Project Report

on

WEAPON DETECTION

In partial fulfillment of requirements for the degree

of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

SANSKRITI SHARMA [21100BTCSAII09438]

HARSHIT RAJPAL [20100BTCSAII07162]

VARUN KILLEDAR [20100BTCSAII07196]

Under the guidance of

MR. AVDESH KUMAR SHARMA

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JUL-DEC-2023

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We here declare that work which is being presented in the project entitled "**Weapon Detection System**" in partial fulfilment of degree of **Bachelor of Technology** in **Computer Science & Engineering** is an authentic record of our work carried out under the supervision and guidance of **Mr.Avdesh Kumar Sharma** of Computer Science & Engineering. The matter embedded in this project has not been submitted for the award of any other degree

Sanskriti Sharma
Harshit Rajpal
Varun Killedar

Date:

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROJECT APPROVAL SHEET

The following team has done the appropriate work related to the "**Weapon Detection System**" in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science & Engineering** of "SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY" and is being submitted to SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE.

Team:

1. Sanskriti Sharma
2. Harshit Rajpal
3. Varun Killedar

Internal Examiner **External Examiner**

Date:

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that **Sanskriti Sharma , Harshit Rajpal** and **Varun Killedar** working in a team have satisfactorily completed the project entitled "**WEAPON DETECTION SYSTEM** " under the guidance of **Mr.Avdesh Kumar Sharma** in the partial fulfilment of the degree of **Bachelor of Technology** in **Computer Science & Engineering** awarded by SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY affiliated to SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA , INDORE during the academic year **Jul-Dec-2022**.

Mr.Avdesh Kumar Sharma
Project Guide

Mr. Vishwas Dixit
Project Coordinator

Dr. Anand Rajavat

Director & Head,
Department of Computer Science & Engineering

ACKNOWLEDGEMENT

We are grateful to a number of persons for their advice and support during the time of complete our project work. First and foremost, our thanks goes to **Dr. Anand Rajavat** Head of the Department of Computer Science & Engineering and **Mr.Avdesh Kumar Sharma** the mentor of our project for providing us valuable support and necessary help whenever required and also helping us explore new technologies by the help of their technical expertise. His direction, supervision and constructive criticism were indeed the source of inspiration for us.

We would also like to express our sincere gratitude towards our Director **Dr. Anand Rajavat** for providing us valuable support.

We are really indebted to **Mr. Vishwas Dixit** project coordinator for helping us in each aspect of our academic's activities. We also owe our sincere thanks to all the faculty members of Computer Science & Engineering Department who have always been helpful

.
We forward our sincere thanks to all **teaching and non-teaching staff** of Computer Science & Engineering department, SVVV Indore for providing necessary information and there kind co-operation.

We would like to thanks our parents and family members, our classmates and our friends for their motivation and there valuable suggestion during the project. Last, but not the least, we thank all those people, who have helped us directly or indirectly in accomplishing this work. It has been a privilege to study at SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE.

ABSTRACT

In recent years, concerns related to public safety and security have driven significant advancements in weapon detection technologies. This abstract presents a state-of-the-art Weapon Detection System (WDS) utilizing cutting-edge deep learning algorithms and computer vision techniques. The proposed system aims to enhance security measures by accurately identifying and classifying weapons in real-time scenarios.

The WDS leverages Convolutional Neural Networks (CNNs) to extract intricate features from input images or video frames, enabling precise detection of swords, knives, handguns, and other dangerous objects. Through the integration of advanced deep learning architectures, such as YOLO (You Only Look Once), the system achieves high accuracy rates and rapid processing speeds, making it suitable for deployment in high-security environments.

One of the key features of the proposed system is its ability to perform real-time weapon detection, allowing for instantaneous response to potential threats. By utilizing parallel processing and optimized algorithms, the system achieves low latency, enabling quick and effective decision-making in critical situations. The WDS is designed to be adaptable and scalable, facilitating seamless integration with existing security infrastructure and surveillance systems.

Furthermore, the WDS emphasizes the importance of privacy preservation. By addressing ethical concerns and privacy regulations, the system ensures the protection of individuals' privacy while maintaining a high level of security.

LIST OF FIGURES

TABLE OF CONTENT

Chapter - 1

Introduction

1.1 Introduction

The crime rate across the globe has increased mainly because of the frequent use of handheld weapons during violent activity. For a country to progress, the law-and-order situation must be in control. Whether we want to attract investors for investment or to generate revenue with the tourism industry, all these needs is a peaceful and safe environment. The crime ratio because of guns is very critical in numerous parts of the world. It includes mainly those countriesin which it is legal to keep a firearm. The world is a global village now and what we speak or write has an impact on the people. Even if the news they heard is crafted having no truth but as itgets viral in a few hours because of the media and especially social media, the damage will be done. People now have more depression and have less control over their anger, and hate speeches can get those people to lose their minds. People can be brainwashed and psychological studies show that if a person has a weapon in this situation, he may lose his senses and commit a violent activity. High incidents were recorded in past few years with the use of harmful weapons in public areas. Starting with the past year's attacks on a couple of Mosques in New Zealand, on March 15, 2019 at 1:40 pm, the attacker attacks the Christchurch AL-Noor Mosque during a Friday prayer killing almost 44 innocent and unarmed worshippers. On the same day just after 15minutes at 1:55 PM, another attack happened killing seven more civilians [1]. Active shooter incidents had also occurred in USA and then in Europe. The most significant cases were those at Columbine High School (USA, 37 victims), Andreas Broeivik's assault on Uotya Island (Norway, 179 victims) or the Charlie Hebdo newspaper attack killing 23. According to stats provided by the UNODC, among 0.1 Million people of a country, the crimes involving guns are very high i-e. 1.6 in Belgium, United States having 4.7 and Mexico with a number of 21.5 [2].

1.2 Problem Statement

The term The need for a reliable and accurate weapon detection system arises from the increasing number of violent incidents and security threats around the world. Traditional security measures such as metal detectors and manual searches are time-consuming and can be easily bypassed by individuals who intend to harm others. Therefore, there is a growing need for automated systems that can detect weapons in real-time and alert security personnel when a potential threat is detected. The proposed solution involves using the YOLOv5 object detection algorithm to train a deep neural network to identify weapons in images and video streams. The system will be trained on a large dataset of images and videos that contain weapons, including handguns, rifles, and other types of firearms. Once trained, the system can be deployed in real-world scenarios, where it can process live feeds from surveillance cameras or other sensors and detect weapons in real- time.The ultimate goal of this project is to create an accurate and reliable weapon detection system that can help enhance security measures and reduce the risk of violent incidents in public spaces.

1.3 Need For the Proper System

As Implementing a robust weapon detection system is imperative in today's world, where ensuring public safety and security is paramount. These systems play a crucial role in preventing violence by swiftly identifying individuals carrying firearms or dangerous weapons in public spaces and sensitive environments such as airports, schools, government buildings, corporate offices, and public events. The mere presence of weapon detection systems acts as a powerful deterrent, dissuading potential wrongdoers from attempting violent acts due to the high risk of detection. Additionally, these systems are instrumental in averting mass shootings, especially in educational institutions and crowded gatherings, by denying entry to unauthorized persons with firearms. From a law enforcement perspective, these technologies aid in crime prevention, provide essential evidence for investigations, and enhance overall situational awareness. Leveraging advanced technologies like artificial intelligence and machine learning, coupled with adherence to legal mandates, ensures accurate detection and rapid response, ultimately safeguarding human lives and instilling a sense of security within communities.

1.4 Objective

The objectives of our application is:-

- Enhanced Security: The use of YOLOv5 for weapon detection can significantly enhancethe security of sensitive areas such as airports, public events, and government facilities. The system can quickly detect concealed weapons, making it difficult for potential attackers to bring in weapons and cause harm.
- Customizable System: YOLOv5 is highly customizable, allowing developers to fine-tunethe system for specific applications. This means that the system can be trained to detect different types of weapons, such as pistols, rifles, and knives, with high accuracy.
- Real-time Detection: YOLOv5 provides real-time detection of weapons, allowing security personnel to respond quickly to potential threats. This is particularly important inhigh-risk scenarios, such as mass shootings, where every second counts. The real-time detection capabilities of YOLOv5 can help to prevent violent incidents from escalating and save lives.

1.5 Modules of the System

1. Input Module (Video Feed):

- Video Input: Capture video feed from surveillance cameras or existing security systems.

2. Image and Video Processing:

- Image Capture: Extract frames from the video feed for analysis.
- Image Enhancement: Enhance image quality, adjust brightness, contrast, and apply filters to improve object visibility.
- Object Detection: Utilize computer vision algorithms (e.g., YOLO, SSD) to identify and locate objects, including weapons, within video frames.
- Motion Detection: Identify moving objects within the video stream to detect suspicious activities.

3. Machine Learning and Deep Learning:

- Feature Extraction: Extract relevant features from images or video frames that are useful for weapon detection.
- Training Data Preparation: Prepare labeled datasets for training machine learning models.
- Model Training: Train machine learning or deep learning models (e.g., convolutional neural networks) to recognize weapons based on extracted features.
- Model Evaluation: Evaluate the trained models using validation datasets to ensure accuracy and adjust hyperparameters if necessary.

4. Decision-Making Module:

- Threshold Setting: Establish thresholds for the confidence level of detected weapons to minimize false positives and false negatives.
- Alert Generation: Generate alerts or notifications when a weapon is detected above the specified confidence threshold.
- Integration with Security Systems: Integrate the system with security systems to trigger alarms, notifications to authorities, or other security protocols.

5. User Interface:

- Visualization: Provide a user-friendly interface to display real-time or recorded video feeds with detected weapons highlighted.
- Configuration: Allow users to configure system parameters, thresholds, and other settings.
- Alert Management: Enable users to view and manage alerts, review historical data, and generate reports.

6. Database and Logging:

- Data Storage: Store processed data, detected objects, and alerts in a secure and organized database for future reference and analysis.
- Logging: Maintain logs of system activities, including detections, alerts, and user interactions, for auditing and troubleshooting purposes.

7. Maintenance and Update:

- Health Monitoring: Implement monitoring mechanisms to detect system failures or degradation in performance.
- Software Updates: Provide a mechanism for system updates to incorporate improvements, bug fixes, and new threat detection techniques.

1.6 Scope

1. **Video Analysis and Object Recognition:** The system analyzes live CCTV footage using advanced computer vision algorithms to detect suspicious objects, specifically focusing on identifying weapons based on their shape, size, and appearance.

2. **Behavioral Analysis:** Utilizing machine learning, the system identifies suspicious behaviors or movements exhibited by individuals that could indicate the presence of a concealed weapon.

