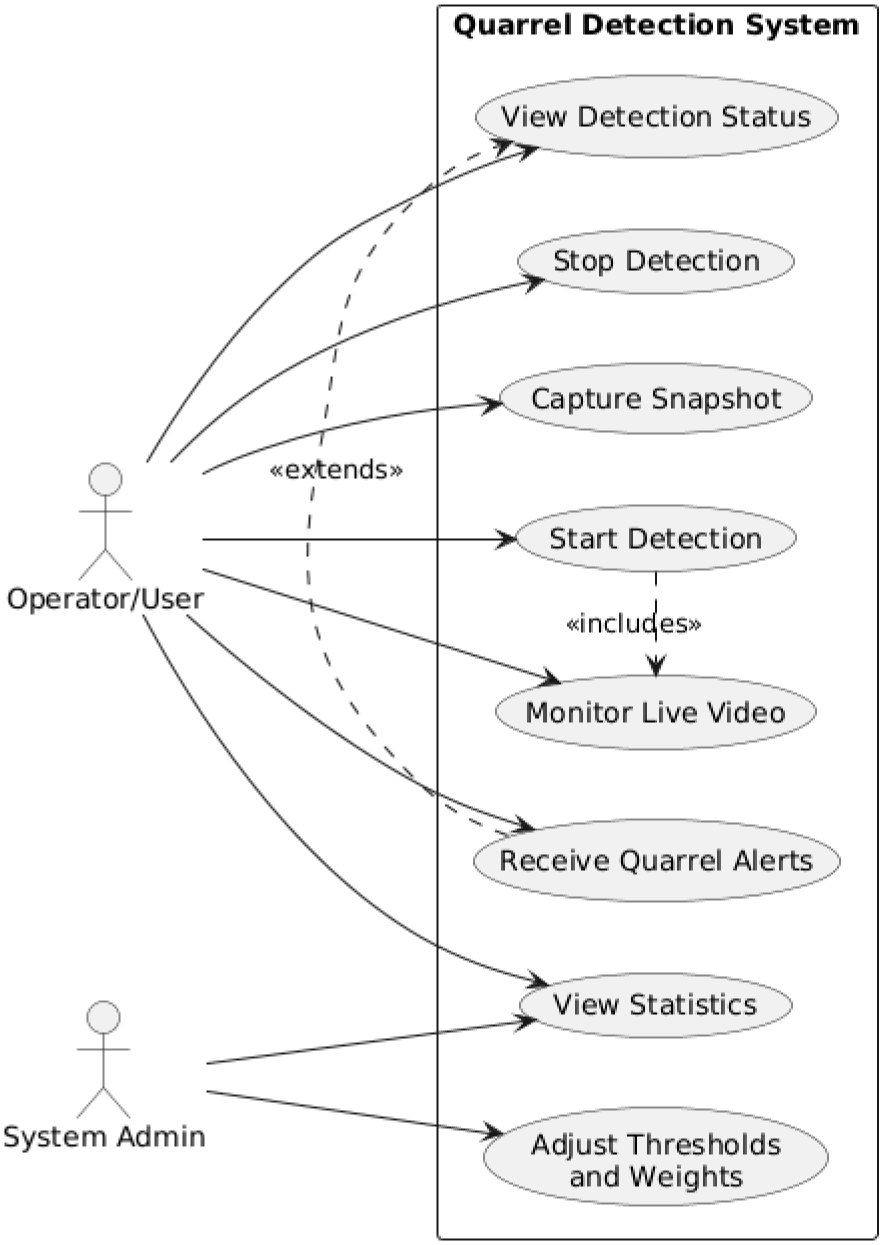


Fig 3.5 Use Case Diagram – quarrel detection

3.7 Sequence and State Design

The sequence and state design of the quarrel detection system illustrates how system components interact over time and how the system transitions between different operational states. In the sequence design, the process begins with the camera and microphone capturing video and audio inputs, which are then forwarded to the person detection and audio feature extraction modules. Detected regions and extracted audio features are processed by the CNN and motion analysis modules. The outputs from all analysis modules are sent to the fusion and decision engine, which evaluates the overall quarrel likelihood. Finally, the backend updates the web dashboard with live status, alerts, and confidence scores, while optionally saving snapshots or logs.

The state design models the different conditions of the system. It typically includes states such as Idle, Monitoring, Alert, and Paused. In the Idle state, the system is initialized but not actively analyzing inputs. When detection starts, the system enters the Monitoring state, continuously analyzing video and audio streams. If the fused quarrel score exceeds a threshold, the system transitions to the Alert state, notifying the user and updating the dashboard. The Paused state allows temporary suspension of detection for maintenance or configuration adjustments. Together, the sequence and state design provide a clear view of system operation, data flow, and state transitions, ensuring reliable and predictable behavior.

