Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan, Karnataka, India – 573202

Capstone Project Phase-II (22AI706)

Report on

"AI-Powered Smart Surveillance for Campus Security and Monitoring"

Submitted by:

Ankita	4MC22CI003
Dhavan H S	4MC22CI011
Harshitha B V	4MC22CI012
Kumuda D P	4MC22CI014
Manikya K	4MC22CI016

Under the Guidance of

Dr. Swathi H Y

Associate Professor

Dept. of CSE (AI and ML)

MCE, Hassan

Department of Computer Science and Engineering

(Artificial Intelligence and Machine Learning)

Malnad College of Engineering

PB# 21, Hassan, Karnataka, India – 573 202

2025-26

Malnad College of Engineering

Hassan-573202, Karnataka, India

(An autonomous Institute affiliated to Visvesvaraya Technological University, Belagavi)

Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning)

CERTIFICATE

This is to certify that, the Capstone Project entitled **"AI-Powered Smart Surveillance for Campus Security and Monitoring"** is a bonafied work carried out by **Ankita, Dhavan H S, Harshitha B V, Kumuda D P, Manikya K,** bearing **USN: 4MC22CI003, 4MC22CI011, 4MC22CI012, 4MC22CI014, 4MC22CI016,** bonafied students of Malnad College of Engineering, Hassan, in partial fulfillment for the award of Degree of Bachelor of Engineering in Computer Science and Engineering (Artificial Intelligence and Machine Learning), Malnad College of Engineering, Hassan, an autonomous Institute affiliated to Visvesvaraya Technological University, Belagavi, during the academic year 2025-2026.

Guide:
Dr. Swathi H Y
Associate Professor
Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning)
Malnad College of Engineering, Hassan - 573202

Dr. Arjun B C
Professor & HOD
Department of CSE(AI&ML)
Malnad College of Engineering
Hassan - 573202

Dr. Amarendra H J
Principal
Malnad College of Engineering
Hassan - 573202

Examiners: 1.

2.

Malnad College of Engineering

(An autonomous Institution under Visvesvaraya Technological University, Belagavi)
Hassan – 573202, Karnataka, India

Vision of the Institute

To be an institute of excellence in engineering education and research producing socially responsible professionals.

Mission of the Institute

- ❖ To create a conducive environment for learning and research.
- ❖ Establish industry academic collaborations.
- ❖ Ensure professional and ethical values in all institutional endeavours.

Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning)

Vision of the Department

To be a Center of Excellence for innovative teaching, learning and research to produce socially responsible professionals in the field of Artificial Intelligence and Machine Learning to address real-world problems.

Mission of the Department

❖ Fostering innovation through cutting-edge teaching, transformative learning, and innovative research in the field of artificial intelligence and machine learning with foundations of Computer Science and Engineering.

❖ Impart latest technology by establishing industry–academia collaboration.

❖Maintain high standards of ethical values involved in AI and ML applications with transparency of operations for moral concerns of the society.

ACKNOWLEDGEMENT

We present with immense pleasure this work titled "**AI-Powered Smart Surveillance for Campus Security and Monitoring**". An endeavor over a long period can be successful with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them. The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning the people who made it possible. So, with gratitude we acknowledge all those whose guidance and encouragement made to successfully complete this project.

We would like to express sincere thanks to our **Principal Dr. Amarendra H J**, Malnad College of Engineering for his encouragement in successful completion of the project work. We wish to express our gratitude to **Dr. Arjun B C**, **Professor and Head**, Department of Computer Science & Engineering (Artificial Intelligence and Machine Learning) for providing a good working environment and for his constant support and encouragement. It gives great pleasure in placing on record a deep sense of gratitude to our guide **Dr. Swathi H Y**, **Associate Professor,** Department of Computer Science & Engineering (Artificial Intelligence and Machine Learning) for his/her regular evaluation of the work and for providing us constant encouragement with unflinching support and valuable guidance throughout this project. We would also like to thank all the staff of the department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) who have directly or indirectly helped us in the completion of the project work. At last, we would hereby acknowledge and thank our parents who have been a source of inspiration and instrumental in the successful completion of the project work.

Ankita (4MC22CI003)

Dhavan H S (4MC22CI011)

Harshitha B V (4MC22CI012)

Kumuda D P (4MC22CI014)

Manikya K (4MC22CI016)

DECLARATION

Ankita (4MC22CI003), Dhavan H S (4MC22CI011), Harshitha B V (4MC22CI012), Kumuda D P (4MC22CI014), Manikya K (4MC22CI016), students of B.E. Computer Science and Engineering (Artificial Intelligence and Machine Learning), Malnad College of Engineering, Hassan, hereby declare that the capstone project entitled **"AI-Powered Smart Surveillance for Campus Security and Monitoring"** is an authentic work carried out under the supervision and guidance of **Dr. Swathi H Y**, **Associate Professor**, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Malnad College of Engineering, Hassan. We have not submitted the matter embodied to any other university or institution for the award of any other degree.

Place: Hassan
Date: 30-12-2025

Name	USN	Signature
ANKITA	4MC22CI003	
DHAVAN H S	4MC22CI011	
HARSHITHA B V	4MC22CI012	
KUMUDA D P	4MC22CI014	
MANIKYA K	4MC22CI016	

ABSTRACT

Campus security is a growing concern due to increasing incidents such as unauthorized access and vehicle intrusions. This project presents an AI-based campus surveillance system that enables real-time monitoring using computer vision techniques. The system processes live CCTV video to detect humans and vehicles using the YOLOv8 model. Face detection and recognition are performed using Haar Cascade and the LBPH algorithm to identify known individuals. Vehicle number plates are extracted using Optical Character Recognition (OCR). The system displays annotated video output and stores detection logs for analysis. This automated approach reduces manual monitoring and enhances campus safety effectively.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1. **AI** – Artificial Intelligence
2. **ANPR** – Automatic Number Plate Recognition
3. **CNN** – Convolutional Neural Network
4. **CPU** – Central Processing Unit
5. **CCTV** – Closed Circuit Television
6. **DL** – Deep Learning
7. **FPS** – Frames Per Second
8. **GPU** – Graphics Processing Unit
9. **HDD** – Hard Disk Drive
10. **HOG** – Histogram of Oriented Gradients
11. **IoT** – Internet of Things
12. **IR** – Infrared
13. **KNN** – K-Nearest Neighbors
14. **LAN** – Local Area Network
15. **LBPH** – Local Binary Patterns Histogram
16. **ML** – Machine Learning
17. **OCR** – Optical Character Recognition
18. **OS** – Operating System
19. **RAM** – Random Access Memory
20. **SSD** – Solid State Drive
21. **UI** – User Interface
22. **YOLO** – You Only Look Once
23. **YOLOv8** – You Only Look Once Version 8

CHAPTER 1

INTRODUCTION

1.1 Overview

Security has become a major concern in educational institutions due to increasing incidents such as unauthorized access, theft, vandalism, and inappropriate student behavior. These security challenges not only put students and staff at risk but also disrupt the learning environment and damage institutional property. Traditional surveillance systems rely heavily on manual monitoring of CCTV footage, making them dependent on human operators whose attention and performance naturally decline over long periods of observation.

Manual surveillance has significant limitations. Human attention spans are limited, and monitoring multiple camera feeds for extended durations often leads to delayed responses or missed events. As campuses expand in size and population, continuously monitoring all areas becomes even more challenging using conventional approaches.

To overcome these limitations, modern surveillance solutions increasingly incorporate Artificial Intelligence (AI) and Computer Vision. By automating the process of analyzing CCTV footage, AI-powered systems can detect people, vehicles, and faces in real time without requiring constant human supervision. These intelligent systems are not only faster and more accurate but also capable of operating continuously without fatigue.

In this context, the integration of AI into campus surveillance offers a shift from passive recording to active, intelligent monitoring. Through real-time object detection using deep learning models, face detection and recognition using LBPH, and automated vehicle number plate extraction using OCR, educational institutions can greatly enhance their ability to maintain a safe, secure, and well-monitored campus environment.

1.2 Motivation

Ensuring the safety of students, staff, and campus infrastructure is a major priority for every educational institution. Conventional CCTV systems capture video effectively but depend entirely on human operators to analyze the footage in real time. This approach becomes

inefficient and impractical, especially in large campuses with multiple entry points and high movement.AI-based surveillance systems provide a more reliable and automated alternative. Real-time detection of people and vehicles helps identify unusual activity instantly, while face recognition can verify individuals entering restricted areas. Automated number plate recognition (ANPR) assists in maintaining accurate records of vehicle movement within the campus.This project is motivated by the need to reduce manual effort, improve incident detection speed, enhance security accuracy, and build a more intelligent, responsive surveillance system by leveraging modern AI technologies.

1.3 Problem Statement

Traditional CCTV surveillance systems operate passively and rely heavily on manual monitoring, which limits their ability to provide real-time threat detection and timely responses. Since human operators cannot continuously observe multiple camera feeds with full attention, critical events such as unauthorized entry, suspicious vehicle movement, or the presence of unknown individuals may go unnoticed.Most existing systems do not include automated face recognition, object detection, or number plate identification. As a result, identifying individuals or tracking vehicles entering and exiting the campus becomes difficult and highly time-consuming. Reviewing hours of recorded footage for specific events is inefficient and prone to errors. There is a need for an intelligent surveillance solution that can automatically detect people, vehicles, faces, and vehicle number plates directly from live CCTV footage. Integrating AI-based models such as YOLOv8 for object detection, LBPH for face recognition, and OCR for number plate extraction can significantly enhance the accuracy and speed of campus security operations. The goal is to minimize manual effort, reduce missed incidents, and provide real-time automated analysis to improve overall campus safety and monitoring efficiency.

1.4 Scope of the Project

The scope of this project includes the development of an AI-powered campus surveillance system capable of real-time monitoring using CCTV camera feeds. The system focuses on automating the detection of people and vehicles using deep learning models, reducing the need

for continuous manual observation. It also incorporates license plate detection and Optical Character Recognition (OCR) to extract alphanumeric information from vehicle number plates for improved access control and vehicle monitoring. In addition, the system performs face detection and recognition using the LBPH algorithm, enabling the identification of authorized individuals and helping to detect unknown or unauthorized persons in restricted areas. The solution is designed to be scalable and compatible with existing CCTV infrastructure, allowing institutions to integrate AI-based intelligence without major hardware upgrades. The project supports optional data logging of recognized faces and detected number plates, facilitating future verification and enhancing investigation capabilities. While factors such as low lighting conditions, occlusion, camera angle, and video quality may affect accuracy, these limitations can be mitigated through improved hardware and optimized model configurations in future enhancements. Overall, the system aims to provide a reliable, automated, and intelligent approach to campus surveillance by combining real-time object detection, face recognition, and number plate extraction.

1.5 Objective

The main objective of this project is to develop an intelligent AI-driven surveillance system that enhances campus safety through automated, real-time monitoring of CCTV feeds. The system aims to accurately detect people and vehicles on campus using advanced deep learning models, thereby reducing reliance on manual surveillance. A key objective is to detect vehicle number plates and extract alphanumeric information using Optical Character Recognition (OCR), enabling easier identification of vehicles entering or exiting the campus. The system also includes face detection and face recognition using the LBPH algorithm to help identify authorized or unauthorized individuals in restricted or sensitive areas. By integrating YOLOv8 for object detection, OCR for number plate recognition, and LBPH for face recognition, the system ensures real-time, reliable, and automated surveillance. This reduces manual effort, supports faster response to security events, and provides a scalable monitoring solution suitable for deployment across various campus locations. The system also aids in maintaining digital records of recognized faces and detected number plates for future security analysis.

CHAPTER 2

LITERATURE SURVEY

Building on the foundational understanding provided in the previous chapter, which outlined the motivation, objectives, and scope of our project, this chapter presents a comprehensive review of related work carried out in the domain of facial recognition and CCTV-based surveillance systems. The goal is to understand existing approaches, identify gaps, and highlight techniques that align with or contrast our project objectives.

2.1 Detailed Literature Survey

Building on the foundational understanding provided in the previous chapter, which outlined the motivation, objectives, and scope of our project, this chapter presents a comprehensive review of related work carried out in the domain of facial recognition and CCTV-based surveillance systems. The goal is to understand existing approaches, identify gaps, and highlight techniques that align with or contrast our project objectives.

2.1.1 Literature Review

The rapid advancement of artificial intelligence and computer vision has enabled the development of increasingly sophisticated surveillance systems, particularly those incorporating facial recognition for enhanced security and monitoring. Researchers have proposed a variety of approaches, models, and tools aimed at addressing challenges such as recognition accuracy, environmental variability, and system scalability. The following review presents selected contributions in this domain, outlining their methodologies, performance results, and technical limitations, thereby providing a contextual understanding of current trends and innovations in CCTV-based facial recognition systems.

