Chapter 1 – INTRODUCTION

1.1 Introduction
1.2 Objective of the work
1.3 Brief Literature Review
1.4 Problem Definition
1.5 Plan Of The Work

1.1 Introduction

Security and surveillance are important issues in today's world. The recent acts of terrorism have highlighted the urgent need for efficient surveillance. Contemporary surveillance systems use digital video recording (DVR) cameras which play host to multiple channels. The major drawbacks with this model is that it requires continuous manual monitoring which is infeasible because of factors like human fatigue and cost of manual labour. Moreover, it is virtually impossible to search through recordings for important events in the past since that would require a playback of the entire duration of video footage. Hence, there is indeed a need for an automated system for video surveillance which can detect unusual activities on its own.

1.2 Objective of the System

- ✓ Design and Development: Describe the process of designing and developing an Intelligent Video Surveillance System (IVSS) using computer vision, machine learning, and real-time data processing techniques.
- ✓ System Integration: Explain the integration of various components, including object detection and tracking algorithms, deep learning models, and video analytics capabilities, to create a cohesive and efficient surveillance system.
- ✓ Dataset Collection and Preparation: Outline the methodology for collecting and preparing a diverse dataset for training the deep learning models used in the IVSS. Emphasize the importance of representative data in achieving accurate and robust surveillance capabilities.
- ✓ Model Training and Fine-tuning: Discuss the training process for the deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), including techniques for optimizing model performance and adaptability to specific surveillance contexts.

1.3 Brief Literature Review

- ✓ Object Detection and Tracking: Discuss the advancements in object detection and tracking algorithms, including popular techniques such as deep learning-based approaches (e.g., Faster R-CNN, YOLO), feature-based methods (e.g., SURF, SIFT), and multi-object tracking algorithms. Highlight the strengths and limitations of each approach and their applicability to intelligent video surveillance systems.
- ✓ Activity Recognition: Review research papers focused on activity recognition in video surveillance. Discuss different approaches, including handcrafted feature-based methods (e.g., HOG, HOF) and deep learning-based techniques (e.g., 3D Convolutional Neural Networks, Recurrent Neural Networks), highlighting their effectiveness in recognizing specific activities and events.
- ✓ Anomaly Detection: Explore studies that address anomaly detection in video surveillance systems. Discuss statistical models, pattern recognition techniques, and machine learning algorithms used to identify unusual or suspicious behaviors. Highlight approaches such as abnormality modeling, behavior profiling, and outlier detection.
- ✓ System Architecture: Discuss research papers that propose system architectures for intelligent video surveillance. Include studies on distributed systems, edge computing, and cloud-based architectures. Examine the integration of hardware components, real-time data processing.

1.4 Problem Definition

The problem definition section of an Intelligent Video Surveillance System project report should clearly articulate the specific challenges or issues that the project aims to address. It sets the foundation for the project by identifying the problem statement and the motivation behind developing an intelligent video surveillance system. The problem definition can be outlined as follows:

Introduction: Provide a brief overview of the importance of video surveillance systems in ensuring safety, security, and monitoring in various environments such as public spaces, transportation hubs, retail establishments, and critical infrastructure. Highlight the limitations and challenges of traditional video surveillance systems.

Problem Statement: Clearly state the problem that the project aims to address. For example:

- ✓ Traditional video surveillance systems rely heavily on manual monitoring, which is time-consuming, prone to human error, and often ineffective in real-time threat detection.
- ✓ Conventional surveillance systems struggle with the sheer volume of video data generated, making it challenging to analyze and extract meaningful information efficiently.
- ✓ The lack of intelligent analysis capabilities hinders the identification of suspicious activities, abnormal behaviors, and potential security threats.

1.5 Plan Of The Work

- Research and Literarture Review.
- System Design and Architecture.
- Dataset collection and Annotation.
- Algorithm Development.
- System Integration and Development.
- Training and Evaluation.
- Documentation and Report Writing.
- Project Management and Collaboration.

.

Chapter 2 – Tools and Technology

2.1 Tools and Technology

2.1.1 HTML, CSS, BOOTSTRAP

2.1.2 PYTHON, DEEPLEARNING

2.2 Literature Review

2.1.1 HTML, CSS, Bootstrap

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating WebpagesHTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is aheading","this is a paragraph", "this is a link", etc.

What is CSS ?

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

What is Bootstrap ?

Bootstrap is a popular open-source front-end framework that helps in designing responsive and mobile-first websites and web applications. It provides a collection of CSS and JavaScript components, as well as pre-built templates and styles, that enable developers to quickly and easily create visually appealing and functional user interfaces.

Benefits of Using Bootstrap:

1. Time Efficiency: Bootstrap's pre-built components and styles save development time by providing ready-to-use UI elements that can be easily customized and integrated into projects.
2. Consistency: Bootstrap promotes consistency in design and layout, ensuring that websites and applications have a cohesive look and feel across different pages and devices.
3. Responsiveness: With its responsive design features, Bootstrap simplifies the task of creating mobile-friendly websites and applications, eliminating the need for writing separate code for different devices.
4. Cross-browser Compatibility: Bootstrap is designed to work well across different web browsers, reducing compatibility issues and ensuring a consistent experience for users.

What Does Python Mean?

Python is a high-level, interpreted programming language that emphasizes code readability and simplicity. It was created by Guido van Rossum and initially released in 1991. Python is known for its clean syntax, which allows developers to express concepts in fewer lines of code compared to other programming languages. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming styles.

