CAPSTONE PROJECT - 1

CAMPUS SURVEILLANCE

AND

ANALYSIS SYSTEM

By

Dijin Dominic, Dennis P James, Anuja Alice Thomas

(2348027, 2348025, 2348013)

Under the Guidance of

Dr. Dalvin Vinod Kumar

A capstone project synopsis submitted in partial fulfillment of the requirements for the award of the degree of Master of Science (Data Science) of CHRIST (Deemed to be University)

May – 2024

Software Requirements Specification

for

Campus Surveillance and Analysis System

Version 1.0

Prepared by

Dijin Dominic, Dennis P James, Anuja Alice Thomas

07/072024

Table of Contents

1. Introduction

1.1 Background

The Campus Surveillance and Analysis System aims to improve security and monitoring abilities in a campus setting. This system utilizes sophisticated computer vision methods, such as YOLOv8, to recognize, monitor, and assess student actions and distinguish between individuals with ID cards and those without. The surveillance system can thus improve the overall security within the campus by preventing entry of intruders and rule violators.

1.2 Project Scope

The Campus Surveillance and Analysis System project aims to create a sophisticated, automated security system specifically designed for schools. The system's range includes gathering live video feeds from strategically positioned IP cameras around the campus and processing these feeds before using advanced YOLOv8 and Deep SORT algorithms for object detection and tracking. Essential features include keeping track of the number of students coming in and leaving specific zones, recognizing people with and without identification cards, and differentiating between hostel residents based on the color of their ID cards. The system has a robust data storage system and an easy-to-use web-based dashboard for live monitoring, analysis, and notifications. This project guarantees top-notch performance, dependability, expandability, and data protection while tackling possible risks and limitations linked to hardware needs and privacy issues.

2. Literature Review

2.1 Related Works

Researchers have explored the use of vehicle trajectory data in traffic conditions, utilizing uncrewed aerial vehicles (UAV) and videos for analysis [1]. They proposed a vehicle trajectory model incorporating CNN and YOLO v3 for vehicle detection, yielding accurate results. Stahl et al. addressed CNN limitations by designing a YOLOv3-based model for fast detection on the PASCAL VOC dataset, focusing on reducing power consumption [1].

A new method for real-time crowd detection from videos was introduced, emphasizing the importance of physical distancing in indoor spaces, especially during events like the COVID-19 pandemic.[2]

Machine learning and computer vision are crucial in data sensing, understanding, and action-taking based on past outcomes. Various methods like supervised, unsupervised, semi-supervised, and machine learning algorithms are employed for prediction and analysis. Object detection, traffic detection models, and cloud-based machine learning services are highlighted in the research[3].

Computer vision and machine learning are crucial areas of research that have seen significant advancements in recent years. Computer vision involves using images and pattern mappings to automate monitoring, inspection, and surveillance tasks. It treats images as arrays of pixels and has led to the automatic analysis and annotation of videos, showcasing capabilities like classification, object detection, and instance segmentation. The integration of machine learning paradigms such as support vector machines, neural networks, and probabilistic graphical models has further enhanced the capabilities of computer vision systems. The evolution of machine learning has transformed computer applications from simple data processing to more sophisticated systems that can learn and improve over time. Western countries have shown a keen interest in machine learning, computer vision, and pattern recognition, evident through various conferences, workshops, and real-life implementation. Recent applications of machine learning in computer vision include object detection, object classification, and extracting relevant information from images and videos, showcasing the practical implications of these technologies [4]

Tools like TensorFlow, Faster-RCNN-Inception-V2 model, and Anaconda have been utilized to identify objects like cars and persons in images, highlighting the practical implementations of machine learning in computer vision.