To enhance the performance of campus monitoring and individual tracking, Sarikonda Poojitha and Bhutada (2020) [1] proposed a "Customized Campus Surveillance System". This system utilized enrollment databases of students, staff, and visitors, combined with image capture for entry-exit logging. The preprocessing steps included registration and database creation for face image recognition. Technologies such as TensorCam, TensorFlow, Keras, and Caffe2 were

employed for facial recognition tasks. The approach enabled efficient tracking of individuals within the campus environment. However, the system lacked numerical performance metrics, and its reliance on pre-registered users presents a challenge in terms of scalability.

To improve attendance tracking through real-time surveillance, Rakshitha (2021) [2] utilized CCTV footage to automate attendance monitoring. The preprocessing steps included face detection, image normalization, and grayscale conversion. Facial recognition and feature extraction were performed using the Local Binary Pattern Histogram (LBPH) technique. The system achieved a high accuracy of 99.69%, highlighting its effectiveness for attendance automation. However, challenges such as poor lighting conditions, pose variations, limited training data, and concerns regarding user privacy affect its broader applicability.

To advance the accuracy of attendance systems through deep learning, Alhanaee et al. (2021) [3] employed a student image dataset to develop a smart face recognition model. Face detection was carried out using the Haar Cascade Classifier, while recognition was achieved through pre-trained convolutional neural networks—ResNet50, VGG16, and InceptionV3—using transfer learning and fine-tuning techniques. The system demonstrated high prediction accuracy and efficient training time. Although the study did not explicitly mention any drawbacks, practical considerations such as integration challenges may still arise in real-world deployments.

To enhance the intelligence and accuracy of attendance systems, Setialana et al. (2021) [4] implemented a model based on Deep Convolutional Neural Networks (DCNN) using a customized student image dataset. The preprocessing phase involved face detection and alignment to ensure consistent input. Recognition was carried out through DCNN, and although no specific accuracy value was reported, the system was described as highly effective. Nonetheless, challenges such as sensitivity to lighting conditions, variations in facial pose, high computational demands, and user privacy concerns were acknowledged.

To strengthen campus security and real-time monitoring, Junaid et al. (2021) [5] implemented a smart surveillance system using live CCTV streams. Video frames were processed in real time, with face and object recognition carried out using the YOLO algorithm along with supporting image processing techniques. The integration of IoT-enabled devices allowed automation of tasks such as data collection and alert generation, thereby improving responsiveness and

oversight. However, the continuous nature of surveillance through IoT networks raises concerns related to privacy and data protection.

To improve safety monitoring in educational environments, Ye et al. (2021) [6] developed a system for detecting campus violence through analysis of surveillance video sequences. The initial stage involved frame extraction and normalization. Human figure detection and interaction analysis were used to identify violent acts, followed by action recognition using deep learning models and temporal analysis. By interpreting actions over time, the system effectively leveraged sequential data for enhanced accuracy in identifying suspicious behavior. While the system demonstrates strong potential, addressing real-time constraints and false detection rates remains important for practical deployment.

To optimize surveillance video analysis for smart campuses, Muhammad et al. (2021) [7] proposed a summarization framework based on activity attention modeling using deep learning. Using the Campus Activities Dataset (CAD), activities were labeled and recorded in video sequences. Deep features were extracted through a CNN model, while an activity attention mechanism helped highlight key events. This approach enabled conversion of long surveillance videos into concise summaries, improving monitoring efficiency. However, the model's reliance on a specific dataset and the requirement for retraining in different environments present challenges for broader scalability.

To strengthen surveillance capabilities in campus environments, Keerthana et al. (2024) [8] developed a smart system integrating multiple face recognition and feature extraction techniques. Preprocessing steps involved face decoding, landmark detection, normalization, and time-based filtering. Surrounding features were extracted using HOG, SIFT, SURF, and DNN encoding methods. For recognition, combinations such as HOG + SVM + DNN were employed, along with CONV networks for mask detection. Implementation made use of Python libraries including OpenCV, TensorFlow, and cvzone. While technically diverse, the system’s effectiveness in dynamic real-world scenarios may depend on computational efficiency and environmental conditions.

To enable real-time monitoring and proactive response in surveillance systems, Ullah et al. (2022) [9] developed a framework for human face detection and recognition using CCTV

images. The system analyzed movement and abnormal behavior through AI-based crowd density monitoring. Video clarity was enhanced using filters, and footage was gathered from multiple CCTV sources across the university. An integrated alert mechanism notified authorities promptly upon detecting unusual activity. Although no specific accuracy metrics were reported, the framework demonstrated functional capability. A key challenge lies in the need for continuous system updates to adapt to evolving threats and manage potential false positives effectively.

To streamline attendance tracking through intelligent surveillance, Kagona (2022) [10] implemented a facial recognition system using CCTV cameras, combining OpenCV and deep learning techniques. The system utilized the Campus Activities Dataset (CAD), where activities were labeled and recorded from video footage. Deep features were extracted using a CNN model, supported by an activity attention mechanism to identify key events. This enabled transformation of lengthy surveillance footage into concise highlights for efficient monitoring. However, dependence on a specific dataset and the necessity for retraining in varied environments present challenges to generalizing the system effectively.

To enable low-cost, hardware-based face recognition, Zamir et al. (2022) [11] developed a system using CNN models deployed on Raspberry Pi. Preprocessing steps involved face decoding, landmark detection, normalization, and temporal filtering. Feature extraction was carried out using HOG, SIFT, SURF, and DNN encoding methods. For recognition, combinations such as HOG + SVM + DNN were applied, along with convolutional models for mask detection. The implementation leveraged Python libraries including OpenCV, TensorFlow, and cvzone. While the system shows promise in embedded environments, performance may be influenced by hardware limitations and processing delays in real-time scenarios.

To support real-time IoT-based surveillance, Chattopadhyay et al. (2022) [12] developed a face recognition framework using a custom dataset of 67 subjects. Preprocessing steps included normalization, contrast enhancement, background removal, and lighting correction. Face detection and recognition were performed using Haar Cascade and LBPH, with MTCNN employed to improve detection accuracy. The system achieved an accuracy of 73.41% with a test duration of 0.241 seconds, enabling timely monitoring and automated email alerts. Despite its

utility, the system faced challenges such as moderate recognition accuracy, sensitivity to lighting conditions, and obstruction of facial features.

To enhance facial recognition for forensic applications, Verma et al. (2022) [13] employed the Likelihood Ratio (LR) approach using facial landmark indices from the FaceScrub dataset. Preprocessing involved face warping, frontal normalization, and landmark detection using Dlib. Twelve facial indices were computed based on landmark distances, and recognition was performed using LR, with HoG for face detection and CNN for landmark matching. The system achieved a true-positive rate of 85.75% with a 25% false positive rate, resulting in a cllr of 0.26. While the model demonstrated interpretability and resilience to pose and lighting variations, its performance remains lower compared to deep learning-based methods.

To improve recognition accuracy under challenging visual conditions, Seshaiah Merikapudi et al. (2022) [14] proposed a CNN-based approach trained with a histogram equalization image enhancement scheme. The model was evaluated on the YouTube Face and IJB-A datasets, with preprocessing steps including histogram equalization, face alignment, and resizing images to 224×224 pixels. A GoogleNet CNN integrated with an attention-based feature aggregation module was used for recognition. The system achieved an accuracy of 98.55% and an AUC exceeding 99% across both datasets, showing robustness against occlusions and low-quality CCTV footage. However, the reliance on high-end GPUs and deployment challenges on edge devices remain important concerns.

To address the challenges of recognizing masked faces in real-time scenarios, Golwalkar and Mehendale (2022) [15] developed a system using deep metric learning with a custom CNN model named FaceMaskNet-21. The approach utilized the LFW and RMFRD datasets along with a proprietary dataset containing approximately 2,000 images. Preprocessing included cropping, rotation, resizing, and contrast adjustment. Recognition was based on 128-dimensional encodings generated through a quadruplet-based metric learning framework. The model achieved an overall accuracy of 88.92%, with 82.22% on RMFRD and 100% for single-face video frame detection, making it suitable for embedded applications. Nonetheless, reduced performance was observed in multi-face scenarios, with additional challenges in detecting children’s faces and non-frontal angles.

To enable reliable face recognition in diverse real-world conditions, Nurpeisova et al. (2022) [16] explored mathematical models and algorithms within a proctoring system using Python. A custom dataset of 400 images from 40 students was compiled, with image collection performed under varying lighting and angle conditions. Preprocessing involved resizing, brightness correction, and face detection using the Haar Cascade method, followed by model training. While the study effectively demonstrated the use of both traditional and deep learning techniques, its limited dataset size and constrained subject diversity may affect generalizability in larger deployment scenarios.

To examine the broader implications of facial recognition technologies, Smith and Miller (2022) [17] conducted a theoretical analysis focusing on ethical, societal, and legal dimensions rather than technical implementation. The study explored the operations of organizations like Clearview AI and AFR Locate across the US, UK, and Australia, emphasizing the urgent need for well-defined regulatory frameworks, oversight mechanisms, and legal accountability. As the work was conceptual in nature, it did not involve specific models, datasets, or experimental metrics. The absence of empirical validation and a structured analytical methodology were noted as key areas requiring further development.

To evaluate the role of CCTV in enhancing campus safety, Prasetyo (2022) [18] conducted a qualitative study at IAIN Kendari, relying on manual observation and structured interviews for data collection. The study did not incorporate image datasets or AI models. Surveillance relied on analog CCTV systems with DVRs, which limited functionality. Despite this, notable reductions in incidents such as theft, vandalism, and unauthorized access were observed in the monitored areas. However, the absence of intelligent analytics, real-time monitoring capabilities, and remote control access highlights operational limitations that could impact scalability and responsiveness.

To improve facial recognition accuracy on low-resolution CCTV images, Gondosiswojo and Kusuma (2023) [19] combined super-resolution techniques with face recognition models using the LFW dataset. Preprocessing included MTCNN-based face detection, resizing images from 128×128 to 32×32, and class-based sorting. Super-resolution methods such as U-Net, EDSR, and Bicubic interpolation were integrated with recognition models like ResNet50 and MobileNetV2.

The U-Net and ResNet50 combination achieved the best results with 85% accuracy, 87% precision, 85% recall, and a processing time of 114.54 ms per image. Challenges noted include the relatively lower performance of MobileNetV2 on low-resolution images and reduced accuracy with Bicubic interpolation.

To enhance home surveillance for child safety, Firmasyah et al. (2023) [20] developed a face recognition system aimed at preventing child kidnapping using CCTV and the YOLO algorithm. The Kaggle face dataset with 918 images was used, and preprocessing involved face classification into child/known/stranger categories, frame segmentation, and distance filtering. Recognition and detection were performed using CNN, LBPH, SVM, and YOLOv8. YOLOv8 achieved high detection performance with mAP@0.5 values of 0.995 and 0.976 at 100 and 200 epochs, respectively. While CNN delivered the highest accuracy, it incurred longer runtimes (17–24 seconds), compared to LBPH's faster performance and SVM's slower processing. Notable challenges include misclassifications in single-camera setups and sensitivity to lighting and viewing angles.

To evaluate the comparative effectiveness of face recognition models in attendance systems, Budiman et al. (2023) [21] conducted a systematic review using datasets of 1,050 images for CNN and 165 for LBPH. Preprocessing included frame extraction, grayscale conversion, resizing, and both manual and automatic feature extraction. LBPH was combined with HOG for manual recognition, while CNN was responsible for automatic detection and classification. Among the evaluated models, CNN achieved the highest accuracy at 99%, followed by LBPH at 92%, and MLP and SVM at 87% and 88%, respectively. Performance was influenced by factors such as limited dataset size, computational demands, and vulnerability to occlusions and varying lighting conditions.

To explore the potential of 3D face recognition techniques, Hangaragi et al. (2023) [22] utilized the LFW dataset along with 20 real-time images, amounting to 1,700 samples. Preprocessing steps included facial landmark detection using MediaPipe Face Mesh, normalization, and 3D reconstruction, where each face was mapped with 468 (x, y, z) coordinates. BlazeFace and Face Mesh were integrated with a deep neural network for recognition. The system achieved an accuracy of 94.23%, showing strong resilience to variations in pose and lighting. Nonetheless, its

recognition accuracy decreased in the presence of extreme facial expressions and sharply angled face positions.

To promote privacy in surveillance-based face recognition, Henry et al. (2023) [23] utilized a lensless imaging approach using the FlatCam Face Dataset, which included 23,838 images from 87 subjects. The original high-resolution inputs (1280×1024) were resized and converted into RGB and DCT subband representations (32×32×15) as part of the preprocessing. Recognition was performed using VGG16 and its attention-based variant, VGG-ATT. On unprocessed input, the models achieved 46.59% and 42.81% accuracy, respectively, which significantly improved to 73.60% and 87.71% with DCT-based preprocessing. Although effective in preserving privacy, the system relied on specialized lensless hardware and demonstrated comparatively lower accuracy than traditional webcam-based methods due to inherent imaging constraints.

To strengthen real-time surveillance applications, Irfan et al. (2024) [24] developed a facial recognition system integrated with CCTV using deep learning techniques. The model was trained on diverse datasets and evaluated using live CCTV footage. Preprocessing steps included face detection, alignment, normalization, grayscale conversion, and feature extraction using deep convolutional neural networks. Haar Cascade classifiers supported efficient real-time tracking. The system effectively recognized individuals in live video streams; however, performance was affected by low-light conditions and varying facial orientations during operation.

