Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Safewatch AI : Real-time Weapon Detection System for CCTV Surveillance

Sajina Salim, Ahsana Latheef T. A, Ihlas N Jabbar, Muhamad Saheel, Ubaidha Ummer
Department of Computer Science Engineering
KMEA Engineering College
Ernakulam, India

***Abstract*—In recent years, the need for better security systems has grown due to increasing crime and fraud. traditional CCTV setups rely heavily on continuous human monitoring, which often results in delays, missed incidents, and errors. To address these issues, SafeWatch AI introduces an Ai powered intelligent surveillance system that can automatically detect suspicious activities and alert authorities in real time. The system uses computer vision and deep learning to analyze live CCTV footage, identify faces, detect actions such as theft, intrusion, or violence, and instantly notify administrators through a dashboard or mobile application. It also includes features like blacklist management and an incident reporting module that logs events and provides useful insights. By reducing human involvement and improving response time, SafeWatch AI offers a more reliable and efficient approach to security for institutions, public areas, and private organizations.**

I. Introduction

In today's fast paced world, road safety has become one of the most serious concerns facing society. Every year, thousands of lives are lost due to delayed medical assistance after road accidents. According to a report by the World Health Organization (WHO), nearly 1.3 million people die each year from traffic accidents, and a timely emergency response could significantly reduce this number [1]. Most of these deaths could have been prevented if victims had received immediate emergency support. This challenge inspired the development of SafeWatch AI, an intelligent road monitoring and accident detection system designed to enhance emergency response and save lives. The system combines modern technologies such as artificial intelligence (AI), machine learning (ML), and cloud integration to automatically detect accidents and alert the nearest emergency units within seconds [2]. SafeWatch AI is built with the goal of transforming how accidents are identified and reported. In many cases, accidents go unnoticed for a long period especially on highways, isolated routes, or during late night hours when traffic is minimal [3]. Even if a passerby notices an accident, manually informing authorities can take precious minutes. SafeWatch AI aims to bridge this critical time gap by using real time video surveillance and automated accident detection algorithms. By continuously analyzing live footage from roadside cameras, the system can identify unusual events such as collisions or sudden halts, and immediately notify emergency services through a connected alert system [4]. The project's vision goes beyond simple accident detection. It represents a step toward creating smart roads that can think, analyze, and respond intelligently. With the help of advanced image processing and deep learning techniques, SafeWatch AI can distinguish between normal traffic patterns and accident scenarios [5]. Once an accident is confirmed, it generates an instant alert containing the exact location, time, and type of accident. This information is then sent to the nearest hospital or ambulance service, enabling faster action and potentially saving numerous lives. Another important motivation behind SafeWatch AI is to reduce human dependency in accident reporting. Traditional systems rely heavily on manual calls or witness reports, which are often delayed or inaccurate [6]. The automated nature of SafeWatch AI minimizes such issues and ensures consistent, real time detection. Additionally, the system can be integrated with cloud based storage and traffic management servers, allowing authorities to monitor accident trends, analyze data for future improvements, and even predict high risk zones using statistical insights [7]. From a technical standpoint, SafeWatch AI is a combination of hardware and software intelligence. The hardware involves surveillance cameras and communication modules, while the software side uses machine learning models trained to identify accident features in real time. The use of technologies like OpenCV, TensorFlow, and iot based alert systems makes the solution robust, scalable, and suitable for real world deployment [8]. The overall framework is designed to be modular, so it can be easily integrated into existing smart city infrastructures or highway monitoring systems [9]. In short, SafeWatch AI is not just a technological innovation it's a life saving initiative. It represents the future of intelligent transportation and emergency management systems. By blending automation, artificial intelligence, and human concern, this project aims to make roads safer and ensure that help reaches accident victims faster than ever before. Through

Identify applicable funding agency here. If none, delete this.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

this report, we explore how SafeWatch AI is developed, and the potential impact it can have on our society.