3. **Pattern Recognition:** Recognizing patterns associated with weapons, allowing the system to detect concealed items even if partially obscured or concealed under clothing.

4. **Real-time Alert Generation:** Generating immediate alerts in real-time when a potential weapon is detected, ensuring rapid response by security personnel.

5. **Integration with Access Control:** Seamless integration with access control systems to automate security responses, such as denying entry or triggering alarms, when a weapon is identified on CCTV.

6. **Continuous Monitoring:** 24/7 monitoring of CCTV feeds, ensuring constant surveillance and threat detection regardless of the time of day, providing a proactive security approach.

7. **Remote Monitoring and Management:** The system allows for remote monitoring and management, enabling security personnel to access live CCTV feeds and respond to potential threats from a centralized location.

8. **Data Logging and Reporting:** Logging and storing data related to weapon detection incidents, allowing for analysis, compliance reporting, and improving system performance over time.

9. **Privacy Protection:** Implementing privacy measures to anonymize personal information and comply with data protection regulations, ensuring the privacy of individuals while performing weapon detection.

10. **Customization and Adaptability**: Customizing the system to suit various environments, including public transportation, commercial spaces, government buildings, and public events, adapting to specific security needs and threat levels of each location.

Chapter - 2

Literature Survey

2. Literature Survey

1. Automated detection of firearms and knives in a CCTV image:

Closed circuit television systems (CCTV) are becoming more and more popular and are being deployed in many offices, housing estates and in most public spaces. Monitoring systems have been implemented in many European and American cities. This makes for an enormous load for the CCTV operators, as the number of camera views a single operator can monitor is limited by human factors. In this paper, we focus on the task of automated detection and recognition of dangerous situations for CCTV systems. We propose algorithmsthat are able to alert the human operator when a firearm or knife is visible in the image. We have focused on limiting the number of false alarms in order to allow for a real-life application of the system. The specificity and sensitivity of the knife detection are significantly better than others published recently. We have also managed to propose a version of a firearm detection algorithm that offers a near-zero rate of false alarms. We have shown that it is possible to create a system that is capable of an early warning in a dangerous situation, which may lead to faster and more effective response times and a reduction in the number of potential victims.

2. A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery:

local density descriptor, through local region histograms and three-dimensional (3D) extensions to both to the RIFT descriptor and the seminal SIFT feature descriptor. We show that, in the complex CT imagery domain containing a high degree of noise and imaging artefacts, a specific instance object recognition system using simpler descriptors appears to outperform a more complex RIFT/SIFT solution. Recognition rates in excess of 95% are demonstrated with minimal false-positive rates for a set of exemplar 3D objects.

3. A computer vision based framework for visual gun detection using Harris interest point detector:

Today's automatic visual surveillance is prime need for security and this paper presents first step in the direction of automatic visual gun detection. The objective of our paper isto

develop a framework for visual gun detection for automatic surveillance. The proposed framework exploits the color based segmentation to eliminate unrelated object from an image using k-mean clustering algorithm. Harris interest point detector and Fast Retina Keypoint (FREAK) is used to locate the object (gun) in the segmented images. Our framework is robust enough in terms of scale, rotation, affine and occlusion. We have implemented and tested the system over sample images of gun, collected by us. We got promising performance of our system to detect a gun. Further, our system performs very well under different appearance of images. Thus our system is rotation, scale and shape invariant.

2.1 Existing System

The modern world is very concerned with security and safety. A nation's ability to attract foreign investment and tourism depends on how safe and secure its environment is. However, even if Closed Circuit Television (CCTV) cameras are employed for surveillance and to keep an eye on actions like robberies, they still need human oversight and involvement. We require a system that can instantly identify these criminal acts. Even with cutting-edge deep learning algorithms, quick processing power, and sophisticated CCTV cameras, real-time weapon detection remains a significant barrier. The challenge is made more difficult by observing from different angles and being obscured by the gun's owner and those nearby.

2.2 Proposed System

The goal of this effort is to create a safe environment by utilising CCTV footage as a source to identify dangerous weapons using cutting-edge open-source deep learning algorithms. To eliminate false positives and false negatives, we have built binary classification using pistol class as the reference class and created relevant confusion items inclusion idea. Since there was no standard dataset for the real-time scenario, we created our own by taking shots of weapons with our own cameras, manually gathering images from the internet, extracting data from YouTube CCTV recordings, and using GitHub repositories. Sliding window/classification and regionproposal/object detection are the two methods applied. The algorithms YOLOV5, YOLOV6, YOLOV&, and Faster RCNN are a few that are employed.

2.3 Feasibility Study

2.3.1 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

2.3.2 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.

2.3.3 Operational Feasibility

Operational feasibility for a Weapon Detection System involves assessing its compatibility with existing systems, user acceptance, resource availability, potential operational impact, compliance with regulations, scalability, and the availability of maintenance and support services. It ensures the system can be effectively implemented and operated within the organization's existing framework.

Chapter - 3

Requirement Analysis

3.1 Method Used for Requirement Analysis

A software requirement is a capability needed by the user to solve a problem or to achieve an objective. In other words, requirement is a software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation. Ultimately, what we want to achieve is to develop quality software that meets customers' real needs on time and within budget. Perhaps the greatest challenge being faced by software developers is to share the vision of the final product with the customer. All stakeholders in a project – developers, end users, software managers, customer managers – must achieve a common understanding of what the product will be and do, or someone will be surprised when it is delivered. Surprises in software are almost never good news. Therefore, we need ways to accurately capture, interpret, and represent the voice of customers when specifying the requirements for a software product. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Conceptually, requirements analysis includes four types of activity:

A. Eliciting requirements: the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.

B. Analyzing requirements: determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.

C. Requirements modeling: Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

1. Use Cases

A UML use case diagram is the primary form of system/software requirements for a new software program under development. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (such as UML). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

2. User Stories

A user story is a note that captures what a user does or needs to do as part of his/her work. Each user story consists of a short description written from the user's point of view, with natural language. Unlike the traditional requirement capturing, the user story focuses on what the user needs instead of what the system should deliver. This leaves room for further discussion of solutions and the result of a system that can really fit into the customers' business workflow, solving their operational problems and most importantly adding value to the organization. User stories are well compatible with the other agile software development techniques and methods, such as scrum and extreme programming.

D. Review and retrospective: Team members reflect on what happened in the iteration and identify actions for improvement going forward.

Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people. Here are the main activities involved in requirement analysis:

- Identify customer's needs.
- Evaluate systems for feasibility.
- Perform economic and technical analysis.
- Allocate functions to system elements.
- Establish schedule and constraints.
- Create system definitions.

3.2 Data Requirements

1. Video Data: High-quality video feeds from CCTV cameras are fundamental. These visual inputs provide the system with real-time imagery for analysis.
2. Image Processing Data: Detailed image processing data is necessary for analyzing still frames or video frames to identify suspicious objects or patterns associated with weapons.
3. Machine Learning Training Data: An extensive dataset of labeled images containing weapons and non-weapons is crucial for training machine learning algorithms. This dataset helps the system recognize and differentiate weapons from other objects accurately.
4. Metadata: Information about the date, time, and location of the video feeds is essential for contextual analysis and tracking suspicious activities over time.
5. Access Control Data: Integration with access control systems provides data about authorized personnel, enabling the system to differentiate between authorized individuals and potential threats.
6. Alert History: Records of past alerts and system responses provide valuable data for system performance analysis and improvement.
7. System Configuration Data: Information about the system's configuration, including camera placements, calibration data, and algorithm parameters, is necessary for optimal performance.

3.3 Functional Requirements

1. Data Collection:

Data collection is the process of collecting, measuring, and analysing data from various sources to gain insights. Data can be collected through various sources, such as social media monitoring, online tracking, surveys, feedback, etc. In fact, there are three main categories of data that businesses endeavour to collect.

2. Data Pre-processing:

Data pre-processing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process. More recently, data pre-processing techniques have been adapted for training machine learning models and AI models and for running inferences against them.

3. Training And Testing:

Training and testing are crucial steps in the development of machine learning models, including those used in computer vision tasks such as image classification, object detection, and object recognition. Training involves using a set of labelled data, called the training set, to teach a machine learning algorithm how to recognize patterns and make predictions. After the model is trained, it is evaluated on a separate dataset called the testing set. The testing set is used to assess how well the model performs on new, unseen data.

4. Modeling :

Modelling is an important aspect of functional requirements in software development. It involves creating abstract representations of a system or software application to help stakeholders understand its behaviour, structure, and functionality

5.Predicting:

Predicting is another important aspect of functional requirements in software development. It involves anticipating the behaviour of the system or software application in response to different inputs or user actions. Predictive modelling can help software developers and stakeholders understand how the system will perform

3.4 Non-Functional Requirements

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allow you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is > 10000. Description of non-functional requirements is just as critical as a functional requirement.

1. Usability requirement: Usability is a non-functional requirement that refers to how easy andefficient it is for users to interact with a system or product.

2. Serviceability requirement: Serviceability is a non-functional requirement that refers to howeasy it is to maintain, repair, and support a system or product. Serviceability requirements help ensure that a product can be easily serviced, minimizing downtime and reducing maintenance costs.

3. Manageability requirement: Manageability is a non-functional requirement that refers tohow easily a system or product can be managed and administered.

4. Recoverability requirement: Recoverability is a non-functional requirement that refers to the ability of a system or product to recover from failures or errors. Recoverability requirements help ensure that a product can continue to function even in the event of unexpected errors or outages

5. Security requirement: Security is a critical non-functional requirement that refers to the protection of a system or product from unauthorized access, theft, damage, or misuse. Security requirements help ensure that a product is secure and can protect sensitive data and information.

6. Data Integrity requirement: Data Integrity Requirement: Data integrity is a non-functionalrequirement that refers to the accuracy, consistency, and reliability of data stored in a system orproduct. Data integrity requirements help ensures that data is not corrupted or lost during processing, storage, or transmission. Some common data integrity requirements include data validation, error handling, and data backup and recovery.

7. Availability requirement: Availability Requirement: Availability is a non-functional requirement that refers to the ability of a system or product to be accessible and operational during specified periods. Availability requirements help ensure that a product is reliable and canbe accessed when needed. Some common availability requirements include uptime percentage, mean time between failures, and mean time to repair.

8. Scalability requirement: Scalability Requirement: Scalability is a non-functional requirement that refers to the ability of a system or product to handle increasing workloads or user demands without sacrificing performance or reliability. Scalability requirements help ensurethat a product can grow and adapt to changing requirements over time. Some common scalabilityrequirements include horizontal scalability, vertical scalability, and load balancing.

