EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 7 No. 1 (2026) 122-129
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Real-Time Cigarette Smoking Detection Using YOLOv8 Convolutional Neural Network (CNN)

Devanthran Perumal[1], Munirah Ab Rahman[1]*

[1] *Department of Electronic Engineering, Faculty of Electrical and Electronics Engineering Universiti Tun Hussein Onn Malaysia, 86400, Batu Pahat, Johor, MALAYSIA*

*Corresponding Author: de28va10@gmail.com, munira@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2026.07.01.017

Article Info

Received: 13 June 2025
Accepted: 2 April 2026
Available online: 30 April 2026

Keywords

Cigarette Smoking Detection, YOLOv8 (Deep Learning), Real-Time Surveillance System

Abstract

Cigarette smoking has major health and environmental effects, especially in public places where second-hand smoke is a concern. Educational and higher education institution areas are among the 28 places categorized as areas effective January 1, 2025, following the implementation of the Smoking Products Control Act for Public Health 2024 (Act 852). To guarantee the efficient enforcement of the law, a real-time cigarette smoking detection system has been developed by using the YOLOv8 object detection model based on Convolutional Neural Networks (CNN). The system was built with Python and Streamlit, while the training and validation were done on a dataset of over 2,300 annotated photos tagged with 'cigarette', 'smoking', and 'face'. The YOLOv8n (nano) model was chosen for its compact architecture and ability to identify small objects. Training was carried out in Google Colab, with further programming and testing in PyCharm. The trained model performed well, with a precision of 93.73%, recall of 90.36%, F1-score of 91.86%, and mean Average Precision (mAP@0.5) of 94.36%, indicating that it is suitable for real-world amazing showed promising performance deployment. The system includes a web interface for detection via camera or video upload, automatic logging in CSV format, PDF report generation and Telegram notifications. Overall, the study successfully shows the practical application of deep learning for scalable and accurate real-time cigarette smoking detection, thus promoting better enforcement of no-smoking laws and improving public health surveillance.

1. Introduction

The detection of cigarette smoking using YOLO v8 Convolutional Neural Networks (CNNs) is an innovative and effective way [1], [2] to assist law enforcement and improve public awareness. Globally, CNN-based systems can detect smoking behaviour from static images, allowing for automated monitoring and lowering false alarms. In Malaysia, where no-smoking laws are strictly enforced in public places like parks, restaurants, and hospitals, YOLOv8 CNN model can be used to improve observance by detecting violations in real time. These tools help authorities monitor restricted zones and evaluate enforcement efficiency.

1.1 Problem Statement

Cigarette smoking in public places causes major health and environmental risks, but existing detection tackles based on manual surveillance are frequently inefficient, error-prone, and lack real-time response. These limitations prevent effective enforcement and monitoring, particularly in dynamic or large-scale settings. An automated system that can reliably identify smoking behavior in real-time from both image and video streams is desperately needed. This project suggests using the YOLOv8 object detection model [3], [4], which is based on YOLOv8 Convolutional Neural Networks (CNNs), to provide a reliable and scalable detection system. Furthermore, assessing the model's performance using important metrics like precision, recall, and mAP is critical to ensuring its efficacy and acceptability for real-world use. All datasets were made publicly available or manually annotated using Roboflow, in accordance with academic licensing rules. Ethical norms were upheld by anonymizing any identifiable elements and ensuring that photographs did not violate personal privacy.

1.2 Objective

The main objectives for this project are to create a system which can detect cigarette smoking behavior from images and video streams using the YOLOv8 object detection model, which is based on Convolutional Neural Networks (CNNs). The project also aims to assess the model's performance using important measures such as precision, recall, and mean Average Precision (mAP) to assure its reliability under varying settings. Finally, the detection system will be implemented in real time using a user-friendly Streamlit interface, with automated alert alerts via Telegram to ensure effective monitoring and reporting.

1.3 Significance of Study

This project presents a real-time cigarette smoking detection system utilizing the YOLOv8 model, which is based on Convolutional Neural Networks (CNNs). It demonstrates the effectiveness of deep learning in accurately identifying smoking behaviors across varied conditions. The system’s integration of detection, alerting, and logging provides a practical solution for public health monitoring and policy enforcement.