Key Features of Python:

- ✓ Readability: Python's syntax is designed to be easy to read and understand, making it particularly suitable for beginners. It utilizes indentation (whitespace) instead of brackets to define code blocks, enhancing code readability and maintaining consistency.
- ✓ Expressive and Concise: Python emphasizes code simplicity and expressiveness, allowing developers to write clear and concise code. It provides built-in data structures and high-level abstractions, reducing the need for complex and boilerplate code.
- ✓ Large Standard Library: Python has a vast standard library that provides a wide range of modules and functions for various purposes. This library includes modules for file I/O, network programming, data manipulation, web development, and more. The extensive standard library eliminates the need to build everything from scratch and accelerates development.
- ✓ Cross-platform Compatibility: Python is available on multiple platforms, including Windows, macOS, Linux, and various other operating systems. This cross-platform compatibility allows developers to write code once and run it on different platforms without modification.

.

What is DeepLearning ?

Deep learning is a subfield of machine learning that focuses on training artificial neural networks to learn and make predictions or decisions in a manner similar to the human brain. It is inspired by the structure and function of biological neural networks.

In deep learning, neural networks with multiple layers are utilized to process and learn from complex data representations. These networks, known as deep neural networks, consist of an input layer, one or more hidden layers, and an output layer. Each layer consists of interconnected nodes, called artificial neurons or units, which perform mathematical operations on the input data..

Applications of Deep Learning:

Deep learning has gained significant popularity and has been successfully applied across various domains, including:

- Computer Vision: Deep learning models have achieved state-of-the-art performance in tasks such as image classification, object detection, image segmentation, and facial recognition.
- Natural Language Processing (NLP): Deep learning has revolutionized NLP tasks, including language translation, sentiment analysis, question answering, text generation, and speech recognition.
- Autonomous Systems: Deep learning is crucial in the development of autonomous vehicles, where models are trained to perceive the environment, make decisions, and navigate safely.
- Healthcare: Deep learning is used in medical imaging analysis, disease diagnosis, drug discovery, genomics research, and personalized medicine.

2.2 Literature review

The literature review section of the project report provides an overview of existing research, methodologies, and advancements related to intelligent video surveillance systems. It aims to identify relevant studies and establish the current state of the field. The literature review can be structured as follows:

- ✓ Evolution of Video Surveillance: Start by discussing the historical development and evolution of video surveillance systems. Explore the transition from traditional analog surveillance to digital systems and the emergence of intelligent video surveillance with advanced computer vision and machine learning techniques.
- ✓ Object Detection and Tracking: Review research papers focused on object detection and tracking in video surveillance. Discuss various approaches, such as background subtraction, motion detection, and object recognition using techniques like Haar cascades, Viola-Jones algorithm, and deep learning-based models. Highlight the strengths, limitations, and performance of these methods.
- ✓ Activity Recognition and Behavior Analysis: Examine studies that address activity recognition and behavior analysis in video surveillance. Discuss approaches like human action recognition, crowd behavior analysis, and abnormal behavior detection. Explore methodologies such as spatio-temporal features, deep neural networks, and recurrent neural networks. Evaluate the effectiveness of these approaches in recognizing complex activities and abnormal behaviors.
- ✓ Real-time Analytics and Event Detection: Explore research papers on real-time video analytics and event detection. Discuss techniques for identifying specific events and anomalies, such as crowd density estimation, abandoned object detection, and violence detection. Highlight the algorithms and models used to enable quick and accurate event detection in surveillance videos.

Chapter 3: System Requirement Study

3.1 User Characteristics
3.2 Hardware and Software Requirements
3.3 Constraints
- 3.3.1 Regulatory Policies
- 3.3.2 Hardware Limitations
- 3.3.3 Interfaces to Other Applications
- 3.3.4 Parallel Operations
- 3.3.5 Higher Order Language Requirements
- 3.3.6 Reliability Requirements
- 3.3.7 Criticality of the Application
- 3.3.8 Safety and Security Considerations

3.4 Assumptions and Dependencies

3.1 User Characteristics

Intelligent video surveillance system may include:

Security Operators and Personnel: Describe the primary users of the intelligent video surveillance system, such as security operators and personnel responsible for monitoring and managing surveillance operations

Security Managers and Administrators: Discuss the user characteristics of security managers and administrators who oversee and manage the intelligent video surveillance system.

Law Enforcement and Emergency Responders: Consider the user characteristics of law enforcement personnel and emergency responders who may interact with or rely on the intelligent video surveillance system.

System Administrators and IT Staff: Discuss the user characteristics of system administrators and IT staff responsible for managing and maintaining the intelligent video surveillance system infrastructure.

3.2 Hardware and Software Requirements

3.2.1 Hardware Requirements

Processor	:	Intel 5 and above
RAM	:	8 GB
Hard disk	:	4.3 GB
Monitor	:	15” Color Monitor
Floppy Disk	:	1.44 MB
Graphics	:	Nvidia Graphic card 2GB

3.2.2 Software Requirements

Front End	:	HTML, CSS, JS,Bootstrap
Back End	:	Python, DeepLearning
Operation System	:	Windows 8 and above