Machine learning techniques like the nearest neighbor algorithm, support vector machine (SVM), decision tree (DT), random forest, and Naïve Bayes classifier have been previously employed for land cover prediction. Input features are typically gathered from satellite images using time-series data after normalized difference vegetation index, with output classes including impervious, forest, grass, water, orchard, and farm. Synthetic minority techniques are applied for data balancing using oversampling, with k-NN providing the highest accuracy. High spatial resolution images from Google

Earth create land cover thematic maps for urban scenes. Different image classification methods are used to classify land usage and cover maps, with maximum likelihood techniques demonstrating better accuracy and kappa coefficient results. Using minimum distance and support vector machines as supervised classifiers, along with maximum likelihood and parallelepiped systems, has improved the kappa coefficient and overall accuracy.[5]

In the context of the COVID-19 pandemic, maintaining physical distance is crucial, leading to the need for limited people in indoor spaces, as the World Health Organization recommended. The capacity of indoor areas varies based on their size, necessitating the measurement of indoor spaces to calculate the maximum number of people allowed. Utilizing computers and cameras can aid in enforcing capacity rules in indoor spaces by measuring specific regions in real time and counting the number of people present. The proposed method involves predetermining the borders of an area in a video, identifying and counting people within that region, estimating its size, and determining the maximum number of people it can accommodate. The study employed the YOLO object detection model, specifically YOLO v3 and YOLO v5s, using Microsoft COCO dataset pre-trained weights to detect and label persons in the specified region, with performance metrics such as mean average precision (mAP), frame per second (fps), and accuracy rate is evaluated. The YOLO v3 model demonstrated the highest accuracy rate and mAP metrics. At the same time, YOLO v5s excelled in fps rate among non-tiny models, showcasing the effectiveness of these models in real-time crowd detection from videos.[1], [2]

Object detection is crucial for real-world applications such as surveillance, security, and automated vehicle systems. The paper compares the effectiveness of two algorithms, YOLOv3 and SSD, for people detection and counting at junctions. The study involves two main tasks: object detection using image datasets and counting objects using video datasets. Results show that SSD outperforms YOLOv3 in precision, recall, and F1 measures for people's detection and counting. Accurate people detection and counting are essential for applications like surveillance and security systems, and people counting at junctions serves various purposes in ensuring task integrity. The research provides a detailed analysis of the efficiency of YOLOv3 and SSD models for detection and counting tasks. The paper contributes by evaluating and comparing YOLOv3 and SSD models, offering insights for future research in people detection and counting.[1]

Computer vision aims to identify and recognize objects of various characteristics such as size, shape, and position. One of the challenges faced in computer vision is dealing with issues like illumination and viewpoint of objects. Deep learning techniques, particularly Convolutional Neural Networks, have shown high accuracy and precision in detecting and recognizing objects under such conditions. The proposed work in this paper focuses on object detection in a college environment, involving identifying individuals wearing ID cards using TensorFlow object detection API and recognizing faces using the Haar cascade method of OpenCV. Deep learning, explicitly using Convolutional Neural Networks, has been widely adopted in various computer vision tasks due to its effectiveness in handling complex visual data and achieving high accuracy and precision. This study's combination of TensorFlow and OpenCV showcases the integration of deep learning and traditional computer vision techniques for efficient object detection and recognition tasks in real-world scenarios.[4]

Security is one of the significant concerns of humans, as well as automated surveillance and security in improving home-based safety through the development of various safety schemes based on technologies such as embedded Linux, Raspberry Pi, OpenCV, and face recognition. It discusses using face recognition technology to develop a safety access control application, utilizing algorithms like Haar-like features, HOG, and SVM for face detection and recognition. The research paper suggests a hybrid expression recognition algorithm incorporating PCA and LDA for improved system control and verification performance. Emphasis is placed on developing efficient and accurate expression recognition schemes for safety controllers, especially in regions like Nigeria, to combat criminal activities. The paper also explores the use of low-cost, real-time structural designs based on FPGA for expression recognition, which is beneficial for identifying faces in various scenarios. It highlights the significance of advancements in face recognition technology for enhancing safety measures and providing flexibility in smart home control systems, mainly through Raspberry Pi boards. Additionally, the research delves into the application of fog computing for expression recognition and resolution outline to assess the impact of safety schemes on system performance.[6]