1.4 Literature Review

Several studies have investigated the use of computer vision algorithms to detect cigarette smoking. Nazari et al. [2] proposed a deep learning-based smoking detection system for surveillance video that performed satisfactorily under controlled conditions. Sharma and Sahu [5] created an automated real-time surveillance system that used object detection for public safety purposes. Alzubaidi et al. [1] conducted a thorough analysis of deep learning and CNN architectures, emphasizing their capability to handle complicated object detection tasks. These experiments indicate the viability of utilizing CNNs to detect cigarettes. However, most earlier models used outdated YOLO versions or insufficient datasets, resulting in a performance gap in real-time, high-accuracy detection. This study extends the previous work by employing the latest YOLOv8n model with enhanced accuracy, speed, and real-time alerting integration via a web interface and Telegram API.

2. Methodology

The methodology outlines each phase involved in the development process, starting from the design of the system architecture to model training, real-time detection, and deployment. It provides detailed explanations of the tools and technologies used, dataset preparation methods, and model configuration techniques. The implementation environment, including both software and hardware platforms, is discussed to illustrate how the system was built and tested. This chapter also covers the deployment process using Streamlit for a user-interactive web application, ensuring that the system functions effectively for real-time smoking detection with features like logging, alerting, and PDF report generation.

2.1 System Architecture

This study's system architecture focuses on the development of a cigarette smoking detection system based on the YOLOv8 Convolutional Neural Network model. The framework's key stages, which include data collecting, preprocessing, model training, detection, and alert or response handling. Initially, datasets containing cigarette smoking photos are gathered and annotated. These datasets are then pre-processed to ensure the best possible results for training, which includes resizing, normalization, and format conversion. The YOLOv8 model is used for feature extraction and classification, which involves processing visual inputs through many convolutional layers to detect items such as cigarettes, faces, and smoking activities. Once detected, the system records frames containing identified smoking behavior and delivers real-time alerts to Telegram, including GPS coordinates. All detected events are logged, and the visual results can be presented or exported as reports. This structured flow ensures accurate detection and timely alerting for real-time monitoring applications.

2.2 Detection Workflow

Fig. 1 illustrates the operational workflow of the proposed cigarette smoking detection system. The workflow begins with user authentication and dashboard access, followed by the selection of either video upload mode or live webcam monitoring. Uploaded videos are processed frame-by-frame, while webcam input is analysed continuously for real-time smoking detection. When smoking behaviour is detected, the system records the event, logs the detection information, and generates alerts. The workflow also provides facilities for downloading detection logs and administrative reports before the user exits the system.