3.3 Constraints

3.3.1 Regulatory Policies

- ✓ Data Protection and Privacy Policies: Explain the measures taken to safeguard the privacy and personal data of individuals captured by the surveillance system. Adhere to relevant data protection laws, such as the General Data Protection Regulation (GDPR) in the European Union or other regional data protection regulations. Describe how data is collected, stored, processed, and anonymized (if necessary) to protect individual privacy rights.
- ✓ Consent and Notification: Address the policies regarding consent and notification to individuals who may be subject to surveillance. Ensure that clear and transparent information is provided to the public about the presence and purpose of the intelligent video surveillance system. Comply with legal requirements for obtaining consent when necessary, especially in areas with public access.
- ✓ Retention and Data Handling: Explain the policies for data retention and handling. Define the duration for which surveillance data is stored and when it should be deleted or securely discarded. Address data access controls and permissions to ensure that only authorized personnel can access and use surveillance data.
- ✓ Use of Facial Recognition and Biometric Data: If the system employs facial recognition or biometric data, address the specific policies and legal requirements for its use. Some jurisdictions have strict regulations concerning the use of biometric data, and it is essential to comply with these regulations.
- ✓ Video Surveillance in Restricted Areas: Consider any specific regulations or restrictions regarding the deployment of video surveillance in certain areas, such as private property, sensitive locations, or areas with high expectations of privacy.
- ✓ Compliance with Industry Standards: Adhere to relevant industry standards and guidelines for video surveillance systems, such as those provided by the International Organization for Standardization (ISO) or other regulatory bodies. These standards can cover aspects like data security, system integrity, and ethical use of surveillance data.
- ✓ Third-Party Data Sharing and Integration: If the system involves third-party data sharing or integration with external systems, ensure compliance with data protection agreements and any necessary legal arrangements.

3.3.2 Hardware limitations

- ✓ **Processing Power:** The intelligent video surveillance system may require significant processing power, especially when using computationally intensive algorithms such as deep learning for object detection, tracking, and activity recognition. Hardware limitations in terms of the number of processing cores, clock speed, or available memory may affect the system's ability to process multiple video feeds in real-time.
- ✓ **Graphics Processing Units (GPUs):** Deep learning-based algorithms often rely on GPUs for accelerated parallel processing. However, the availability of GPUs or the number of GPU units in the hardware infrastructure may be limited, which could impact the system's performance and real-time capabilities.
- ✓ **Storage Capacity:** The storage capacity of the hardware infrastructure determines the amount of surveillance data that can be stored and retained. High-definition video feeds can consume significant storage space, and limitations in storage capacity may require careful management of data retention policies.
- ✓ **Network Bandwidth:** Real-time video feeds generate large amounts of data that need to be transmitted and processed over the network. Limited network bandwidth may lead to delays in video transmission, affecting the timeliness of the system's response to security incidents.
- ✓ **Cameras and Sensors:** The choice of cameras and sensors used in the video surveillance system can impact the quality and accuracy of data captured. Limitations in the resolution, frame rate, or field of view of the cameras may affect the system's ability to detect and track objects accurately.
- ✓ **Integration with Existing Infrastructure:** The hardware limitations in the existing infrastructure, such as compatibility issues, legacy systems, or proprietary hardware, may pose challenges in integrating the intelligent video surveillance system seamlessly.
- ✓ **Power Consumption:** Real-time video processing and data transmission can consume substantial power. For systems deployed in remote or power-constrained locations, managing power consumption becomes a critical concern.

3.3.3 Interfaces to Other Applications

- ✓ **Security Management Systems:** The intelligent video surveillance system may integrate with existing security management systems, such as access control systems or alarm monitoring platforms. This interface allows the surveillance system to trigger alarms or alerts in the security management system when specific events or anomalies are detected.
- ✓ **Incident Response Systems:** Integration with incident response systems enables the intelligent video surveillance system to send alerts or notifications to incident response teams or emergency responders. This interface facilitates quick and coordinated responses to security incidents.
- ✓ **Video Management Software (VMS):** The intelligent video surveillance system may integrate with Video Management Software (VMS) platforms, which provide comprehensive video recording, storage, and management capabilities. This integration allows seamless storage and retrieval of surveillance video feeds from the VMS.
- ✓ **Analytics and Business Intelligence Tools:** Integration with analytics and business intelligence tools allows for deeper analysis of surveillance data. Data from the intelligent video surveillance system can be combined with other data sources, such as customer behaviour data or inventory information, to gain valuable insights for business operations.
- ✓ **Accessible User Interface:** The intelligent video surveillance system may offer an accessible user interface that allows authorized users to access video feeds and surveillance data remotely. This interface ensures that security operators, administrators, or stakeholders can monitor the system and respond to incidents from any location.

3.3.4 Parallel Operations

- ✓ **Multi-Core Processing:** Explain how the intelligent video surveillance system utilizes multi-core processing capabilities of modern CPUs or processors. By distributing computational tasks across multiple cores, the system can process video feeds and perform object detection, tracking, and activity recognition concurrently, improving overall system performance.
- ✓ **GPU Acceleration:** Discuss the use of Graphics Processing Units (GPUs) for parallel processing in the intelligent video surveillance system. GPUs excel at handling large-scale data parallelism, making them well-suited for tasks like deep learning-based object detection and tracking. Detail how the system offloads computationally intensive tasks to GPUs for faster and more efficient processing.
- ✓ **Edge Computing:** Address the implementation of edge computing in the intelligent video surveillance system. Explain how certain processing tasks are performed closer to the surveillance cameras, reducing latency and network bandwidth requirements. Edge computing enables real-time analysis of video feeds at the edge of the network, minimizing data transmission to central processing units.
- ✓ **Distributed Computing**: Describe any distributed computing techniques used in the system. Distributed computing involves dividing tasks into smaller sub-tasks and distributing them across multiple nodes or servers in the network. This approach allows the system to process video feeds and analyze surveillance data in a distributed and parallel manner, improving scalability and fault tolerance.