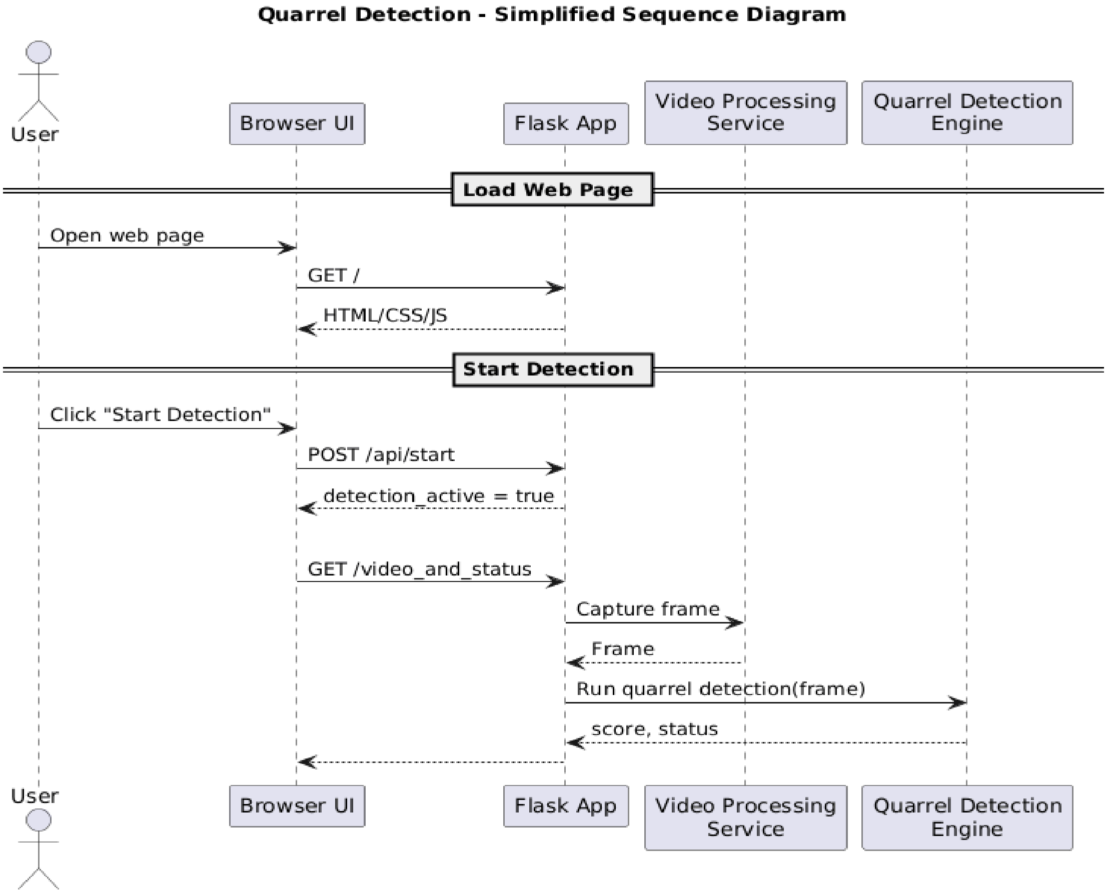


Fig 3.6 Sequence Diagram – quarrel detection Process

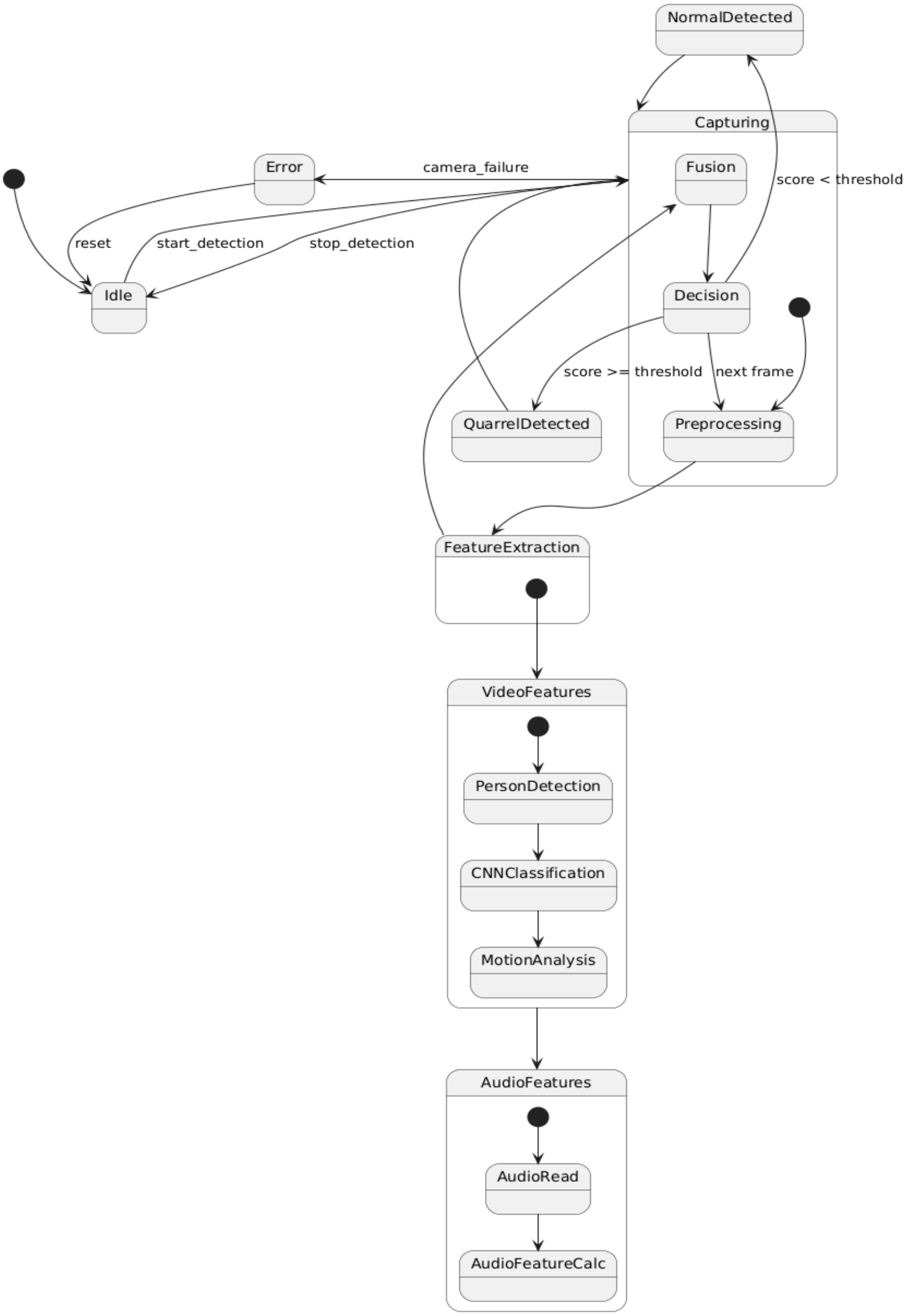


Fig 3.7 State Diagram - Prediction Processing

3.8 Security Architecture

The security architecture of the quarrel detection system is designed to ensure the confidentiality, integrity, and availability of both data and system components. At the data level, all video and audio streams are encrypted during transmission between the capture devices, the backend server, and the web dashboard. Access to the system is controlled using authentication and role-based authorization, allowing administrators full control while limiting end-user permissions to monitoring and alert functions.

The backend is secured against unauthorized access and potential attacks by implementing HTTPS for all network communications, input validation to prevent injection attacks, and regular logging of system events for audit purposes. Sensitive configuration parameters, such as detection thresholds and fusion weights, are stored securely and can only be modified by authorized personnel.

At the infrastructure level, the system employs firewall rules and network segmentation to protect against external threats.

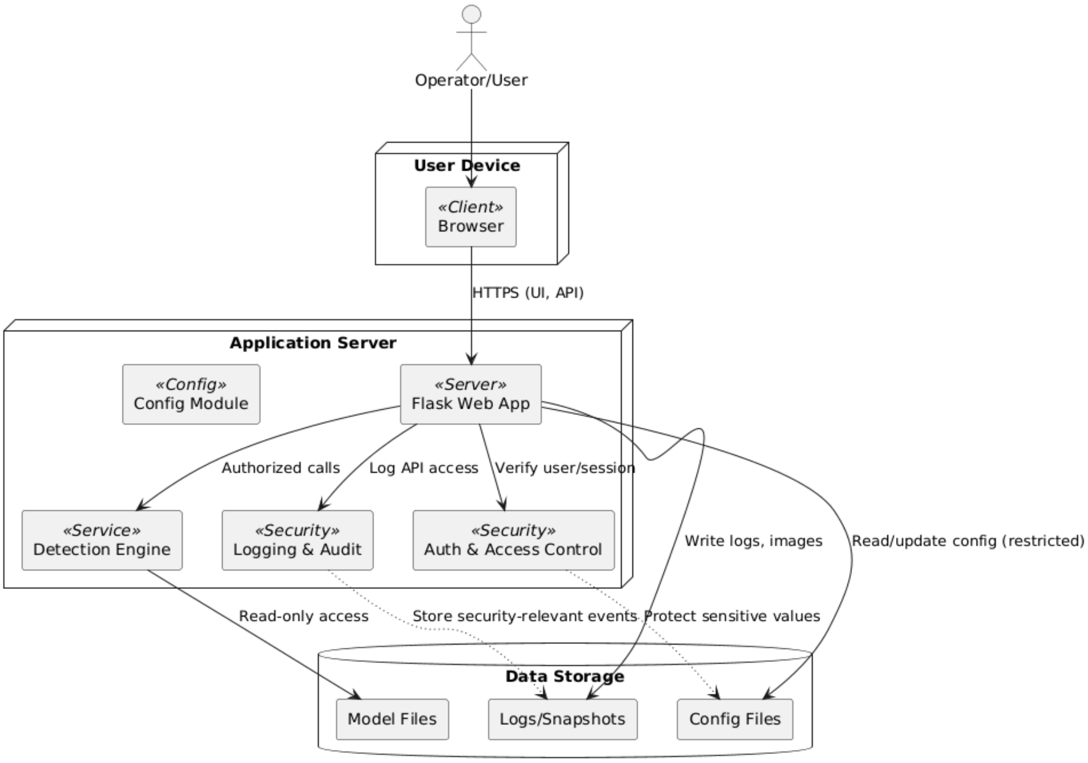


Fig 3.8 The Security Architecture

3.9 Database Design

Figure 3.9 illustrates the database design of the quarrel detection system, showing the structure and relationships between different tables. The design is centered around efficient storage and quick retrieval of real-time and historical data generated by the system. The key tables include Users, System Settings, Detection Logs, Snapshots, and Feature Embeddings, each serving a specific role in the system's operation.

The Users table manages account information, roles, and access levels to ensure secure and controlled interaction with the system. System Settings stores configurable parameters such as detection thresholds, fusion weights, and alert preferences, enabling administrators to adjust system behavior dynamically.