3.5 System Specification

3.5.1 Hardware Specification

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or

Processing power – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored.

Memory – All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Secondary storage – Hard-disk requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display adapter – Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements.

1.1 Peripherals –

- Operating System : Windows Only
- Processor : i7 and above
- Ram : 4gb and above
- Hard Disk : 50 GB

application. The following sub- sections discuss the various aspects of hardware requirements.

3.5.2 Software Specification

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform – In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.

Operating system is one of the first requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

APIs and drivers – Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms.

Chapter - 4

Design

4.1 Software Requirement Specification

4.1.1 Glossary

- **Machine Learning -** Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.
- **Weapon Detection System (WDS):** The comprehensive software solution developed to enhance public safety and security by accurately identifying weapons in various environments.
- **Real-time Analysis:** The process of analyzing video feeds and images instantaneously to identify weapons.
- **Surveillance Cameras:** Devices that capture live video feeds used as input for the Weapon Detection System.
- **Access Control Systems:** Security systems that control access to physical or virtual resources, validating detected threats against authorized personnel databases.
- **Machine Learning Algorithms:** Algorithms that enable the system to improve its accuracy and adaptability to new threat patterns.
- **Computer Vision:** A field of study enabling computers to interpret visual information from the world, used in the identification of weapons.
- **API (Application Programming Interface):** A set of protocols and tools for building software and applications, allowing different systems to communicate with each other.
- **RTSP (Real Time Streaming Protocol):** A network control protocol used in video streaming.
- **ONVIF (Open Network Video Interface Forum):** A global and open industry forum with the goal of facilitating the development and use of a global open standard for the interface of physical IP-based security products.
- **SMTP (Simple Mail Transfer Protocol):** An internet standard for electronic mail (email) transmission.

4.1.2 Use Case Model

A Use Case consists of use cases, persons, or various things that are invoking the features called as actors and the elements that are responsible for implementing the use cases. Use case diagrams capture the dynamic behaviour of a live system. It models how an external entity interacts with the system to make it work. Use case diagrams are responsible for visualizing the external things that interact with the part of the system.

Following are the common notations used in a use case diagram:

1. Use-case: Use cases are used to represent high-level functionalities and how the user will handle the system. A use case represents a distinct functionality of a system, a component, a package, or a class. It is denoted by an oval shape with the name of a use case written inside the oval shape. The notation of a use case in UML is given below:

FIGURE 4.1 Use Case Notation

Actor: It is used inside use case diagrams. The actor is an entity that interacts with the system. A user is the best example of an actor. An actor is an entity that initiates the use case from outside the scope of a use case. It can be any element that can trigger an interaction with the use case. One actor can be associated with multiple use cases in the system. The actor notation in UML is given below.