3.3.5 Higher Order Language Requirements

- ✓ **Python:** Python is a popular and versatile programming language commonly used in artificial intelligence and computer vision applications. Its ease of use, rich libraries (e.g., OpenCV, TensorFlow, PyTorch), and strong community support make it an excellent choice for developing the core functionalities of the intelligent video surveillance system, including object detection, tracking, and activity recognition.
- ✓ **C++:** C++ is known for its performance and efficiency, making it suitable for implementing computationally intensive algorithms and real-time video processing. C++ is often used in conjunction with Python for critical components that require high-speed execution, such as low-level image processing or GPU-accelerated operations.
- ✓ **JavaScript:** JavaScript is essential for developing web-based user interfaces for the intelligent video surveillance system. Using JavaScript frameworks like React or Vue.js, developers can create interactive and responsive web applications that allow security operators to monitor video feeds, receive alerts, and interact with the system from various devices.
- ✓ **HTML and CSS:** HTML and CSS are essential for designing the layout and appearance of the user interface. They define the structure and styling of the web pages, ensuring a visually appealing and user-friendly interface for accessing and managing surveillance data**.**

3.3.6 Reliability Requirements

- ✓ **System Uptime and Availability:** Specify the desired system uptime and availability for the intelligent video surveillance system. This refers to the percentage of time the system is expected to be operational and accessible to users. High availability ensures continuous monitoring and timely response to security incidents.
- ✓ **Error Handling and Fault Tolerance:** Detail the strategies and mechanisms in place to handle errors and faults within the system. The intelligent video surveillance system should be resilient to failures, gracefully recover from errors, and continue functioning with minimal disruption.
- ✓ **Redundancy and Backup:** Describe the use of redundancy and backup measures to safeguard against hardware or software failures. Redundant components, such as backup servers or redundant power supplies, can ensure uninterrupted surveillance operations in case of primary system failures.
- ✓ **Video Feed Recovery:** Explain how the system handles video feed recovery in case of interruptions, such as camera disconnections or network outages. The system should be able to automatically recover video feeds and continue processing without data loss.
- ✓ **Object Tracking Accuracy:** Define the expected accuracy of object tracking in the surveillance system. The reliability of object tracking directly impacts the effectiveness of security operations, and high accuracy ensures that objects are correctly identified and tracked over time.
- ✓ **False Alarm Reduction:** Specify the requirements for false alarm reduction to minimize unnecessary alerts and notifications. The intelligent video surveillance system should employ advanced algorithms and filters to distinguish genuine security threats from false positives.
- ✓ **Continuous Monitoring and Diagnostics:** Detail the continuous monitoring and diagnostics mechanisms implemented in the system. Regular system health checks, performance monitoring, and log analysis can help detect anomalies and proactively address potential issues.

3.3.7 Criticality of the Application

- ✓ Security Enhancement:
- ✓ Real-time Response:
- ✓ Proactive Threat Detection:
- ✓ Event Documentation and Forensics:
- ✓ Integration with Security Ecosystem:
- ✓ Public Safety and Asset Protection:
- ✓ Privacy and Ethical Considerations:
- ✓ Reducing Human Error:
- ✓ Scalability and Adaptability:
- ✓ Emergency Response Support:

3.3.8 Safety and Security Considerations

- ✓ **Data Privacy and Protection:** Emphasize the measures taken to safeguard the privacy and protection of data collected by the intelligent video surveillance system. Comply with relevant data protection laws and regulations, such as GDPR, and implement encryption and access controls to prevent unauthorized access to sensitive data.
- ✓ **Ethical Use of Surveillance Data:** Discuss the ethical considerations related to the collection and use of surveillance data. Ensure that data is used only for its intended purpose, and measures are in place to prevent misuse or unauthorized sharing of surveillance footage.
- ✓ **Secure Communication:** Detail the security protocols and encryption used to ensure secure communication between the intelligent video surveillance system components and external applications. Secure communication prevents interception and tampering of data during transmission.
- ✓ **Authentication and Access Control:** Explain the mechanisms used for user authentication and access control. Strong authentication methods, such as multi-factor authentication, help prevent unauthorized access to the surveillance system and its data.
- ✓ **Vulnerability Assessments and Penetration Testing:** Address how vulnerability assessments and penetration testing are conducted to identify and address potential security weaknesses in the system. Regular assessments help mitigate security risks and ensure the system's resilience against cyber threats.
- ✓ **Monitoring and Logging:** Describe the system's monitoring and logging capabilities to track and record system activities. Monitoring helps detect potential security breaches, while logging provides an audit trail for accountability and forensic analysis.
- ✓ **Physical Security:** Highlight the physical security measures in place to protect the hardware components of the intelligent video surveillance system. Secure data centers, access controls, and surveillance of critical infrastructure prevent unauthorized physical access.

3.4 Assumptions and Dependencies

Assumptions for Intelligent Video Surveillance system could include:

1. Availability of Sufficient and Representative Dataset.
2. Adequate Hardware Resources.
3. Relevance of Existing Research Papers.
4. Collaboration and Data Sharing.

Dependencies for Intelligent Video Surveillance system could include:

1. Availability of Software Libraries and Tools.
2. External APIs and Services.
3. System Integration and IT Infrastructure.
4. Regulatory and Privacy Compliance.
5. Stakeholder Support and Collaboration.

Chapter 4 –System Design

4.1 Class Diagram
4.2 ER Diagram
4.3 Use-Case Diagram
4.4 DFD Diagram

4.1 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable of the software application.