To address the challenges of low-resolution facial recognition in surveillance settings, Surantha et al. (2024) [25] developed a system aimed at enhancing recognition performance for CCTV and edge-powered smart attendance solutions. The study employed the Labeled Faces in the Wild (LFW) and SCface datasets, with preprocessing steps including image resizing, alignment, and grayscale transformation. Feature extraction was conducted using LightCNN and ArcFace embeddings, while classification was handled by LightCNN, MobileNet-V2, and ArcFace models. The system demonstrated satisfactory accuracy across varying resolutions; however, implementation in real-world scenarios was hindered by limited adaptability to low-resolution inputs, pose variations, and constrained dataset diversity.

For automated class attendance monitoring using surveillance footage, Ambre et al. (2024) [26] developed a system that analyzed classroom video recordings to detect and recognize student

faces. Preprocessing involved extracting frames, applying Haar Cascade-based face detection, grayscale conversion, and resizing. Facial embeddings were generated using FaceNet, followed by classification with the K-Nearest Neighbors (KNN) algorithm. The system achieved an accuracy of 92.85%. Despite its effectiveness, the model's performance was affected by suboptimal lighting conditions, partial face occlusions, and non-frontal camera angles.

To enhance suspect identification from surveillance video, Sari et al. (2024) [27] developed a facial recognition system using a Hybrid Bat Algorithm (HBA) for feature optimization. The system utilized the ORL and Yale face datasets, with preprocessing steps including histogram equalization and face detection through Haar classifiers. Feature extraction was performed using Local Binary Patterns (LBP), followed by optimization with HBA and classification using the K-Nearest Neighbors (KNN) algorithm. The model achieved accuracies of 94.4% on ORL and 92.3% on Yale datasets. While the results were promising, the system's effectiveness was constrained by its reliance on static images from controlled settings, limiting its application in real-world dynamic surveillance environments.

In the context of real-time criminal face detection, Bhatt et al. (2024) [28] developed a system capable of identifying suspects from live surveillance video streams. The approach employed Haar Cascade for face detection and used Histogram of Oriented Gradients (HOG) for feature extraction. Classification was performed using Support Vector Machines (SVM). While the system demonstrated functional real-time face identification, specific performance metrics were not provided. Its effectiveness was constrained by a dependence on high-resolution video input and vulnerability to low-light conditions.

For deep learning-based criminal tracking, Singh et al. (2024) [29] developed *AIGuard*, a system designed to identify suspects in CCTV footage using MTCNN and ResNet. A custom CCTV dataset was created for training and evaluation. Preprocessing steps included frame extraction, grayscale conversion, image resizing, and normalization. MTCNN was used for face detection, while ResNet handled both feature extraction and classification. The system achieved an accuracy of 92.56%. Although it showed strong performance, its effectiveness was reduced when processing occluded or blurry faces, limiting the model’s ability to generalize across varied real-world scenarios.

To support law enforcement through facial recognition, Chittibomma et al. (2024) [30] developed a system integrating Haar Cascade classifiers with the Local Binary Patterns Histogram (LBPH) algorithm. The model was trained on a dataset of law enforcement images, with preprocessing steps including grayscale conversion and face alignment. Feature extraction and recognition were conducted using LBPH, with Haar Cascade assisting in initial face detection. The system exhibited high accuracy in controlled environments; however, its reliability decreased under low-light conditions and limited dataset diversity, highlighting constraints in generalizability and model complexity.

To achieve high-accuracy facial recognition in unconstrained environments, Rathod et al. (2025) [31] enhanced the VGGFace-16 model using transfer learning techniques. The system was trained on VGGFace2 and LFW datasets, with preprocessing steps that included data augmentation through image rotation, flipping, and contrast enhancement. Architectural modifications included the addition of global average pooling and PReLU activation functions. The model achieved outstanding performance, reporting 99.9% in accuracy, precision, recall, and F1 score. However, its deployment required substantial computational resources and access to large, diverse datasets to sustain robustness under varied real-world conditions.

In an effort to aid missing child identification, Sk et al. (2025) [32] developed a facial recognition system incorporating neural networks and age progression techniques. The system utilized real-world facial images and inputs from law enforcement databases, with preprocessing involving facial aging simulations and enhancement methods. CNN-based facial embeddings were generated using transfer learning for training and matching. While the system was reported to achieve high precision and recall, exact performance metrics were not provided. Key concerns included increased computational requirements, privacy implications, and potential misidentification due to variations in real-world facial changes.

To support NGOs and law enforcement in reuniting lost children with their families, Sanju et al. (2025) [33] developed a facial recognition system that utilized photos from public uploads, users, and NGOs. Preprocessing involved face detection, normalization, image enhancement, and augmentation using OpenCV. A pre-trained CNN was used to generate 128-dimensional facial embeddings, and matching was performed using cosine and Euclidean distance metrics. While

formal accuracy metrics were not reported, the system supported both mobile and CCTV integration, along with multilingual functionality. System effectiveness could be challenged by low-quality images, privacy concerns, and the impact of facial aging over time.

To promote fairness and security in academic assessments, Arumugam (2025) [34] developed a deep learning-based smart invigilation system aimed at enhancing exam integrity. A custom dataset of 5,000 surveillance images was utilized, with preprocessing steps including frame extraction, face detection using SSD, gesture recognition through YOLOv5, and emotion detection via C3D networks. The integrated system achieved a high overall accuracy of 98.8%. Despite its effectiveness, large-scale deployment faced challenges due to the system's computational intensity and the need for specialized hardware in real-time classroom environments.

Focusing on real-time access control at airport terminals, Norhashim et al. (2025) [35] developed a facial recognition system integrated with Internet of Things (IoT) and cloud technologies. The system used a custom dataset comprising grayscale images of passengers, along with associated flight and passport details. Although the feature extraction model was not explicitly specified, embedded recognition was performed using Raspberry Pi devices. Performance metrics were not provided, but practical deployment faced challenges due to hardware constraints and limited design sophistication, affecting consistent real-time recognition in varied airport conditions.

To evaluate the effectiveness of deep learning techniques for face recognition, Nemavhola et al. (2025) [36] conducted a scoping review focused on CNN-based methods. The study analyzed benchmark datasets such as Labeled Faces in the Wild (LFW), YouTube Faces, and CelebA, encompassing both clean and occluded facial images. Preprocessing involved denoising, normalization, and data augmentation to enhance input consistency. Feature extraction utilized architectures like VGG16, FaceNet, and ResNet, with reported accuracy ranging from 95% to 99.85%. The review emphasized that over-reliance on clean datasets could impede the robustness of models in real-time and occlusion-prone environments.

To address growing concerns about data privacy in facial recognition systems, Laishram et al. (2025) [37] conducted a comprehensive survey highlighting privacy-preserving strategies such as image blurring, anonymization, and encryption. Rather than focusing on specific datasets, the

study analyzed deep learning approaches including CNNs, GANs, and Autoencoders for facial obfuscation, as well as security frameworks like Federated Learning and Homomorphic Encryption. While quantitative metrics were not reported, the review provided broad insights into privacy-focused architectures. One major concern was the lack of standardized benchmarks and experimental validation to assess these methods effectively in real-world scenarios.

Focusing on intelligent edge-based surveillance, Chahande et al. (2025) [38] developed an automated video surveillance system combining IoT sensors with AI capabilities. The system utilized real-time camera feeds activated through motion detection using PIR and IR sensors. Although specific AI models were not specified, face and object recognition were performed via edge AI techniques. Preprocessing included real-time video segmentation and sensor-triggered data acquisition. While the system improved operational efficiency and reduced false alarms, it lacked reported quantitative performance metrics. Noted concerns included maintaining privacy safeguards and the need for frequent system calibration to adapt to changing environmental conditions.

To develop a robust and integrated surveillance solution, Umasankaran et al. (2025) [39] introduced a system combining biometrics with AI-driven monitoring technologies. Datasets such as the NIST Mugshot and Fall Detection Dataset v2 were used, with preprocessing involving normalization, denoising, and scrambling to enhance privacy. Feature extraction was performed using models like FaceNet, YOLOv8, ResNet, and 3D-CNN, while classification was handled by KNN, SVM, and Random Forest algorithms. The system achieved 97.1% accuracy in face recognition and approximately 92% accuracy in fall detection using YOLOv8. Despite strong overall performance, challenges included reduced efficiency under occlusion or poor lighting and concerns about biometric data privacy.

Focusing on low-cost and hardware-integrated surveillance, Egbe et al. (2024) [40] developed a real-time campus security system combining CCTV cameras with PIR motion sensors and Arduino-based processing. Preprocessing included motion-triggered capture and sensor calibration. Anomaly detection was performed using K-means clustering, and the system utilized Arduino Uno boards connected to OV7670 cameras. The model achieved accuracy between

94.75% and 98.75%, with a precision of 96.77%. However, performance was constrained by the limited range of PIR sensors and the restricted memory capacity of the microcontroller.

With a focus on AI-driven real-time alerting, Farooq and Khan (2024) [41] developed *Safe-Campus*, a surveillance system leveraging CCTV footage from NED University. Preprocessing involved frame extraction, annotation through Roboflow, segmentation, and data augmentation. YOLOv8 was used for detecting people, vehicles, and stray dogs, and was integrated with a MERN stack to enable live alerts. The system achieved an accuracy of approximately 80.56%, showing strong performance particularly in stray dog identification. However, it faced challenges in detecting small objects and handling class imbalance within the dataset.

In an effort to automate rule enforcement through visual monitoring, Deepalakshmi and KR (2024) [42] developed a deep learning-based system for campus surveillance focusing on rule violation detection. A dataset comprising over 7,000 images was curated, capturing faces, ID cards, shoes, and grooming-related attributes. Preprocessing steps included annotation, resolution standardization, and database structuring. YOLOv7 was used for object detection, while a CNN handled face recognition. The system achieved 80.8% mAP, 90.4% accuracy, 76.7% recall, and 92% CNN testing accuracy. Output consistency was challenged by varying image quality and difficulty in detecting occluded or small-sized objects.

Targeting wireless deployment in administrative surveillance, Ehiagwina et al. (2024) [43] developed a reliable and secure wireless CCTV camera network for the main administration building at Federal Polytechnic Offa. The system processed real-time wireless CCTV footage, with preprocessing involving motion parameter tuning, NVR configuration, and comprehensive site surveys. Key technologies included motion detection, secure remote access, and H.265 video compression, which collectively enhanced data efficiency by 25–50%. However, system performance was occasionally affected by wireless signal interference and reliance on consistent network connectivity.

To enhance the performance of campus surveillance systems, Mishra and Jabin (2024) [44] proposed a model using the JMI-UCF dataset, which integrates CCTV footage with UCF-Crime data. Preprocessing included video clipping, brightness correction, resizing to 224×224 resolution, MPEG-4 encoding, and segment-based analysis. The study utilized pretrained I3D

and C3D models, along with a Temporal Self Attention (TSA) Vision Transformer Hybrid. The TSA model demonstrated strong performance with 96.26% average precision, 95.9% accuracy, a 79.5% F1 score, and an Equal Error Rate (EER) of 2.9%. However, challenges remained in terms of GPU dependency, poor generalization to low-resolution videos, and difficulties in handling low-quality contextual inputs.

To enhance the performance of security measures in academic libraries, Olorunyomi et al. (2025) [45] developed a deep learning-based CCTV surveillance system using annotated video data from the Federal Polytechnic Ile-Oluji Library. Preprocessing steps involved annotation, frame extraction, dataset splitting into training and testing sets, and adjustments for lighting variations and crowd density. The system employed YOLOv4 with CSPDarknet53 for real-time anomaly detection. It achieved 92% precision, 90% recall, a 91% F1 score, and a mean Average Precision (mAP) of 91.73%. Despite its strong performance, challenges included adherence to privacy regulations, inconsistent lighting, and occasional processing delays during real-time operations.

To enhance the performance of unsafe behavior detection in surveillance systems, Wu et al. (2025) [46] introduced a model utilizing the UCSD Anomaly Detection Dataset, UCF-QRNF, and SAIVT-QUT. Preprocessing steps included normalization, contrast correction, frame extraction, region masking, and object alignment to ensure high-quality inputs. Region-based features such as edges, textures, and pixel intensity were extracted using convolutional neural networks. The proposed Masked Region-Centric Disparity Detection (MRCDD) model achieved 96.15% precision, a 98.469% classification rate, and a false detection rate of just 0.045%. Despite its accuracy, the model's effectiveness could decline in low-light or highly dynamic environments and it requires substantial computational resources due to its complexity.