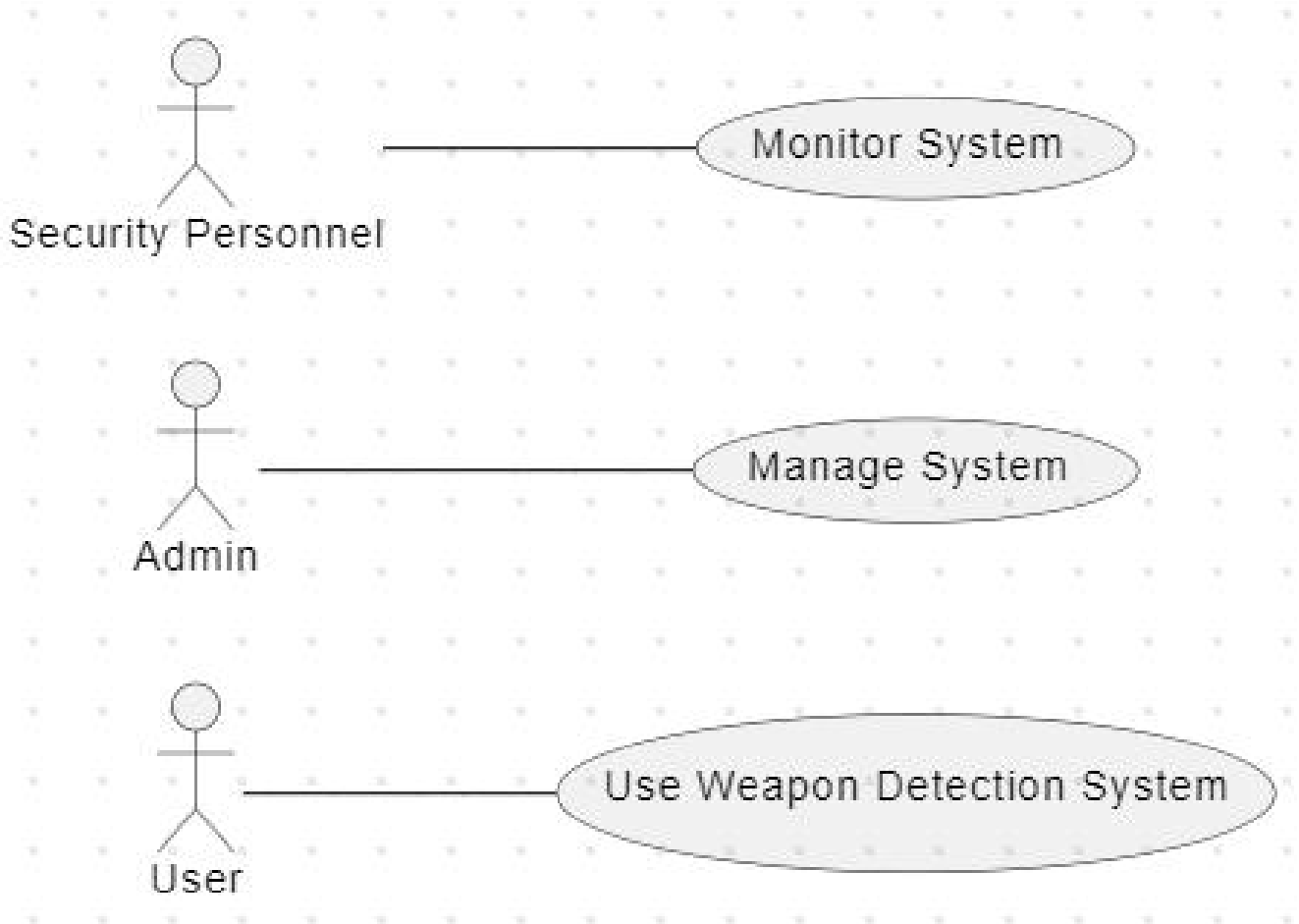

FIGURE 4.2 ACTOR NOTATIONS

USE CASE MODEL OF PROJECT

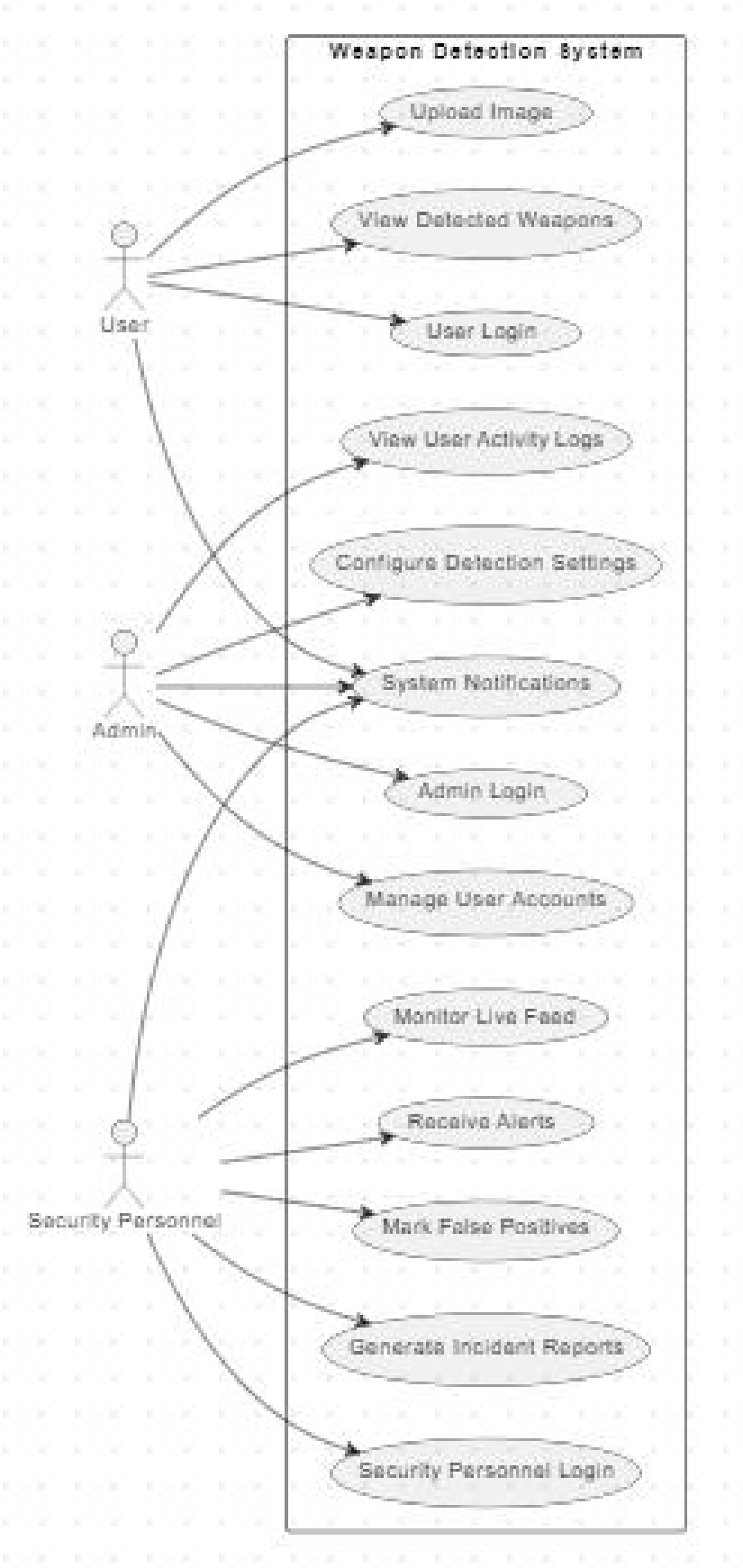

FIGURE 4.3 USE CASE MODEL

4.2 Conceptual level class diagram

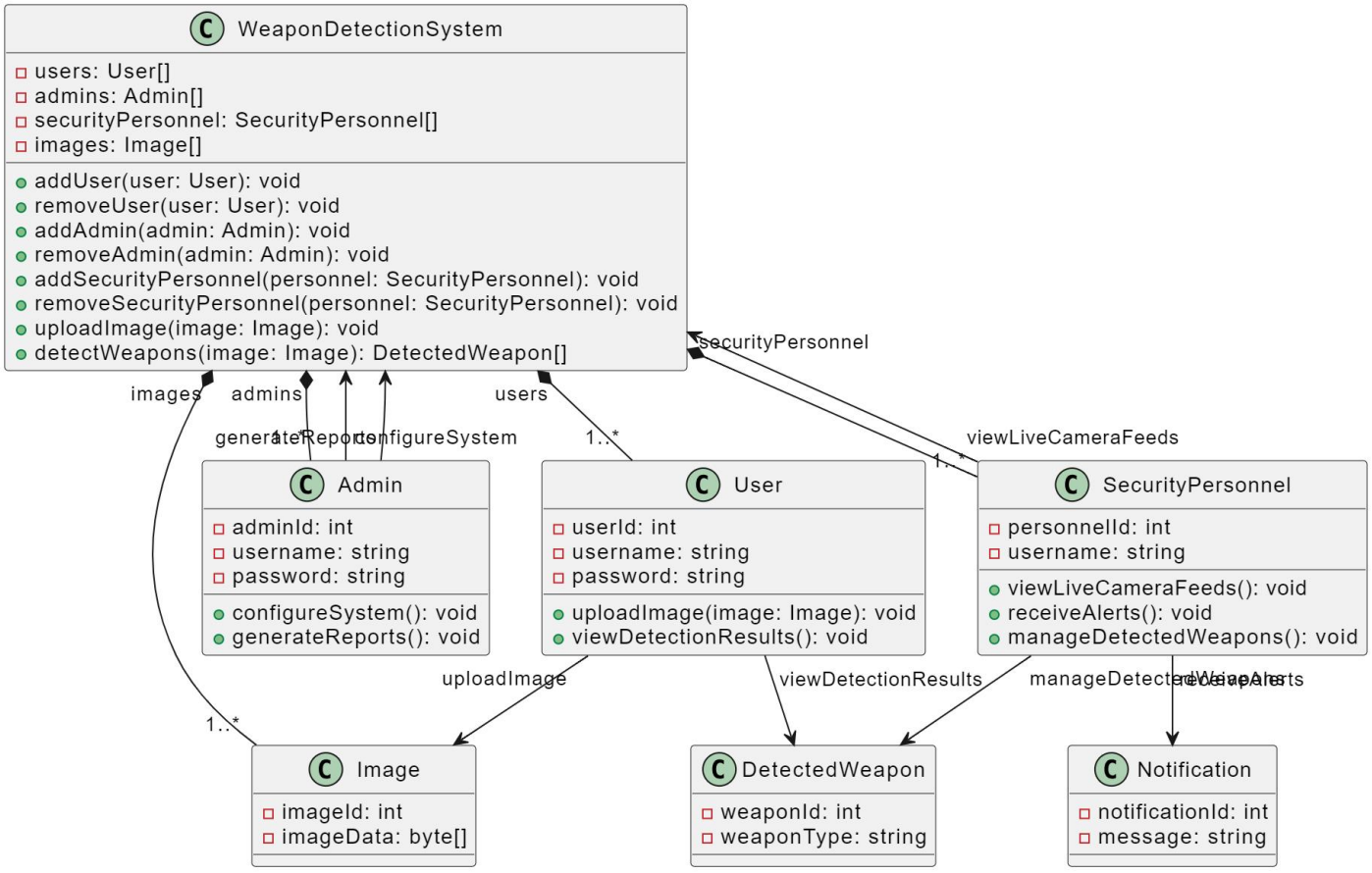

FIGURE 4.4 CONCEPTUAL LEVEL CLASS DIAGRAM

4.3 Conceptual level activity diagram

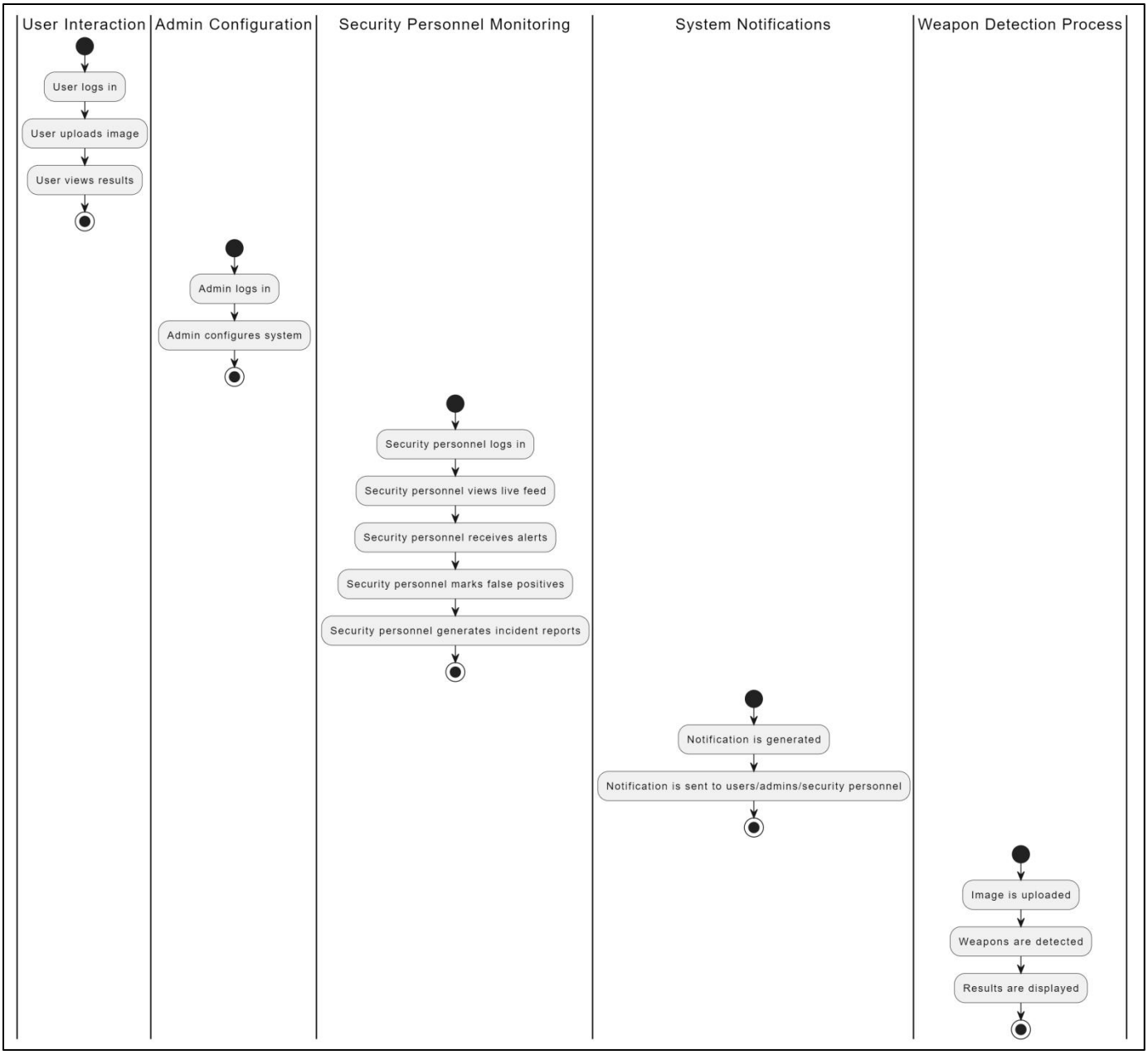

FIGURE 4.5 CONCEPTUAL LEVEL ACTIVITY DIAGRAM

4.4 Data flow diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.

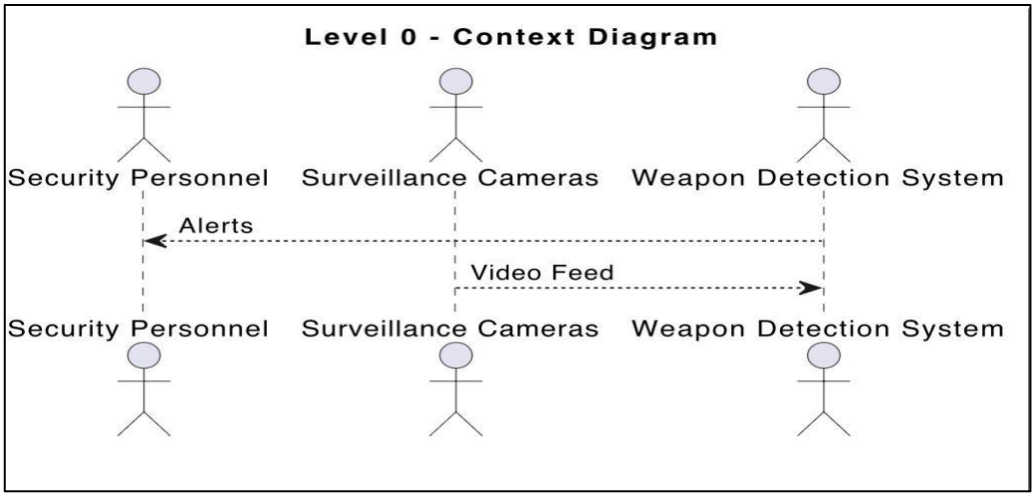

FIGURE 4.6.1 DATA FLOW DIAGRAM LEVEL 0

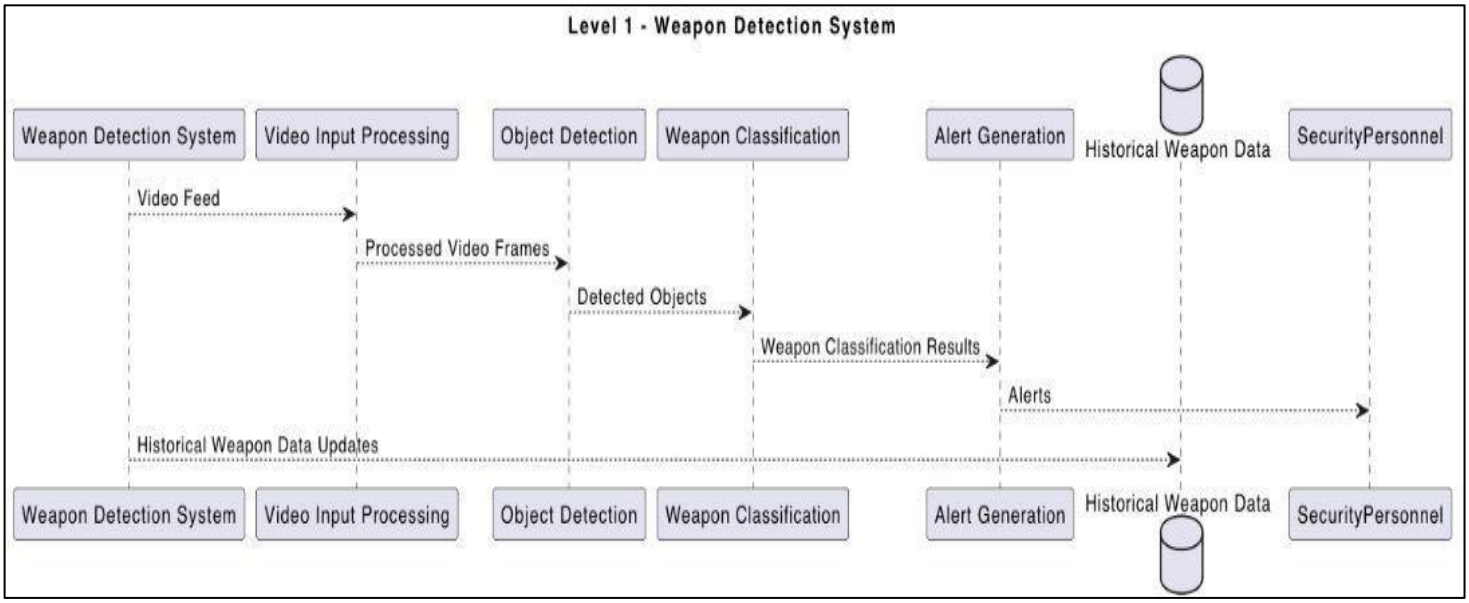

FIGURE 4.6.2 DATA FLOW DIAGRAM LEVEL 1

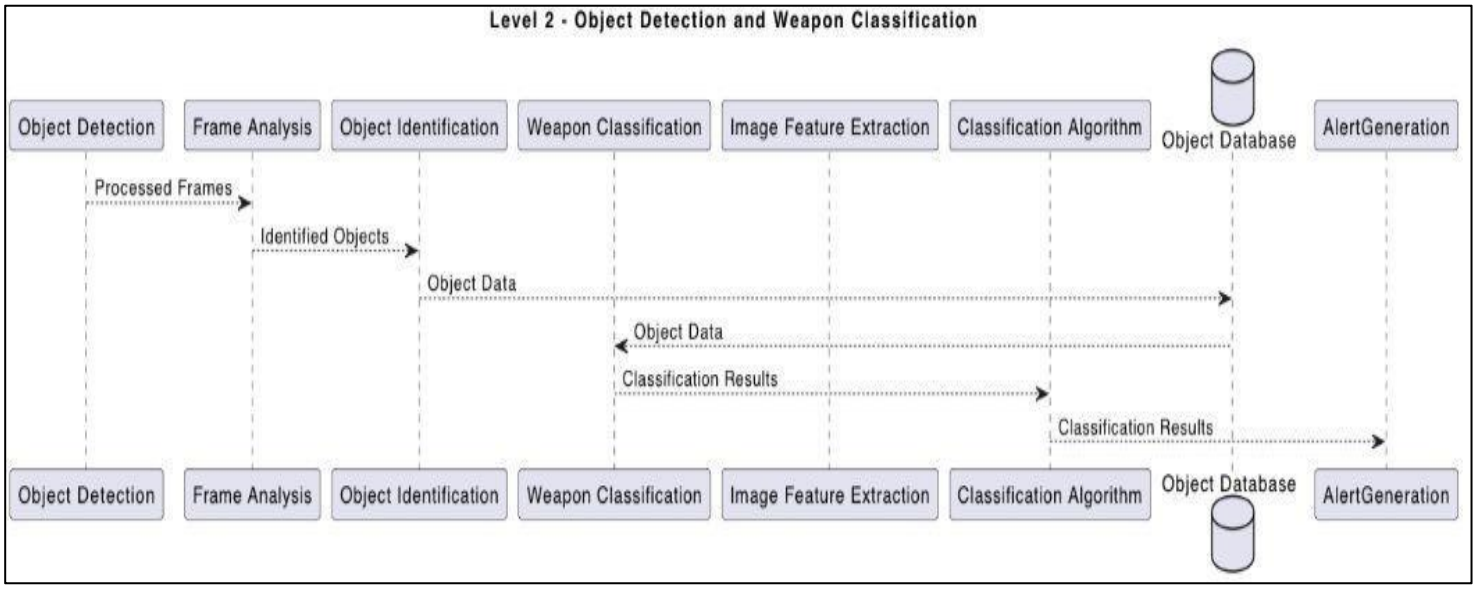

FIGURE 4.6.1 DATA FLOW DIAGRAM LEVEL 2

Chapter - 5