Class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

CLASS DIAGRAM SYMBOLS

Class Diagram Relationship Type	Notation
Association	
Inheritance	
Realization/ Implementation	
Dependency	
Aggregation	
Composition	

Table 4.1 Class Diagram Symbol

CLASS DIAGRAM

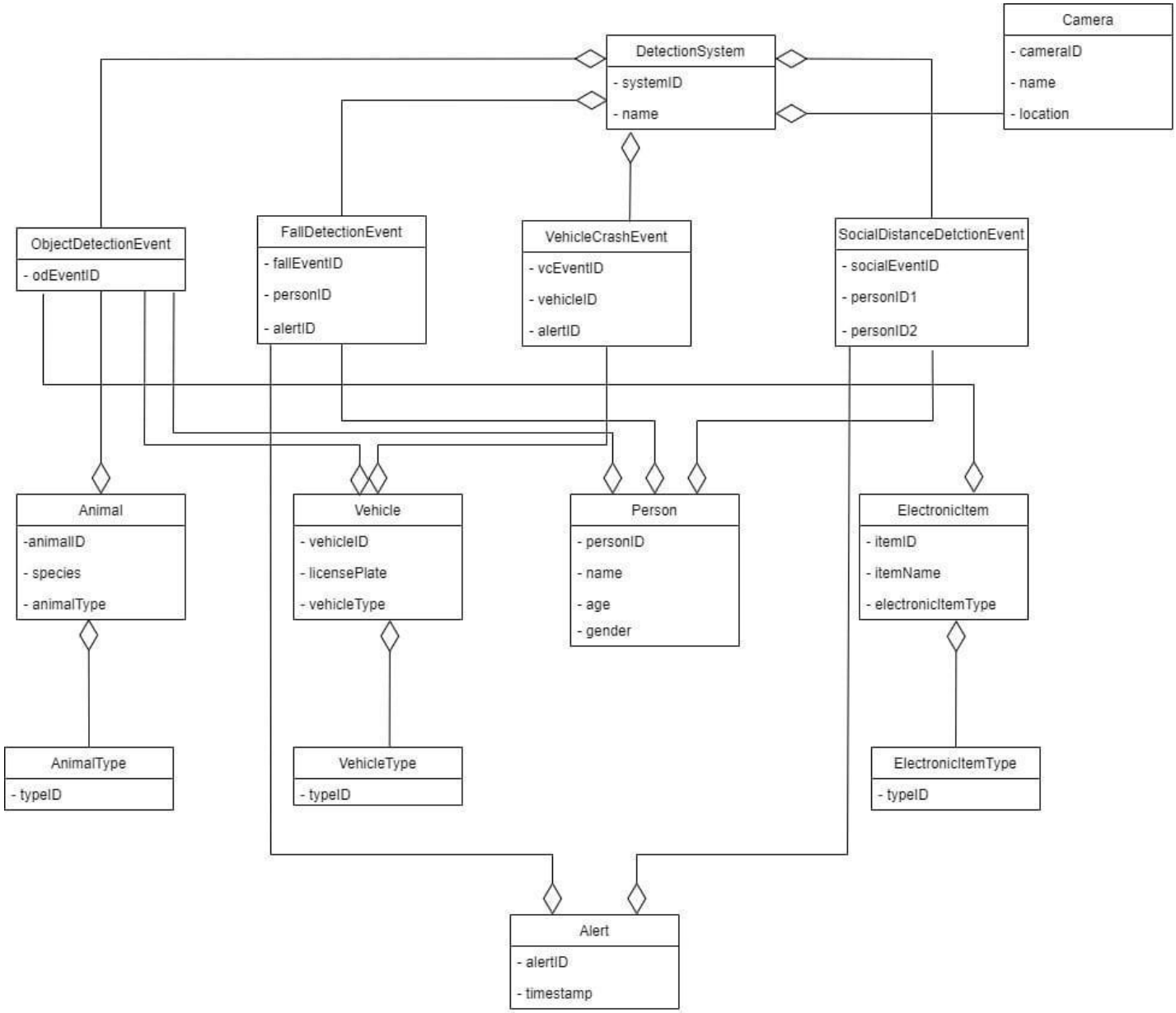


FIG 4.1 Class Diagram

4.2 ER DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system.

ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

They mirror grammatical structure, with entities as nouns and relationships as verbs. Entity- relationship diagrams (ERDs) are a logical representation of data that describes the relationships among entities and attributes.

The central reasons for choosing an entity relation (ER) model over physical design are as follows (Al-Shemaiah, 2015):

- ✓ Conceptual simplicity: If relationships between entities and attributes are known, an ERD can be quickly drawn.
- ✓ Explicit visual representation. The database structure can be easily comprehended after consulting the diagram.
- ✓ Communication effectiveness. Standard symbols representing different information facilitate understanding of the working of the database after completion.
- ✓ Immense flexibility. ER data model can be easily converted into any other data model with minor manipulations. The characteristics mentioned above are also the primary benefits of ERDs. However, it is necessary to state the disadvantages of the concept to acquire a holistic picture of the matter:
- ✓ Limited relationship representation. The model is limited in relationships as compared to other data models.