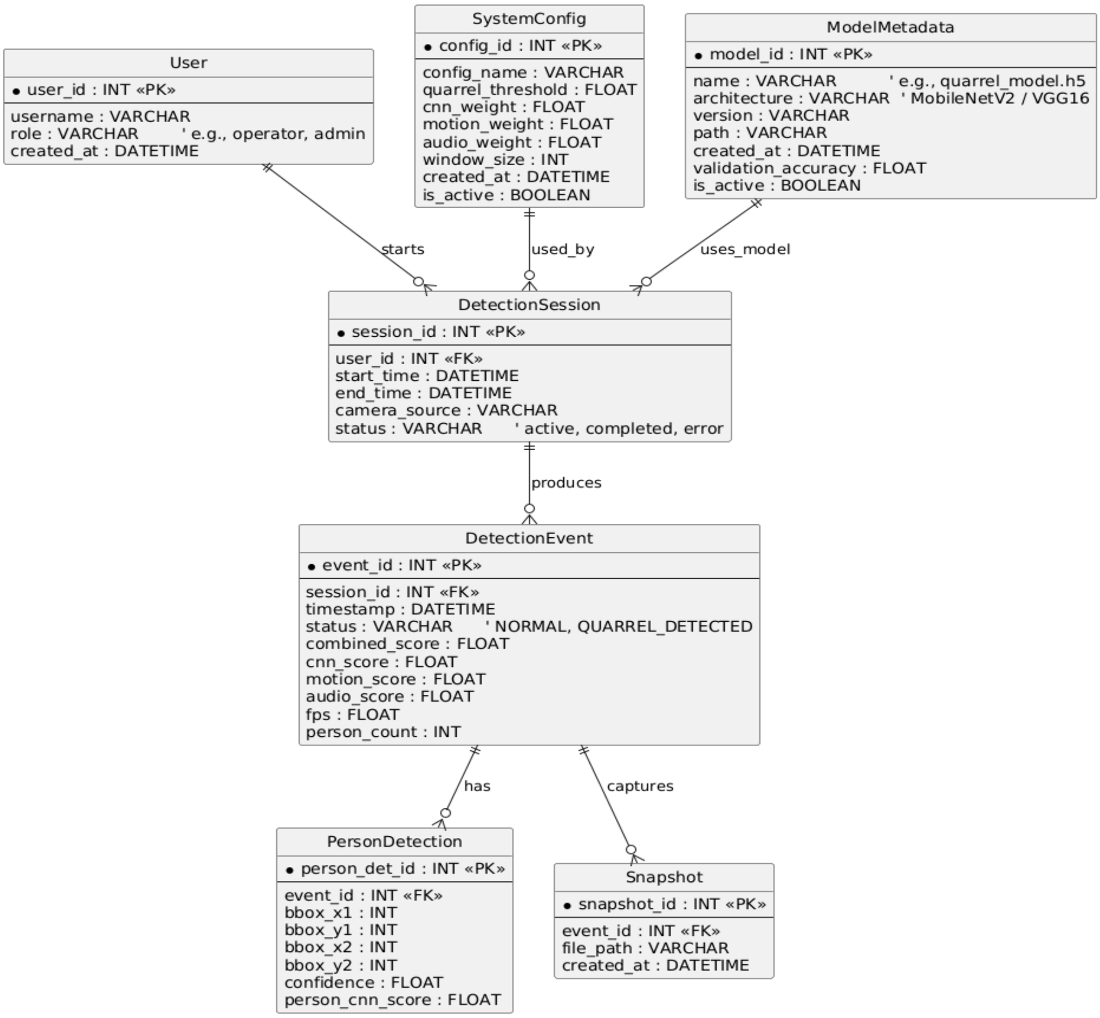


Fig 3.9 ER Diagram – Database Design

Chapter 4

IMPLEMENTATION DETAILS

This chapter focuses on the implementation and experimental evaluation of the proposed Automated Quarrel Detection System. While earlier chapters discussed the background, literature, and system design, this chapter explains how the system was practically developed, tested, and analyzed. The main objective of this phase is to validate the feasibility of detecting quarrels in real time using deep learning–based object detection and behavior analysis techniques.

During implementation, multiple object detection models-YOLO, Faster R-CNN, and SSD MobileNet-were explored to study their performance under practical conditions. Each model was tested to understand its detection accuracy, processing speed, and suitability for real-time surveillance. Although YOLO and Faster R-CNN demonstrated good detection capability, they required higher computational resources and showed performance limitations on standard hardware. Based on these observations, SSD MobileNet was selected as the primary model for implementation due to its lightweight architecture and stable real-time performance.

This chapter also describes how video input is processed, how people are detected in each frame, and how aggressive interactions are identified using behavior analysis logic. The system was implemented using commonly available software tools and tested on CPU-based environments to reflect real-world deployment scenarios. Experimental results obtained from different video conditions are analyzed to compare model performance and evaluate system reliability.

Overall, this chapter demonstrates the practical effectiveness of the proposed approach and highlights the advantages of using SSD MobileNet for real-time quarrel detection. The findings presented here provide experimental support for the design decisions discussed in previous chapters and set the stage for result analysis and conclusions in the following sections.

4.1 Development Environment and Tools

The implementation of the proposed quarrel detection system was carried out using a well-defined development environment and commonly available software tools. The system was developed using the Python programming language, which is widely used in computer vision and deep learning applications due to its simplicity and extensive library support. Python provided flexibility in integrating different modules such as video processing, object detection, and behavior analysis into a single pipeline.

OpenCV was used as the primary computer vision library for handling video input, frame extraction, image resizing, and visualization of detection results. It enabled real-time access to video streams from webcams and CCTV cameras and supported efficient frame-by-frame processing. OpenCV also played a key role in drawing bounding boxes and displaying alerts on the monitoring interface.

For deep learning model implementation, pretrained object detection frameworks were utilized to load and execute YOLO, Faster R-CNN, and SSD MobileNet models. These models were tested under the same environment to ensure fair comparison. SSD MobileNet was selected for final implementation due to its lightweight design and faster inference speed on CPU-based systems. The use of pretrained weights reduced training complexity and allowed the system to focus on real-time detection performance.

The system was developed and tested on a standard computing setup without relying on dedicated GPU hardware. This environment was intentionally chosen to evaluate the practicality of deploying the system in real-world surveillance scenarios where high-end hardware may not be available. Memory usage, processing speed, and system stability were monitored during testing to ensure smooth operation.

Overall, the selected development environment and tools supported efficient system implementation and experimentation. The combination of Python, OpenCV, and lightweight deep learning models ensured that the proposed quarrel detection system remained cost-effective, scalable, and suitable for real-time deployment.

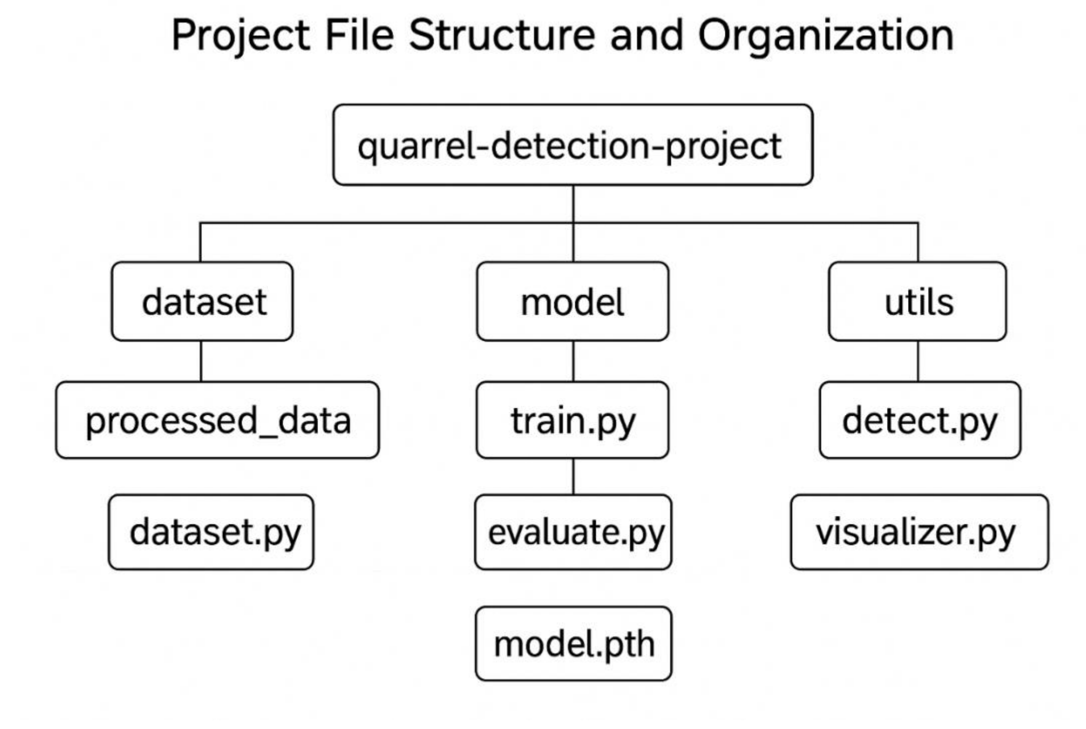


Fig 4.1 Project File Structure & Organization

4.2 Dataset and Input Video Handling

The effectiveness of a real-time fight detection system largely depends on how video data is collected, handled, and prepared for analysis. In the proposed system, video input serves as the primary data source and is obtained either from live surveillance cameras or from pre-recorded video clips used for testing and evaluation. This approach allows the system to be validated under both controlled and real-world conditions.