System Modeling

5.1 Detailed Class Diagram

UML CLASS DIAGRAM gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.

Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations

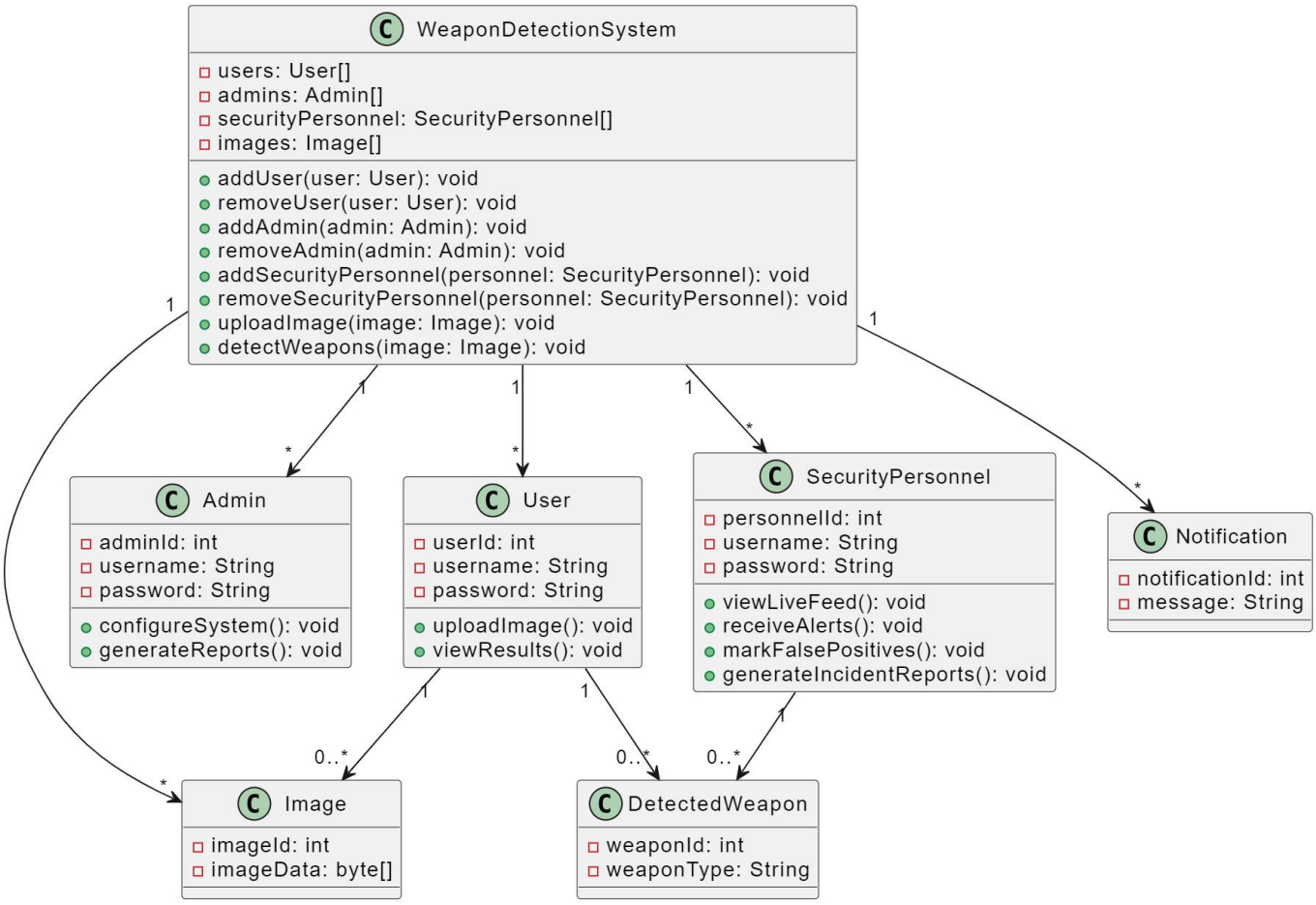

FIGURE 5.1 DETAILED CLASS DIAGRAM

5.2 Interaction Diagram

INTERACTION DIAGRAMS are used in UML to establish communication between objects. It does not manipulate the data associated with the particular communication path. Interaction diagrams mostly focus on message passing and how these messages make up one functionality of a system. Interaction diagrams are designed to display how the objects will realize the particular requirements of a system. The critical component in an interaction diagram is lifeline and messages.