A comprehensive analysis of the reviewed literature reveals that the majority of surveillance and facial recognition systems rely heavily on deep learning techniques, particularly Convolutional Neural Networks (CNNs), for feature extraction and classification. Prominent models such as ResNet, VGG16, FaceNet, YOLO (v7/v8), and LightCNN were frequently utilized, either standalone or in hybrid combinations with classifiers like SVM, KNN, and LBPH. For face detection, techniques like Haar Cascade, MTCNN, and SSD were commonly employed. Preprocessing techniques including grayscale conversion, normalization, histogram equalization,

image resizing, and face alignment were vital in enhancing data quality. Datasets such as LFW, VGGFace2, ORL, Yale, SCface, and various custom-built datasets from academic institutions formed the basis for model training and testing. Some systems also incorporated IoT devices, PIR sensors, or Raspberry Pi to enable real-time, low-cost, or edge-based deployments.

Despite achieving high accuracies often ranging from 85% to over 99% across many studies, several limitations persist. These include sensitivity to lighting variations, occlusion, pose changes, and image quality, as well as computational limitations in real-time processing. A few works addressed additional factors like privacy, ethics, and the legal implications of widespread facial recognition use. Furthermore, while certain models achieved high precision and robustness in controlled environments, their effectiveness often declined in dynamic or uncontrolled real-world scenarios. Overall, the literature highlights the growing potential of AI-integrated CCTV surveillance systems while also emphasizing the need for improved generalizability, real-time adaptability, and ethical considerations in their deployment.

2.2 Summary of the literature survey

To provide a structured overview of the existing research in the domain of CCTV-based facial recognition systems, the following table summarizes key aspects of the reviewed literature. Each entry highlights the paper title, the specific problem addressed, the approach or methodology adopted by the authors, and the outcomes or results reported. This comparative summary aids in identifying common techniques, recurring challenges, and performance benchmarks across various implementations.

Tabel 2.1: Survey Summary

Ref. No	Problem Addressed	Authors' Approach / Method	Results
[1]	Tracking and logging individual entries/exits	Face registration + TensorCam with TensorFlow, Keras, Caffe2	Efficient face tracking; accuracy not quantified; limited to pre-registered users
[2]	Real-time attendance via CCTV	Face detection + LBPH; grayscale conversion and normalization	99.69% accuracy; reliable in controlled environments
[3]	Efficient attendance	Haar Cascade for detection + fine-	High accuracy; short training

	with limited training	tuned ResNet50/VGG16/InceptionV3	time; specific metrics not reported
[4]	Secure and scalable attendance marking	Deep CNN for recognition after face alignment	High accuracy claimed; exact value not provided
[5]	Enhancing campus safety with AI and IoT integration	YOLO for object detection; IoT for alerts and data flow	Improved response time; effectiveness dependent on network stability
[6]	Detecting violent interactions in campus surveillance	Deep learning + temporal analysis on human behavior	High accuracy in identifying violent events; real-time alerts supported
[7]	Summarizing long CCTV videos into key activity events	CNN with activity attention modeling on CAD dataset	Effectively summarizes academic activity videos; accurate filtering of key events
[8]	Improving face/mask detection during surveillance	HOG, SIFT, DNN, CNN; integrated using OpenCV and Dlib	Accurate face recognition; handles unusual face/mask detection
[9]	Recognizing individuals in real-time CCTV streams	PCA for features; CNN, KNN, RF for classification	CNN achieved 97.5% accuracy; PCA slower and less accurate
[10]	Real-time small-scale attendance system	LBPH with OpenCV; grayscale conversion + face detection	High accuracy for small datasets; real-time functionality in controlled setup
[11]	Real-time face recognition with limited hardware	CNN via Dlib; face alignment and cropping; compared with HoG	CNN achieved 98.3% accuracy; outperformed HoG; real-time on Raspberry Pi
[12]	Affordable IoT-enabled face recognition from CCTV	MTCNN + LBPH classifier; brightness & illumination normalization	73.41% accuracy; training ~309 sec; real-time alerts enabled
[13]	Face recognition using landmark-based morphometrics	HOG + Dlib landmark detection; Likelihood Ratio method	85.75% TPR, 25% FPR, cllr = 0.26; interpretable, forensic-friendly

[14]	Robust recognition in CCTV with occlusion/lighting	Histogram equalization + GoogleNet CNN + attention aggregation	98.55% accuracy on YouTube Face Dataset; 99.12% (no occlusion), 98.87% (occluded)
[15]	Face recognition under mask-wearing conditions	FaceMaskNet-21 (CNN + quadruplet loss); LFW, RMFRD datasets	88.92% overall accuracy; 100% for single-face video frames
[16]	Variations in face angles, lighting, and accessories	CNN + Haar, PCA, ICA, SVM; 400 images (students)	CNN+Haar: 98.95%; CNN: 97.83%; Haar: 78.95%; SVM: 93.95%
[17]	Ethical, legal, and privacy issues in facial recognition	Theoretical analysis; case studies (Clearview AI, AFR Locate)	No metrics; proposed legal oversight, transparency, and public consent
[18]	Qualitative evaluation of analog CCTV systems	Manual observation and interviews; no image data	Reduction in theft and vandalism; no smart analytics or digital features
[19]	Improving recognition in low-res CCTV frames	Super-resolution (U-Net, EDSR) + FR (ResNet50, MobileNetV2)	U-Net + ResNet50: 85% accuracy, 87% precision, 114 ms/image
[20]	Identifying children/strangers through face and object detection	YOLOv8 + CNN/LBPH/SVM; face classification by age/mask	YOLO mAP@0.5: 0.995; CNN most accurate; LBPH fastest
[21]	Comparing recognition methods for attendance	Reviewed 30 papers; CNN, LBPH, HOG, Eigenface, SVM	CNN: up to 99% accuracy; LBPH: 92%; CNN > others overall
[22]	Robust facial recognition under pose/illumination changes	MediaPipe Face Mesh + BlazeFace + DNN	Accuracy: 94.23%; robust to illumination and background variation
[23]	Enhancing privacy using lensless camera data	FlatCam dataset; DCT transformation + VGG16, VGG-ATT	VGG-ATT + DCT: 87.71% accuracy; lower performance on raw sensor input

[24]	Security surveillance using real-time facial recognition from CCTV	Preprocessing with face detection and normalization; Deep CNNs + Haar Cascade	High accuracy in real-time face recognition; effective surveillance tracking
[25]	Recognizing faces in low-resolution CCTV footage	LightCNN, MobileNet-V2, ArcFace on LFW and SCface datasets	Achieved high accuracy in low-res conditions; ArcFace enhanced robustness
[26]	Classroom attendance using CCTV-based recognition	Used FaceNet for embeddings, KNN classifier	Achieved 92.85% accuracy in real-time attendance using CCTV
[27]	Suspect identification in security surveillance	HBA for feature selection + KNN on ORL and Yale datasets	Accuracy: 94.4% (ORL), 92.3% (Yale); promising for law enforcement use
[28]	Real-time detection and identification of criminal faces	Face detection via Haar, HOG for features, classified with SVM	Alerts on matches; accuracy metrics not detailed but system worked live
[29]	Difficulty identifying criminals in CCTV under occlusion	Face detection with MTCNN; recognition with ResNet	Achieved 92.56% accuracy; works well in occluded CCTV scenes
[30]	Face identification in law enforcement tasks	Haar Cascade + LBPH algorithm on law enforcement image set	High accuracy; supports field application with enhancements needed
[31]	Facial recognition in unconstrained, real-world settings	Modified VGGFace16 with GAP & PReLU; transfer learning	Achieved 99.9% accuracy, precision, recall, and F1-score on LFW, VGGFace2
[32]	Identifying missing children through facial recognition	CNN with transfer learning; real-world and law enforcement images	High precision and recall; exact values not provided
[33]	Locating missing children via face recognition	128-d embeddings using pre-trained CNNs; cosine/Euclidean comparison	Functional system; accuracy not quantified; works with mobile and CCTV
[34]	Detecting cheating	SSD for face, YOLOv5 for gesture,	98.8% overall accuracy; 99.2%

	behavior during exams	C3D for emotion analysis	for non-cheating detection
[35]	Real-time face recognition at airport checkpoints	Real-time capture via Raspberry Pi; unspecified recognition model	Functional testing completed; no specific accuracy metrics reported
[36]	Summary of face recognition using deep learning	Reviewed VGG16, FaceNet, AlexNet, MTCNN, YOLO, etc.	Reported accuracies from 95% to 99.85%; gaps in occlusion handling noted
[37]	Privacy risks in facial recognition systems	Survey of GANs, autoencoders, encryption, federated learning	No performance metrics; presents solutions for privacy-preserving recognition
[38]	Real-time surveillance without human monitoring	PIR/IR sensors + AI-based object and face recognition on live feed	Improved efficiency and reduced false alarms; metrics not specified
[39]	Improving recognition accuracy and fall detection	FaceNet, ResNet, YOLOv8, 3D-CNN; fusion of multiple classifiers	KNN: 97.1% (face); YOLOv8: ~92% (fall); high accuracy across tasks
[40]	Low-cost real-time surveillance with anomaly detection	PIR sensors + Arduino + K-means clustering on motion events	Accuracy: 94.75%–98.75%; precision: up to 96.77%; scalable solution
[41]	Object/person detection in campus CCTV	YOLOv8 with MERN stack; Roboflow for annotation	~80.56% accuracy; real-time alerting for specific classes
[42]	Identifying violations (ID, shoes) in students	YOLOv7 for object detection; CNN for face recognition	YOLOv7: 90.4% accuracy; CNN testing accuracy: 92%
[43]	Securing wireless surveillance with optimal streaming	H.265 compression + NVR + QoS + motion triggers	Real-time access; 25–50% better data efficiency
[44]	Detecting anomalies in campus video using temporal modeling	I3D, C3D + Vision Transformer-based Temporal Self Attention (TSA)	TSA achieved 95.9% accuracy, 96.26% AP, F1: 79.5%, EER: 2.9%
[45]	Real-time object	YOLOv4 with CSPDarknet53	Precision: 92%, Recall: 90%, F1-

	detection and anomaly detection in libraries	backbone on annotated CCTV data	score: 91%, mAP: 91.73%
[46]	Detecting unsafe behavior in crowds using visual cues	CNN + MRCDD (Masked Region-Centric Disparity Detection)	Precision: 96.15%, Classification Rate: 98.47%, False Detection Rate: 0.045%

The Table 2.1 reveals a dominant reliance on deep learning and computer vision models for facial recognition and surveillance tasks across academic and institutional settings. Convolutional Neural Networks (CNNs) were the most commonly used architecture, employed in works such as [3], [5], [16], [18], [23], [24], [35], and [36], indicating their adaptability in both real-time and complex recognition scenarios. Local Binary Pattern Histogram (LBPH) was another frequently adopted method for feature extraction and recognition in studies like [2], [3], [27], and [33] due to its computational simplicity and suitability for embedded devices like Raspberry Pi. Haar Cascade classifiers were widely used for face detection across several works including [3], [4], [30], [33], and [31]. YOLO (You Only Look Once), particularly versions YOLOv7 and YOLOv8, was utilized for object and behavior detection in papers [15], [20], [44], [45], and [46], owing to its speed and real-time detection capabilities.

Several hybrid approaches emerged, combining detection models with classifiers such as SVM [4], [15], [43], and KNN [30], [29], as well as using attention mechanisms or feature fusion strategies to improve performance. Works such as [48] employed Temporal Self-Attention Transformers, while [50] introduced a Masked Region-Centric Disparity Detection (MRCDD) model for unsafe behavior. Notable datasets included LFW [4], [13], [28], VGGFace2 [34], ORL and Yale [30], and many custom-built datasets tailored to campus and security applications. Accuracy scores in most systems ranged between 85% to over 99%, with models such as enhanced VGGFace16 [34] and FaceMaskNet-21 [17] achieving peak performance in constrained conditions. Challenges consistently reported include performance degradation under low lighting, occlusion, pose variation, and the need for computational resources in real-time applications.

2.3 Research Gap Identified

Although significant progress has been made in AI-based surveillance, several gaps still exist in current campus security systems. Many existing solutions focus on a single surveillance task—such as either face recognition or object detection—without integrating multiple capabilities like face recognition, vehicle detection, and number plate extraction into a unified real-time system. This lack of integration reduces their effectiveness in complex and dynamic environments such as educational campuses.

Another major limitation is the low accuracy of number plate recognition under challenging conditions such as low lighting, motion blur, and partial occlusion. Many existing ANPR solutions struggle to extract clear alphanumeric information when plates are angled, moving quickly, or partially obstructed. Similarly, traditional face recognition systems often fail under variations in illumination, orientation, or camera quality, making them less reliable in real-world CCTV deployments.

Most conventional surveillance systems also lack automation and rely heavily on manual monitoring. They do not provide real-time analysis of video streams and cannot automatically detect people, vehicles, or known individuals. This results in delayed responses, missed events, and heavy reliance on human operators. Additionally, existing systems often do not maintain automated logs for face recognition or number plate detection, making post-incident investigations time-consuming.

Scalability presents another challenge: many existing models are developed as standalone applications and cannot be easily integrated into existing CCTV infrastructure. Performance inconsistencies across different hardware setups and environmental conditions further limit the usability of current solutions.