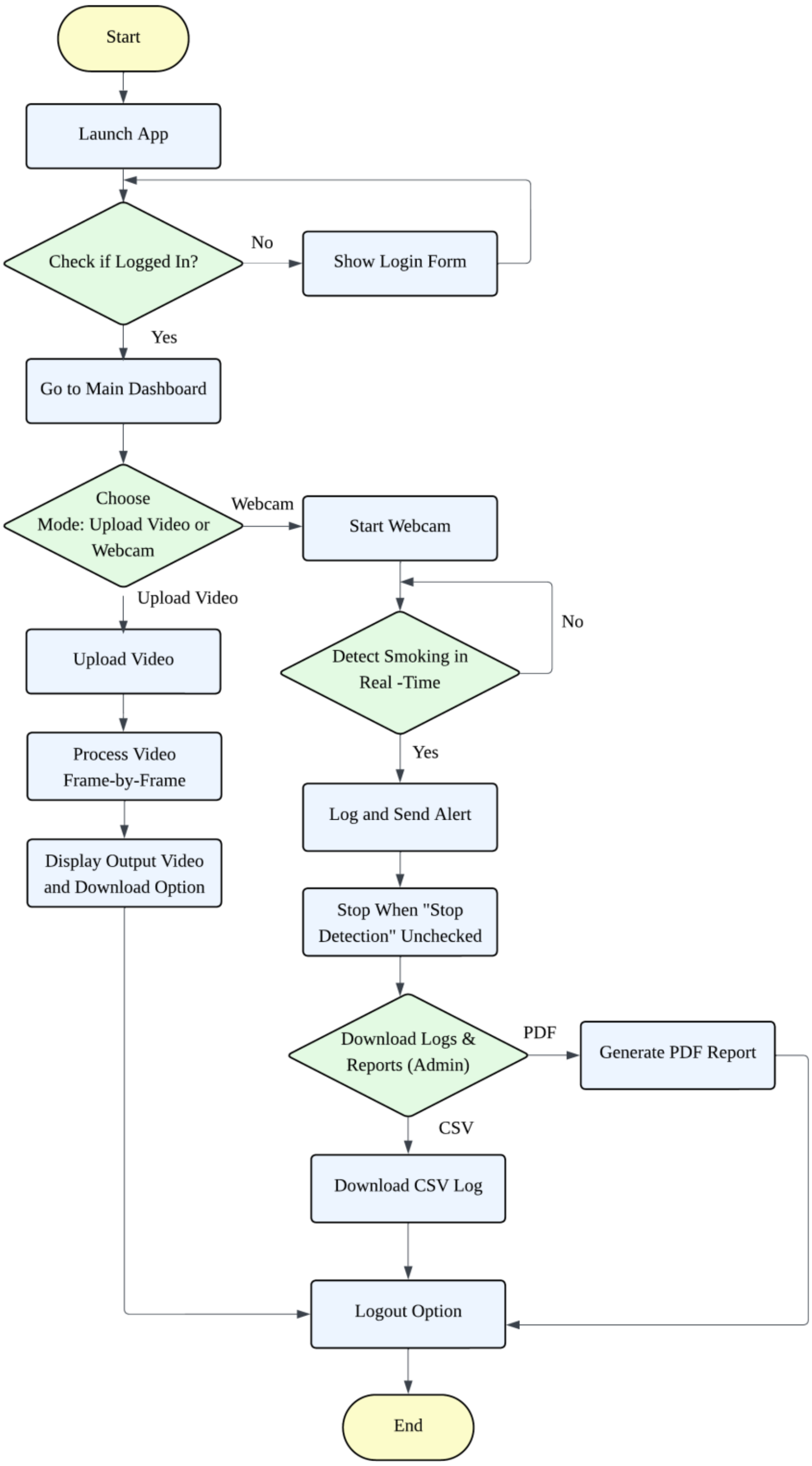


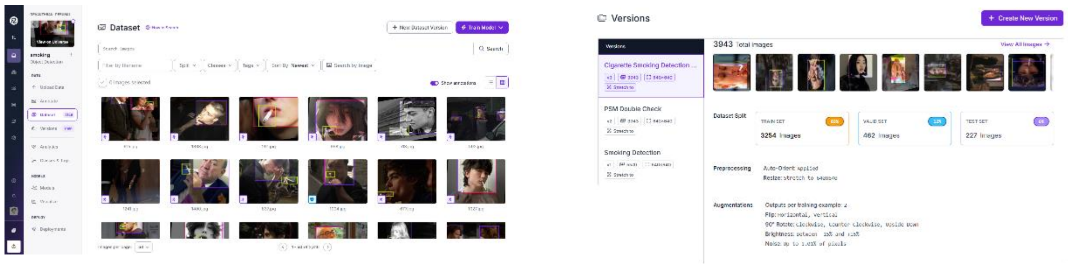
Fig. 1 *Flowchart of the proposed cigarette smoking detection system using YOLOv8 CNN*

As shown in Fig. 1, the application supports two detection modes, namely uploaded video analysis and real-time webcam monitoring. Both modes utilise the same YOLOv8 detection engine to identify smoking-related activities. Detection results are recorded automatically and can be exported as CSV logs or PDF reports for administrative review. The inclusion of authentication, logging, reporting, and alert mechanisms enables the system to function as a complete monitoring platform rather than a standalone object detection application.

