ELKOM (Jurnal Elektronika dan Komputer)
p-ISSN : 1907-0012 (print) e-ISSN : 2714-5417 (online)
Vol. 19, No. 1, Juli 2026
https://journal.stekom.ac.id/index.php/elkom

Real-Time Human Detection and Face Recognition System Using CCTV Stream and Localhost-Based Monitoring Dashboard

M. A. Racka Eratama[1], Ade Silvia handayani[2*], Aryanti Aryanti[3] Asriyadi Asriyadi[4]
[1] Telecommunication Engineering Study Program, Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Indonesia
[2,3] Telecommunication Engineering Study Program, Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Indonesia
[4] Electronics And Communication Engineering Department, King Abdul Aziz University, Arab saudi
Email author: rackaeratama07@gmail.com[1], ade_silvia@polsri.ac.id[2], aryanti@polsri.ac.id[3], aasriyadi@stu.kau.edu.sa[4]

Article Info

Article history:

Received June 26, 2026
Revised June 29, 2026
Accepted July 3, 2026
Available: July 13, 2026
Published: July 30, 2026

Keywords:

Computer Vision;
Human Detection;
Face Recognition;
Closed-Circuit Television;
Localhost Dashboard

ABSTRACT

This study presents a real-time human detection and face recognition system that utilizes CCTV video streams and a localhost-based monitoring dashboard. Using a Research and Development (R&D) approach, a computer vision application was designed and implemented on a laptop platform. Human detection was performed using YOLO11n, facial regions were localized with YuNet, and face recognition was carried out using SFace to distinguish registered individuals from unknown persons. The detection results were displayed through a web-based dashboard that provided live video streaming, AI status information, face registration, and detection history records. Performance evaluation was conducted under various conditions, including human and non-human scenarios, registered and unknown faces, dark-room environments, and night-vision mode. The dashboard maintained a preview rate of approximately 20–30 FPS. Experimental results showed that human detection achieved an accuracy of 80%, while face recognition achieved 72% accuracy under the tested conditions. Alert Level 1 was triggered when a person was detected, whereas Alert Level 2 was activated for unknown-face events. The findings demonstrate the potential of integrating lightweight computer vision models into a local surveillance system without relying on cloud infrastructure. Nevertheless, system performance remained dependent on factors such as lighting conditions, camera distance, face orientation, image quality, and available computing resources.

Corresponding Author:

Ade Silvia Handayani,
Politeknik Negeri Sriwijaya, Jl. Srijaya Negara, Bukit Lama, Kec. Ilir Barat I, Kota Palembang, Sumatera Selatan 30139, Indonesia.
Email: ade_silvia@polsri.ac.id

1. INTRODUCTION

Closed- Closed-circuit television (CCTV) has become a fundamental surveillance technology in residential, academic, industrial, and public environments. However, conventional CCTV systems still mainly function as passive recording tools that depend on continuous human observation. This limitation can reduce monitoring effectiveness because operators may lose attention during long-duration surveillance, causing important events such as human presence, unknown individuals, or suspicious activity to be missed. Therefore, intelligent video analytics, edge artificial intelligence, anomaly detection, and face recognition have become important approaches to support automatic visual interpretation, real-time event detection, and faster security response in surveillance systems (Awad et al., 2024; Badidi et al., 2023; Duja et al., 2024; Duong et al., 2023; Rahim et al., 2023; Zennayi et al., 2023). Deep learning has significantly improved computer vision because it can learn visual features directly from image and video data. In surveillance systems, object detection is essential for identifying the presence and location of objects before further analysis is performed. Previous studies explain that deep learning-based object detection has evolved from traditional feature-based methods into convolutional neural network architectures with one-stage and two-stage detection pipelines (Jiao et al., 2019; Sarker, 2021; Zhao et al., 2019). However, practical surveillance environments still face challenges such as occlusion, illumination variation, object scale, and camera viewpoint, which can affect detection reliability (El-Alami et al., 2024; Ouardirhi et al., 2024). The You Only Look Once (YOLO) family is widely used for real-time object detection because it performs localization and classification in a single-stage era process. Several studies have discussed the development of YOLO architectures, including improvements in accuracy, speed, training strategy, and deployment efficiency (Hussain, 2024; Sapkota et al., 2025; Terven et al., 2023). Lightweight object detection models are also important for edge and local devices because surveillance applications require low-latency inference with limited computing resources (Mittal, 2024). Recent studies further show that YOLO-based models can support person detection, real-time visual analysis, and dashboard-based monitoring in practical environments (Safre et al., 2025; Septria et al., 2025; Wang et al., 2025; Zhang et al., 2025). Therefore, this study employed YOLO11n to detect the person class from CCTV frames because the nano variant is suitable for local laptop-based implementation and real-time monitoring requirements. After human detection, face detection and face recognition are required to determine whether the detected person is registered or unknown. Previous research has shown that web-based real-time face recognition using YOLO can support building access security by identifying registered and unregistered individuals automatically (Indra et al., 2025). YuNet was selected because it is a lightweight face detector designed for real-time and resource-constrained environments (Wu et al., 2023). For recognition, embedding-based methods are commonly used because they represent facial identity in a compact feature space and compare similarity between detected and registered faces. Previous studies show that robust facial embeddings can improve identity matching under different visual conditions (Deng et al., 2022; Gururaj et al., 2024; Minaee et al., 2023; Wang & Deng, 2021). In this study, SFace was applied to extract facial features and perform similarity matching because it supports robust face recognition through discriminative hypersphere-based representation learning (Zhong et al., 2021). Although computer vision-based surveillance provides significant benefits, face recognition systems must consider privacy, ethical issues, and responsible deployment because visual identity data are sensitive and may create risks if processed without proper control (Wang et al., 2024). In addition, previous studies commonly focused on separate tasks such as object detection, anomaly detection, or face recognition, whereas the integration of human detection, face detection, face recognition, event logging, and dashboard visualization in a local laptop-based CCTV system remains a practical research gap. Based on this gap, this study proposes a real-time CCTV-based human detection and face recognition system integrating YOLO11n, YuNet, SFace, and a localhost monitoring dashboard. The main contribution of this research is the implementation of lightweight computer vision models within a single laptop-based surveillance framework, while its novelty lies in local real-time monitoring without requiring cloud infrastructure.