II. Related Works

[1] Ruiz-Santaquiteria et al. proposed a handgun detection framework that integrates human pose estimation with weapon appearance features to improve contextual understanding in surveillance scenes. The system combines pose-based cues with deep convolutional feature extraction to enhance firearm detection, particularly in scenarios involving occlusion and complex backgrounds. Evaluated on surveillance datasets, the model demonstrated improved detection robustness compared to appearance-only approaches, highlighting the importance of contextual reasoning in weapon recognition tasks.

[2] Bhatti et al. developed a deep learning–based weapon detection system for real-time CCTV video analysis. The authors employed convolutional neural networks to detect firearms in surveillance footage and evaluated their system under various environmental conditions. The model achieved reliable real-time performance while maintaining competitive detection accuracy. However, the study indicated sensitivity to illumination changes and background clutter, emphasizing the need for improved generalization strategies.

[3] Tahir et al. introduced a real-time event-driven road traffic monitoring system using CCTV video analytics. Although the primary focus was traffic monitoring rather than weapon detection, the framework utilized deep learning–based object detection and event analysis techniques to process live surveillance streams. The system demonstrated scalable real-time performance in urban environments, achieving high detection reliability across dynamic scenes. Their work underscores the effectiveness of deep learning models in large-scale surveillance applications.

[4] Thakur et al. presented a real-time accident detection system using CCTV footage analysis. The approach leveraged deep learning models for detecting abnormal events in traffic scenarios, enabling automated monitoring and rapid response. Experimental evaluations demonstrated effective real-time detection capability, validating the applicability of computer vision techniques in intelligent surveillance systems. However, the study primarily focused on motion-based anomaly detection rather than fine-grained object localization.

[5] Koca proposed a real-time security risk assessment framework using hand gesture recognition in CCTV streams. The system aimed to enhance situational awareness by identifying potentially threatening gestures through deep learning–based classification. The model achieved reliable recognition accuracy in controlled surveillance environments. While the approach contributes to proactive security monitoring, it does not directly address precise localization and classification of handheld weapons such as guns and knives.

III. Methodology

A. Existing Systems

Surveillance systems currently employ the use of traditional CCTV cameras and manual observation. These systems are not proactive, demanding perpetual human observation, and tend to miss out on unusual activity in real time. Most systems perform poorly in low light, dense, or sophisticated environments, resulting in lost events or late response. There are advanced solutions with AI or motion detection but at a high price, as they are rigid and offer limited functionality. In summary, current systems do not provide intelligent analysis, automated decision making, and smooth integration, hence the need for a smart and proactive solution such as SafeWatch AI.

B. Proposed Systems

SafeWatch AI is a sophisticated, AI based surveillance platform that extends beyond the conventional monitoring. It is based on state of the art computer vision and machine learning that identifies abnormal activities in real time, even in harsh conditions such as dim light or dense crowds. The system can automatically notify authorities or users in case of suspicious activity, minimizing response time and human intervention. With intelligent activity analysis, pattern recognition, and robust integration with other security systems as features, SafeWatch AI offers a proactive, efficient, and affordable solution for contemporary surveillance requirements.

C. System Requirements

The SafeWatch AI system needs both stable hardware and compatible software in order to operate effectively. On the hardware front, it requires a high end pro cessor like Intel i5 or higher, at least 8 GB RAM (16 GB recommended for optimal performance), a dedicated graphics card for AI processing like NVIDIA GTX se ries, high res cameras for sharp video input, and adequate storage for holding video recordings. On the software front, it runs on Windows 10/11 or Linux and requires Python 3.x along with basic libraries like OpenCV, TensorFlow, and Keras. It also needs a database system like MySQL or MongoDB to hold alerts and logs, and a web server and interface for remote viewing and notifications. These specifications are important to provide SafeWatch AI with the capability to handle video feeds in real time, achieve precise detection, and have seamless integration with other security systems.