2.3 Dataset Preparation (Roboflow)

The dataset for this project was created using Roboflow, a web-based application that simplifies image annotation and dataset management for computer vision applications. Images depicting both smoking and non-smoking conditions were gathered and divided into three object classes: cigarette, face, and smoking. For ideal balance in learning and evaluation, the dataset was separated into three sets: training (70%), validation (20%), and testing (10%). Roboflow also used preprocessing techniques including resizing and data augmentation to boost model generalization [6]. The final annotated dataset was produced in YOLOv8 format and put into the training pipeline via Google Colab.

Fig. 2 illustrates the dataset preparation process performed using Roboflow. The platform was utilised for image annotation, class management, dataset organisation, and dataset splitting. Through this interface, labelled images were prepared according to the defined object classes before being exported in YOLOv8 format for model training and evaluation.



(a) **(b)**

Fig. 2 *Dataset preparation in Roboflow: (a) image annotation interface; (b) dataset split configuration and version summary*

2.4 Model Training YOLOv8 in Google Colab

The model for detecting cigarette smoking was trained in Google Colab using the YOLOv8 object detection framework [2], [7], which allows for GPU acceleration for faster processing. The annotated dataset, prepared with Roboflow, was imported into the Colab environment and setup with the proper YAML file that defined detection classes and pathways. The lightweight version of the model, YOLOv8n, was chosen for its balance of speed and accuracy, making it suitable for real-time detection. The key hyperparameters were a batch size of 16, an image size of 640x640 pixels, and a learning rate of 0.001. The training phase lasted 50 epochs, allowing the model to learn complicated features from the data. During training, critical performance parameters like as loss, precision, recall, and mean Average Precision (mAP) were recorded and shown to track the model's progress over epochs.

2.5 Detection Interface (Streamlit)

The system's detecting interface was built with Streamlit, a Python-based open-source framework for creating responsive and interactive web apps. This interface is the frontend of the cigarette smoking detection system [8], allowing users to input footage and enable webcam-based detection in real time. It works directly with the trained YOLOv8 model to process each video frame and instantly display detection results, highlighting recognized objects like cigarettes, faces, and smoking behavior with bounding boxes. The interface also contains features for user login authentication, live result display, and access-controlled downloads of CSV logs and PDF reports. Streamlit's lightweight deployment capabilities make the system easily available through a web browser without the need for extensive backend work. To support the implementation of the detection interface, several Python libraries and functional modules were integrated into the system. Each module performs a specific role ranging from user interface development, video processing, object detection, data logging, authentication, and report generation. Table 1 summarises the software modules used and their respective functions within the proposed cigarette smoking detection system.

Table 1 *Functional description of software modules used in the proposed system*

Functional Modules	Description
Streamlit	Used to build the web-based Streamlit interface
cv2(OpenCV)	Used for handling video processing and frame-by-frame reading
ultralytics.YOLO	Loads the trained YOLOv8 model (best.pt) performs object detection on incoming frames
tempfile, os, time	Handles file saving, temporary paths for videos, and timing operations

requests	Used to fetch geolocation information via IP
datetime	Adds timestamps to detections
csv	Logs detection metadata such as location, timestamp, event type, and image path
hashlib	Used for password hashing in role-based login system
reportlab	Utilized to generate structured PDF reports for administrators

2.6 Alert and Reporting System (Telegram API, CSV logs, PDF reports)

The project's alert and reporting system enhances real-time responsiveness and data management by integrating Telegram API, CSV logging, and PDF report generation [9]. When the YOLOv8 model detects cigarette smoking, it sends an automated alarm to a specified Telegram channel, which includes a timestamp, location information, and the detected image, allowing monitoring workers to receive a quick notification. Each detection event is simultaneously stored in a CSV file, which includes critical parameters such as the date, time, detection class, and picture path, enabling systematic data recording and future analysis. A PDF report creation module gathers this information into a readable way, including captured frames, detection summaries, and location data.

2.7 Hardware Specifications

Table 2 shows the hardware requirements to execute this work.