p-ISSN: 2714-5417
e-ISSN: 1907-0012

2. METHOD

This study used the Research and Development (R&D) method. The method was used to design and develop a local computer vision monitoring application for real-time human detection and face recognition using CCTV stream input and a localhost-based dashboard.

2.1. System Design

Metode The system was designed as a local computer vision monitoring application that received input from a closed-circuit television stream and processed it on a laptop. The video stream was displayed through a web dashboard running on a localhost server. The dashboard consisted of a live preview area, artificial intelligence status panel, face registration form, and detection history table.).

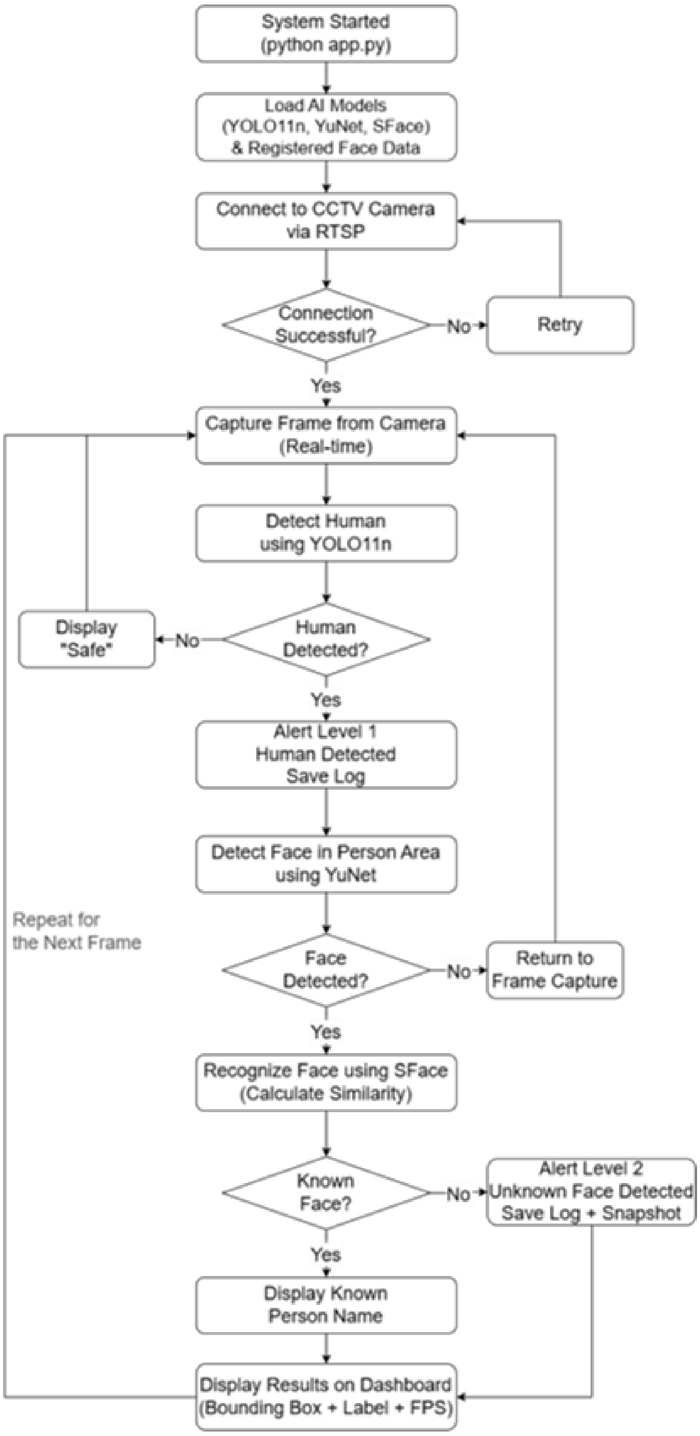


Figure 1. Flowchart System

Based on the flowchart, the process started when the system was executed through python app.py. After that, the system loaded the AI models, namely YOLO11n, YuNet, and SFace, together with the registered face data. The next step was connecting the system to the CCTV camera through RTSP. If the connection failed, the system retried after 3 seconds. If the connection was successful, the system captured frames from the camera in real time. Each captured frame was first processed for human detection using YOLO11n. If no human was detected, the system displayed the status "Safe" and continued to the next frame. If a human was detected, the system activated Alert Level 1 and saved the log. After that, the system performed face detection in the human area using YuNet. If no face was detected, the system returned to frame capture. If a face was detected, the system performed face recognition using SFace by calculating similarity. If the face was not recognized, the system activated Alert Level 2 and stored the log and snapshot. If the face was recognized, the system displayed the name of the known person. Finally, the detection results were shown on the dashboard in the form of a bounding box, label, and FPS. This process was repeated continuously for the next frame.