D. System Architecture

The architecture is divided into five structured layers, each responsible for a specific stage in the detection pipeline.

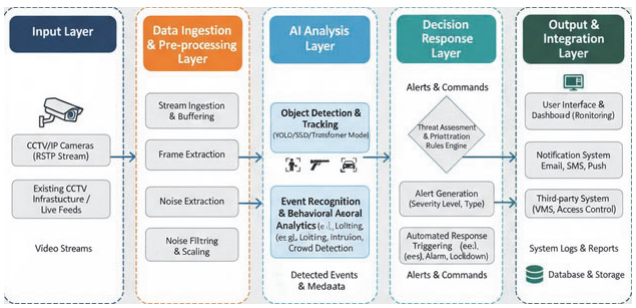

Fig. 1. SafeWatch AI System Workflow Block Diagram

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

1) Input Layer: The Input Layer is responsible for capturing live video data from surveillance cameras and preparing it for further processing within the system. The system connects to CCTV or IP cameras using the Real Time Streaming Protocol (RTSP), which enables continuous transmission of video data over a network. Once the live stream is received, the video is divided into individual frames through a frame extraction process. Instead of processing every frame, images are extracted at fixed intervals, such as every one to three seconds, in order to reduce computational overhead while maintaining effective detection performance. These extracted frames are then forwarded to the Processing Layer for enhancement and preprocessing. This layer serves as the entry point of the entire AI pipeline, ensuring stable real-time monitoring and consistent frame acquisition for accurate threat detection.

2) Processing Layer: The Processing Layer is responsible for preparing the extracted video frames for accurate analysis by the AI detection model. After frames are received from the Input Layer, they undergo several preprocessing steps to enhance image quality and ensure compatibility with the detection system. Initially, noise filtering is applied to remove visual disturbances such as blur, grain, and low-light distortions that may affect detection performance. This improves the clarity of the frames and enhances the model's ability to identify objects accurately.

Following noise removal, normalization is performed to adjust brightness, contrast, and pixel intensity values. Pixel values are scaled to a standardized range, typically between 0 and 1, which helps maintain consistency across different lighting conditions and camera qualities. The frames are then converted into the required format, such as transforming color channels from BGR to RGB, and resized to match the input dimensions expected by the YOLO model (for example, 640×640 pixels).

Additionally, image preprocessing techniques such as scaling, resizing, and sharpening are applied to ensure that the frames maintain uniform structure and quality before being passed to the Intelligence Layer. This layer plays a crucial role in improving detection accuracy, minimizing false positives, and ensuring that all input data meets the technical requirements of the deep learning model.

3) Intelligence Layer: The Intelligence Layer serves as the core artificial intelligence component of the SafeWatch AI system. It is responsible for analyzing the preprocessed frames using a YOLO (You Only Look Once) based object detection model to identify potential weapons in real time. The YOLO model processes the entire image in a single pass, enabling fast and efficient detection suitable for real-time surveillance applications. During this process, the model scans the frame and identifies objects such as guns and knives by generating bounding boxes around detected items.

Each detected object is assigned a confidence score that represents the probability of the prediction being accurate. For example, a confidence score of 0.85 indicates an 85 percent certainty that the detected object belongs to a specific class. To improve reliability and reduce false detections, confidence filtering is applied. Only detections that exceed a predefined threshold, such as 70 percent, are considered valid, while weaker or uncertain predictions are discarded.

If a weapon is detected with sufficient confidence, the system classifies it as a potential threat. Once confirmed, the relevant details are recorded in the threat database. This includes the timestamp of detection, the confidence score of the prediction, and a snapshot of the frame in which the object was identified. By combining real-time object detection with confidence-based filtering and systematic logging, the Intelligence Layer acts as the decision-making engine of the system, ensuring accurate and efficient threat identification.