Addressing these gaps—through a system that integrates real-time person detection, vehicle detection, face recognition, and number plate extraction into a single automated framework—is essential for developing a more intelligent, accurate, and reliable surveillance solution suitable for modern campus environments.

.CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

3.1 Software Requirements

The AI-based campus surveillance system relies on a set of essential software components that enable real-time video processing, object detection, face recognition, and number plate extraction. The system can operate on commonly used operating systems such as Windows 10/11 or Ubuntu 20.04, both of which provide strong compatibility with computer vision and deep learning libraries. The core implementation is done in Python, which serves as the primary programming language due to its extensive AI and image-processing ecosystem.

The system uses OpenCV for video capture, frame preprocessing, face detection with Haar Cascade classifiers, and other image-processing operations. PyTorch or Ultralytics YOLO is required to run the YOLOv8 model used for detecting people, vehicles, and number plates. For face recognition, the system uses the LBPH (Local Binary Patterns Histogram) algorithm, implemented through OpenCV's face recognition module. Number plate text extraction is handled through OCR tools, such as EasyOCR or Tesseract, which process the detected plate region to extract alphanumeric characters.

Additional Python libraries—including NumPy, Pandas, and Matplotlib—support data handling and optional logging functionalities. A simple graphical or console-based interface is used to display real-time detection results, eliminating the need for web frameworks or complex dashboards. If logging is enabled, lightweight file-based storage (CSV or text files) or a simple database such as SQLite may be used.

Overall, this software stack provides a practical and efficient foundation for deploying the real-time surveillance system while ensuring compatibility, ease of installation, and smooth integration with existing CCTV infrastructure.

3.2 Hardware Requirements

The AI-based campus surveillance system requires a reliable hardware setup capable of handling real-time video processing and running deep learning models efficiently. The system can operate using the existing CCTV infrastructure, provided that the cameras support standard IP streaming and deliver clear video suitable for detection and recognition tasks. While higher resolution cameras improve accuracy, the system can function effectively with commonly available 720p or 1080p CCTV feeds.

For processing, a standard workstation or laptop with a multi-core CPU—such as an Intel Core i5/i7 or an AMD Ryzen 5/7—is sufficient for running the YOLOv8 model, face detection using Haar Cascade, face recognition using LBPH, and OCR operations. Although a dedicated GPU can significantly accelerate object detection, the system can still operate on a CPU-based machine with moderate performance expectations. A minimum of 8–16 GB RAM is recommended to ensure smooth execution of real-time video analysis.

In terms of storage, an SSD with at least 256–512 GB capacity is preferred for faster data access and model loading, along with optional HDD storage if long-term logs or large video segments are to be maintained. Stable network connectivity is required to receive live camera feeds, with a standard 1 Gbps LAN or stable Wi-Fi connection being adequate for most deployments. This hardware configuration ensures that the system can perform detection and recognition reliably while remaining practical, cost-effective, and suitable for deployment within typical campus environments.

3.3 Functional Requirements

The AI-based campus surveillance system is designed to automate monitoring by processing live CCTV footage and performing intelligent detection and recognition tasks in real time. The system must continuously capture video streams from all connected CCTV cameras and make them available for analysis without interruption. It should preprocess incoming frames to ensure they are suitable for detection models and maintain steady performance under varying lighting or environmental conditions.

The object detection module must accurately identify people, vehicles, and number plates using the YOLOv8 model. The system should also detect faces using Haar Cascade classifiers and recognize individuals using the LBPH algorithm by comparing them with stored facial datasets. For vehicles, the number plate region must be correctly localized and passed through OCR to extract readable alphanumeric text.

Once detection and recognition are completed, the system must display the annotated results—such as bounding boxes, recognized face labels, and extracted number plate text—on the user interface in real time. The system should also support optional data logging, allowing recognized faces, number plates, and related timestamps to be stored in the database for future reference or post-event verification.

The system must operate continuously, provide clear visual outputs to security personnel, and maintain reliable performance throughout its operation. Its modular design should ensure easy scalability and integration with existing CCTV infrastructure.

3.4 Non-functional Requirements

The non-functional requirements define the quality attributes that the AI-based campus surveillance system must satisfy to ensure dependable and efficient operation. The system should be capable of processing real-time CCTV video streams smoothly, maintaining low latency so that detections and recognitions appear on the user interface without noticeable delay. It must operate reliably for extended durations, supporting continuous 24/7 surveillance without frequent crashes or interruptions.

To ensure meaningful monitoring, the detection and recognition modules—including YOLOv8, Haar Cascade, LBPH, and OCR—should maintain consistent performance across varying lighting and environmental conditions. Although absolute accuracy cannot be guaranteed in all scenarios, the system should aim to provide stable and dependable results suitable for campus-level surveillance needs.

Scalability is important; the architecture should allow additional cameras or processing modules to be integrated in the future without significantly affecting performance. The user interface must

remain simple, clear, and easy for security staff to understand, requiring minimal training. Compatibility with standard CCTV and IP camera feeds is also essential so that the system can be deployed without major hardware changes.

These non-functional requirements ensure that the surveillance system remains usable, reliable, and adaptable for long-term campus security needs.

CHAPTER-4

PROJECT DESIGN

4.1 System Design

The AI-based campus surveillance system is designed as an intelligent and fully automated framework capable of analyzing live CCTV footage in real time. The system begins by acquiring continuous video input from surveillance cameras installed across the campus. This raw video stream undergoes preprocessing, where each frame is extracted, resized, normalized, and prepared to ensure that detection models receive consistent and high-quality input data. In the case of face recognition, the frames are further converted into grayscale and filtered to reduce noise, improving the reliability of recognition results.

Once preprocessing is complete, the frames are passed through several integrated detection components that operate simultaneously. YOLOv8 serves as the core model for both person and vehicle detection, identifying individuals and vehicles present in the camera's view with high accuracy. Alongside this, facial regions within the video frames are detected using Haar Cascade, after which the LBPH algorithm performs face recognition to distinguish between known and unknown individuals. The system also incorporates automatic number plate recognition, where vehicle plates detected by YOLOv8 are isolated and processed through an OCR engine to extract their alphanumeric details.

All processed outputs—including detected persons, recognized faces, identified vehicles, and extracted number plates—are presented through a unified interface that allows security personnel to monitor activity in real time. The system can optionally store these outputs, creating digital records for later verification or investigation. Through this integrated approach, the proposed design ensures accurate, scalable, and real-time surveillance, making it suitable for deployment within modern campus environments where automation and rapid response are essential.

4.1.1 System Architecture

The system architecture integrates real-time video processing with intelligent detection using modern computer vision and deep learning techniques. It follows a modular design that ensures smooth data flow from CCTV input to the final output on the user interface. Each component—such as object detection, face recognition, and number plate extraction—performs a specific task within the pipeline.

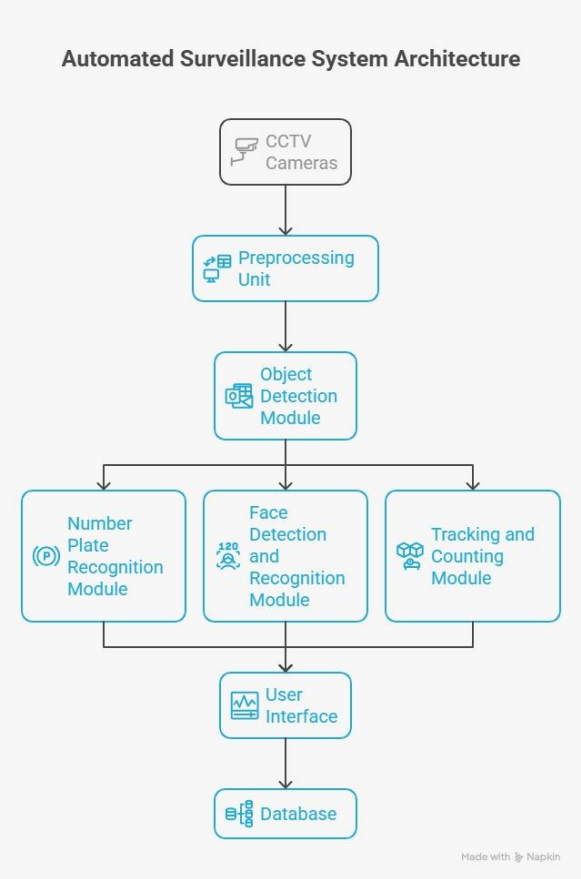

Fig 4.1 Automated Surveillance System Architecture for Campus Security

The proposed AI-based surveillance system is composed of several interconnected modules, each designed to perform a specific task within the overall architecture. These modules work together to enable real-time detection, recognition, and extraction of useful information from live CCTV feeds.

CCTV Input Module : This module represents the video acquisition layer, where CCTV cameras installed across the campus continuously capture live footage. The video stream is forwarded to the processing system, forming the raw input for the detection and recognition pipeline. The quality of input frames directly influences detection performance, making stable camera placement and adequate lighting important.

Preprocessing Module : In this stage, the incoming video is divided into individual frames, which are then resized, normalized, and enhanced to prepare them for further analysis. Preprocessing may include grayscale conversion, noise reduction, and contrast adjustments, especially for tasks like face recognition and OCR. These steps ensure that the input data remains consistent and suitable for the different deep learning models used in the system.

Object Detection Module (YOLOv8) : This module is responsible for identifying people and vehicles in the video frames. It uses the YOLOv8 deep learning model, which performs fast and accurate object detection by generating bounding boxes and class labels for each detected object. The module plays a central role in monitoring campus activity, as it automatically identifies the presence of individuals and vehicles without requiring manual observation.

Face Detection and Recognition Module (Haar Cascade + LBPH) : This component first detects facial regions within the frames using Haar Cascade classifiers. Once a face is detected, the LBPH (Local Binary Patterns Histogram) algorithm is used to perform face recognition. LBPH compares the detected face with stored face data to determine whether the person is authorized or unknown. This module aids in identifying individuals entering sensitive or restricted areas and enhances overall security.

Number Plate Detection and OCR Module : This module extracts vehicle registration numbers from video frames. YOLOv8 is used to detect and localize number plates, after which the detected plate region is cropped and passed to an Optical Character Recognition (OCR) engine. The OCR system reads and extracts the alphanumeric characters from the plate, enabling automatic logging and verification of vehicles entering or leaving the campus.

User Interface Module : The system includes an interface that displays the live video feed along with annotated outputs, such as detected people, recognized faces, vehicle detections, and

extracted number plate information. The interface provides real-time visualization to assist security personnel in monitoring campus activity efficiently. It also serves as a central display for system results without requiring specialized technical knowledge.

Database and Storage Module : This module stores recognized face data, extracted number plates, timestamps, and detection logs for later review or analysis. It supports post-incident investigation, verification, and long-term security monitoring. Although optional, this component enhances the system's utility by providing historical data for campus security administration.

4.1.2 Module Design

he proposed campus surveillance system leverages artificial intelligence to convert traditional CCTV setups into smart, automated monitoring systems. By integrating modules for object detection, face recognition, and number plate extraction, the system enhances campus security through real-time and automated analysis of CCTV footage.

The system is divided into the following modules for structured development, testing, and maintenance:

Video Capture Module: This module interfaces with CCTV or external cameras and continuously fetches real-time video streams for processing.

Preprocessing Module: It handles frame extraction, resizing, normalization, and image enhancement to ensure that the input data is suitable for accurate detection and recognition.

People and Vehicle Detection Module: Using the YOLOv8 deep learning model, this module detects and classifies people and vehicles present in each video frame with high accuracy.

Number Plate Detection and OCR Module: YOLOv8 is used to localize vehicle number plates, after which OCR techniques (such as Tesseract or EasyOCR) extract the alphanumeric text for identification and logging.

Face Detection and Recognition Module: Faces are detected using Haar Cascade classifiers, and identity recognition is performed using the LBPH algorithm by comparing detected faces with stored images in the database.

Data Management Module : This module stores recognized face details, extracted number plate information, and optional detection logs for future reference and verification.

User Interface Module : A simple display interface shows the live video feed along with real-time annotations of detected people, vehicles, faces, and number plates, allowing security personnel to easily monitor campus activity.