5.2.1 Sequence Diagram

SEQUENCE DIAGRAM simply depicts interaction between objects in a sequential order. The purpose of a sequence diagram in UML is to visualize the sequence of a message flow in the system. The sequence diagram shows the interaction between two lifelines as a time-ordered sequence of events. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

SEQUENCE DIAGRAMS OF PROJECT

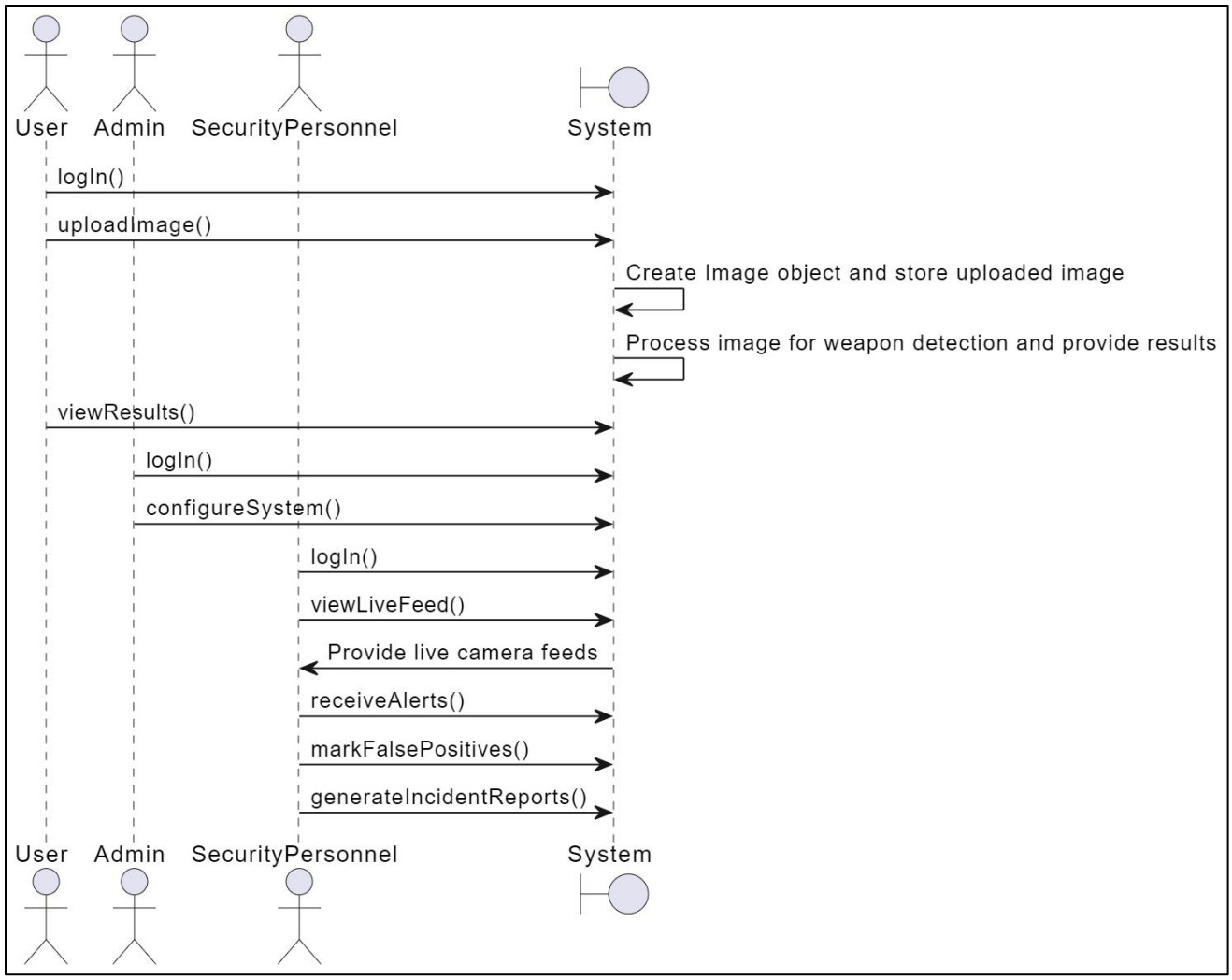

FIGURE 5.2 SEQUENCE DIAGRAM

5.3 State Diagram

Statechart diagram is used to capture the dynamic aspect of a system. State machine diagrams are used to represent the behavior of an application. An object goes through various states during its lifespan. The lifespan of an object remains until the program is terminated. The object goes from multiple states depending upon the event that occurs within the object. Each state represents some unique information about the object.
Statechart diagrams are used to design interactive systems that respond to either internal or external events. Statechart diagram visualizes the flow of execution from one state to another state of an object. It represents the state of an object from the creation of an object until the object is destroyed or terminated.

Following are the various notations that are used throughout the state chart diagram. All these notations, when combined, make up a single diagram.

1. Initial state: The initial state symbol is used to indicate the beginning of a state machine diagram.

2. Final state: This symbol is used to indicate the end of a state machine diagram.

3. Decision box: It contains a condition. Depending upon the result of an evaluated guard condition, a new path is taken for program execution.

4. Transition: A transition is a change in one state into another state which has occurred because of some event. A transition causes a change in the state of an object.

STATE DIAGRAM OF PROJECT:

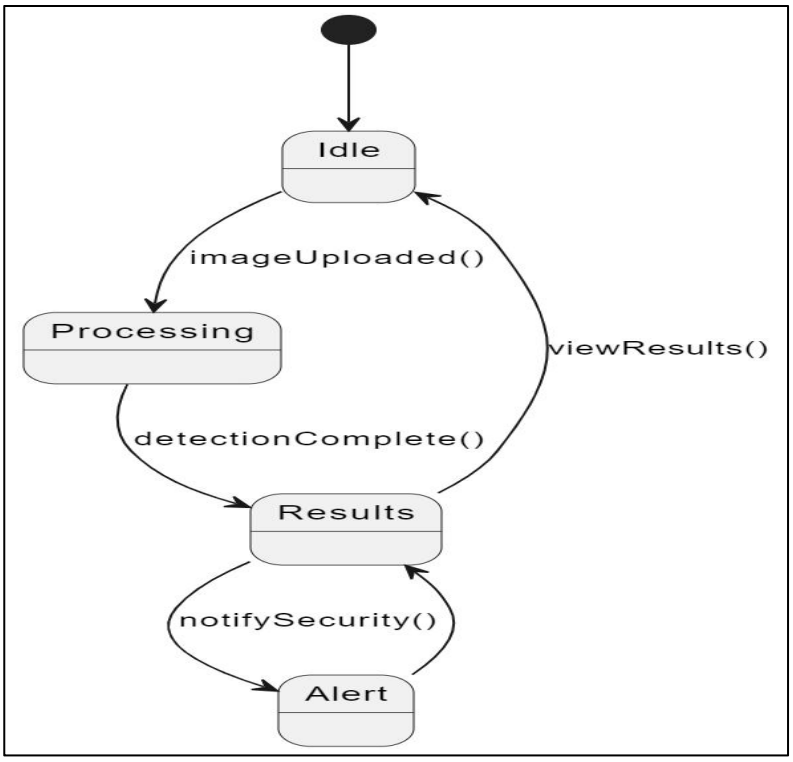

FIGURE 5.3 STATE DIAGRAM

5.4 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

ACTIVITY DIAGRAM OF PROJECT

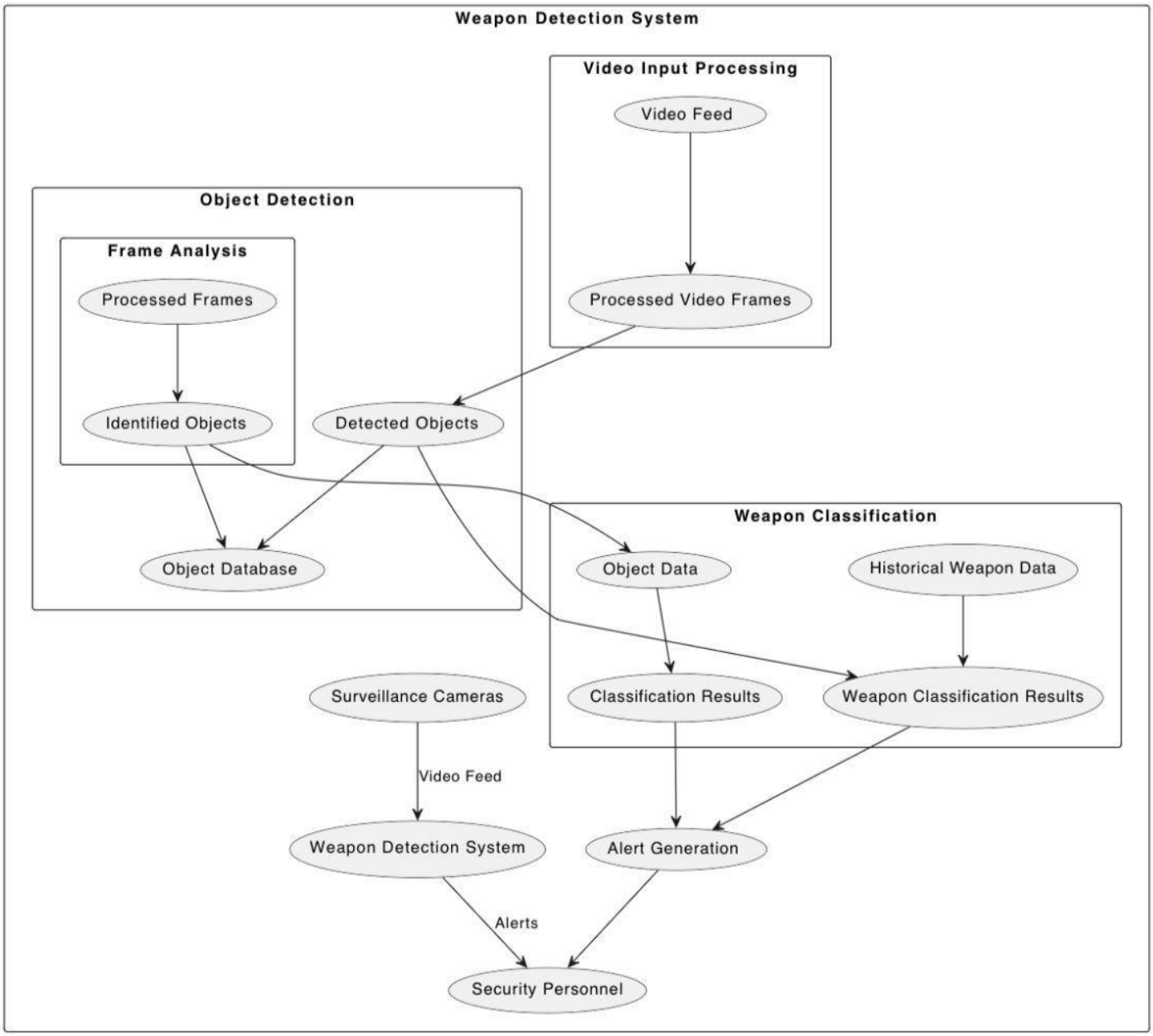

FIGURE 5.4 ACTIVITY DIAGRAM

5.5 Object Diagram

Objects are the real-world entities whose behavior is defined by the classes. Objects are used to represent the static view of an object-oriented system. We cannot define an object without its class. Object and class diagrams are somewhat similar. The difference between the class and object diagram is that the class diagram mainly represents the bird's eye view of a system which is also referred to as an abstract view. An object diagram describes the instance of a class. It visualizes the particular functionality of a system.