ER SYMBOLS:

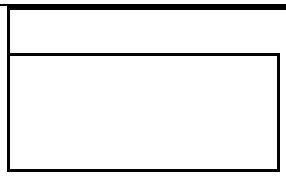
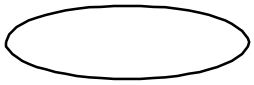
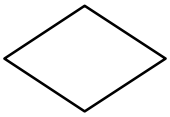
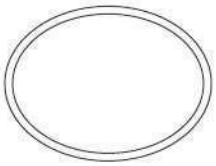
	ENTITY
	ATTRIBUTE
	RELATIONSHIP
	Key ATTRIBUTE/SINGLE VALUED ATTRIBUTE
	Multivalued Attribute

Table 4.2 ER diagram symbols

ER Diagram

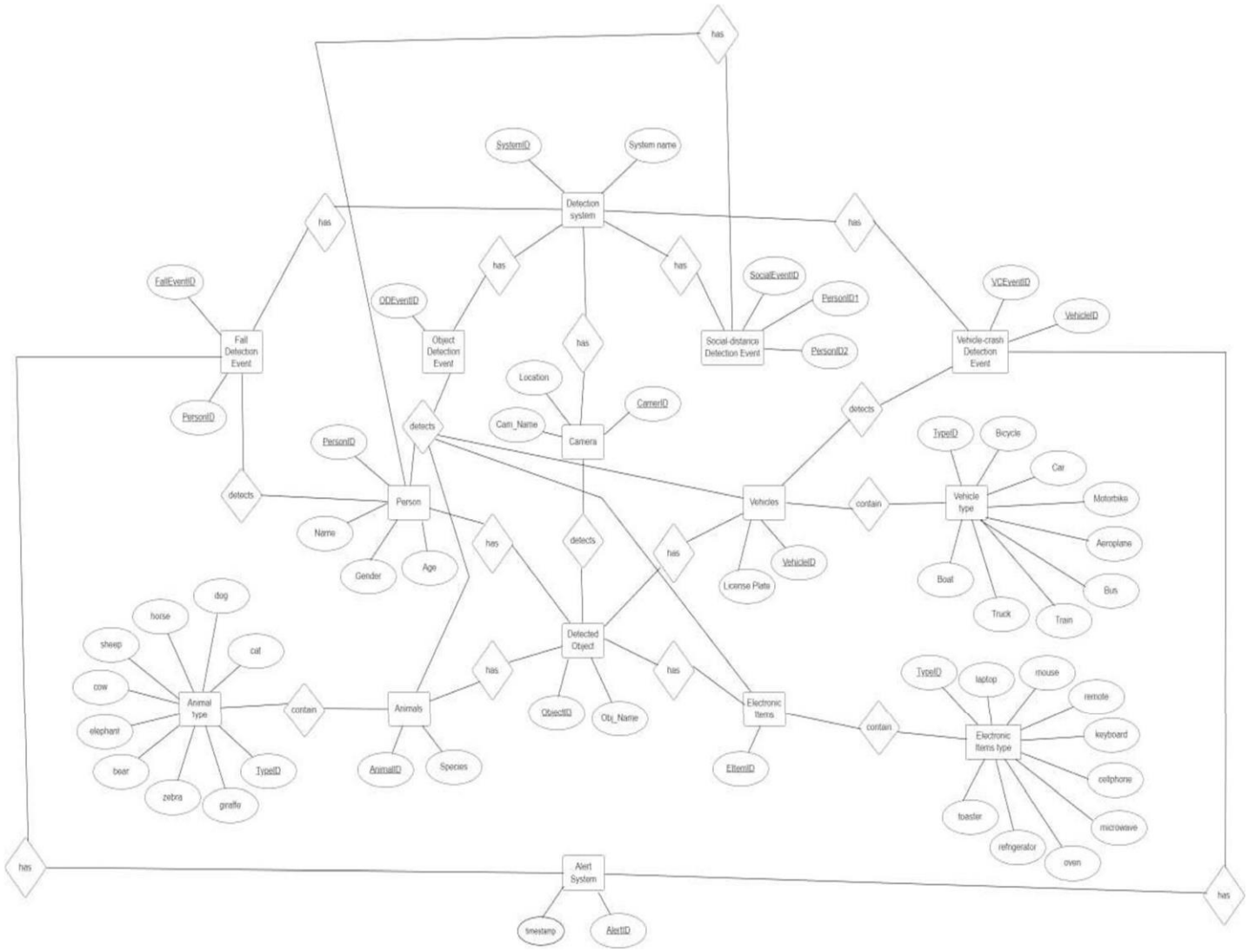


FIG 4.2 ER diagram symbol

4.3 USE CASE DIAGRAM

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them.

Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, number of use case diagrams are used.

The purposes of use case diagrams can be said to be as follows –

- ✓ Used to gather the requirements of a system.
- ✓ Used to get an outside view of a system.
- ✓ Identify the external and internal factors influencing the system.
- ✓ Show the interaction among the requirements are actors.

USE CASE DIAGRAM SYMBOLS

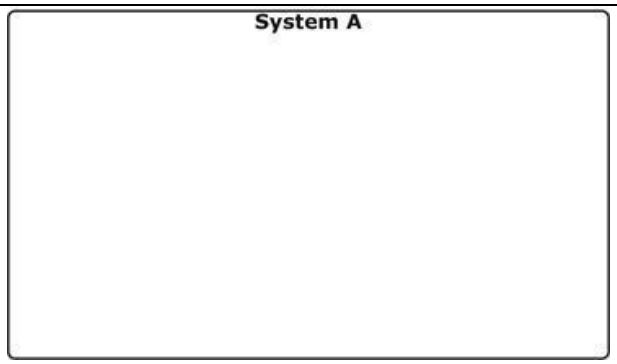
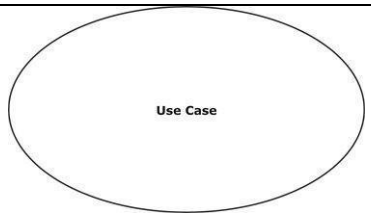
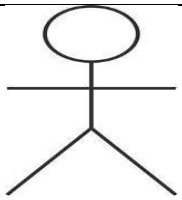
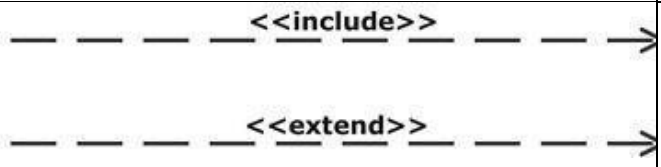
	**SYSTEM**
	**USE CASE**
	**ACTOR**
	**RELATIONSHIPS**