Table 2 Hardware Requirements

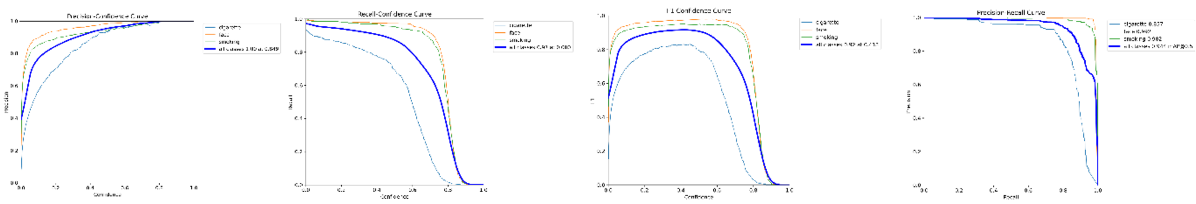
Hardware	Specifications
MSI GF63 12UCX 422 Thin	CPU: Intel Core i5-12450H Installed Memory: 16GB Graphics Card: RTX 2050 4GB Operating System: Windows 11 64-bit

3. Result and Discussion

This chapter examines and evaluates the trained YOLOv8n model for detecting cigarette smoking actions. It discusses the results achieved during the training and validation processes, as well as the model's performance using conventional metrics such as precision, recall, and mAP. Additionally, visual results from test images and video feeds are examined to determine the model's performance in real-life scenarios.

3.1 Model Performance Metrics

The performance of the YOLOv8n-based cigarette smoking detection system was evaluated using key object detection metrics, and it demonstrated high reliability for real-time applications. As shown in Fig. 3. the model achieved a precision of 0.903, recall of 0.768, and mAP@0.5 of 0.857 for cigarette detection, demonstrating efficient recognition with few false positives. For example, it achieved a mAP@0.5 of 0.982 for smoking activity detection, demonstrating its capacity to detect complex visual cues such as smoke patterns and hand-to-mouth gestures. The average inference speed of 2.2 milliseconds per image demonstrates its applicability for real-time surveillance. Furthermore, continuous monitoring using precision, recall, F1-score, and confusion matrix analysis confirmed the model's balanced performance with no significant bias or overfitting [2], [5].

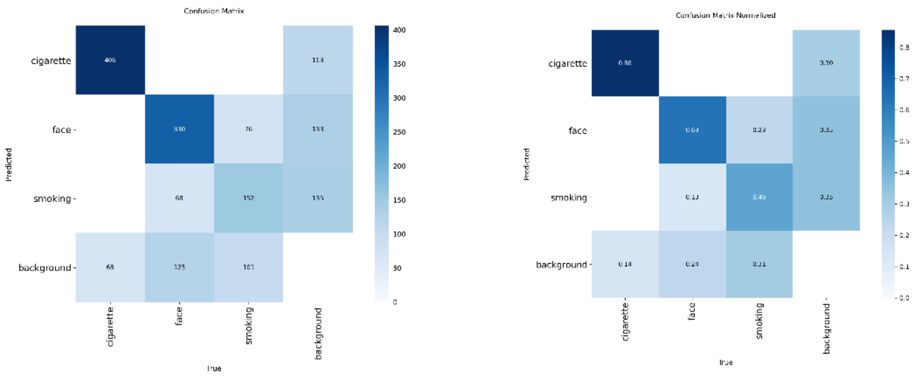


(a) (b) (c) (d)

Fig. 3 *Model Performance Metrics (a) P Curve; (b) R Curve; (c) F1 Curve; (d) PR Curve*