4.2 Detailed Design

4.2.1 Class Diagram

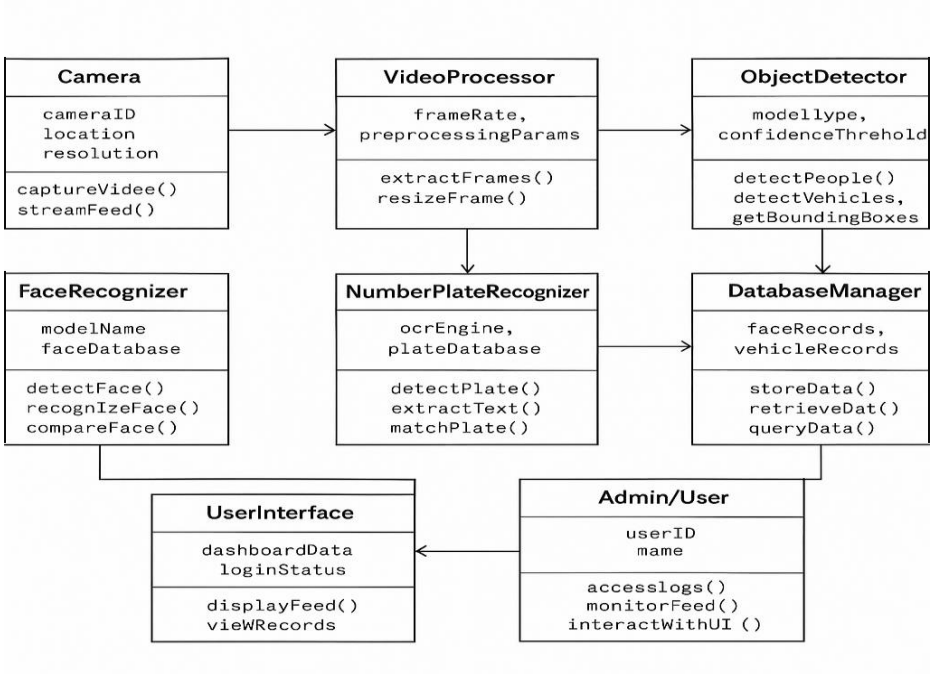

Fig 4.2 Class Diagram for Campus Security

4.2.2 Activity Diagram

The activity diagram illustrates the dynamic workflow of how CCTV footage is processed for person detection, vehicle tracking, face recognition, and number plate reading. It shows the sequence of actions from video input to alert generation and storage.

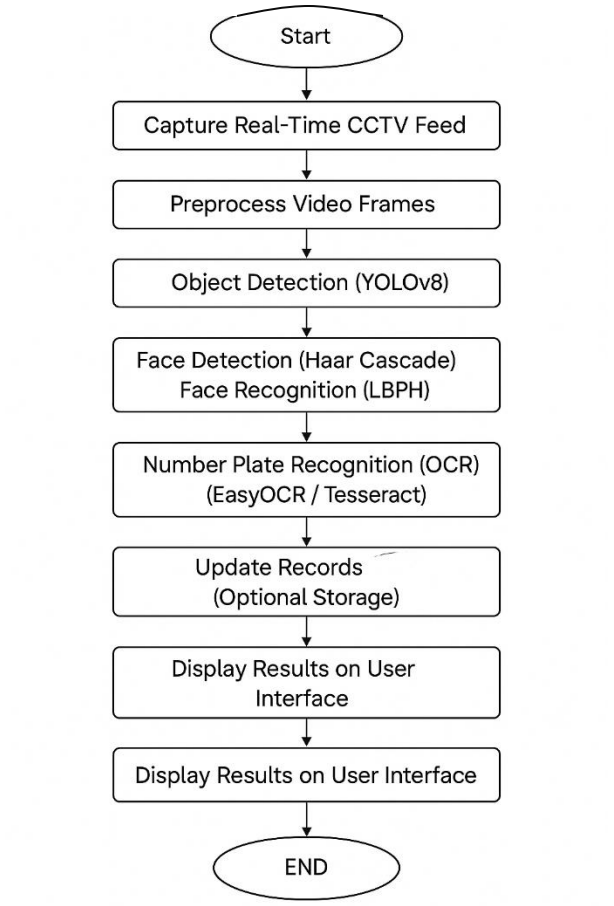

Fig 4.3 Activity Diagram of the Proposed Campus Surveillance System

The workflow of the proposed AI-based surveillance system describes the sequence of operations performed from the moment live CCTV footage is captured until the final processed output is displayed. The process begins with continuous video acquisition from CCTV cameras positioned across the campus. These video streams are converted into individual frames, which are then preprocessed through resizing, normalization, and noise reduction to ensure optimal quality for analysis.

The preprocessed frames are next passed to the object detection module, where the YOLOv8 model identifies people, vehicles, and vehicle number plates. Simultaneously, the face detection module uses Haar Cascade classifiers to locate human faces within the frames. Once a face is extracted, the LBPH algorithm performs recognition by comparing the detected face with stored images in the database. For vehicles, the detected number plate region is isolated and processed using an OCR engine, such as EasyOCR or Tesseract, to extract the alphanumeric characters.

After detection and recognition, the processed results—including detected individuals, recognized faces, and number plate text—are compiled and displayed in real time through the user interface. Optional logging allows the system to store detected outputs for later verification. This structured workflow enables the system to operate continuously, providing automated and efficient surveillance without requiring manual monitoring.

4.2.3 Use Case Diagram

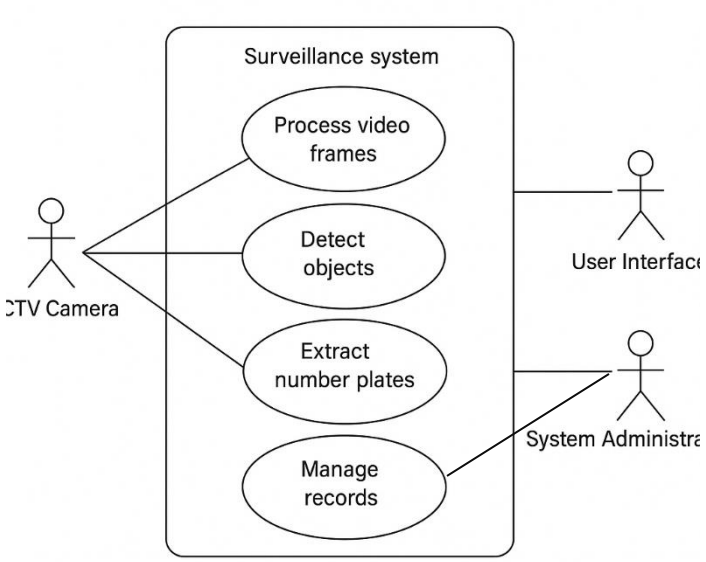

Fig 4.4 Use Case Diagram of the Proposed Campus Surveillance System

4.2.4 Scenarios

This section presents practical use-case scenarios demonstrating how the proposed surveillance system performs in real-world campus environments. These scenarios illustrate the system's automated responses and real-time functionalities in various security-related situations.

Scenario 1: Vehicle Entering the Campus at Night

When a vehicle enters the campus during late hours, the CCTV feed is automatically processed by the system. YOLOv8 detects the vehicle, and the number plate region is extracted and passed to the OCR module to read its alphanumeric characters. The system records the detected vehicle and stores the number plate information in the database for verification. Security personnel can view the recorded details through the user interface the next morning to confirm whether the entry was authorized.

Scenario 2: Monitoring a High-Activity Zone

During events or busy hours, the surveillance system continuously analyzes CCTV footage in crowded areas such as corridors or parking zones. YOLOv8 detects people and vehicles in real time, and the interface displays all detections with bounding boxes. Although the system does not estimate crowd size or generate alerts, it provides clear visual monitoring and maintains logs of detected individuals and vehicles for later review.

Scenario 3: Recognizing Known Personnel

When a registered faculty member or staff passes through a monitored area, the system detects their face using Haar Cascade and performs recognition using the LBPH algorithm. If the face matches a stored entry in the database, the system logs the recognition event with a timestamp. Since the individual is authorized, the system simply records the information without further action.

Scenario 4: Unknown Person Detected Near a Restricted Zone

If a person moves through a sensitive area such as laboratory premises or staff-only sections, the system detects their presence and attempts face recognition using LBPH. If the face does not

match any stored record, the system logs the unidentified person's appearance. Security staff reviewing the logs can inspect the footage to determine whether further investigation is required.

Scenario 5: Night-Time Surveillance and Logging

During low-light or night-time conditions, the system continues operating with IR-supported CCTV cameras. It detects people, vehicles, and faces as usual, and records these detections in the database. Although the system does not classify nighttime behavior or raise alerts, it ensures uninterrupted monitoring and provides a complete log of activity for later assessment by campus security.

CHAPTER- 5

EXPERIMENTAL SETUP

The experimental setup describes the sequence of operations and configurations used to implement and evaluate the AI-based campus surveillance system. The system is implemented using Python and tested on live video streams and recorded video files to validate detection, recognition, and OCR performance.

5.1 Data Collection

Data collection for the proposed campus surveillance system is performed using live video streams and image data rather than structured datasets. The primary data source consists of real-time video feeds captured from a webcam or CCTV camera installed in the campus environment. These video streams provide continuous frames containing humans, vehicles, faces, and vehicle number plates under real-world conditions.

For the face recognition module, a separate face image dataset is created by capturing multiple facial images of known individuals. Images are collected under different lighting conditions and facial orientations to improve recognition performance. Each individual's images are labeled and organized into separate folders to support LBPH face recognition training.

For vehicle and number plate detection, the system uses live video frames where vehicles naturally appear in the camera view. No manual annotation is required, as the YOLOv8 model uses pre-trained weights to detect humans, vehicles, and number plate regions directly from the video stream.

5.2 Pre-processing

The input video stream obtained from a webcam or CCTV camera is first converted into individual frames. Each frame is resized to a fixed resolution to ensure faster processing and consistent input to the detection models. For face recognition tasks, frames are converted to

grayscale to improve the effectiveness of Haar Cascade face detection and LBPH recognition. Basic noise reduction and normalization techniques are applied to enhance image quality and improve OCR accuracy for number plate recognition.

5.3 Feature Extraction

Feature extraction in this project is performed using a combination of deep learning–based and traditional computer vision techniques. The following features are considered:

- Deep Visual Features (YOLOv8):
  YOLOv8 automatically extracts deep convolutional features from video frames to detect and classify humans, vehicles, and number plate regions. These features capture object shape, size, edges, and spatial patterns.
- Facial Region Features (Haar Cascade):
  Haar Cascade uses Haar-like features based on edge, line, and intensity differences to detect facial regions within video frames.
- Texture-Based Facial Features (LBPH):
  The LBPH algorithm extracts Local Binary Pattern histograms from detected face regions. These texture-based features represent local pixel intensity variations and are robust to lighting changes, making them suitable for face recognition.
- Character-Level Features (OCR):
  For number plate recognition, OCR engines extract character-level features such as contours, strokes, and pixel patterns from the cropped number plate image to identify alphanumeric text.

These extracted features are used to identify objects, recognize faces, and read number plate numbers accurately in real time.

5.4 Model Selection

Model selection in this project is based on accuracy, computational efficiency, and suitability for real-time campus surveillance. YOLOv8 is selected for object and number plate detection due to its high detection accuracy and fast inference speed on standard hardware. For face detection,

Haar Cascade classifiers are chosen because of their lightweight nature and reliable performance on grayscale images. LBPH (Local Binary Patterns Histogram) is selected for face recognition saved as a model file (face_model.yml). as it performs well with limited training data and is robust to lighting variations. For number plate text extraction, EasyOCR or Tesseract OCR is used due to their effectiveness in recognizing alphanumeric characters from real-world images.

5.5 Training Phase

The training phase is applied primarily to the face recognition component of the system. A dataset of face images for known individuals is collected and organized class-wise. Each face image is converted to grayscale and resized to a fixed dimension. The LBPH face recognizer is trained using these processed face images to generate a trained face recognition model, which is YOLOv8 uses a pre-trained model (.pt file) for object and number plate detection, and no custom training is performed in this project. Haar Cascade classifiers are also pre-trained and used directly without additional training.

5.6 Model Loading and Initialization

After the training phase, all the required trained and pre-trained models are loaded into the system to prepare it for real-time execution. The trained LBPH face recognition model generated during training is loaded along with the pre-trained YOLOv8 model used for object and number plate detection. In addition, the Haar Cascade classifier for face detection and the OCR engine (EasyOCR or Tesseract) for number plate text extraction are initialized. This step ensures that all models and components are ready for inference on live or recorded video data.

5.7 Inference and Testing Phase

In the inference phase, the loaded models are applied to new, unseen video frames obtained from the live camera feed or video files. YOLOv8 processes each frame to detect humans, vehicles, and number plates. Simultaneously, the Haar Cascade classifier detects facial regions, and the LBPH model predicts the identity of detected faces by comparing them with the trained face data. The detected number plate regions are passed to the OCR module, which extracts the

alphanumeric characters. This phase evaluates how effectively the system performs on real-time surveillance data.

5.8 Post-processing

Post-processing is carried out to refine the raw outputs obtained from the detection and recognition models. This includes filtering detections based on confidence thresholds, formatting the output labels such as person names and number plate numbers, and handling cases where faces are not recognized by labeling them as “Unknown.” These steps improve the clarity, consistency, and usability of the system’s output.

5.9 Annotation and Visualization

In this stage, the processed detection and recognition results are visually presented on the video frames. Bounding boxes are drawn around detected humans, vehicles, faces, and number plates. Recognized face names and extracted number plate numbers are displayed alongside the corresponding objects. The annotated frames are shown in real time, enabling effective visual monitoring for surveillance purposes.

5.10 Logging and Result Storage

The system optionally records the detection and recognition results for further analysis. Information such as recognized face names, detected number plate numbers, object types, and timestamps is stored in CSV files. These logs provide a record of surveillance activity and can be used for verification, analysis, and documentation.