2.2 Literature Summary

Year of Publication	Author / Company	Techniques/ Algorithms	Gaps/drawbacks

2017	Machine Learning in Computer Vision, Asharul Islam Khan, Salim Al-Habsib	Techniques: supervised, unsupervised, semi-supervised. Algorithms: neural networks, k-means clustering, support vector machine.	Lack of focus on real-time applications in machine learning. Limited discussion on the challenges of data annotation in computer vision.
2019	Face Detection and Recognition for use in Campus Surveillance, B. Kranthikiran, Padmaja Pulicherla	Techniques: Geometric, feature-based, template matching. Algorithms: PCA, ICA, LDA, SVM, deep neural networks.	Lack of discussion on algorithm performance evaluation and comparison. Absence of exploration on scalability and adaptability of the system.
2020	Computer Vision Based Campus Surveillance, Mahesh Mahajan, Lokesh Sharma	Machine learning techniques: SVM, KNN, Naive Bayes, Decision Tree.	Lack of comparison with existing surveillance systems
2020	People detection and counting using YOLOv3 and SSD models	YOLOv3 and SSD are object detection algorithms used in the research. Object detection and counting are tasks performed by the algorithms.	Lack of exploration of real-world applications beyond surveillance and security. There is limited discussion on the impact of dataset variations on model performance.

2020	ID Card Detection with Facial Recognition using Tensorflow and OpenCV Kushal M Kushal Kumar B V Charan Kumar M J Prof.Pappa M	Object detection using tensor flow for ID card recognition. Haar cascade method in OpenCV for face detection. Mobile-based face recognition system using Euclidean distance calculation.	Lack of discussion on real-world implementation challenges. Absence of comparison with existing ID card detection methods
2023	A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models Mehmet Şirin Gündüz· Gültekin Işık	YOLO models were used for real-time crowd detection in the study. Deep learning algorithms are applied in various fields, including image processing.	Lack of comparison with other object detection models. There is an absence of discussion on the impact of varying video resolutions.

3. Overall Description

3.1 Problem Statement

An automated surveillance system is crucial to guarantee the safety and security of students on campus. Conventional manual monitoring techniques are inadequate for addressing the growing need for extensive security coverage. The system plans to tackle these issues by recognizing and monitoring students, spotting individuals without ID cards, and differentiating between hostel residents and day scholars according to the color of their ID cards. Implementing advanced computer vision methods, the system will offer security personnel instant alerts and analysis, facilitating quick reactions to possible security risks. Moreover, the system will improve campus management by providing in-depth insights and the ability to make decisions based on data.

3.2 Objectives

The following are the key objectives of the project:

1. **Real-Time Surveillance:** Monitor live video feeds to track individuals entering and leaving the campus.
2. **ID Card Detection:** Identify individuals not wearing their ID cards.
3. **Categorization by Tag Color:** Differentiate between students, faculty, and hostelers based on the color of their ID tags.
4. **Entry/Exit Analysis:** Provide detailed reports on the number of individuals entering and exiting the campus.
5. **Hostel Intrusion Detection:** Detect and alert unauthorized intrusions in hostel premises based on tag color.
6. **Gender Identification:** Optionally identify individuals as male or female to enhance security measures.

3.3 Operating Environment

The Campus Surveillance and Analysis System will function within the educational institution's network infrastructure, utilizing high-quality IP cameras strategically positioned throughout the campus to ensure full coverage. The system will operate on specialized servers with strong GPUs for real-time video processing and analysis. It will connect to a central database for storing and accessing data and work with a secure web-based dashboard that can be accessed through network connections. The operational setting needs to enable consistent, dependable functioning with little to no interruptions, guaranteeing that security staff can have uninterrupted access to real-time alerts and analytics. Adherence to privacy and data protection laws will be crucial to protect the gathered information.