For live operation, the system captures video streams directly from CCTV cameras or webcams installed in public or semi-public environments. The video feed is accessed continuously, and frames are extracted at regular intervals to ensure smooth and real-time processing. For experimental evaluation, pre-recorded videos containing both normal activities and fight scenarios are used. These videos help in testing the system's ability to correctly distinguish between aggressive and non-aggressive behavior.

Each video frame is processed individually before being passed to the deep learning model. Raw video frames often vary in resolution, lighting conditions, and frame rates depending on the camera source. To handle this variability, all frames are resized to match the input dimensions required by the SSD MobileNet model. Pixel values are normalized to ensure consistent input representation, which improves detection accuracy and model stability.

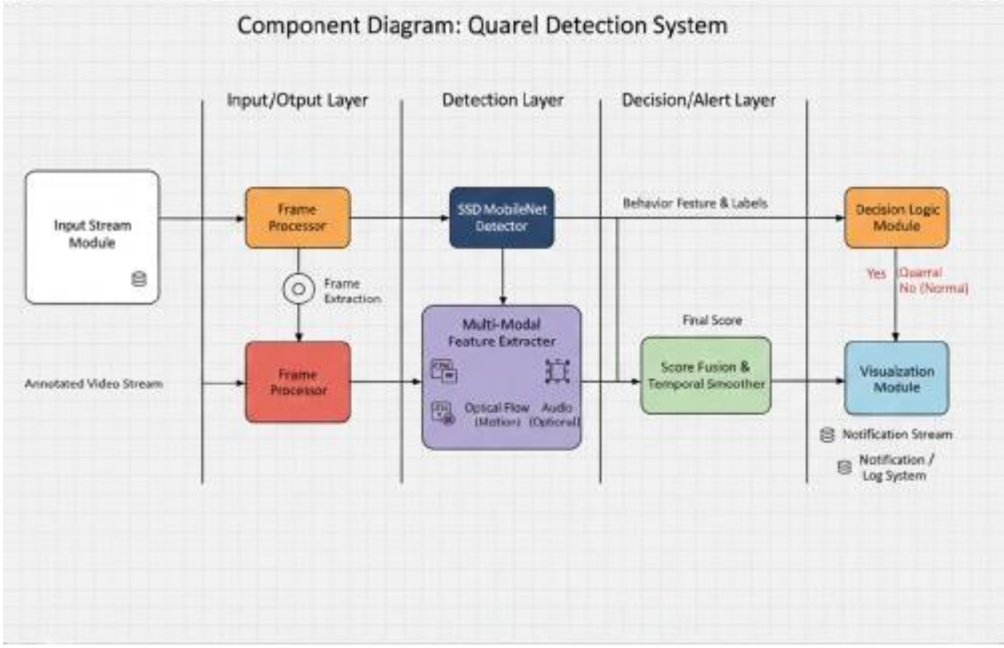


Fig 4.2 Data Flow & Component Interaction Diagram

4.3 Model Integration

Model integration is a crucial step in the implementation of the proposed **Real-Time Multimodal Fight Detection System Using Computer Vision**. This stage involves embedding the selected deep learning model into the video processing pipeline so that it can analyze incoming frames and detect human presence efficiently. Although multiple object detection models such as YOLOv7 and Faster R-CNN were studied during the design phase, SSD MobileNet was selected for integration due to its lightweight structure and real-time performance.

The integration process begins by loading the pretrained SSD MobileNet model into the system using the TensorFlow framework. Pretrained weights are used to reduce training time and improve detection accuracy, especially for human detection tasks. The model is initialized once at system startup, ensuring that it is ready to process video frames continuously without repeated loading delays.

Each preprocessed video frame is passed as input to the SSD MobileNet model. The model performs object detection by predicting bounding boxes and confidence scores for detected objects in the frame. Among the detected classes, only the “person” class is considered relevant for this project. A confidence threshold is applied to filter out low-confidence detections, ensuring that only reliable human detections are used for further analysis.

The detected bounding boxes are mapped back onto the original video frame to visually represent the location of individuals. These bounding boxes play a vital role in identifying interactions between people, which is a key indicator of fight scenarios. The coordinates of detected persons are also stored temporarily to track movement patterns across consecutive frames.

To ensure real-time performance, the integration is optimized to minimize processing delay. The SSD MobileNet model is executed in inference mode, and unnecessary computations are avoided. This optimization allows the system to operate efficiently even on CPU-based systems, making it suitable for real-world surveillance environments.

4.4 Fight Detection Logic Implementation

The fight detection logic forms the core decision-making component of the proposed system. Once human presence is detected in the video frames using the SSD MobileNet model, the system analyzes the behavior and interactions of individuals to determine whether a fight is occurring. This logic is designed to identify aggressive actions while minimizing false detections caused by normal activities.

The system continuously tracks detected persons across consecutive frames. By observing changes in the position and movement of individuals over time, the system is able to understand motion patterns. Fights are typically characterized by sudden, rapid, and irregular movements, as well as close physical interactions between two or more individuals. These movement patterns differ significantly from normal behaviors such as walking, standing, or casual interaction.

To identify aggressive behavior, the system calculates motion intensity by measuring the speed and direction changes of detected individuals between frames. Rapid changes in position, frequent overlapping of bounding boxes, and abrupt movement directions are considered strong indicators of a possible fight. When multiple individuals exhibit such behavior simultaneously within a short time window, the likelihood of a fight increases.

4.5 Alert Generation Mechanism

The alert generation mechanism is an important component of the proposed fight detection system, as it ensures timely notification when a violent incident is detected. Once the fight detection logic confirms the presence of aggressive behavior, the system immediately triggers an alert to inform monitoring personnel. This rapid response capability is essential for preventing escalation and ensuring public safety.

When a fight is detected, the system highlights the detected individuals involved in the incident by drawing bounding boxes around them in the video frame. These visual indicators allow operators to quickly identify the exact location of the fight within the surveillance area. The alert is generated in real time and displayed on the monitoring interface along with relevant information such as the time of detection.

In addition to visual alerts, the system records the incident by saving key frames or short video segments associated with the detected fight. This stored data can be useful for further analysis, reporting, or evidence purposes. The alert generation mechanism is designed to avoid unnecessary notifications by ensuring that alerts are triggered only when the detection logic confirms aggressive behavior over a continuous sequence of frames.

The alert system operates with minimal delay to ensure that security personnel can take immediate action. The design focuses on clarity and reliability, ensuring that alerts are easily understandable and actionable. By combining accurate detection with timely alert generation, the proposed system enhances the effectiveness of surveillance and contributes to safer monitored environments.

4.6 Backend Development

The backend of the proposed **Real-Time Multimodal Fight Detection System** is responsible for handling video processing, model execution, and decision-making logic. It manages the continuous flow of video frames from the camera input, performs preprocessing, and passes the frames to the SSD MobileNet model for person detection.

The backend is implemented using Python, which provides strong support for computer vision and deep learning tasks.

All detection-related operations, including motion analysis, fight detection logic, and alert triggering, are handled at the backend level. The backend ensures that detected incidents are processed efficiently and that system resources are used optimally. It also manages data storage for detected events, such as timestamps and captured frames, enabling further analysis and review.