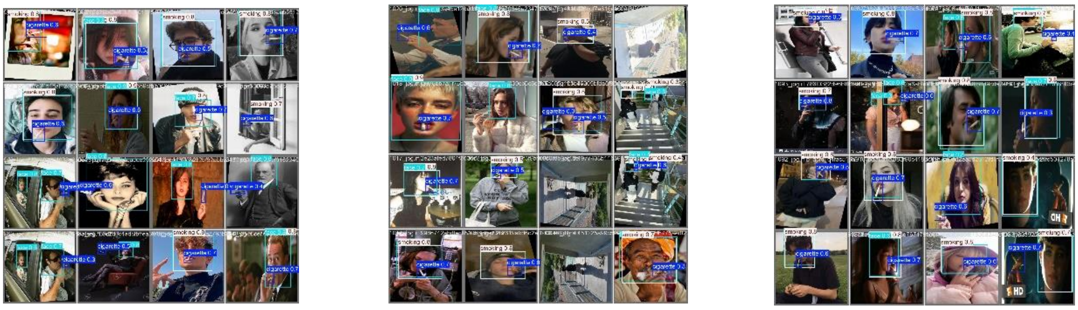
3.2 Visual Results

The system's visual results in Figures 4 and 5 confirm the efficiency of the YOLOv8 model in detecting cigarette smoking behavior. In both uploaded videos and real-time webcam streams, sample outputs show bounding boxes that are precisely drawn around cigarettes, faces, and smoking movements. These visuals not only show good multi-class identification, but also the model's ability to pinpoint objects under different lighting situations and angles. Furthermore, the confusion matrix [2] created from the validation set has a high true positive rate and few inaccurate classifications, demonstrating the model's reliability across all three detection classes. These images provide strong qualitative evidence to support the quantitative metrics obtained during the evaluation process.



(a) **(b)**

Fig. 4 *Visual Results (a) Confusion Matrix; (b) Confusion Matrix Normalized*

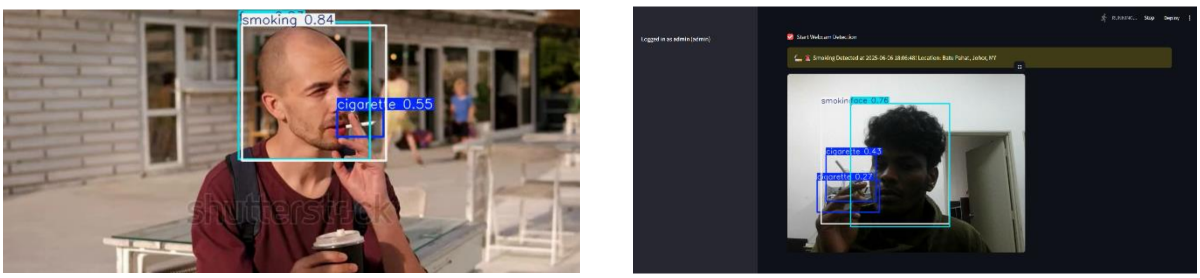


(a) **(b)** **(c)**

Fig. 5 *Visual Results (a) Test Image Batch 0; (b) Test Image Batch 1; (c) Test Image Batch 2*

3.3 Real-Time Detection Demo Results

The system's real-time detection capacity was shown utilizing both uploaded video files and live webcam feeds transmitted over a Streamlit-based interface. The trained YOLOv8n model detected smoking-related activities such as holding a cigarette and doing smoking motions with great accuracy. As shown in Fig. 6, during testing, bounding boxes were drawn precisely around cigarettes, faces, and visible smoke in a variety of lighting conditions and angles, suggesting strong model generalization. The system responded quickly, demonstrating its suitability for practical use in surveillance applications [5], [10]. Furthermore, upon detection, automated Telegram warnings were sent, and the corresponding detection events were recorded for future reporting. This real-time feature proved the system's capacity to produce consistent and fast monitoring results.

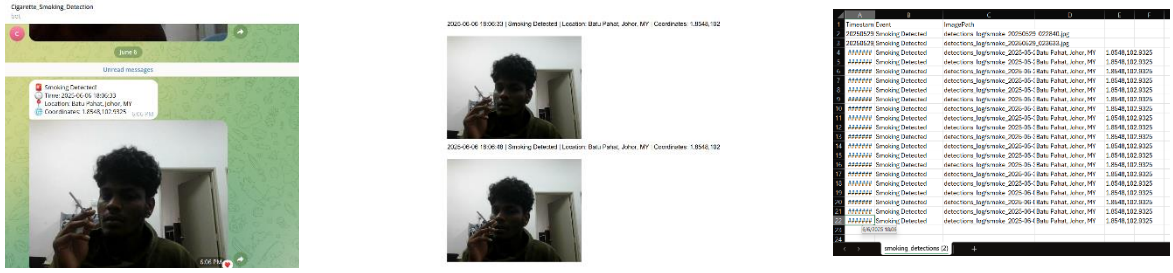


(a) **(b)**

Fig. 6 *Real-Time Detection Results (a) Real-Time Video File Detection; (b) Real-Time Webcam Detection*