3.4 Design and Implementation Constraints

The following are some of the significant constraints in the project:

- **Real-Time Processing:** The system needs significant computational power to handle video feeds in real-time, requiring strong GPUs and practical algorithms.
- The **accuracy of detecting and tracking** objects may be impacted by lighting and camera angles, potentially causing changes in performance.
- **Privacy Concerns:** To ensure the ethical use of captured data, the implementation of video surveillance must consider privacy concerns and adhere to applicable regulations.
- Good **network infrastructure** is essential for transmitting video feeds without delay, but it may be lacking in certain parts of the campus.
- **Scalability** is necessary for the system to manage numerous video feeds and rising data amounts as the campus expands.
- **Maintenance and updates** must be done regularly to maintain system performance and implement new technologies and enhancements.
- **Efficient storage solutions** are required for managing large amounts of video and analytical data, which should be easily accessible and secure.
- **Integration with Current Systems:** The new system needs to seamlessly integrate with the current campus security and IT infrastructure for smooth operation and data exchange.

3.5 Assumptions and Dependencies

A few assumptions in the project are:

- Cameras are strategically placed to cover all entry and exit points.
- All students are required to carry ID cards.
- Adequate network infrastructure exists for continuous video feed transmission.
- Adequate lighting conditions are maintained in monitored areas to optimize detection accuracy.

3.6 Potential Risks

- **Hardware Failure Leading to Downtime**: Potential breakdown of cameras, servers, or network infrastructure can cause interruptions in surveillance.
- **False Positives/Negatives in Detection**: Inaccuracies in detecting and identifying students can lead to security lapses or unnecessary alerts.
- **Data Privacy and Compliance Issues**: Risks related to the unauthorized access, use, or disclosure of surveillance data, as well as ensuring compliance with data protection regulations.
- **Network Reliability**: Dependence on a stable and high-speed network; any network disruptions can affect the system's real-time capabilities.
- **Integration Issues**: Potential challenges in integrating the surveillance system with existing campus IT and security systems.
- **Environmental Factors**: Adverse weather conditions or physical obstructions can impact camera visibility and detection accuracy.
- **Maintenance and Operational Costs**: Ongoing costs related to maintaining and upgrading hardware and software components of the system.

4. System Requirements

The functional and non-functional system requirements for the project are defined as:

4.1 Functional Requirements

1. **Video Feed Acquisition and Preprocessing**
   - **Description**: Capture real-time video feeds from IP cameras and preprocess frames to enhance quality.
   - **Inputs**: Video streams from IP cameras.
   - **Outputs**: Enhanced video frames for analysis.
   - **Processing**: Continuous capture, noise reduction, resolution adjustment, and frame normalization.

2. Entry and Exit Line Management

- **Description**: Define virtual entry and exit lines and detect crossing events within the video feed.
- **Inputs**: Configuration data for entry and exit points.
- **Outputs**: Identified entry and exit events.
- **Processing**: Analyze video frames to detect crossing events and update counters for student counts.

3. Object Detection and Tracking

- **Description**: Detect students and ID cards using YOLOv8 and track movements using the Deep SORT algorithm.
- **Inputs**: Preprocessed video frames.
- **Outputs**: Detected objects with bounding boxes, classification labels, and tracked paths.
- **Processing**: Apply YOLOv8 for detection and Deep SORT for continuous tracking across frames.

4. ID Card and Color-Based Identification

- **Description**: Identify students with and without ID cards and classify ID cards by color to distinguish hostelers from day scholars.
- **Inputs**: Detected objects and classification labels.
- **Outputs**: List of students with and without ID cards and classification labels indicating hosteler or day scholar status.
- **Processing**: Analyze detected objects for ID card presence and perform color analysis for classification.

5. Real-Time Alerts and Notifications

- **Description**: Send real-time alerts and notifications for detected anomalies, such as students without ID cards.
- **Inputs**: Detected anomalies and predefined rules.
- **Outputs**: Alerts and notifications to security personnel.