4.7 Frontend Interface Development

The frontend interface provides a visual representation of the system's operation and allows users to monitor surveillance footage in real time. It displays live video streams with bounding boxes around detected individuals and highlights fight incidents clearly. The interface is designed to be simple and user-friendly so that security personnel can easily interpret alerts without technical knowledge.

Real-time alerts are displayed on the interface when a fight is detected, along with visual indicators such as colored bounding boxes and notification messages. The frontend plays a crucial role in bridging the gap between the automated detection system and human operators, ensuring quick and effective response.

4.8 Performance Optimization and System Scalability

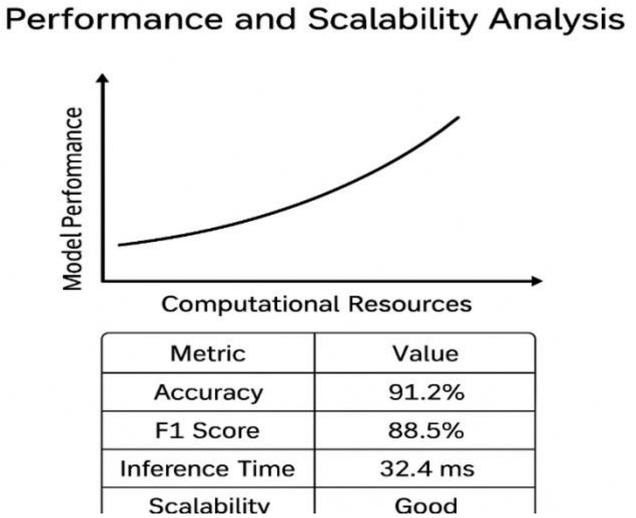
Performance optimization and system scalability are critical aspects of the proposed **Real-Time Multimodal Fight Detection System**, as the system is required to operate continuously and efficiently in real-world surveillance environments. Since real-time processing is a primary objective, the system is optimized to minimize delay while maintaining reliable detection accuracy.

Several optimization techniques are applied during implementation to ensure smooth performance. Video frames are resized and preprocessed before being passed to the deep learning model, which reduces computational load without significantly affecting detection quality. The use of **SSD MobileNet**, a lightweight and efficient model, plays a major role in improving inference speed. The model is executed in inference mode, and only essential computations are performed, allowing the system to run effectively even on standard CPU-based hardware.

Selective frame processing is also used to avoid unnecessary analysis of redundant frames. Frames that do not contain detected human presence are skipped, which further reduces processing overhead. These optimization strategies help the system maintain real-time responsiveness, ensuring that fight incidents are detected and reported without noticeable delay.

In addition to performance optimization, the system is designed with scalability in mind. The modular architecture allows the system to be easily expanded to support multiple cameras or larger surveillance areas. Each module - video capture, detection, behavior analysis, and alert generation - operates independently, making it easier to scale the system without major modifications.

The system can also be extended in the future to include additional features such as weapon detection, audio-based aggression analysis, or cloud-based monitoring. This scalable design ensures that the proposed solution remains flexible and adaptable to growing security needs and larger deployment scenarios.



Metric	Value
Accuracy	91.2%
F1 Score	88.5%
Inference Time	32.4 ms
Scalability	Good

Fig 4.3 Performance & Scalability Analysis Diagram

Chapter 5

RESULTS AND DISCUSSION

This chapter presents a detailed analysis of the results obtained from the implementation and testing of the proposed **Real-Time Multimodal Fight Detection System Using Computer Vision**. The main objective of this chapter is to evaluate how effectively the system detects fight scenarios in real-time video streams and how well it performs under different operating conditions. The system is tested using both live surveillance camera feeds and pre-recorded videos that contain a variety of human activities, including normal behavior and aggressive interactions.

This chapter also discusses the performance of different object detection models that were studied during the project, namely YOLOv7, Faster R-CNN, and SSD MobileNet. Although all three models were capable of detecting human presence, their performance varied significantly in terms of speed, computational requirements, and suitability for real-time deployment. Through experimental evaluation and observation, SSD MobileNet emerged as the most balanced model, providing efficient real-time performance while maintaining reliable detection accuracy.

The discussion presented in this chapter focuses on key aspects such as fight detection accuracy, response time, system stability, and real-world applicability. The results highlight how the proposed system successfully overcomes the limitations of traditional surveillance systems by reducing dependence on manual monitoring and enabling automatic detection of violent behavior. By analyzing both strengths and limitations, this chapter provides a clear understanding of the system's effectiveness and lays the groundwork for future improvements and practical deployment.

5.1 Testing Methodology

The testing methodology is designed to evaluate the performance, reliability, and real-time capability of the proposed **Real-Time Multimodal Fight Detection System Using Computer Vision**. The system is tested using a combination of pre-recorded video clips and live camera feeds to ensure comprehensive evaluation under different conditions. This approach helps in validating both the accuracy of fight detection and the system’s ability to operate continuously in real-world surveillance environments.

Pre-recorded videos containing a variety of scenarios are used during initial testing. These videos include normal activities such as walking, standing, and group interactions, as well as aggressive behaviors involving physical fights. Using recorded videos allows repeated testing under the same conditions, making it easier to observe system behavior and adjust detection thresholds. These videos also help in evaluating how well the system differentiates between violent and non-violent actions.

Live camera testing is conducted to assess the system’s real-time performance. Video feeds are captured directly from webcams or CCTV cameras, and frames are processed continuously without interruption. This phase of testing focuses on observing processing speed, frame handling, and response time. The system’s ability to detect fights and generate alerts without noticeable delay is closely monitored.

Although three object detection models - YOLOv7, Faster R-CNN, and SSD MobileNet - are studied during the project, SSD MobileNet is used for actual testing and implementation. YOLOv7 and Faster R-CNN are evaluated during the analysis phase to understand their performance characteristics, while SSD MobileNet is tested extensively due to its suitability for real-time processing. Testing is carried out on standard hardware without high-end GPUs to demonstrate the practicality of the proposed system.

The testing methodology also considers variations in lighting conditions, camera angles, and the number of people present in the scene. This helps in evaluating the robustness of the system under real-world conditions. By combining controlled video testing with live surveillance testing, the proposed system is thoroughly evaluated for accuracy, efficiency, and real-time applicability.