2.2. Human Detection Using YOLO11n

Human detection was performed using YOLO11n, a lightweight object detection model from the YOLO family. The model architecture consists of four main stages: input pre-processing, backbone, neck, and detection head. CCTV frames were first resized and normalized before being processed by the model. The backbone extracted visual features from the frame, while the neck combined multi-scale features to improve object localization. The detection head then predicted the bounding box, class label, and confidence score. In this study, the detection output was filtered only to the person class because the system focused on identifying human presence in the monitored area.

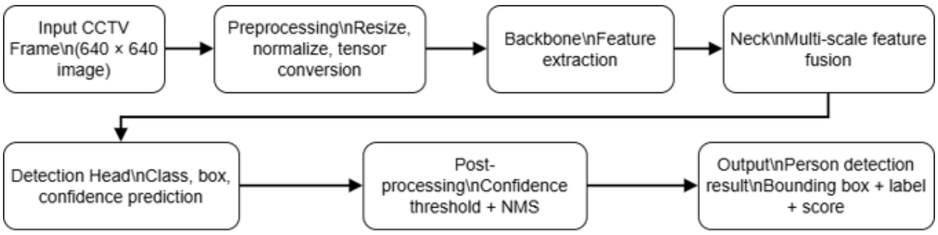


Figure 2. YOLO11n architecture for human detection

The YOLO11n architecture for human detection consists of several sequential stages, as illustrated in Figure 2. The process begins with an input CCTV frame with a resolution of 640 × 640 pixels. The frame is first passed through a preprocessing stage, where resizing, normalization, and tensor conversion are performed to make the image compatible with the model input. The processed frame is then forwarded to the backbone, which extracts important visual features such as edges, shapes, and object patterns. These features are further processed in the neck layer, which combines multi-scale feature information to improve object localization under different object sizes and distances. The detection head then predicts the object class, bounding box coordinates, and confidence score. In this research, the output was filtered only to the person class because the system focused on human detection. After post-processing using a confidence threshold and non-maximum suppression, the final result was displayed as a human bounding box, label, and confidence score on the monitoring dashboard.

2.3. Face Recognition

Face recognition was applied after the human detection process to identify whether the detected person belonged to a registered individual or an unknown person. In this system, the face recognition stage used YuNet for face detection and SFace for facial feature extraction and identity matching. YuNet was responsible for locating the facial region from the detected human area, while SFace was used to generate facial embeddings and compare them with the registered face database.

p-ISSN: 2714-5417
e-ISSN: 1907-0012

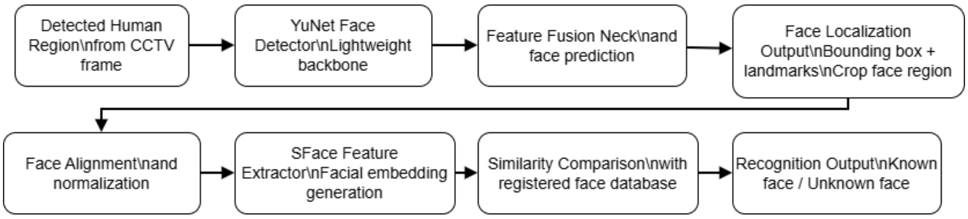


Figure 3. Face recognition architecture using YuNet and Sface

Figure 3 shows the face recognition architecture used in the proposed CCTV monitoring system. The process begins with the detected human region from the CCTV frame. YuNet then detects the facial area using a lightweight backbone and feature fusion neck, producing the face localization output in the form of a bounding box and facial landmarks. The detected face is cropped, aligned, and normalized before being processed by SFace. SFace extracts facial embeddings that represent the identity features of the detected face. These embeddings are then compared with the registered face database using a similarity comparison process. If the detected face matches the registered data, the system displays it as a known face. Otherwise, the system classifies it as an unknown face and activates Alert Level 2 on the localhost dashboard.

2.4. Testing Scenario

The system was tested using several visual conditions, including no-human condition, human detection condition, registered-face condition, unknown-face condition, dark-room condition, and night-vision condition. Each testing scenario was used to observe whether the system could display the correct detection status, bounding box, face label, alert level, and detection history. The evaluation focused on the correctness of human detection, face recognition, dashboard display, and detection history recording.

3. RESULT AND ANALYSIS

3.1. Localhost Dashboard Output

The developed dashboard displayed the live preview from the surveillance camera with an estimated website preview speed of 20-30 FPS. The right-side panel presented the artificial intelligence status, including alert status, number of detected people, number of detected faces, and last update time. The dashboard also provided a face registration feature that allowed users to upload frontal facial images for recognition data.