4) Detection Threat Classification Layer: The Detection and Threat Classification Layer is responsible for determining whether a detected object qualifies as a genuine threat and for managing backend logging and system responses. Once the Intelligence Layer identifies a potential weapon, this layer performs a decision check to verify whether the detection meets the required confidence threshold. If no weapon is detected, the frame is discarded to avoid unnecessary processing. However, if a weapon is confirmed, the threat classification process is initiated.

Upon confirmation, detailed information about the detected threat is recorded in the system. This includes the timestamp of detection, the confidence score generated by the model, the captured image frame, and additional metadata such as camera identification and location details. The backend server, typically implemented using a Django-based framework, handles data storage, API communication, and alert triggering. All detection records, including event logs, image snapshots, and metadata, are securely stored in the database. This structured storage enables historical analysis, report generation, and future reference. Overall, this layer acts as a bridge between the AI detection system and the application interface, ensuring that only verified threats generate alerts while maintaining organized and secure data management.

5) Application Layer: The Application Layer is the user-facing component of the system that delivers alerts and provides monitoring capabilities. Once a threat is confirmed and logged by the backend, notifications are immediately sent to registered users through mobile applications or web dashboards. Push notifications are generated in real time and typically include a preview image of the detected threat to provide immediate visual context. In addition to push alerts, detailed threat reports may also be sent via email, containing information such as the time of detection, camera location, and captured snapshot.

The application dashboard allows users to view live alerts, review past threat history, and monitor confidence levels associated with each detection. This interface ensures continuous situational awareness and enables users to respond quickly to potential dangers. By providing real-time monitoring and interactive alert management, the Application Layer enhances overall system usability and ensures effective communication between the detection system and its users.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

E. Dataset Description

The dataset used in this study was constructed by merging four publicly available weapon detection datasets obtained from Kaggle and Roboflow. These datasets contain annotated images of firearms and knives captured in diverse real-world environments, including indoor surveillance scenes, outdoor public areas, and CCTV camera footage. The objective of combining multiple datasets was to increase diversity, improve generalization capability, and enhance real-time detection performance.

The following publicly available datasets were utilized:

1) **CCTV WEAPON - v1 CCTV Weapon Detection V1** https://universe.roboflow.com/sami-kuprn/cctv-weapon-yshhr
2) **Weapon-Detect - v2** https://universe.roboflow.com/deakin-07shj/weapon-detect-41htd
3) **CCTV WEAPON Computer Vision Dataset** https://universe.roboflow.com/sami-kuprn/cctv-weapon-yshhr
4) **Weapon-Detection-1cj3j Dataset** https://universe.roboflow.com/nick-payew/weapon-detection-1cj3j

The datasets were preprocessed through manual data cleaning, duplicate image removal, and class label alignment to ensure consistency across all sources. Class remapping was performed to unify different naming conventions under two primary categories: *gun* and *knife*. Additionally, noisy or incorrect annotations were removed after manual verification to improve dataset reliability.

To address class imbalance, oversampling techniques were applied to ensure a more uniform distribution between weapon categories. Data augmentation techniques such as horizontal flipping, rotation, scaling, and brightness adjustment were also implemented to enhance model robustness and improve generalization across varying environmental conditions.

After preprocessing and merging, the final unified dataset contained 25,473 gun instances and 22,408 knife instances. Although the base images were sourced from publicly available datasets, significant preprocessing and refinement were conducted to create a structured, balanced, and deployment-ready weapon detection dataset suitable for real-time surveillance applications.

F. Model Selection

Object detection for real-time weapon surveillance requires a model that achieves a balance between detection accuracy and inference speed. Considering these constraints, the YOLO (You Only Look Once) architecture was selected due to its single-stage detection mechanism, which enables real-time performance while maintaining competitive accuracy.

Among the YOLO variants, the YOLOv8 architecture was chosen because of its anchor-free detection head, improved feature pyramid network (FPN), and optimized loss functions (CIOU-based box loss and Distribution Focal Loss). These improvements enhance localization accuracy and classification robustness compared to earlier YOLO versions.The selected model variant provides an effective trade-off between computational efficiency and detection performance, making it suitable for deployment in surveillance systems and edge-based environments. The model is capable of detecting two classes: *gun* and *knife*.