3.4 Alert and reporting system (Telegram API, CSV logs, PDF reports)

To improve real-time response and ensure valid documentation, the system includes an automated alarm and reporting mechanism [9]. The Telegram Bot API is used to alert administrators immediately when smoking behavior is identified, with alerts containing the timing, approximate position (using IP-based geolocation), and detected image. Simultaneously, each detection event is saved in a structured CSV file that includes the date, time, object class, and image location for traceability and further analysis. In addition, the system includes a PDF report creation feature that uses the ReportLab library, allowing administrators to compile detection logs and related photos into a downloadable, organized format. The alert and reporting system for this proposed work is shown in Fig. 7.



(a) **(b)** **(c)**

Fig. 7 *Alert and Reporting System (a) Telegram Notification; (b) PDF Report; (c) CSV Detection Log*

3.5 Comparison Between Existing Work and Proposed System

This implementation outperformed previous efforts by Nazari et al. [2] and Sharma et al. [5] in terms of detection accuracy across a wide range of illumination and camera angles as shown in Table 3. YOLOv8n also had better latency performance and could handle smaller object sizes due to its decoupled head architecture. Smoke opacity and extremely low-light conditions did, however, occasionally interfere with detection.

Table 3 *Comparison*

Criteria	Nazari et al.2020 [4]	YOLOv8
Precision	90.02%	93.73%
Recall	87.53%	90.36%
F1-Score	88.75%	91.86%
Accuracy/mAP@0.5	91.47%	94.36%

4. Conclusion

In summary, the YOLOv8 object detection model combined with a Streamlit web interface and Telegram alerting features allowed this project to successfully create a real-time cigarette smoking detection system. With excellent precision, recall, and mAP scores, the model proved that it could reliably identify smoking-related behaviors like cigarette presence and smoke gestures in a variety of environmental and lighting conditions. Features like video upload processing, CSV logging, real-time webcam detection, and automated PDF report generation were added to improve the system's usability and suitability for actual surveillance situations. Even though the system showed promising performance, it could be improved in the future by adding additional events to the dataset and increasing the detection speed for edge deployment. Future study could focus on improving the system for embedded platforms such as the Jetson Nano or Raspberry Pi, allowing for city-wide deployment through integration with CCTV systems. Furthermore, enabling the system to detect other forbidden actions such as vaping and littering will improve its utility in public policy enforcement.

Acknowledgement

The author gratefully acknowledges the guidance and support and the facilities made available by the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia.

Conflict of Interest

The author state that there are no conflicts of interest in the publishing of this research. All actions, findings, and conclusions are given honestly, with no personal or financial links that could influence the study's outcomes.

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation

References

[1] L. Alzubaidi *et al*., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Art. no. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[2] A. A. Nazari, M. A. Khan, and T. Saba, "Cigarette smoking detection in surveillance videos using deep learning," *IEEE Access*, vol. 8, pp. 120104–120114, 2020, doi: 10.1109/ACCESS.2020.3005076.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[4] G. Jocher *et al*., "YOLO by Ultralytics," *GitHub Repository*, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Nov 30, 2025]

[5] P. Sharma and S. S. Sahu, "Automated real-time surveillance system using object detection," *Procedia Computer Science*, vol. 167, pp. 2594–2603, 2020, doi: 10.1016/j.procs.2020.03.316.

[6] A. Jain and P. Ghosh, "Roboflow: Computer vision workflow made easy," *Roboflow Blog*, 2021. [Online]. Available: https://blog.roboflow.com/. [Accessed: Sept 30, 2025].

[7] J. Howard and S. Gugger, *Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD*. Sebastopol, CA, USA: O'Reilly Media, 2020.

[8] Streamlit Inc., "Streamlit Documentation." [Online]. Available: https://docs.streamlit.io/. [Accessed: Sept 30, 2025].

[9] Telegram, "Telegram Bot API," [Online]. Available: https://core.telegram.org/bots/api. [Accessed: Sept 29, 2025].

[10] OpenCV, "OpenCV Library," [Online]. Available: https://opencv.org/. [Accessed: Sept 20, 2025].