CHAPTER 6

RESULTS

6.1 Performance Analysis

The performance analysis of the proposed campus surveillance system was carried out to evaluate the effectiveness and reliability of the models used for object detection, face recognition, and number plate recognition. The evaluation focuses on assessing how accurately the system detects humans and vehicles, identifies known individuals, and extracts vehicle number plate information from live video streams. Since the system is designed for real-time surveillance, a practical sample-based evaluation approach was adopted by testing the system on live camera feeds and recorded video data. The performance of each model was analyzed individually to understand its contribution to the overall system performance. The following sections present the accuracy analysis of the object detection, face recognition, and number plate recognition modules used in the system.

6.1.1 Object Detection Accuracy (YOLOv8)

The object detection performance of the system was evaluated using the YOLOv8 model by testing it on live video streams and recorded campus videos. The model was assessed by manually comparing the detected humans and vehicles against the actual objects present in selected video frames. During experimentation, YOLOv8 was able to correctly detect most humans and vehicles under normal lighting and camera angles. Based on sample-based evaluation, the object detection accuracy was observed to be approximately 94%, with minor detection failures occurring in cases of occlusion or rapid movement. Overall, the model demonstrated reliable real-time object detection suitable for campus surveillance.

6.1.2 Face Recognition Accuracy (LBPH)

The face recognition accuracy was evaluated using the LBPH (Local Binary Patterns Histogram) algorithm by testing it with known individuals whose face images were included in the training dataset. The system was tested under varying lighting conditions and different facial orientations.

The recognized face labels were compared with the actual identities to calculate accuracy. Experimental results showed that the LBPH-based face recognition module achieved an accuracy of approximately 88% for known individuals. Recognition errors mainly occurred due to poor lighting, partial face visibility, or changes in facial appearance. Despite these limitations, the model performed effectively for identifying authorized individuals in controlled campus environments.

6.1.3 Number Plate Recognition Accuracy (OCR)

The number plate recognition accuracy was evaluated by testing the OCR module on detected vehicle number plates obtained from live video feeds. The extracted plate text was manually verified against the actual vehicle registration numbers. The OCR system was able to accurately recognize alphanumeric characters when the number plate was clearly visible and properly illuminated. Based on experimental observations, the number plate recognition accuracy was found to be approximately 82%. Errors were primarily due to low-resolution plates, motion blur, or non-standard fonts. However, the OCR module provided satisfactory performance for practical surveillance and logging purposes.

Tabel 6.1: Comparative Analysis of Model Accuracy

Model / Module	Purpose	Evaluation Method	Observed Accuracy(%)
YOLOv8	Human & Vehicle Detection	Manual comparison of detected objects vs actual objects in video frames	94%
LBPH	Face Recognition	Correctly recognized known faces vs total recognition attempts	88%
OCR (EasyOCR / Tesseract)	Number Plate Recognition	Correct plate text extraction vs actual plate numbers	82%

Table 6.1 presents a comparative analysis of the accuracy achieved by different models used in the proposed campus surveillance system. The YOLOv8 model demonstrates high detection accuracy for humans and vehicles, indicating its suitability for real-time object detection tasks. The LBPH face recognition model shows reliable performance in identifying known individuals, with accuracy influenced by lighting conditions and facial visibility. The OCR module achieves moderate accuracy in extracting vehicle number plate text, with errors mainly occurring due to motion blur and low-resolution plates. Overall, the results indicate that the combination of YOLOv8, LBPH, and OCR provides an effective and balanced solution for real-time campus surveillance applications.

6.2 Comparison with Existing Object Detection Models

To justify the selection of YOLOv8 for the proposed campus surveillance system, its object detection accuracy was compared with other commonly used object detection models reported in earlier surveillance systems, such as Haar Cascade, SSD, Faster R-CNN, YOLOv3, and YOLOv5. These models have been widely used for detecting humans and vehicles in CCTV-based applications.

Traditional models like Haar Cascade rely on handcrafted features and perform poorly in complex environments, resulting in lower accuracy, especially under varying lighting conditions and occlusions. SSD provides moderate accuracy but struggles with detecting smaller or partially occluded objects. Faster R-CNN achieves higher accuracy but is computationally expensive and not suitable for real-time surveillance. YOLOv3 and YOLOv5 improved detection speed and accuracy; however, their performance slightly degrades in crowded scenes or fast-moving objects.

In comparison, YOLOv8 demonstrated superior accuracy of approximately 94%**,** outperforming earlier models due to improved feature extraction, better bounding box regression, and enhanced handling of real-time video streams. This makes YOLOv8 more reliable and efficient for campus surveillance applications requiring continuous monitoring.

Tabel 6.2: Comparative study with other object detection models

Object Detection Model	Approximate Accuracy (%)	Remarks
Haar Cascade [12]	70–75%	High false positives, sensitive to lighting
SSD [34]	75–82%	Moderate accuracy, weak for small objects
Faster R-CNN [3]	85–88%	High accuracy but slow inference
YOLOv3 [5]	82–85%	Faster but less accurate than newer versions
YOLOv5 [34]	88–90%	Good performance, struggles in crowded scenes
Proposed YOLOv8	**~94%**	High accuracy, real-time performance

6.3 User Interface Design

The user interface of the proposed AI-based campus surveillance system is designed to provide a clear, simple, and effective means of interacting with the system. The primary objective of the interface is to allow users to monitor campus activities in real time while clearly visualizing detection and recognition results produced by the underlying AI models. Emphasis is placed on usability and clarity so that security personnel and administrators can operate the system without requiring specialized technical knowledge.

As shown in Fig 6.1, the main interface displays the live video feed captured from a connected webcam or CCTV camera. This interface serves as the central monitoring window through which the entire surveillance process is observed. Once the system is initiated, the video stream is continuously processed without the need for manual intervention. The interface automatically loads the detection and recognition modules, enabling smooth and uninterrupted monitoring. Simple execution controls such as start and stop functionality are included to manage system operation effectively.

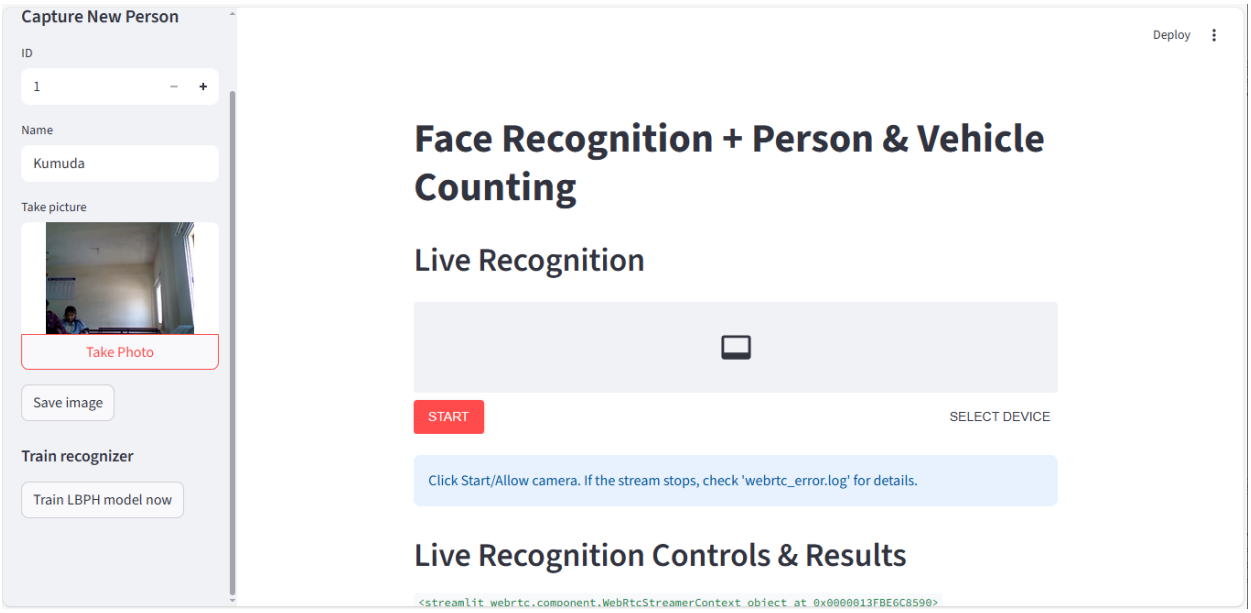

Fig 6.1 User Interface

The processed output of the system is illustrated in Fig 6.2, which shows the real-time detection and recognition results overlaid on the video frames. The system draws bounding boxes around detected humans, vehicles, faces, and number plates. Each bounding box is accompanied by descriptive labels, such as the detected object type, recognized face name, or extracted vehicle number plate. This visual annotation enables users to instantly understand the scene and identify important details without referring to external logs or reports.

Recognized individuals are clearly labeled with their corresponding names when a match is found in the face database. Faces that are not recognized are marked as “Unknown,” helping distinguish authorized individuals from unidentified persons. Similarly, detected vehicles display their extracted number plate text, allowing quick verification of vehicle identity. These visual indicators enhance situational awareness and assist security personnel in making timely decisions.

Fig 6.2 Annotated Output of the Proposed Campus Surveillance System

The interface ensures real-time performance by updating annotated frames continuously with minimal delay. The smooth rendering of video and detection results demonstrates the system's ability to operate efficiently on standard hardware. In addition to real-time visualization, the interface works in coordination with the logging module, enabling detected information to be stored automatically for later review and analysis.

Overall, the user interface design prioritizes simplicity, clarity, and real-time responsiveness. By combining live video monitoring with clear visual annotations, the interface plays a crucial role in making the proposed surveillance system practical and effective for campus security applications.

CHAPTER 7

CONCLUSION

7.1 Conclusion

In this project, an AI-based campus surveillance system was designed and implemented to improve security through automated real-time monitoring. The system aims to overcome the limitations of traditional manual surveillance by integrating computer vision and machine learning techniques capable of detecting, recognizing, and analyzing activities from live video streams.

The proposed system captures live video from a webcam or CCTV camera and processes frames in real time. Preprocessing operations such as resizing, grayscale conversion, and noise reduction are applied to enhance detection and recognition accuracy. YOLOv8 is employed as the primary object detection model to identify humans, vehicles, and number plate regions efficiently. Its speed and accuracy make it well suited for real-time surveillance applications.

Face detection is performed using Haar Cascade classifiers, and known individuals are identified using the LBPH face recognition algorithm. A dataset of authorized faces is collected and used to train the LBPH model, enabling reliable identification of known persons while labeling unknown individuals appropriately. Vehicle number plate recognition is achieved using OCR techniques through EasyOCR or Tesseract, allowing the system to extract alphanumeric text from detected number plates.

The system includes a simple and user-friendly interface that displays live annotated video output. Detected objects, recognized faces, and extracted number plate numbers are clearly visualized using bounding boxes and labels. Additionally, the system supports optional logging of detection results such as face identities, vehicle numbers, and timestamps in CSV format for future reference and analysis.

Performance evaluation indicates that YOLOv8 provides high object detection accuracy, the LBPH model performs reliably for face recognition under suitable conditions, and the OCR

module effectively reads number plates when visibility is clear. Although environmental factors such as lighting and motion may affect accuracy, the overall system demonstrates stable and efficient real-time performance.

In conclusion, the developed system successfully fulfills the objective of creating an intelligent and automated campus surveillance solution. It reduces reliance on manual monitoring, enhances situational awareness, and provides a scalable foundation for future improvements such as alert generation, multi-camera support, and advanced analytics. The proposed system offers a practical and effective approach to modern campus security management.

7.2 Future Enhancement

Developing systems that can simultaneously and seamlessly analyze faces, behaviors, and vehicle and adaptively are the most important systems to design. Edge computing and federated learning will improve speed at the cost of enhanced user privacy by Explainable AI should be included to improve automated trust systems to gain trust in the systems built.

There is a strong need for richer and more diverse campus surveillance datasets so that models can better handle real-world variations. With the growth of IoT and 5G, faster data transfer and instant alerts will support more proactive and responsive monitoring. Finally, future systems must balance security with privacy by using privacy-preserving techniques and strong ethical governance frameworks to ensure responsible deployment in educational environments.