Table 4.3 Use Case Diagram Symbol

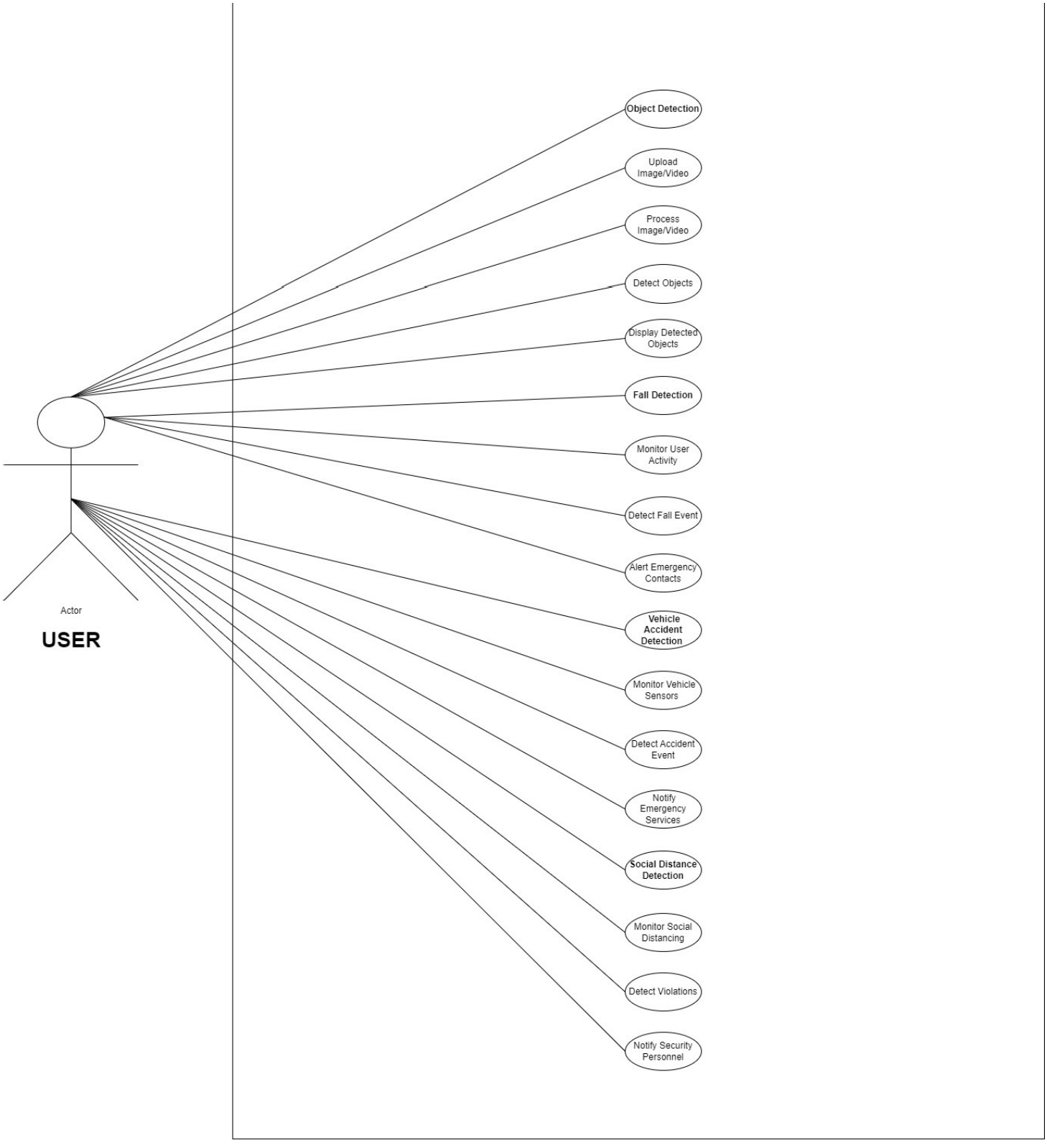


FIG 4.3 Use Case Diagram

4.4 UML DFD DIAGRAM

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself.

DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart.Data Flow Diagram can be represented in several ways.

The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

Levels of DFD uses hierarchy to maintain transparency thus multilevel. DFD's can be created. Levels of DFD are as follows:

- 0-level DFD
- 1-level DFD:
- 2-level DFD:

Advantages of DFD:

- ✓ It helps us to understand the functioning and the limits of a system.
- ✓ It is a graphical representation which is very easy to understand as it helps visualize contents.
- ✓ Data Flow Diagram represent detailed and well explained diagram system.
- ✓ It is used as the part of system documentation file.
- ✓ Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.