The choice of YOLOv8 is further justified by:

- Real-time inference capability.
- High detection accuracy for small and medium-sized objects.
- Reduced latency compared to two-stage detectors such as Faster R-CNN.
- Efficient deployment in GPU-enabled and embedded environments.

G. Training Procedure

The training dataset was constructed by merging four publicly available weapon detection datasets obtained from Kaggle and Roboflow. The datasets were cleaned to remove duplicate images, incorrect annotations, and noisy labels. Class labels were unified to ensure consistency across all merged datasets.To address potential data imbalance and improve generalization, oversampling techniques and data augmentation were applied. The augmentation strategies included:

- Random horizontal flipping
- Random scaling and cropping
- Color jitter (brightness and contrast adjustments)
- Mosaic augmentation

The final merged dataset contained 25,473 gun instances and 22,408 knife instances.The dataset was split into training and validation subsets using an 80:20 ratio. Training was performed for 50 epochs with early stopping to prevent overfitting.The optimization was carried out using Stochastic Gradient Descent (SGD) with momentum. The initial learning rate was set to 0.01, with a cosine decay scheduler applied during training. A batch size of 16 was used.

The model performance was evaluated using:

- Precision
- Recall
- F1-score
- mAP@0.5
- mAP@0.5:0.95

The best model weights were selected based on the highest validation mAP@0.5 score.

H. Implementation Details

The model was implemented using the Ultralytics YOLOv8 framework in Python. Training and evaluation were performed on a system equipped with an NVIDIA GPU to accelerate computation.The experiments were conducted on the following hardware configuration:

- GPU: NVIDIA RTX series
- CPU: Intel Core i7
- RAM: 16 GB

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The software environment included Python 3.x , PyTorch , YOLOv8 , CUDA-enabled GPU drivers . Images were resized to a fixed resolution before being fed into the network. During inference, a confidence threshold of 0.48 was used, corresponding to the highest F1-score observed during validation.The system supports real-time detection and can be integrated into surveillance applications for automated threat monitoring.

IV. Results and Discussion

A. Precision–Recall curve

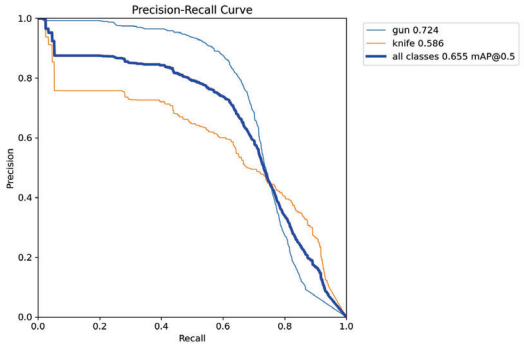

Fig. 2. Precision–Recall curves for gun and knife classes showing class-wise AP and overall mAP@0.5 performance.

Fig. 2 illustrates the Precision–Recall (PR) curves for both gun and knife classes. The proposed model achieves an Average Precision (AP@0.5) of 0.724 for gun detection and 0.586 for knife detection, with an overall mAP@0.5 of 0.655.

The PR curve for the gun class maintains high precision across a wide range of recall values, indicating reliable detection and minimal false positives. In contrast, the knife class exhibits a steeper decline in precision as recall increases, suggesting higher misclassification rates and greater sensitivity to background noise.

The performance gap between the two classes can be attributed to variations in object size, orientation, and visual similarity between knives and surrounding objects in surveillance scenarios. These findings highlight the need for improved small-object detection and enhanced background suppression techniques, which will be addressed through hard negative mining in future work.