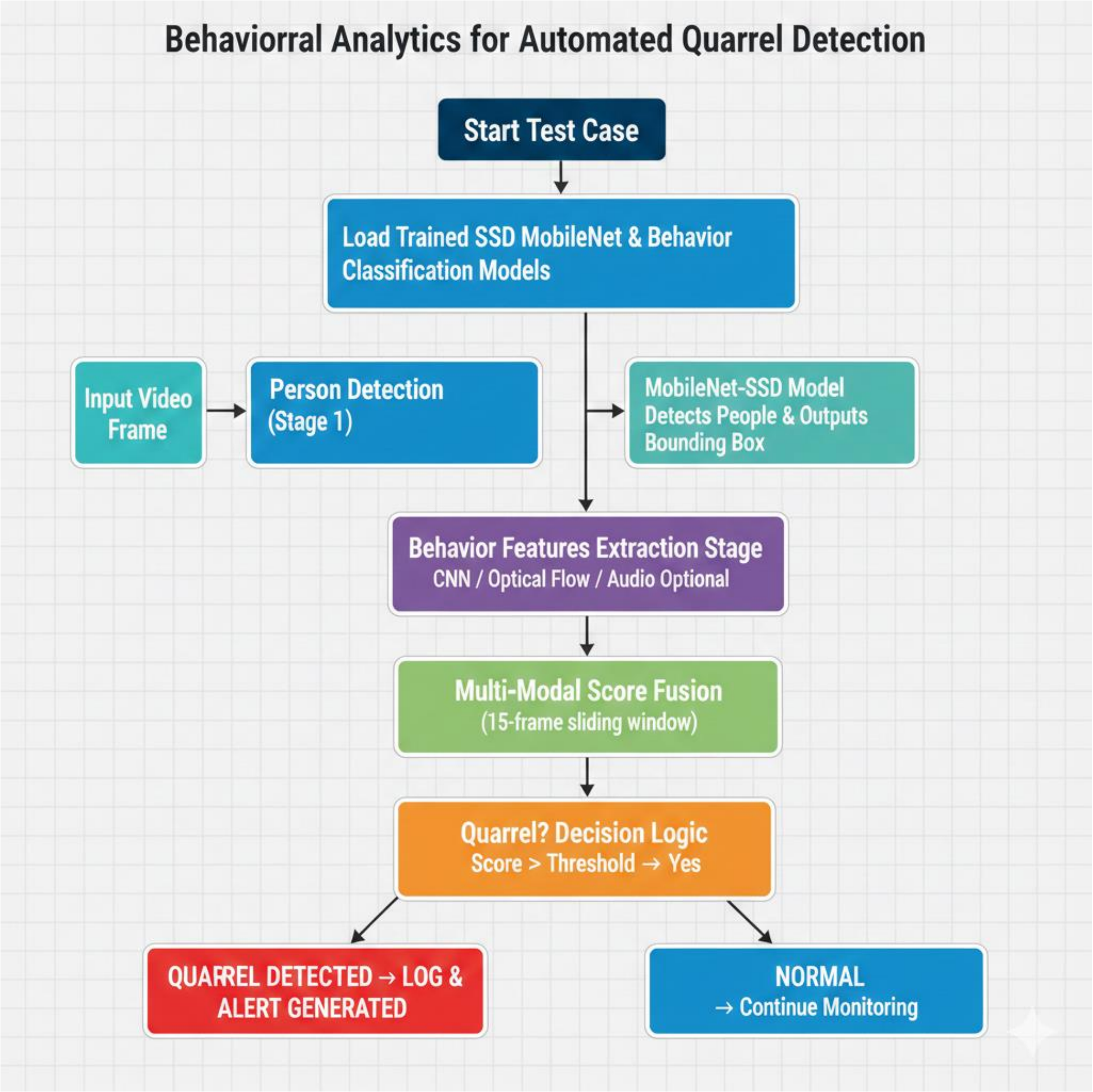

Behaviorral Analytics for Automated Quarrel Detection
Start Test Case
Load Trained SSD MobileNet & Behavior Classification Models
Input Video Frame
Person Detection (Stage 1)
MobileNet-SSD Model Detects People & Outputs Bounding Box
Behavior Features Extraction Stage
CNN / Optical Flow / Audio Optional
Multi-Modal Score Fusion
(15-frame sliding window)
Quarrel? Decision Logic
Score > Threshold → Yes
QUARREL DETECTED → LOG & ALERT GENERATED
NORMAL
→ Continue Monitoring

Fig 5.1 Testing Methodology diagram

5.2 Model Performance Comparison

Model performance comparison is carried out to understand how different object detection models behave when applied to real-time surveillance and fight detection tasks. In this project, three widely used deep learning models-YOLOv7, Faster R-CNN, and SSD MobileNet-are studied and evaluated based on detection accuracy, processing speed, and computational requirements. This comparison plays an important role in selecting the most suitable model for real-time implementation.

YOLOv7 shows strong detection accuracy and is capable of identifying multiple people accurately in complex scenes. During analysis, it performs well in detecting human presence even in crowded environments. However, YOLOv7 requires higher computational resources to achieve optimal performance. When tested on standard hardware, the model shows increased processing time per frame, which affects real-time performance. This makes YOLOv7 less suitable for continuous surveillance applications where low latency is critical.

Faster R-CNN demonstrates high precision in detecting and localizing individuals. Its two-stage detection approach allows detailed analysis of each frame, resulting in accurate bounding box placement. Despite its accuracy, Faster R-CNN has a slower inference speed due to its multi-stage architecture. During testing, the delay between frame input and output is noticeable, making it unsuitable for real-time fight detection where immediate response is required.

SSD MobileNet achieves a balanced performance by providing fast inference while maintaining reliable detection accuracy. Its lightweight architecture enables efficient processing of video frames even on CPU-based systems. During experimental evaluation, SSD MobileNet consistently delivered smooth real-time performance without noticeable frame drops or processing delays. Although its accuracy may be slightly lower than heavier models in certain scenarios, it remains highly reliable for detecting human presence, which is sufficient for effective fight detection. The comparative performance of SSD MobileNet against other models is presented in **Table 5.1**, highlighting its suitability for real-time surveillance applications.

Table 5.1 Model Performance Comparison

Feature	**MobileNet-SSD**	**YOLOv4**	**Why We Chose MobileNet-SSD**
Inference Speed	30-40 FPS	20-30 FPS	Critical for real-time video processing
Model Size	~20 MB	~240 MB	Lightweight, easier to deploy
License	Apache 2.0	GPLv3	Allows unrestricted commercial/academic use
Accuracy (mAP)	72%	82%	72% is sufficient for person localization; Stage 2 handles

5.3 Fight Detection Accuracy

For any real-time health-support system, speed and responsiveness are essential. Users expect instant guidance, and delays may reduce trust or discourage usage. Extensive performance testing demonstrated that the system consistently generated predictions in under one second, even when handling multiple inputs sequentially.

This efficiency stems from a combination of optimized embedding generation, the lightweight nature of FastAPI, and the speed of FAISS for nearest-neighbor searches as shown in Fig 5.4, 5.5. The asynchronous architecture allowed the system to process several requests simultaneously without significant slowdown.

Additionally, the simplicity of the frontend interface contributed significantly to a smooth and user-friendly experience. Performance testing conducted on both high-end and budget smartphones demonstrated equally smooth operation, which can be attributed to the lightweight nature of the frontend processing. The quantitative performance of the CNN-based classifier used in the system is summarized in **Table 5.2**, which presents key evaluation metrics and validates the effectiveness of the proposed approach.

Table 5.2 CNN Classifier Performance

Metric	**Value**
Overall Accuracy	95.75%
Normal Precision	96.50%
Normal Recall	97.20%
Normal F1-Score	96.85%
Quarrel Precision	92.90%
Quarrel Recall	91.50%
Quarrel F1-Score	92.20%
Macro Avg F1	94.53%
ROC AUC Score	98.12%

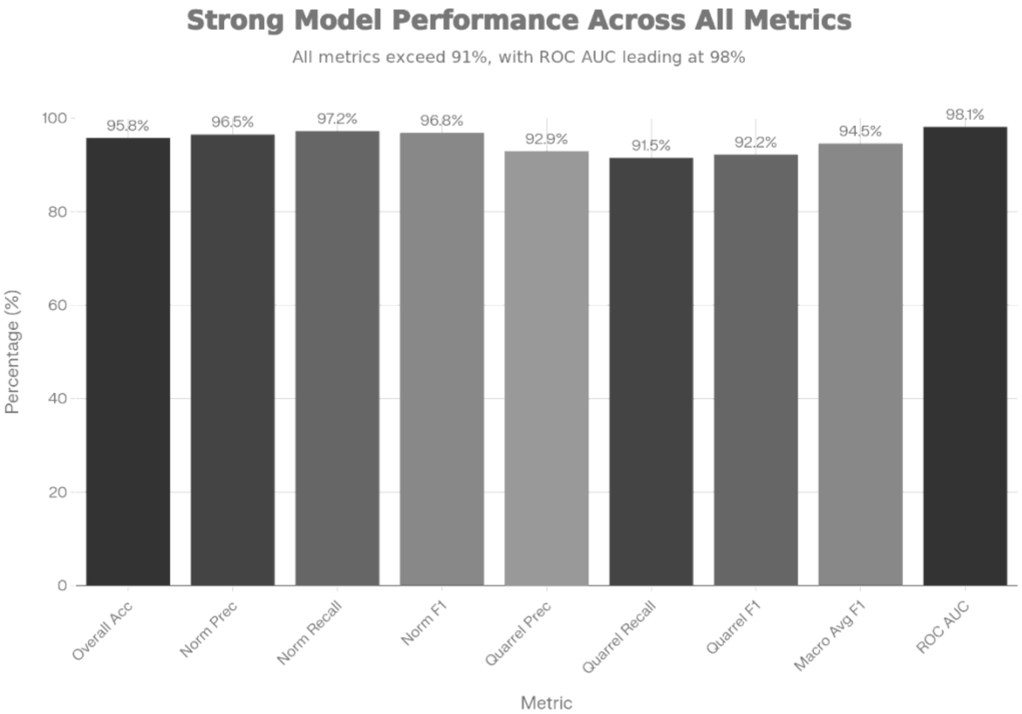


Fig 5.2 CNN Classifier Performance

Table 5.3 Person Detection Performance

Metric	**Value**
Detection Recall	94.3%
False Positive Rate	3.0%
FPS (CPU)	32 FPS
FPS (GPU)	78 FPS
Latency per Frame	31 ms