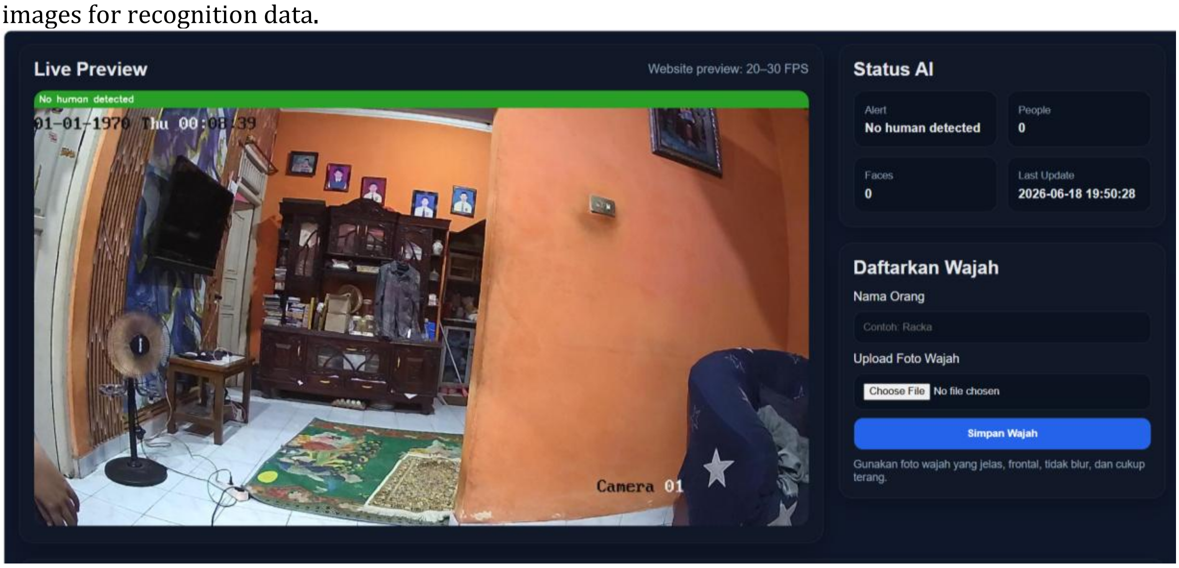


Figure 4. Dashboard condition when no human object was detected

Figure 4 shows the condition when no person was detected in the monitored area. The dashboard displayed the status “No human detected”, with the number of people and faces equal to zero. This result indicates that the system was able to represent an empty monitoring condition through the dashboard status panel.

3.2. Human Detection Result

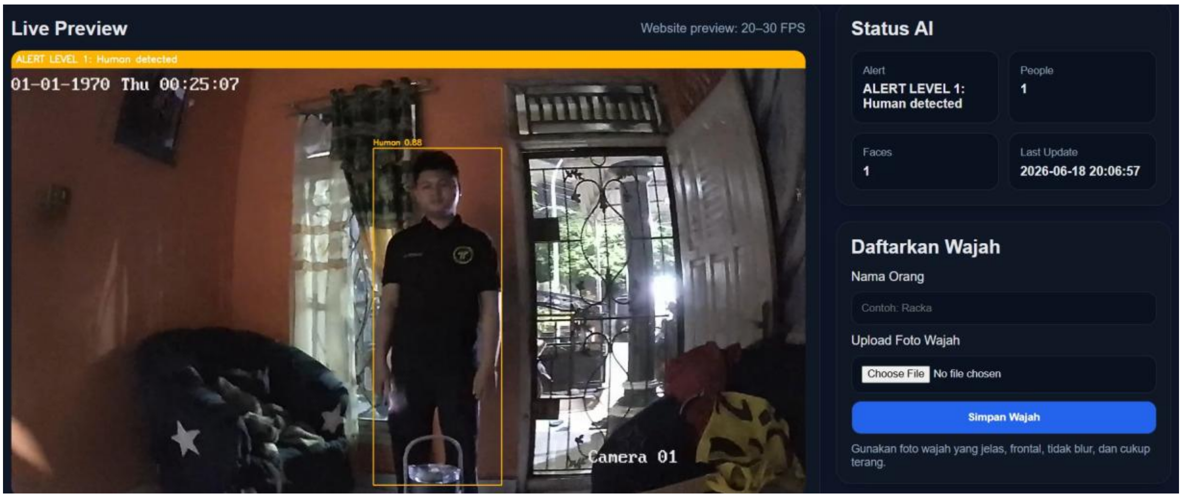


Figure 5. Example of human detection testing using YOLO11n

Figure 5 presents an example of human detection using YOLO11n. The system detected one human object and displayed a yellow bounding box containing the detected class and confidence score. This output shows that the model was able to identify the presence of a person in the CCTV frame in real time. The dashboard status also changed to Alert Level 1, indicating that a human object was detected in the monitored area and that the system successfully generated an early warning response.