- **Processing**: Trigger alerts based on predefined conditions.

6. **Data Storage and Management**

- **Description**: Store detection and tracking data in a centralized database for further analysis.
- **Inputs**: Detection and tracking data.
- **Outputs**: Stored records in the database.
- **Processing**: Insert, update, and retrieve data efficiently.

7. **Analysis Dashboard**

- **Description**: Provide a web-based dashboard for real-time monitoring, visual analytics, and reporting.
- **Inputs**: Stored data from the database.
- **Outputs**: Visual analytics and reports.
- **Processing**: Data visualization and report generation.

8. **Future Scope Features**

1. **Hosteler Identification (Color Classification)**: Extend color-based ID card identification to provide additional features.
2. **Intruder Detection**: Enhance the system to detect and alert unauthorized individuals.

4.2 Non- Functional Requirements

The non-functional requirements are:

- **Performance**: The system must process real-time video feeds with minimal latency.
- **Reliability**: The system should have high uptime and handle hardware failures gracefully.
- **Scalability**: The system should be able to handle multiple video feeds simultaneously.
- **Usability**: The dashboard should be intuitive and easy to use for security personnel.
- **Security**: Ensure data privacy and secure access to the system.

5. External Interface Requirements

5.1 Hardware Interfaces

The following are the hardware requirements for the project:

- High-resolution IP cameras
- Dedicated GPU for real-time processing (e.g., NVIDIA RTX series)
- Server with at least 16GB RAM and multi-core processor
- Storage solution for video feeds and analytical data (e.g., SSDs)

5.2 Software Interfaces

The following are the software requirements for the project:

- Operating System: Linux/Windows
- YOLOv8 for object detection
- Deep SORT for tracking
- Python 3.x for development
- OpenCV for video processing
- Database: MySQL/MongoDB
- Dashboard framework: Flask/Django for web interface

6. System Description

6.1 Diagram Description

The flowchart illustrates the process of the Campus Surveillance and Analysis System. The system begins with video feed acquisition from strategically placed IP cameras. These feeds are preprocessed to enhance quality, followed by determining entry and exit lines within the video frames. Object detection using YOLOv8 is then performed to identify students and their ID cards. The Deep SORT algorithm tracks these students through the monitored areas. The system outputs include the count of entered students, ID card detection, and potential future enhancements like hosteler identification via color classification and intruder detection. All outputs are saved to a centralized database, which supports real-time alerts and notifications and provides data for an analysis dashboard. An ID card dataset created for model training and detection, supplemented with existing datasets for student detection, is used to improve system accuracy.