B. Confusion Matrix

Fig. 3 presents the confusion matrix of the proposed weapon detection model. The results indicate strong classification performance for the gun class, with 1972 correctly detected instances and minimal confusion with the knife class. However, 487 gun instances were misclassified as background, indicating occasional missed detections.

In contrast, knife detection demonstrates a higher misclassification rate, with 1612 knife instances incorrectly predicted as background and 178 misclassified as guns. This suggests that knife detection is more sensitive to variations in object

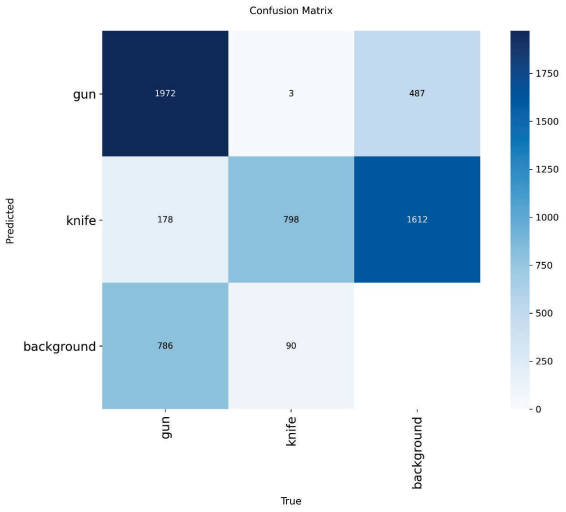

Fig. 3. Confusion matrix of the proposed weapon detection model highlighting class-wise true positives, false positives, and false negatives.

size, orientation, and background complexity.Additionally, a number of background regions were falsely predicted as gun (786 instances) and knife (90 instances), contributing to false positives. These observations highlight the need for improved background suppression and hard negative mining strategies to enhance model robustness.

The presence of false positives and background confusion suggests that incorporating hard negative samples during retraining may further improve precision and reduce misclassification rates.

C. Training and validation performance

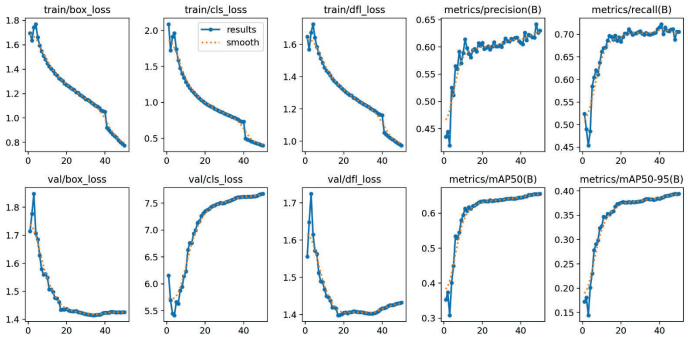

Fig. 4. Training and validation loss curves along with precision, recall, and mAP metrics demonstrating model convergence.

Fig. 4 illustrates the training and validation performance of the proposed model across epochs. The box loss, classification loss, and distribution focal loss demonstrate a consistent decreasing trend during training, indicating effective feature learning and localization refinement. The validation losses closely follow the training losses without significant divergence, suggesting minimal overfitting and good generaliza-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

tion capability. The mAP@0.5 metric increases progressively and stabilizes at approximately 65.5%, while mAP@0.5:0.95 converges near 39%. Precision and recall metrics also exhibit stable convergence behavior, reaching 63% and 71%, respectively. The stabilization of performance metrics after later epochs indicates that the model has reached convergence, and additional training would yield marginal improvements.

D. Detection sample

Fig. 5. Qualitative detection results of the proposed model: (a) knife detection in outdoor surveillance scene, (b) gun detection in indoor environment.