DFD SYMBOLS

	**Entity**
	**Process**
	**Data Store**
	Data Flow

Table 4.4 DFD Diagram Symbol

DFD: LEVEL 0

DFD: LEVEL 1

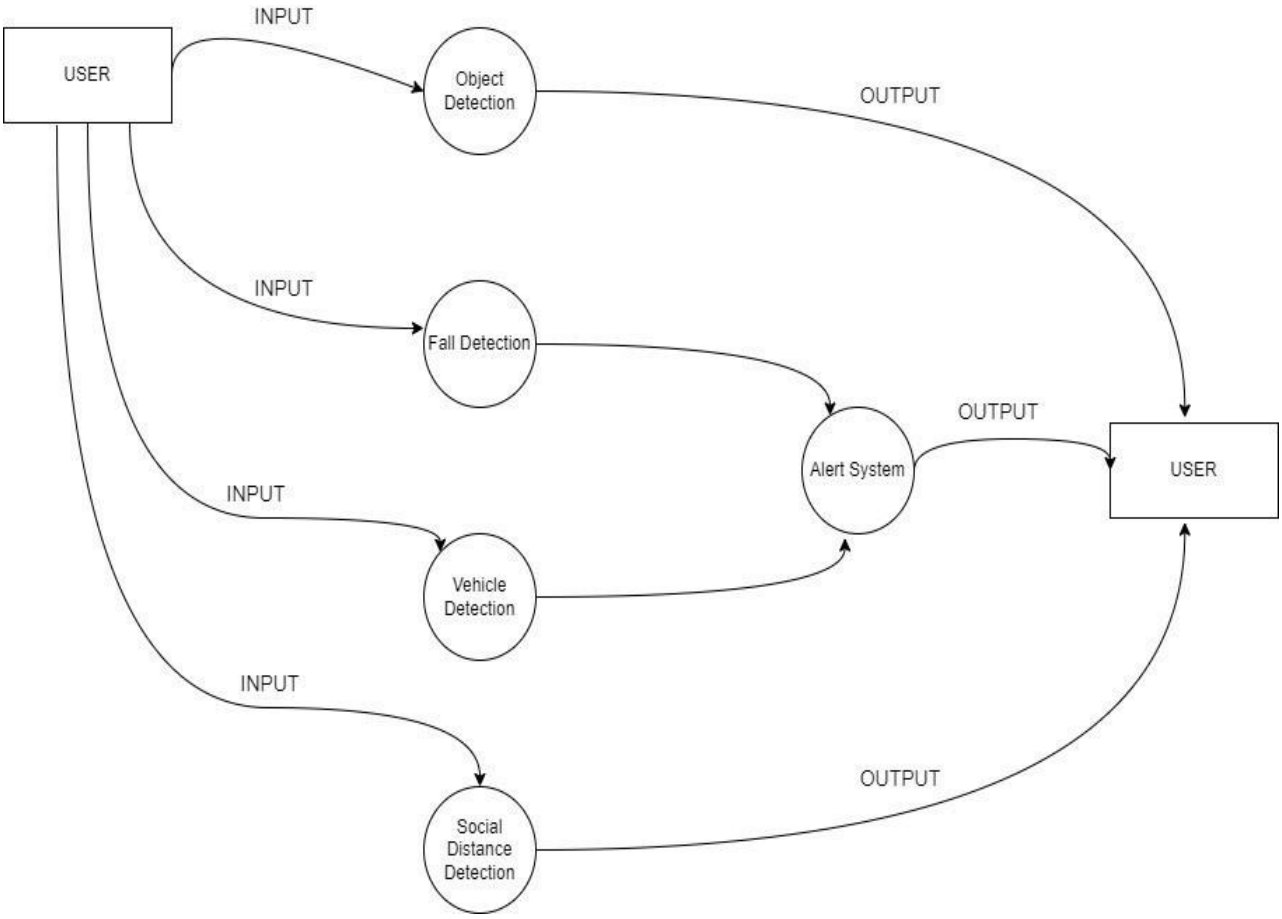


FIG 4.4 DFD Diagram Symbol

Chapter 5 – Result , Discussion and Conclusion

Result:

The results section of the project report presents the findings and outcomes of the implemented intelligent video surveillance system. It highlights the performance, effectiveness, and capabilities of the system based on various metrics and evaluation measures. The specific results presented may vary depending on the project's objectives and the conducted experiments or tests.

Discussion:

The discussion section provides a comprehensive analysis and interpretation of the results obtained from the intelligent video surveillance system. It involves a deeper exploration of the findings, comparison with existing research or systems, and an examination of the implications and limitations of the results.

Conclusion:

In conclusion, the intelligent video surveillance project has successfully developed and implemented an advanced system that improves security measures through the integration of computer vision, machine learning, and real-time data processing techniques. The system demonstrates accurate object detection, tracking, activity recognition, and event detection capabilities. The user-friendly interface ensures efficient monitoring and response to security incidents. The project contributes to the field of intelligent video surveillance, highlighting the potential for enhanced threat detection and surveillance efficiency. Future research can focus on addressing challenges and further optimizing the system for scalability and privacy concerns. Overall, the project showcases the value and potential of intelligent video surveillance.

Chapter 6 – References

7.1 Bibliography

For Python
https://www.w3schools.com/python/default.asp
https://www.sitepoint.com/php/https://www.php.net/

For DeepLearning
https://www.tutorialspoint.com/tensorflow/tensorflow_machine_learning_deep_learning://
https://www.javatpoint.com/deep-learning/

For Bootstrap
https://www.tutorialspoint.com/bootstrap/index.htm
https://www.javatpoint.com/bootstrap-tutorial