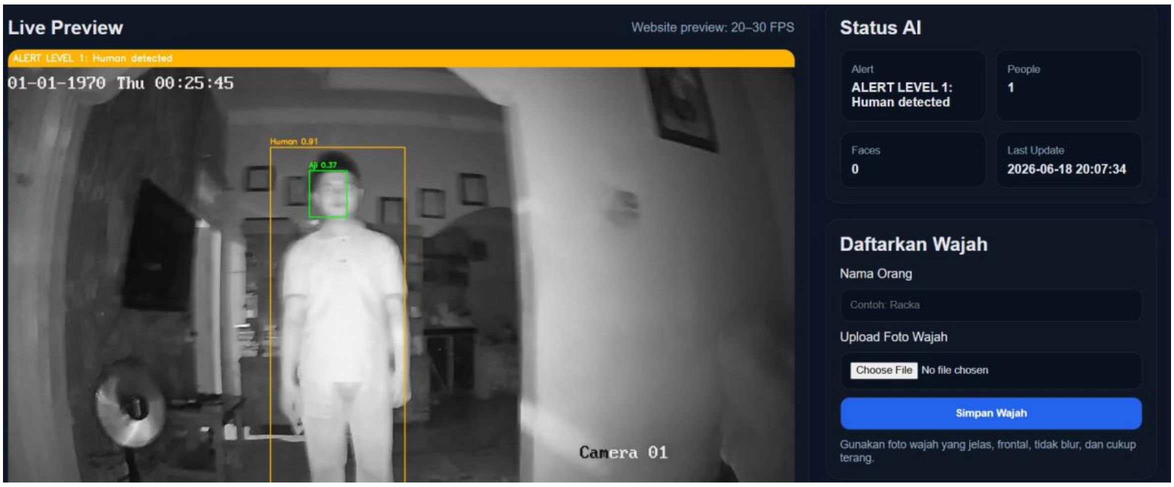


Figure 6. Human detection result in night-vision condition

Figure 6 shows the system output in night-vision mode. The human object was still detected although the image was displayed in grayscale. This result shows that the human detection module could still operate under low-light monitoring conditions, although lighting quality affected the clarity of the visual output. Therefore, the system remained functional for nighttime surveillance despite reduced image detail. This indicates that YOLO11n can still support monitoring tasks even when the input image quality is not optimal.

p-ISSN: 2714-5417
e-ISSN: 1907-0012

The accuracy of human detection was calculated using the following formula:

Accuracy = (Correct Detection / Total Test) x 100%

Human Detection Accuracy = 20 / (20 + 5) x 100% = 80%

Table 1. Human detection accuracy result

Test Type	Correct Detection	Incorrect Detection	Total Test	Accuracy
Human Detection using YOLO11n	20	5	25	80%

Based on Table 1, the human detection module achieved an accuracy of 80%. The incorrect detections were influenced by practical factors such as lighting variation, camera angle, object distance, and partial body visibility. These factors affected the ability of the detection model to identify human objects consistently in every test condition.

3.3. Face Recognition Result

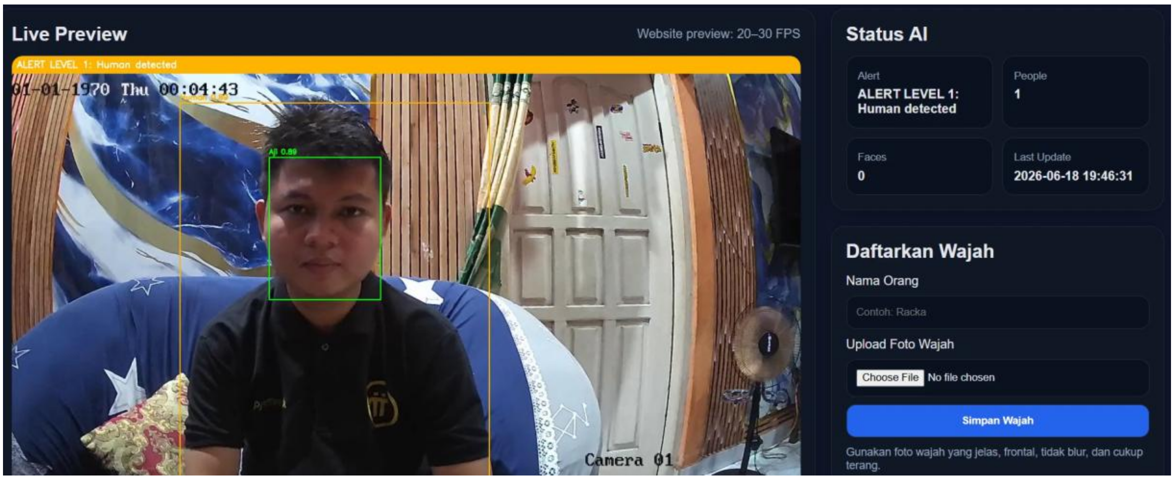


Figure 7. Example of registered face recognition testing

Figure 7 shows the face recognition result when a registered face was detected. The system displayed a face bounding box and label on the detected facial region. This output indicates that the face recognition module was able to match the detected face with registered facial data in the system.