Fig. 5 presents representative detection results obtained using the proposed model. The examples demonstrate successful localization and classification of both gun and knife objects in diverse environments. In Fig. 8(a), the model accurately detects a knife in an outdoor surveillance-like scene despite slight motion blur and background clutter, achieving a confidence score of 0.76. This indicates robustness to real-world variations in lighting and resolution. In Fig. 8(b), the model correctly identifies a firearm in an indoor environment with precise bounding box localization and high confidence. The consistent detection performance aligns with the higher AP observed for the gun class in the Precision–Recall analysis. These qualitative results confirm the model's capability to operate effectively in practical surveillance scenarios.

E. Real Time Performance Evaluation

To evaluate real-time performance, the trained model was deployed on a live video stream obtained through an IP webcam application running on a mobile device. The stream simulated a practical surveillance camera environment under indoor lighting conditions. The system processed incoming frames at an input resolution of 640×480, with inference performed at 640×640. Experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3050 (6GB Laptop GPU) with CUDA acceleration enabled. The proposed model achieved an average inference latency of approximately 15–21 ms per frame, corresponding to a real-time throughput of 48–65 frames per second (FPS) under live streaming conditions. Performance variations were primarily attributed to network latency and frame decoding overhead from the IP stream.

Comparing with CPU execution, CPU-only execution achieved approximately 8 FPS, highlighting the significant acceleration gained through GPU deployment. These results demonstrate that the proposed system satisfies real-time constraints required for intelligent surveillance and rapid threat detection applications.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide, Sajina Salim, for the valuable guidance, continuous support, and insightful suggestions provided throughout this work. The authors also thank the faculty of the Department of Computer Science and Engineering, KMEA Engineering College, Ernakulam, for their support.

REFERENCES

[1] J. Ruiz-Santaquiteria, A. Velasco-Mata, N. Vallez, G. Bueno, J. A. Álvarez-García and O. Deniz, "Handgun Detection Using Combined Human Pose and Weapon Appearance," *IEEE Access*, vol. 9, pp. 123815-123826, 2021, doi: 10.1109/ACCESS.2021.3110335.

[2] M. T. Bhatti, M. G. Khan, M. Aslam and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," *IEEE Access*, vol. 9, pp. 34366-34382, 2021, doi: 10.1109/ACCESS.2021.3059170.

[3] M. Tahir, Y. Qiao, N. Kanwal, B. Lee and M. N. Asghar, "Real-Time Event-Driven Road Traffic Monitoring System Using CCTV Video Analytics," *IEEE Access*, vol. 11, pp. 139097-139111, 2023, doi: 10.1109/ACCESS.2023.3340144.

[4] P. B. H. K. S. Thakur, S. Moseen and V. Deepa, "Real-Time Accident Detection Using CCTV Footage Analysis," in *Proc. 3rd Int. Conf. Artificial Intelligence, Computational Electronics and Communication System (AICECS)*, Manipal, India, 2024, pp. 1-6, doi: 10.1109/AICECS63354.2024.10956241.

[5] M. Koca, "Real-Time Security Risk Assessment From CCTV Using Hand Gesture Recognition," *IEEE Access*, vol. 12, pp. 84548-84555, 2024, doi: 10.1109/ACCESS.2024.3412930.

[6] M. Grega, S. Lach and R. Sieradzki, "Automated recognition of firearms in surveillance video," in *Proc. IEEE Int. Multi-Disciplinary Conf. Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, Feb. 2013, pp. 45-50.

[7] M. Grega, A. Matiolański, P. Guzik and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image," *Sensors*, vol. 16, no. 1, p. 47, Jan. 2016.

[8] J. C. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 761-774, Aug. 2006.

[9] I. T. Darker, P. Kuo, M. Y. Yang, A. Blechko, C. Grecos, D. Makris, J.-C. Nebel and A. Gale, "Automation of the CCTV-mediated detection of individuals illegally carrying firearms: Combining psychological and technological approaches," *Proc. SPIE*, vol. 7341, Apr. 2009, Art. no. 73410P.

[10] M. T. Bhatti, M. G. Khan, M. Aslam and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," *IEEE Access*, vol. 9, pp. 34366-34382, 2021.