REFERENCES

[1]Sarikonda, P., & Bhutada, S. (2020). Customized campus surveillance system. *International Research Journal of Engineering and Technology (IRJET)*, *7*(1), 737. https://www.irjet.net/archives/V7/i1/IRJET-V7I1143.pdf

[2]Sreekantha, B., Vijayalakshmi, S. A., Prasad, H. B., Hrithik, C., Nagajyothi, M. S., & Rakshitha, S. (2021). Face-based CCTV attendance monitoring system using deep face recognition. *International Research Journal of Engineering and Technology (IRJET)*, *8*(6), 3763. https://www.irjet.net/archives/V8/i6/IRJET-V8I6714.pdf

[3]Alhanaee, K., Alhammadi, M., Almenhali, N., & Shatnawi, M. (2021). Face Recognition Smart Attendance System using Deep Transfer Learning. *Procedia Computer Science*, *192*, 4093–4102. https://doi.org/10.1016/j.procs.2021.09.184

[4] Nurkhamid, Setialana, P., Jati, H., Wardani, R., Indrihapsari, Y., & Norwawi, N. M. (2021). Intelligent attendance system with face recognition using the deep convolutional neural network method. *Journal of Physics: Conference Series*, *1737*, 012031. https://doi.org/10.1088/1742-6596/1737/1/012031

[5] Junaid, N., Khalid, M., Saunshi, N., Mehta, P., & Thippeswamy, M. N. (2021). Smart College Camera Security System using IOT. In *Lecture notes in electrical engineering* (pp. 295–309). https://doi.org/10.1007/978-981-16-1338-8_26

[6]Ye, L., Liu, T., Han, T., Ferdinando, H., Seppänen, T., & Alasaarela, E. (2021). Campus violence detection based on artificial intelligent interpretation of surveillance video sequences. *Remote Sensing*, *13*(4), 628. https://doi.org/10.3390/rs13040628

[7]Muhammad, W., Ahmed, I., Ahmad, J., Nawaz, M., Alabdulkreem, E., & Ghadi, Y. (2022). A video summarization framework based on activity attention modeling using deep features for smart campus surveillance system. *PeerJ Computer Science*, *8*, e892. https://doi.org/10.7717/peerj-cs.892

[8]Keerthana, P., Prasath, S., Tamilmani, S., & Veeragokulraj, S. (2024). Enhancing campus security through smart surveillance system. *Proceedings of the 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 14–15 March 2024. IEEE. https://doi.org/10.1109/ICACCS60165.2024.10452631

[9]Ullah, R., Hayat, H., Siddiqui, A. A., Siddiqui, U. A., Khan, J., Ullah, F., Hassan, S., Hasan, L., Albattah, W., Islam, M., & Karami, G. M. (2022). A real-time framework for human face detection and recognition in CCTV images. *Journal of Sensors*, *2022*, Article ID 3276704. https://doi.org/10.1155/2022/3276704

[10]Kagona, E. (2022). Facial recognition attendance scheme on CCTV cameras using Open Computer Vision and deep learning: A case study of International University of East Africa (IUEA). *Advanced Journal of Science, Technology and Engineering*, *2*(1), 1–27. https://doi.org/10.52589/AJSTE-HYVTCZ9E

[11]Zamir, M., Ali, N., Naseem, A., Frasteen, A. A., Zafar, B., Assam, M., Othman, M., & Attia, E. A. (2022). Face detection & recognition from images & videos based on CNN & Raspberry Pi. *Computation*, *10*(9), 148. https://doi.org/10.3390/computation10090148

[12]Framework for Face Recognition in CCTV Based on Internet of Things (IoT) SOURASIS CHATTOPADHYAY1 ,MOSIUR RAHAMAN1 ,CHAYADI OKTOMY NOTO SUSANTO2 ,NIZIRWAN ANWAR3 ,AUGUSTA AN-HAU CHEN4 1Department of Computer Science and Information Engineering, Asia University, Taiwan 2Universitas Muhammadiyah Yogyakarta, Indonesia 3Esa Unggul University of Jakarta, Indonesia 3National Chung Hsing University, Taiwan. (n.d.). *DSIM*.

[13]Verma, R., Bhardwaj, N., Bhavsar, A., & Krishan, K. (2021). Towards facial recognition using likelihood ratio approach to facial landmark indices from images. *Forensic Science International Reports*, *5*, 100254. https://doi.org/10.1016/j.fsir.2021.100254

[14]Merikapudi, S., Math, S., Nandini, C., & Rafi, M. (2020). Face recognition using CNN trained with histogram equalization based image enhancement scheme. *European Journal of Molecular & Clinical Medicine*, *7*(8), 2940. ISSN 2515-8260

[15] Golwalkar, R., & Mehendale, N. (2022). Masked-face recognition using deep metric learning and FaceMaskNet-21. *Applied Intelligence*, *52*(11), 13268–13279. https://doi.org/10.1007/s10489-021-03150-3

[16]Nurpeisova, A., Shaushenova, A., Mutalova, Z., Zulpykhar, Z., Ongarbayeva, M., Niyazbekova, S., Semenov, A., & Maisigova, L. (2022). The study of mathematical models and algorithms for face recognition in images using Python in proctoring system. *Computation*, *10*(8), 136. https://doi.org/10.3390/computation10080136

[17]Smith, M., & Miller, S. (2021). The ethical application of biometric facial recognition technology. *AI & Society*, *37*(1), 167–175. https://doi.org/10.1007/s00146-021-01199-9

[18]Prasetyo, E. W. (2022). The effectiveness of surveillance system through CCTV (Close Circuit Television) in improving campus environment security (*Efektivitas sistem pengawasan melalui CCTV (Close Circuit Television) dalam meningkatkan keamanan lingkungan kampus*). *[Journal Name]*,

[19]Gondosiswojo, A. R. P., & Kusuma, G. P. (2023). Low resolution face recognition on CCTV images using a combination of super resolution and face recognition models. *Journal of Theoretical and Applied Information Technology*, *101*(20), 6353. https://www.jatit.org/volumes/Vol101No20/12Vol101No20.pdf

[20]Firmasyah, G., Joniwan, J., Widodo, A. M., & Tjahjono, B. (2023). Preventing Child Kidnaping at Home Using CCTV that Utilizes Face Recognition with You Only Look Once (YOLO) Algorithm. *Journal of Social Research*, *2*(9), 3291–3304. https://doi.org/10.55324/josr.v2i9.1403

[21] Budiman, A., Fabian, F., Yaputera, R. A., Achmad, S., & Kurniawan, A. (2022). Student attendance with face recognition (LBPH or CNN): Systematic literature review. *Procedia Computer Science*, *217*, 1370–1377. https://doi.org/10.1016/j.procs.2022.12.108

[22]Hangaragi, S., Singh, T., & Neelima, N. (2023). Face detection and recognition using face mesh and deep neural network. *Procedia Computer Science*, *218*, 741–749. https://doi.org/10.1016/j.procs.2023.01.054

[23] Henry, C., Asif, M. S., & Li, Z. (2023). Privacy Preserving Face Recognition with Lensless Camera. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/icassp49357.2023.10096627

[24]Irfan, E., Jacob, C., & Resmi, R. (2024). Facial recognition and CCTV integration for enhanced security using deep learning techniques. In *Proceedings of the 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS)* (pp. [page numbers if available]). IEEE. https://doi.org/10.1109/RAICS61201.2024.10689986

[25]Surantha, N., Yose, E., & Isa, S. M. (2024). Low-Resolution face recognition for CCTV and Edge-Powered smart attendance systems. *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 676–681. https://doi.org/10.1109/compsac61105.2024.00097

[26]Ambre, N., Pitale, R., Rao, P., Mote, H., & Chavan, S. (2024). A-eye: Attendance monitoring using face detection and recognition from CCTV footage. *https://ieeexplore.ieee.org/abstract/document/10638970*, 1–6. https://doi.org/10.1109/esmarta62850.2024.10638970

[27]Sari, S. A., & Sulistiyo, M. D. (2024). Suspect Identification Based on Facial Recognition from CCTV Using Hybrid Bat Algorithm. *https://ieeexplore.ieee.org/abstract/document/10776490?casa_token=HaneQ0dAScQAAAAA:iTG-0Pn95KW1nSibEIfF8aQq4v-ojjNX9TGQa8frI83NPe6tk0zkyf37cH3Zv-YaTzt6MMlQdTo*, 205–210. https://doi.org/10.1109/eecsi63442.2024.10776490

[28]Bhatt, C., Semwal, M., Aswal, P. S., Goswami, V., Rawat, S., & Dhanalakshmi, R. (2024). Real time surveillance criminal detection system. In *Proceedings of the 2024 Second International Conference on Advances in Information Technology (ICAIT)* (pp. [page numbers if available]). IEEE. https://doi.org/10.1109/ICAIT61638.2024.10690333

[29]Singh, A., & Tiwari, R. K. (2024). AIGuard: Criminal Tracking in CCTV Footage Using MTCNN and ResNet. *IEEE*, 31–35. https://doi.org/10.1109/confluence60223.2024.10463292

[30]Chittibomma, S. S., Surapaneni, R. K., & Maruboina, A. (2024). Facial Recognition System for Law Enforcement: An Integrated Approach Using Haar Cascade Classifier and LBPH Algorithm. *IEEE*, 1–6. https://doi.org/10.1109/apci61480.2024.10616450

[31]Rathod, V.M., Patil, A.M., Motekar, H.S. *et al.* Automatic face recognition based on enhanced Vggface- 16 model in an unconstrained environment using transfer learning. *Multimed Tools Appl* (2025). https://doi.org/10.1007/s11042-025-20819-w

[32]Khader Basha, Sk., Malavika, I., Priyanka, V. H., Bhavitha, B. K., & Kiran, G. B. S. (2025). Facial recognition and neural networks for missing child identification: A smart approach. *International Journal for Modern Trends in Science and Technology*, *11*(03), 397–407. https://doi.org/10.5281/zenodo.15174958

[33]Sanju, D. J., Surya, B. R., Sudipta, S., Suhas, T. G., & Suryakiran, N. (2025). Safenest: Finding and reuniting lost children using face recognition. *Journal of Emerging Technologies and Innovative Research (JETIR), 12*(2), d298. https://www.jetir.org/view?paper=JETIR2502335

[34]Arumugam, N. S. (2025). Deep Learning-Based Smart Invigilation System for enhanced exam integrity. *Proceedings of Engineering and Technology Innovation*, *29*, 99–115. https://doi.org/10.46604/peti.2024.14105

[35]Norhashim, N., Shah, S. A., Kamal, N. L. M., Sahwee, Z., Azizan, M. A., & Norizan, M. I. a. A. (2025). Face Recognition System at the Airport Based on Internet of Things and Cloud Technologies. *KARYA*, *1*(1), 31–40. https://doi.org/10.37934/kjaas.1.1.3140

[36]Nemavhola, A., Viriri, S., & Chibaya, C. (2025). A scoping review of literature on deep learning techniques for face recognition. *Human Behavior and Emerging Technologies*. https://doi.org/10.1155/hbe2/5979728

[37] Laishram, L., Shaheryar, M., Lee, J. T., & Jung, S. K. (2024). Toward a privacy-preserving face recognition system: A survey of leakages and solutions. *ACM Computing Surveys*, *57*(6), Article 147, 1–38. https://doi.org/10.1145/3673224

[38]Chahande, M., Yadav, L., & Wadhibhasme, B. (2025). Creation of an automated video surveillance system using IoT and AI systems. *Journal of Interdisciplinary and Multidisciplinary Research*, *11*(2). https://doi.org/10.5281/zenodo.15276889

[39]Anand, P. D., Umasankaran, R., Joshi, N., & Mamtha, K. R. (2025). Optimized CCTV monitoring using biometrics and AI-driven surveillance. *Journal of Informetrics*, *19*(1).

[40]David Egbe, O., Fraser Anwaitu, E., & Memoye Kepeghom, O. (2025). Development and optimization of a real-time campus security surveillance system using Arduino and CCTV camera. *BW Academic Journal*. Retrieved from https://bwjournal.org/index.php/bsjournal/article/view/2615

[41]Farooq, H., & Khan, N. A. (2025). Safe-Campus: Leveraging AI for advanced surveillance and security enhancement. *2024 4th International Conference on Innovations in Computer Science (ICONICS)*, Karachi, Pakistan. IEEE. https://doi.org/10.1109/ICONICS64289.2024.10824618

[42]Deepalakshmi, R., Murugan, K. R. S., Arunnachalam, R. S., & Asif, M. B. (2024). Campus surveillance: Deep learning-based rules violation detection. *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India. IEEE. https://doi.org/10.1109/Confluence60223.2024.10463401

[43]Ehiagwina, F. O., Kadiri, K. O., Yekeen, S. A., Azanubi, J. O., & Mustapha, K. O. (2024). Design and implementation of a reliable and secure wireless CCTV camera network for the main

administration building, Federal Polytechnic Offa. *Engineering and Technology Journal*, *9*(9), 5007–5011. https://doi.org/10.47191/etj/v9i09.04

[44]Mishra, S., & Jabin, S. (2024). Enhancing campus surveillance using temporal self attention. *Available at SSRN*. https://doi.org/10.2139/ssrn.5083118

[45]Olorunyomi, G. T., Babalola, D. A., Tiamiyu, A. T., Aliyu, B. A., & Ojajuni, D. (2025). Exploring the integration of deep learning in CCTV systems for enhanced security measures in academic libraries. *Journal of Engineering Research and Reports*, *27*(2), 310–323. https://doi.org/10.9734/jerr/2025/v27i21411

[46]Wu, Y., Feng, S., Wu, Y., Wang, J., Zhou, S., Yuan, M., Hu, Z., & Wu, C. (2025). Masked region disparity-based unsafe behavior detection via campus monitoring device images. *Knowledge-Based Systems*, *286*, 113051. https://doi.org/10.1016/j.knosys.2025.113051