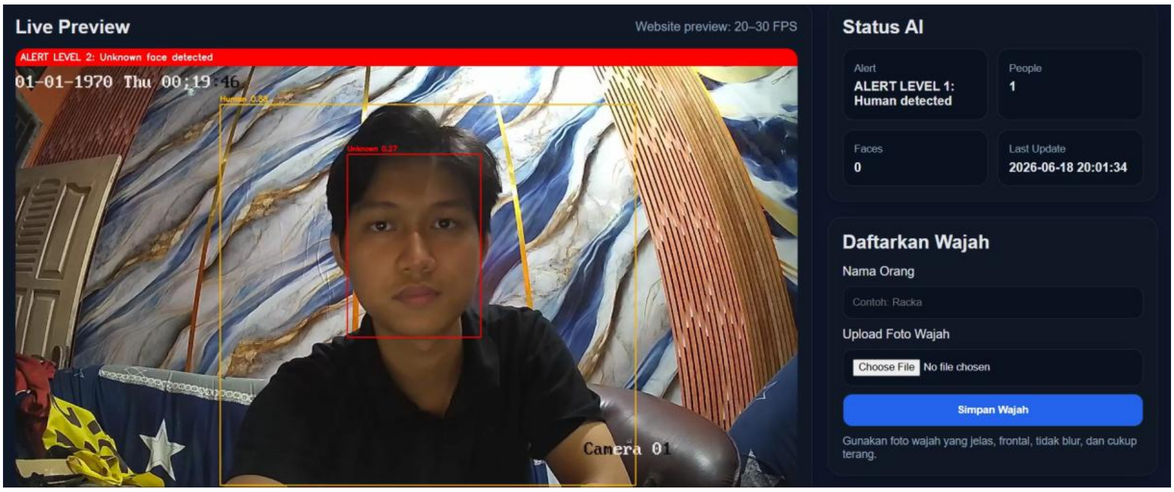


Figure 8. Example of unknown face recognition testing

Figure 8 shows the output when the detected face was not recognized as a registered person. The dashboard displayed Alert Level 2 with the message “Unknown face detected”. The face status was also recorded as unknown with a similarity score, indicating that the detected face did not match the registered facial data.

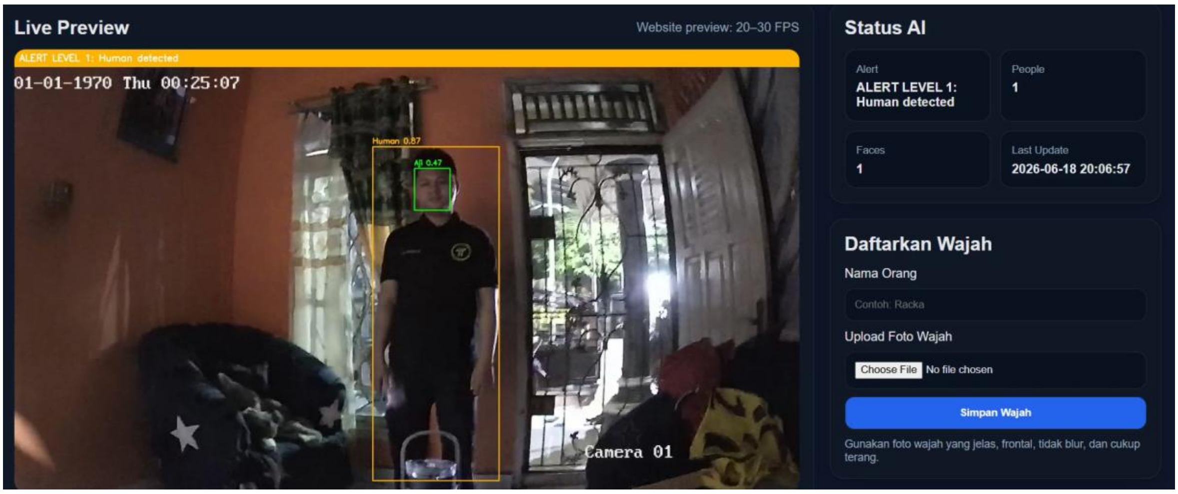


Figure 9. Face recognition result in dark-room condition

Figure 9 presents the face recognition result under darker lighting conditions. The system was still able to detect the human body and facial area, but the recognition result was affected by poor lighting and reduced facial detail. This condition shows that face orientation, image brightness, and face clarity are important factors in recognition performance.
The accuracy of face recognition was calculated using the following formula:

$$\text{Accuracy} = (\text{Correct Recognition} / \text{Total Test}) \times 100\%$$

$$\text{Face Recognition Accuracy} = 18 / (18 + 7) \times 100\% = 72\%$$

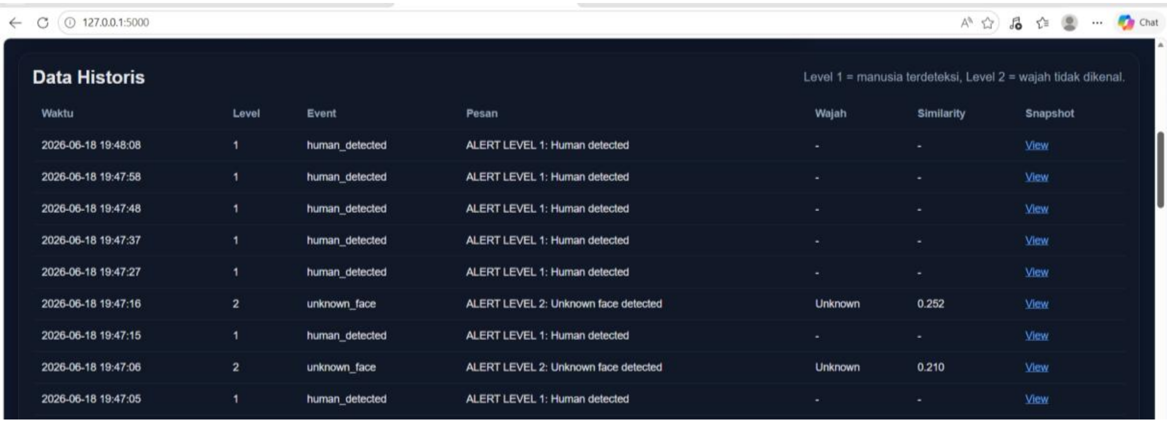
Table 2. Face recognition accuracy result