6.2 Block Diagram

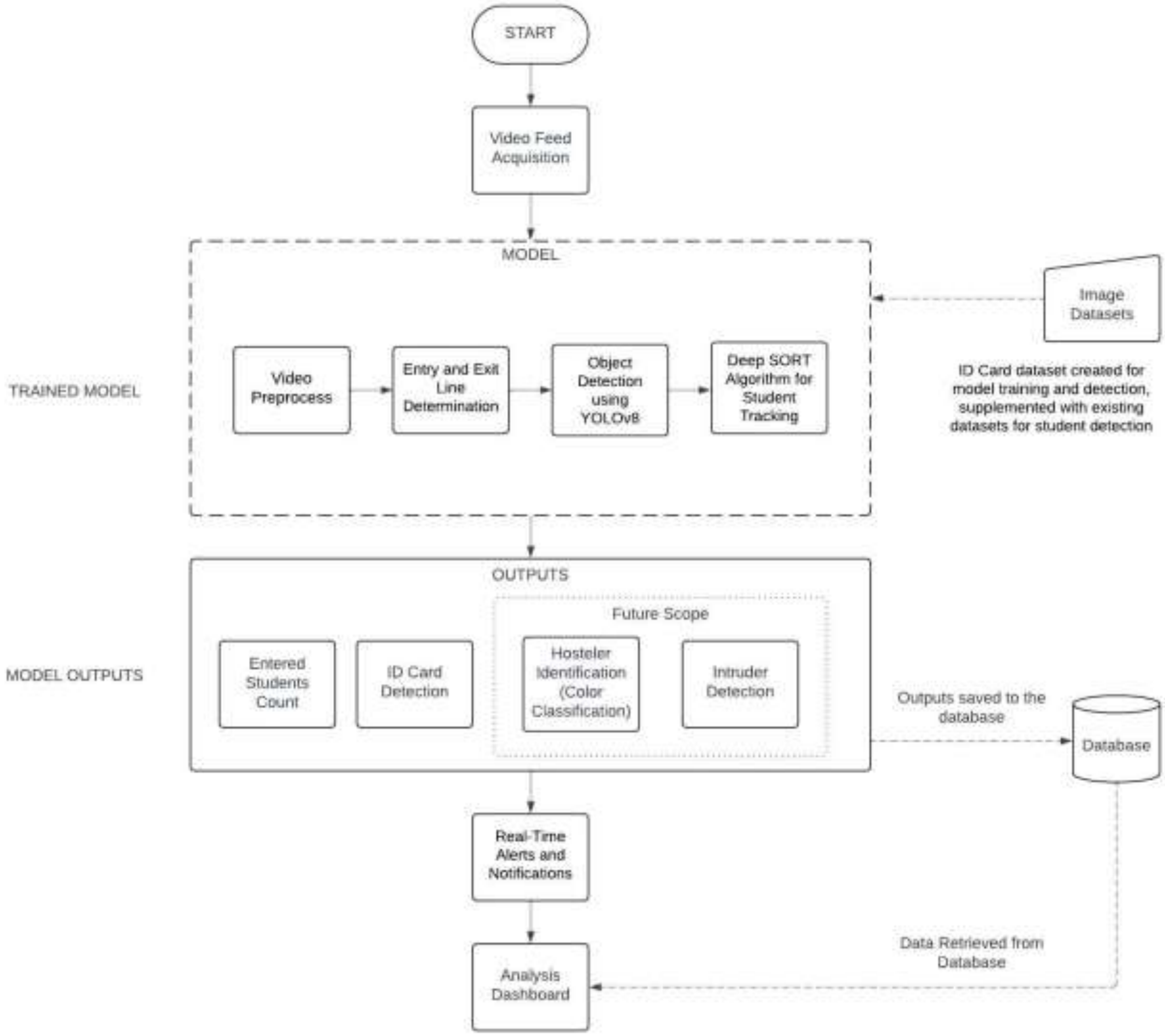

Fig 1: Project Block Diagram

References

[1] P. Gupta, V. Sharma, and S. Varma, “People detection and counting using YOLOv3 and SSD models,” *Mater Today Proc*, 2021, doi: 10.1016/j.matpr.2020.11.562.

[2] M. Ş. Gündüz and G. Işık, “A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models,” *J Real-Time Image Process*, vol. 20, no. 1, Feb. 2023, doi: 10.1007/s11554-023-01276-w.

[3] A. I. Khan and S. Al-Habsi, “Machine Learning in Computer Vision,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1444–1451. doi: 10.1016/j.procs.2020.03.355.

[4] Institute of Electrical and Electronics Engineers and RVS College of Engineering & Technology, *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications (ICIRCA 2020) : 15-17 July, 2020*.

[5] M. Mahajan, R. Rajeshwaribajre, L. Sharma, and S. L. Varma, “Computer Vision Based Campus Surveillance,” in *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1013–1018. doi: 10.1109/ICACCCN51052.2020.9362776.

[6] P. Pulicherla, “Face Detection and Recognition for use in Campus Surveillance International Journal of Innovative Technology and Exploring Engineering (IJITEE),” *Article in International Journal of Innovative Technology and Exploring Engineering*, 2020, doi: 10.35940/ijitee.B7682.019320.