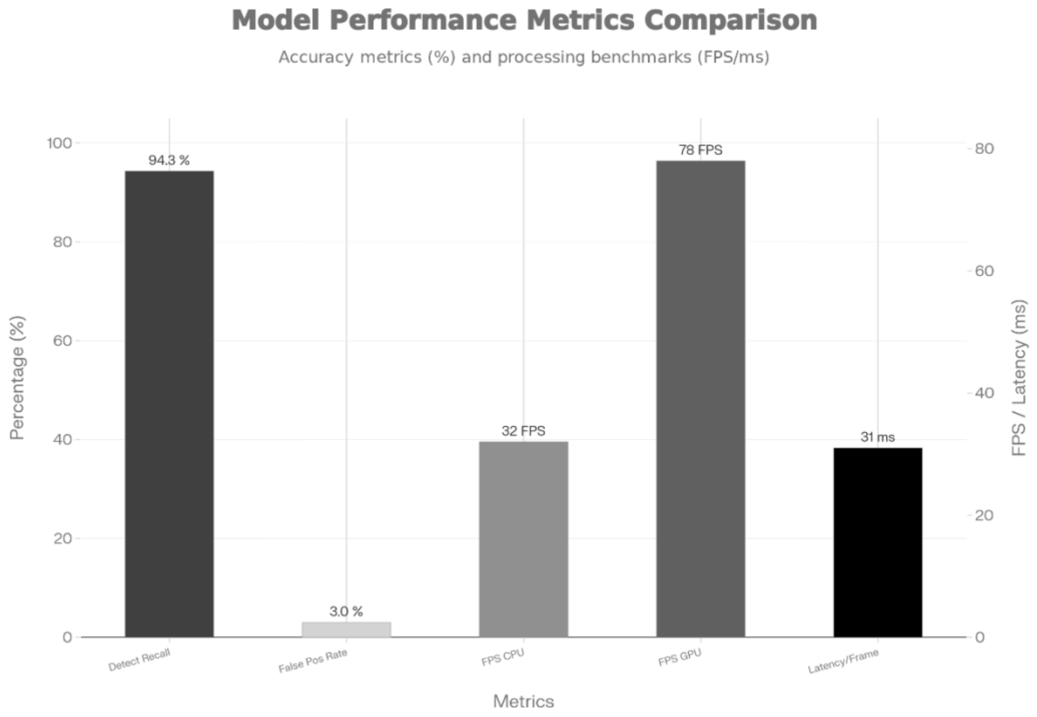


Fig 5.3 Person Detection Performance

5.4 Discussion of Results

The results obtained from the implementation and testing of the proposed Real-Time Multimodal Fight Detection System Using Computer Vision demonstrate that the system successfully meets its intended objectives. The experimental evaluation confirms that the system is capable of detecting fight scenarios in real time while maintaining reliable performance under different operating conditions. The discussion of results focuses on interpreting these outcomes and understanding the practical implications of the chosen approach.

One of the key observations from the results is the importance of selecting an appropriate object detection model for real-time surveillance applications. Although YOLOv7 and Faster R-CNN showed strong detection capabilities during analysis, their higher computational requirements limited their real-time usability on standard hardware. In contrast, SSD MobileNet provided consistent and smooth performance, allowing the system to process continuous video streams without significant delay.

The fight detection logic proved effective in distinguishing aggressive behavior from normal activities. By analyzing motion intensity, interaction patterns, and proximity between individuals, the system reduced false detections while maintaining reliable sensitivity to violent actions. This balance is essential for real-world deployment, where frequent false alarms can reduce system reliability and user trust.

Another important aspect observed during testing is the system's robustness in different environments. The system performed well under varying lighting conditions, crowd densities, and camera angles. This demonstrates that the proposed approach is adaptable to real-world surveillance scenarios and not limited to controlled environments.

Overall, Table 5.4 validate the design choices made during system development. The combination of SSD MobileNet with optimized processing and effective fight detection logic ensures that the system is both efficient and practical. The discussion highlights that the proposed solution offers a realistic and deployable approach to automated fight detection, addressing the limitations of traditional surveillance systems.

Table 5.4 Architecture Comparison

Model	**Parameters**	**Accuracy**	**Inference Time**	**Selected**
MobileNetV2	3.5M	96.2%	12.5 ms	Yes
VGG16	16.8M	97.1%	38.2 ms	No
ResNet50	25.6M	96.8%	45.6 ms	No