Test Type	Correct Recognition	Incorrect Recognition	Total Test	Accuracy
Face Recognition	18	7	25	72%

p-ISSN: 2714-5417
e-ISSN: 1907-0012

Based on Table 2, the face recognition module achieved an accuracy of 72%. The recognition accuracy was lower than human detection because face recognition required clearer facial features. Errors occurred when the face was too far from the camera, the lighting was insufficient, the face was not frontal, or the image was blurred.

3.4. Detection History and System Spesification



Waktu	Level	Event	Pesan	Wajah	Similarity	Snapshot
2026-06-18 19:48:08	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View
2026-06-18 19:47:58	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View
2026-06-18 19:47:48	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View
2026-06-18 19:47:37	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View
2026-06-18 19:47:27	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View
2026-06-18 19:47:16	2	unknown_face	ALERT LEVEL 2: Unknown face detected	Unknown	0.252	View
2026-06-18 19:47:15	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View
2026-06-18 19:47:06	2	unknown_face	ALERT LEVEL 2: Unknown face detected	Unknown	0.210	View
2026-06-18 19:47:05	1	human_detected	ALERT LEVEL 1: Human detected	-	-	View

Figure 8. Detection history recorded by the localhost dashboard

Figure 10 shows the detection history table recorded by the dashboard. The table stored the detection time, alert level, event type, message, face status, similarity score, and snapshot link. This feature helped users review previous detection events, especially human detection and unknown-face alerts.

Table 3. Laptop specification used for running the artificial intelligence system

Component	Specification
Device model	ASUS TUF Gaming F15 FX506HC_FX506HC
Processor	11th Gen Intel(R) Core (TM) i5-11400H @ 2.70 GHz
Installed RAM	8.00 GB
Graphics card	NVIDIA GeForce RTX 3050 Laptop GPU (4 GB) and Intel(R) UHD Graphics
Storage	477 GB
System type	64-bit operating system, x64-based processor

Table 3 shows the laptop specification used to run the artificial intelligence system. The system was tested on a laptop with an 11th Gen Intel Core i5-11400H processor, 8 GB RAM, and NVIDIA GeForce RTX 3050 Laptop GPU with 4 GB graphics memory. The use of a laptop-based processing device shows that the system can be implemented locally without requiring a cloud server.

Overall, the testing results showed that the proposed system successfully displayed real-time human detection and face recognition results through a localhost dashboard. Human detection achieved higher accuracy than face recognition because human detection relied on larger body features, while face recognition depended on smaller and more sensitive facial features. Therefore, the quality of the camera image, lighting condition, face angle, and distance from the camera were important factors influencing system performance.

4. DISCUSSION/CONCLUSION

This study successfully implemented a real-time human detection and face recognition system using YOLO11n, face recognition, closed-circuit television input, and a localhost-based dashboard. The system was able to detect human objects, recognize registered faces, identify unknown faces, display alert levels, and record detection history. Based on the testing results, the human detection module achieved an accuracy of 80%, while the face recognition module achieved an accuracy of 72%. The results indicate that human detection was more stable than face recognition because the detected body area was larger and less sensitive to facial detail. In contrast, face recognition was strongly influenced by lighting conditions, face orientation, camera distance, and image quality. Future development can include additional datasets, improved face registration, alert notification, and deployment on edge computing devices to improve speed and reliability.