OBJECT DIAGRAM OF PROJECT

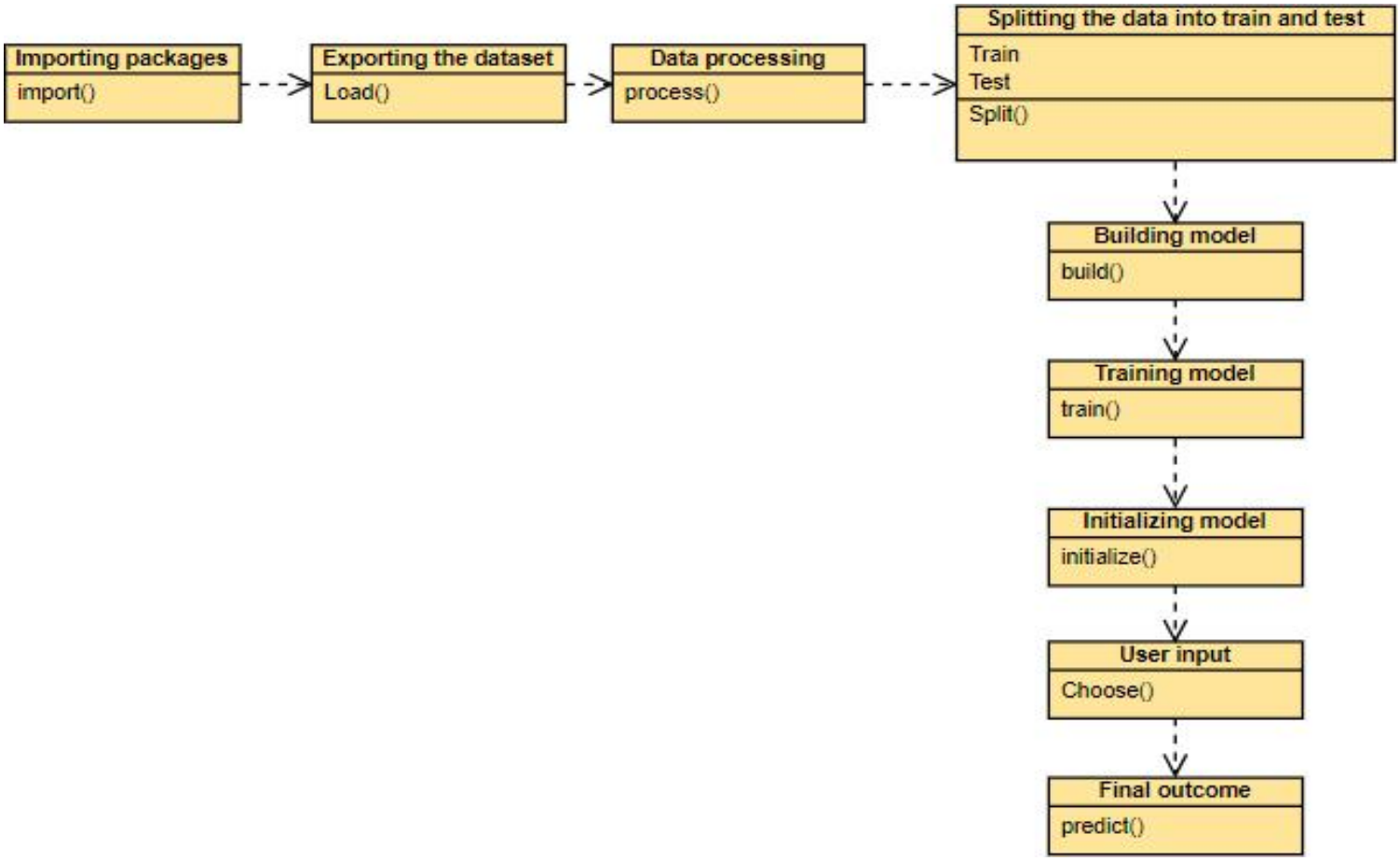

FIGURE 5.5 OBJECT DIAGRAM

5.6 Component Diagram

A component is a replaceable and executable piece of a system whose implementation details are hidden. A component provides the set of interfaces that a component realizes or implements. Components also require interfaces to carry out a function. It is a modular part of a system that encapsulates its contents. They are the logical elements of a system that plays an essential role during the execution of a system. A component is similar to a black box whose external behavior is defined by a provided interface and required interfaces.

A component is represented with classifier rectangle stereotypes as << component >>. Component details are hidden for the outside world. The name of a component is placed at the center of a rectangle. A component icon is displayed at the upper right corner of a rectangle, which is optional.

COMPONENT DIAGRAM OF PROJECT

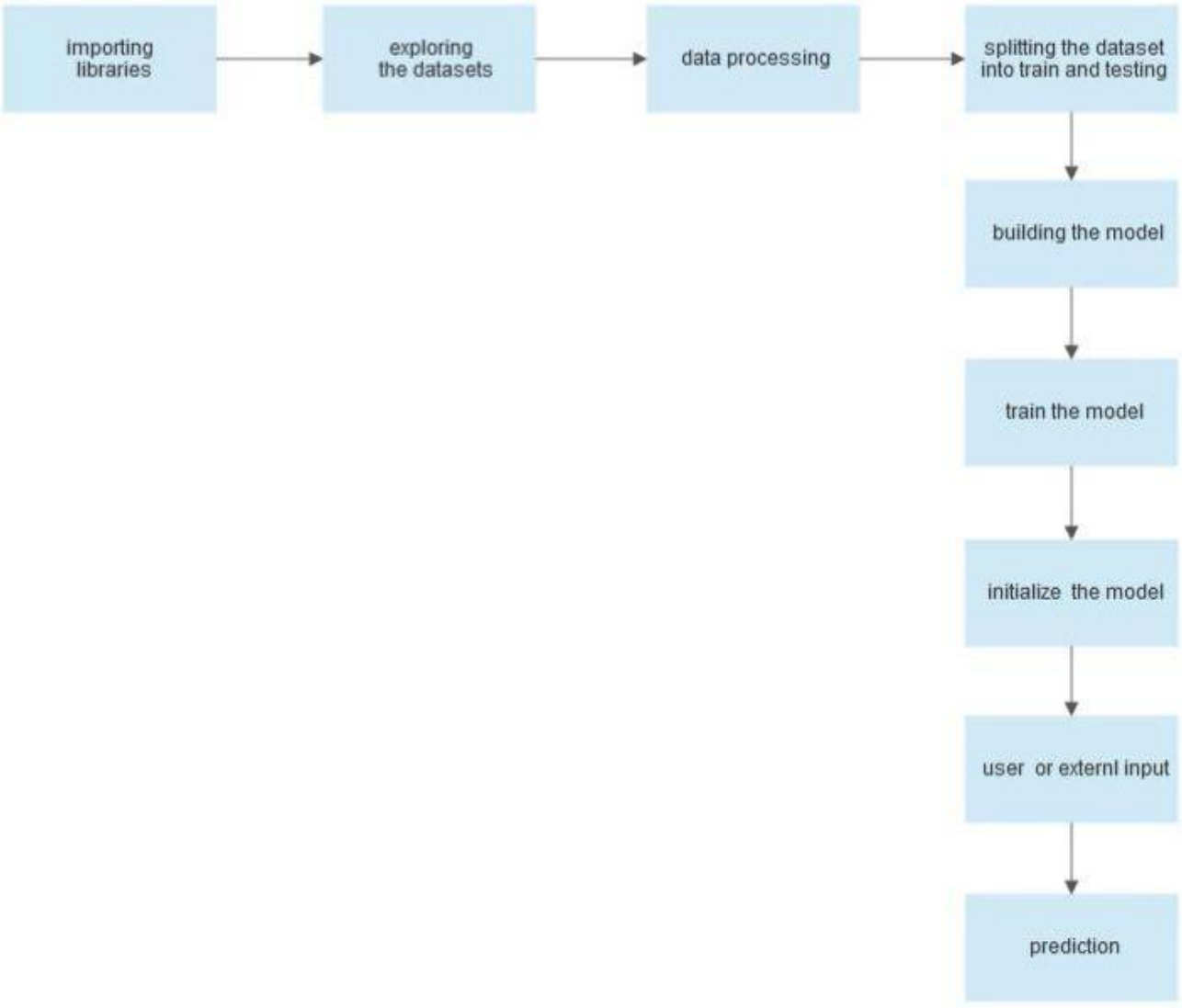

FIGURE 5.6 COMPONENT DIAGRAM

5.7 Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Strategies –

A strategy for software Testing integrates software test case design methods into a well-planned Series of steps that result in successful construction of software. Software testing strategies gives the road map for testing. A software testing strategy should be flexible enough to promote a customized testing approach at same time it must be right enough. Strategy is generally developed by project managers, software engineer and testing specialist.

There are four different software testing strategies.

i. **Unit Testing:** In this testing errors are detected from each software component individually. Various tests are conducted like testing of module interfaces, local data, and boundary conditions.

ii. **Integration Testing:** In this testing interacting components are verified and interface errors are detected. Two types of integration testing are Incremental and Non-incremental. Focus of this testing is to uncover errors in design of software architecture, integrated function or operation, resource integration.

iii. **Acceptance/Validation testing:** It is a testing conducted to ensure that software works correctly in user's working environment. It can be conducted over a period of weeks or months. Two types of acceptance testing are alpha testing and beta testing. It can be performed through a series of black box test.

iv. **System testing:** In system testing all the system elements forming the system is tested as a whole. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system.

Chapter - 6

Conclusion & Future Work

6.1 Limitation of Project

For both monitoring and control purposes, this work has presented a novel automatic weapon detection system in realtime. This work will indeed help in improving the security, law and order situation for the betterment and safety of humanity, especially for the countries who had suffered a lot with these kind of violent activities. This will bring a positive impact on the economy by attracting investors and tourists, as security and safety are their primary needs. We have focused on detecting the weapon in live CCTV streams and at the same time reduced the false negatives and positives. To achieve high precision and recall we constructed a new training database for the real-time scenario, then trained, and evaluated it on the latest state-of-the-art deep learning models using two approaches, i.e. sliding window/classification and region proposal/object detection. Different algorithms were investigated to get good precision and recall. Through a series of experiments, we concluded that object detection algorithms with ROI (Region of Interest) perform better than algorithms without ROI. We have tested many models but among all of them, the state-of-the-art Yolov4, trained on our new database, gave very few false positive and negative values, hence achieved the most successful results. It gave 91.73% mean average precision (mAP) and a F1-score of 91% with almost 99% confidence score on all types of images and videos. We can say that it satisfactorily qualifies as an automatic real-time weapon detector. Looking at the results, we got the highest mean average precision (MAP) F1- score as compared to the research done before for real-time scenarios. The future work includes reducing the false positives and negatives even more as there is still a need for improvement. We might also try to increase the number of classes or objects in the future but the priority is to further improve precision and recall.