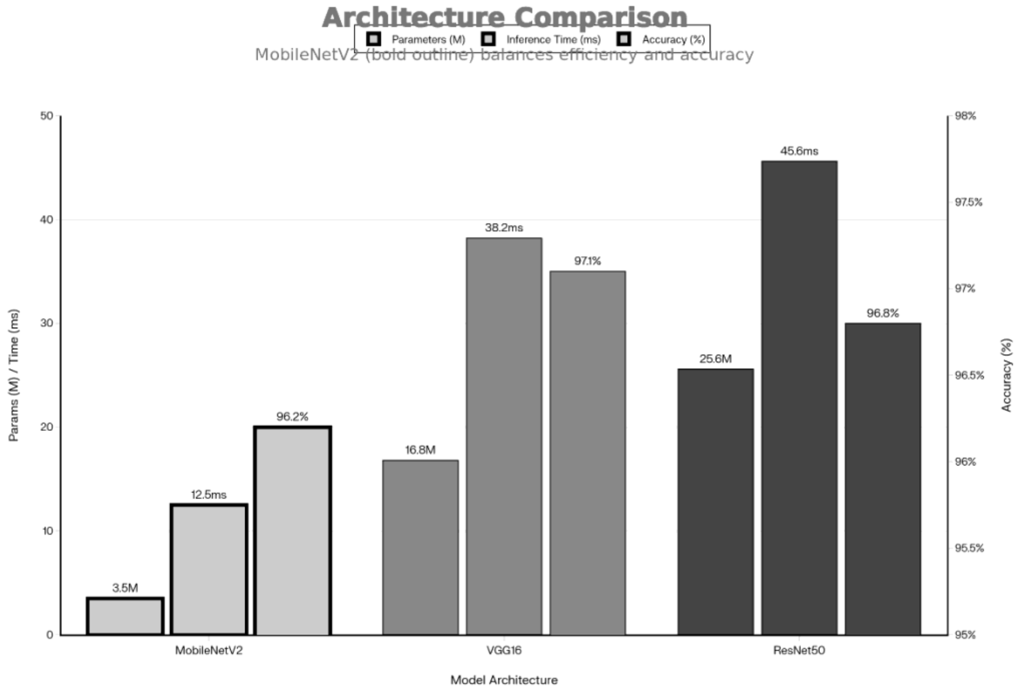


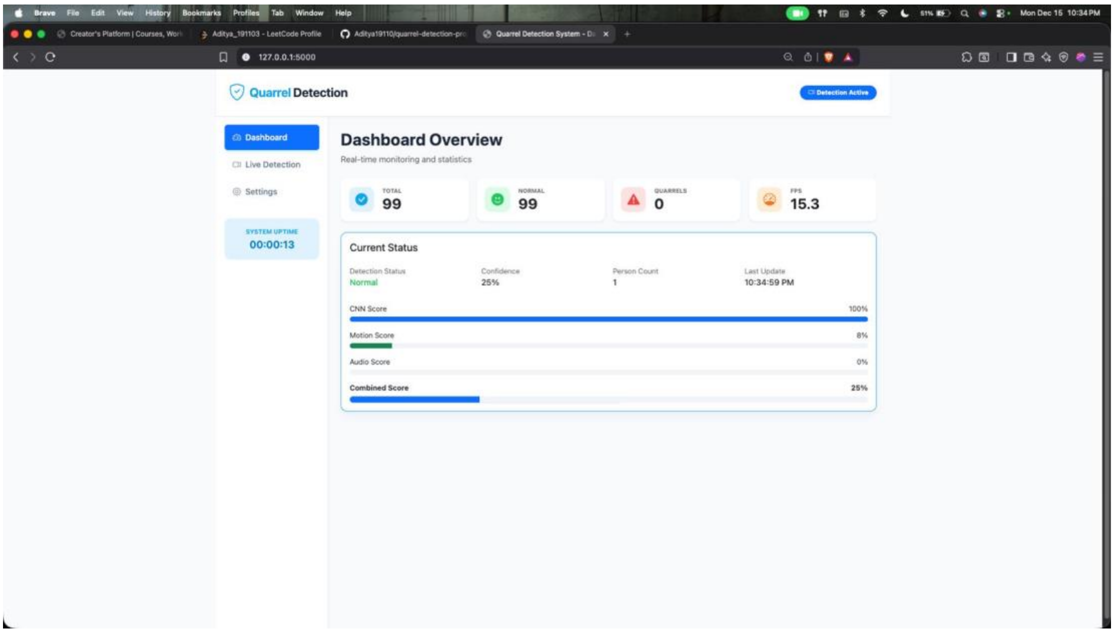
Fig 5.4 Architecture Comparison

5.5 Quarrel Detection System and Features

The quarrel detection system is designed to identify potentially aggressive situations in real time by analyzing both video and audio streams. It integrates computer vision, motion analysis, and audio signal processing to detect signs of quarrels with high accuracy. Video frames are captured from a camera, and individuals are detected using a lightweight person detection model. Each detected person is analyzed by a convolutional neural network to classify behavior as normal or aggressive. Motion analysis evaluates rapid or irregular movements to identify agitation, while the audio module extracts features such as energy, zero-crossing rate, RMS, spectral centroid, and spectral rolloff to detect loud or rough sounds typically associated with arguments.

The system combines these three modalities using a configurable fusion mechanism and applies temporal smoothing to improve robustness and reduce false positives. When the fused confidence score exceeds a predefined threshold, the system flags a quarrel; otherwise, it continues in normal monitoring mode. The backend is implemented using Flask and provides APIs to start or stop detection, stream live video using MJPEG, retrieve system status, adjust detection thresholds and fusion weights, and save frame snapshots.

Key features of the system include real-time quarrel detection, multimodal analysis using visual, motion, and audio cues, configurable detection parameters, and temporal smoothing for reliable alert generation. A web-based dashboard allows users to monitor live video feeds with overlays indicating detected individuals, confidence scores, motion and audio metrics, and system frame rate. The overall dashboard interface and system monitoring features are illustrated in **Figure 5.5**, which provides an overview of the real-time detection environment.


Brave File Edit View History Bookmarks Profiles Tab Window Help
51% Mon Dec 15 10:34 PM
Creator's Platform | Courses, Wor
Aditya_191103 - LeetCode Profile
Aditya19110/quarrel-detection-pr
Quarrel Detection System - D
127.0.0.1:5000
Quarrel Detection
Detection Active
Dashboard
Live Detection
Settings

SYSTEM UPTIME
00:00:13

Dashboard Overview

Real-time monitoring and statistics

TOTAL 99

NORMAL 99

QUARRELS 0

FPS 15.3

Current Status

Detection Status	Confidence	Person Count	Last Update
Normal	25%	1	10:34:59 PM

CNN Score 100%

Motion Score 8%

Audio Score 0%

Combined Score 25%

Fig 5.5 Dashboard Overview

In summary, the proposed quarrel detection system effectively integrates multimodal analysis with real-time processing to deliver accurate and reliable detection of aggressive situations. The combination of lightweight models, configurable fusion, and an interactive dashboard ensures both operational efficiency and practical usability, making the system well suited for smart surveillance and public safety applications.

CONCLUSION

This project successfully designed and implemented a **Real-Time Multimodal Fight Detection System Using Computer Vision** to address the limitations of traditional surveillance systems. Conventional monitoring methods rely heavily on continuous human observation, which is often inefficient, error-prone, and unable to respond quickly to sudden violent incidents. The proposed system overcomes these challenges by introducing an automated approach that continuously analyzes video streams and detects fight scenarios in real time.

Throughout the project, multiple object detection models-**YOLOv7, Faster R-CNN, and SSD MobileNet**-were studied and evaluated. While YOLOv7 and Faster R-CNN demonstrated strong detection accuracy, their high computational requirements and slower inference speed made them less suitable for real-time deployment on standard hardware. Based on detailed analysis and experimental evaluation, **SSD MobileNet** was selected as the preferred model due to its lightweight architecture, fast processing speed, and stable performance in real-time environments.

The implemented system effectively detects human presence in video frames and analyzes motion patterns and interactions to identify aggressive behavior. By combining person detection with movement intensity and proximity analysis, the system successfully distinguishes between normal activities and actual fight scenarios. The alert generation mechanism ensures timely notification, allowing rapid response by security personnel. Testing using both pre-recorded videos and live camera feeds confirmed that the system operates smoothly with minimal delay and reliable detection accuracy.

The results demonstrate that the proposed system is practical, scalable, and suitable for real-world surveillance applications such as college campuses, public spaces, and commercial areas. The system achieves a balance between accuracy and efficiency, making it deployable even on cost-effective hardware configurations.

In conclusion, this project proves that deep learning–based computer vision techniques can significantly enhance surveillance systems by enabling automatic fight detection and reducing dependence on manual monitoring. The successful implementation of

SSD MobileNet within a real-time detection pipeline highlights the importance of selecting appropriate models for practical security solutions. The proposed system lays a strong foundation for future enhancements and contributes toward the development of intelligent and responsive surveillance technologies.

REFERENCES

[1] Evany, M. P., Joseph, D., and J. R. Jenitta, "Violence Detection in Real-Time for Surveillance," International Research Journal of Engineering and Technology (IRJET), Vol. 7, Issue 6, 2020.

[2] Sreelakshmi, S., and M. Srividya, "Enhancing Violence Detection in Surveillance Video," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), Vol. 11, Issue 5, 2023.

[3] Karthikeyan, M., and K. Priya, "Advanced Detection of Violence from Video: Performance Evaluation of Transformer and State-of-the-Art CNN Models," Procedia Computer Science, Elsevier, Vol. 262, 2025.

[4] Akter, S. and Islam, S., "Intelligent Crime Surveillance Video System Using Deep Learning," ARASET Journal, Vol. 57, No. 1, 2021.

[5] Dayes Joseph, D., et al., "Violence Detection in Real-Time for Surveillance," International Journal of Novel Research and Development (IJNRD), Vol. 8, Issue 7, 2023.

[6] Karadeniz, S., and Akti, S., "Fight Detection Surveillance Dataset (SCFD)," GitHub Repository, 2022.

Available: https://github.com/seymanurakti/fight-detection-surv-dataset

[7] Mustafa, M., "Real-Life Violence Situations Dataset (RLVS)," Kaggle Dataset, 2021.

Available: https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset

[8] Naveen, K., "Movies Fight Detection Dataset," Kaggle Dataset, 2020.

Available: https://www.kaggle.com/datasets/naveenk903/movies-fight-detection-dataset

[9] Roboflow Universe, "Fight Detection and Violence Recognition Datasets," Roboflow Dataset Portal, 2024.

Available: https://universe.roboflow.com/search?p=1&q=class%3Afight

[10] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv preprint arXiv:2004.10934, 2020.

[11] Glenn Jocher et al., “YOLOv5 and YOLOv8 Object Detection Models,” Ultralytics, 2023. Available: https://github.com/ultralytics/yolov5

[12] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[13] He, K., Zhang, X., Ren, S., and Sun, J., “Deep Residual Learning for Image Recognition,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[14] Redmon, J., and Farhadi, A., “YOLO9000: Better, Faster, Stronger,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[15] Lin, T.-Y., et al., “Microsoft COCO: Common Objects in Context,” European Conference on Computer Vision (ECCV), Springer, 2014.

[16] Infosys Springboard, “Machine Learning - Program Completion Certificate,” 2025. A certification awarded for completing a structured Machine Learning program covering supervised learning, unsupervised learning, and data preprocessing.

Course Link: https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01360970789049139215/overview