REFERENCES

[1] A. I. Awad, A. Babu, E. Barka, and K. Shuaib, “AI-powered biometrics for Internet of Things security: A review and future vision,” Journal of Information Security and Applications, vol. 82, p. 103748, 2024, doi: 10.1016/j.jisa.2024.103748.
[2] E. Badidi, K. Moumane, and F. E. Ghazi, “Opportunities, applications, and challenges of edge-AI enabled video analytics in smart cities: A systematic review,” IEEE Access, vol. 11, pp. 80543–80572, 2023, doi: 10.1109/ACCESS.2023.3300658.
[3] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 10, pp. 5962–5979, 2022, doi: 10.1109/TPAMI.2021.3087709.
[4] K. U. Duja, I. A. Khan, and M. Alsuhaibani, “Video surveillance anomaly detection: A review on deep learning benchmarks,” IEEE Access, vol. 12, pp. 164811–164842, 2024, doi: 10.1109/ACCESS.2024.3491868.
[5] H. T. Duong, V. T. Le, and V. T. Hoang, “Deep learning-based anomaly detection in video surveillance: A survey,” Sensors, vol. 23, no. 11, p. 5024, 2023, doi: 10.3390/s23115024.
[6] A. El-Alami, Y. Nadir, and K. Mansouri, “A review of object detection approaches for traffic surveillance systems,” International Journal of Electrical and Computer Engineering, vol. 14, no. 5, pp. 5221–5233, 2024, doi: 10.11591/ijece.v14i5.pp5221-5233.
[7] H. L. Gururaj, B. C. Soundarya, S. Priya, J. Shreyas, and F. Flammini, “A comprehensive review of face recognition techniques, trends and challenges,” IEEE Access, vol. 12, pp. 107903–107926, 2024, doi: 10.1109/ACCESS.2024.3424933.
[8] M. Hussain, “YOLOv1 to v8: Unveiling each variant—A comprehensive review of YOLO,” IEEE Access, vol. 12, pp. 42816–42833, 2024, doi: 10.1109/ACCESS.2024.3378568.
[9] F. S. Indra, A. Aryanti, and R. A. Halimatussa’diyah, “Sistem pengenalan wajah real-time menggunakan YOLOv7 untuk akses gedung TVRI Palembang berbasis web,” Journal of Technology and Informatics (JoTI), vol. 7, no. 2, pp. 147–160, 2025, doi: 10.37802/joti.v7i2.1063.
[10] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” IEEE Access, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.
[11] P. Mittal, “A comprehensive survey of deep learning-based lightweight object detection models for edge devices,” Artificial Intelligence Review, vol. 57, p. 242, 2024, doi: 10.1007/s10462-024-10877-1.
[12] Z. Ouardirhi, S. A. Mahmoudi, and M. Zbakh, “Enhancing object detection in smart video surveillance: A survey of occlusion-handling approaches,” Electronics, vol. 13, no. 3, p. 541, 2024, doi: 10.3390/electronics13030541.
[13] A. Rahim, Y. Zhong, T. Ahmad, S. Ahmad, P. Plawiak, and M. Hammad, “Enhancing smart home security: Anomaly detection and face recognition in smart home IoT devices using logit-boosted CNN models,” Sensors, vol. 23, no. 15, p. 6979, 2023, doi: 10.3390/s23156979.
[14] A. L. S. Safre, A. Torres-Rua, B. L. Black, and S. Young, “Deep learning framework for fruit counting and yield mapping in tart cherry using YOLOv8 and YOLO11,” Smart Agricultural Technology, vol. 11, p. 100948, 2025, doi: 10.1016/j.atech.2025.100948.
[15] R. Sapkota, M. Flores-Calero, R. Qureshi, C. Badgujar, U. Nepal, A. Poulose, P. Zeno, U. B. P. Vaddevolu, S. Khan, M. Shoman, H. Yan, and M. Karkee, “YOLO advances to its genesis: A decadal and comprehensive review of the You Only Look Once (YOLO) series,” Artificial Intelligence Review, vol. 58, p. 274, 2025, doi: 10.1007/s10462-025-11253-3.
[16] I. H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions,” SN Computer Science, vol. 2, p. 420, 2021, doi: 10.1007/s42979-021-00815-1.

[17] A. A. Septria, A. S. Handayani, and N. Nasron, “Real-time retail shelf-stock detection with YOLOv7,” Jurnal Teknik Informatika, vol. 18, no. 2, pp. 327–338, 2025, doi: 10.15408/jti.v18i2.46448.
[18] J. Terven, D. M. Cordova-Esparza, and J. A. Romero-Gonzalez, “A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” Machine Learning and Knowledge Extraction, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.
[19] M. Wang and W. Deng, “Deep face recognition: A survey,” Neurocomputing, vol. 429, pp. 215–244, 2021, doi: 10.1016/j.neucom.2020.10.081.
[20] Q. Wang, G. Feng, and Z. Li, “A lightweight person detector for surveillance footage based on YOLOv8n,” Sensors, vol. 25, no. 2, p. 436, 2025, doi: 10.3390/s25020436.
[21] X. Wang, Y. C. Wu, M. Zhou, and H. Fu, “Beyond surveillance: Privacy, ethics, and regulations in face recognition technology,” Frontiers in Big Data, vol. 7, p. 1337465, 2024, doi: 10.3389/fdata.2024.1337465.
[22] W. Wu, H. Peng, and S. Yu, “YuNet: A tiny millisecond-level face detector,” Machine Intelligence Research, vol. 20, pp. 656–665, 2023, doi: 10.1007/s11633-023-1423-y.
[23] Y. Zennayi, S. Benaissa, H. Derrouz, and Z. Guennoun, “Unauthorized access detection system to the equipments in a room based on the persons identification by face recognition,” Engineering Applications of Artificial Intelligence, vol. 124, p. 106637, 2023, doi: 10.1016/j.engappai.2023.106637.
[24] H. Zhang, G. Zhou, W. He, and H. Deng, “Classroom behavior detection method based on PLA-YOLO11n,” Sensors, vol. 25, no. 17, p. 5386, 2025, doi: 10.3390/s25175386.
[25] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.
[26] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li, and D. Wen, “SFace: Sigmoid-constrained hypersphere loss for robust face recognition,” IEEE Transactions on Image Processing, vol. 30, pp. 2587–2598, 2021, doi: 10.1109/TIP.2020.3048632.