6.2 Future Enhancement

Real-time weapons detection is an important application that can enhance public safety and security in a variety of settings, such as airports, train stations, and public venues. Here are some potential future enhancements for real-time weapons detection:

- Fusion of multiple sensing modalities: Integration of different sensing modalities such as audio, visual and infrared sensors can help to improve detection accuracy and reliability by combining the strengths of each modality.
- Use of deep learning algorithms: Deep learning algorithms can be used to identify specific patterns or features associated with different types of weapons, leading to improved detection accuracy.
- Real-time data analysis: Use of edge computing and cloud-based platforms to process and analyze data in real-time can provide faster response times and enhance the accuracy of weapon detection.
- Automated threat assessment: Advanced machine learning algorithms can be used to automatically assess the threat level associated with a detected weapon, leading to faster and more accurate response times.
- Continuous learning: Weapon detection systems can be improved by continuously learning from new data, enabling them to adapt to changing threats and improve their detection capabilities over time.
- Integration with law enforcement systems: Weapon detection systems can be integrated with law enforcement systems to provide real-time alerts and improve situational awareness, helping to prevent potential threats before they escalate.

Chapter - 7

Bibliography & References

7.1 Reference Books

[1] (2019). Christchurch Mosque Shootzings. Accessed: Jul. 10, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Christchurch_mosque_ shootings

[2] (2019). Global Study on Homicide. Accessed: Jul. 10, 2019. [Online]. Available: https://www.unodc.org/unodc/en/data-and-analysis/globalstudy-on-homicide.html

[3] W. Deisman, ‘‘CCTV: Literature review and bibliography,’’ in Research and Evaluation Branch, Community, Contract and Aboriginal Policing Services Directorate. Ottawa, ON, Canada: Royal Canadian Mounted, 2003.

[4] J. Ratcliffe, ‘‘Video surveillance of public places,’’ US Dept. Justice, Office Community Oriented Policing Services, Washington, DC, USA, Tech. Rep. 4, 2006.

[5] M. Grega, A. Matiola“ski, P. Guzik, and M. Leszczuk, ‘‘Automated detection of firearms and knives in a CCTV image,’’ Sensors, vol. 16, no. 1, p. 47, Jan. 2016.

[6] TechCrunch. (2019). China’s CCTV Surveillance Network Took Just 7 Minutes to Capture BBCReporter.Accessed:Jul.15,2019.[Online]Available: https://techcrunch.com/2017/12/13/china-cctv-bbc-reporter/

[7] N. Cohen, J. Gattuso, and K. MacLennan-Brown. CCTV Operational Requirements Manual 2009. St Albans, U.K.: Home Office Scientific Development Branch, 2009.

[8] G. Flitton, T. P. Breckon, and N. Megherbi, ‘‘A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery,’’ Pattern Recognit., vol. 46, no. 9, pp. 2420–2436, Sep. 2013.

[9] R. Gesick, C. Saritac, and C.-C. Hung, ‘‘Automatic image analysis process for the detection of concealed weapons,’’ in Proc. 5th Annu. Workshop Cyber Secur. Inf. Intell. Res. Cyber Secur. Inf. Intell. Challenges Strategies (CSIIRW), 2009, p. 20.

[10] R. K. Tiwari and G. K. Verma, ‘‘A computer vision based framework for visual gun detection using Harris interest point detector,’’ Procedia Comput. Sci., vol. 54, pp. 703–712,Aug. 2015.

7.1 Other Documentation and Resources

[1] (2019). Christchurch Mosque Shootzings. Accessed: Jul. 10, 2019. [Online]. Available:https://en.wikipedia.org/wiki/Christchurch_mosque_ shootings

[2] (2019). Global Study on Homicide. Accessed: Jul. 10, 2019. [Online]. Available:https://www.unodc.org/unodc/en/data-and-analysis/globalstudy-on-homicide.html

[3] W. Deisman, ‘‘CCTV: Literature review and bibliography,’’ in Research and Evaluation Branch, Community, Contract and Aboriginal Policing Services Directorate. Ottawa, ON, Canada: Royal Canadian Mounted, 2003.

[4] J. Ratcliffe, ‘‘Video surveillance of public places,’’ US Dept. Justice, Office Community Oriented Policing Services, Washington, DC, USA, Tech. Rep. 4, 2006.

7.2 Snapshots

Login Window:-

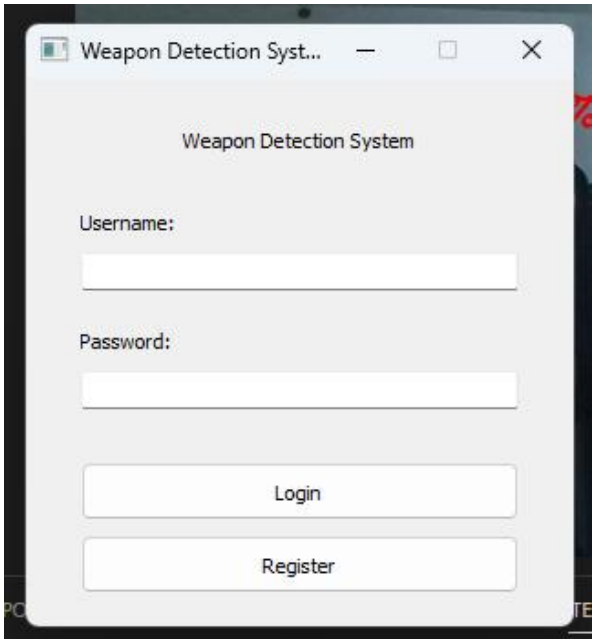

FIGURE 7.1 Login Window

Weapon detection :-

FIGURE 7.2 Weapon Detection

Weapon detection and Alert message :-

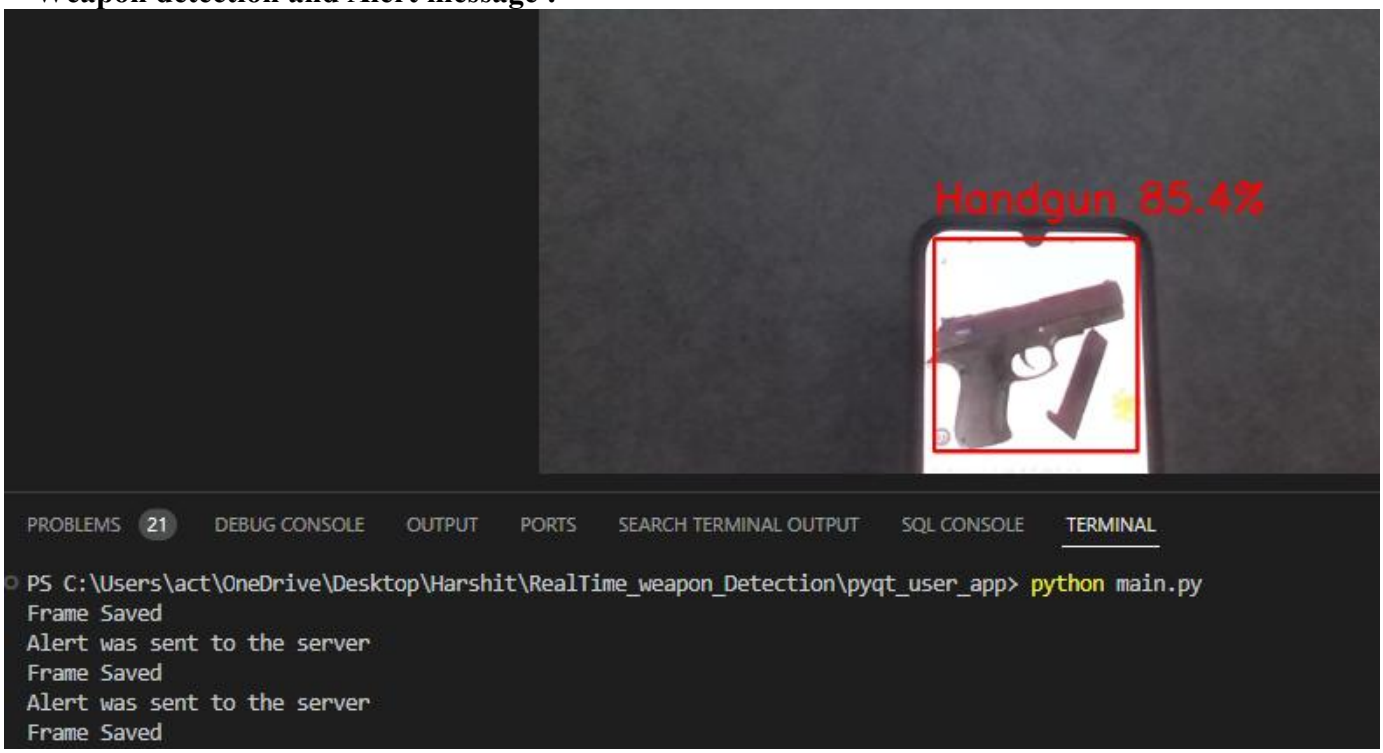

FIGURE 7.3 Alert Message

Server Side :-

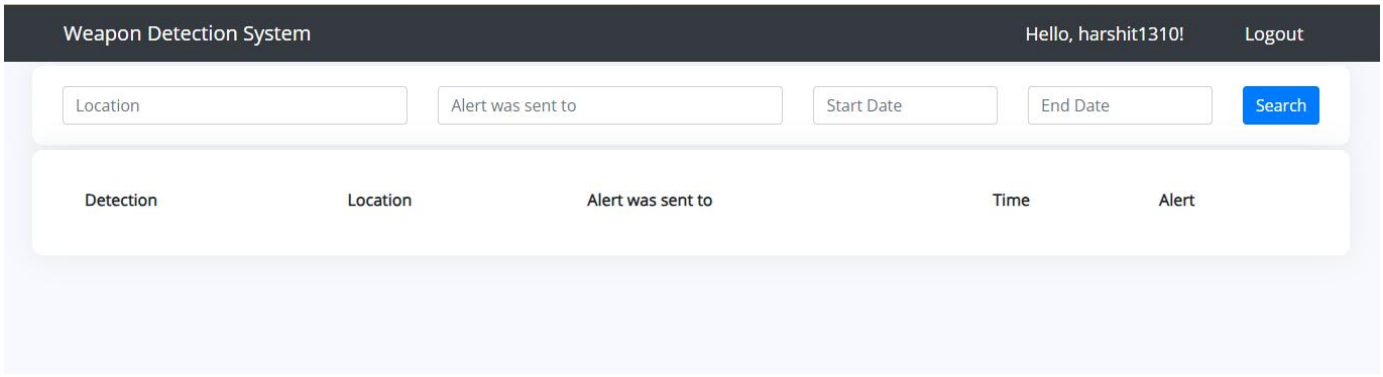

FIGURE 7.4 Server Side