AI-Based Autonomous Drone Surveillance System for Restricted-Area Intrusion Detection

Prof. Pramila R , Prof. Lavanya D

Anjali Dayanand Kamble, Khushal, Prerana Pragathi, Nikhil S

Department of Artificial Intelligence and Machine Learning, Engineering College, Bengaluru, India

{anjali.kamble, Khushal, prerana.pragathi, nikhil.s}@ips.xyz.edu

***Abstract*—*Guarding restricted or hard-to-reach sites — defence installations, industrial facilities, and other critical infrastructure — still depends largely on static CCTV networks and human patrols, both of which leave gaps in coverage and are susceptible to observer fatigue. This work describes an autonomous surveillance drone that brings together GPS-guided waypoint flight, live human detection, face-based authorization screening, cloud-hosted evidence logging, and immediate mobile alerts within one measurable pipeline. Flight is handled by a Pixhawk controller running ArduPilot, which steers the aircraft between preset waypoints; the onboard camera feed is carried over a TS832 5.8 GHz analog transmitter to an RC832H/RC832AV ground receiver, where a ground-station computer runs YOLOv8 to spot people and YuNet plus ArcFace to check their faces against an authorized list, sorting each detected individual as authorized or unauthorized. Whenever an intruder is flagged, the system captures a photo, records the GPS fix and time, pushes the evidence to Firebase, and fires an alert to a Flutter mobile app that also offers live streaming and status monitoring. The aircraft reverts to Return-to-Launch (RTL) automatically once its mission finishes, its battery runs low, or telemetry drops. This paper lays out the mathematics behind each stage of the pipeline — the Haversine distance formula for waypoint tracking, YOLO's objectness/IoU/precision-recall relations, cosine-similarity face matching, and the ArcFace additive-angular-margin loss — and presents performance curves for detection confidence, precision-recall, similarity separation, and behaviour under different lighting, alongside a summary of results by module.*

***Index Terms*—*Autonomous UAV, Pixhawk, ArduPilot, GPS waypoint navigation, YOLOv8, YuNet, ArcFace, face recognition, restricted-area surveillance, intrusion detection, Firebase cloud, Flutter mobile application, emergency alerting.*

I. Introduction

Keeping restricted and remote locations secure is an increasingly pressing challenge for defence sites, industrial plants, and critical-infrastructure operators. The conventional approach — fixed CCTV cameras — only ever sees a limited footprint and needs constant human attention, and consequently suffers from blind spots, fatigue-driven oversights, and slow reactions to break-ins. Earlier research has chipped away at pieces of this problem: CCTV-based human detection with YOLO for spotting anomalies in restricted zones [4], YOLOv8 used to find people in aerial/drone footage for search-and-rescue and reconnaissance work [5], [6], Pixhawk-class autopilots for GPS-directed waypoint flight [7], and Firebase-driven real-time alerting for IoT and phone-based security systems [8], [9].

Progress in AI, computer vision, and UAV hardware now makes it possible to combine these pieces into one mobile, intelligent surveillance platform. Extending the directions noted above, this paper describes an autonomous surveillance drone that folds GPS waypoint flight, live human detection, face recognition, cloud-based evidence storage, and instant mobile alerting into a single working system, and spells out the mathematics driving detection and recognition throughout the pipeline.

The aircraft is guided along preset GPS waypoints by a Pixhawk flight controller running ArduPilot. Video from its onboard camera travels over a TS832 5.8 GHz analog link to an RC832H/RC832AV receiver at the ground station, where YOLOv8 [1] finds people in real time and YuNet with ArcFace [2], [3] checks faces to separate authorized personnel from intruders. Once an intruder is identified, the system captures a photo, logs the GPS position and timestamp, uploads the evidence to Firebase, and immediately notifies a Flutter mobile app that also carries the live video feed. When the mission ends or the battery gets low, the drone automatically returns to launch (RTL).

A. Objectives

- Build an autonomous surveillance drone around a Pixhawk flight controller with GPS waypoint flight.
- Distinguish authorized personnel from intruders using YOLOv8 for detection and YuNet/ArcFace for face recognition.
- Capture intruder photos with GPS coordinates and store them securely in Firebase Cloud.
- Give security staff live video, instant alerts, and remote monitoring through a Flutter mobile application.
- Measure how reliable detection and recognition are using precision, recall, and similarity-separation statistics.

II. Literature Survey

A number of AI- and IoT-driven approaches to automated surveillance and identity recognition have been explored previously, each covering a slice of the problem tackled here. Jocher et al. [1] introduced YOLOv8, a single-pass object detector that has become a standard choice for embedded human/object detection thanks to a strong balance between accuracy and speed; however, the released model is a generic COCO-trained detector, not a drone-integrated identity-verification system.

Deng et al. [2] developed ArcFace, which converts a detected face into an embedding and sharpens the separation between identities through an additive angular margin loss, delivering strong face-verification results; Wu et al. [3] paired this with YuNet, a compact, millisecond-speed face detector built for constrained edge hardware. Neither line of work, though, considers aerial use, GPS-tagged alerts, or cloud syncing.

Abdul Aziz and Bajpai [4] used YOLOv8 for attire-based anomaly spotting in restricted zones from fixed CCTV footage, sharpening automated recognition of unauthorized individuals, but their approach is still tied to the fixed vantage points and blind spots of stationary cameras. Heemskerk et al. [5] tested several YOLOv8 CNN variants for spotting people at long range from drone footage to assist first responders, showing that aerial detection is workable but stopping short of identity classification, GPS-linked alerts, or mobile monitoring. Mummaneni et al. [6] similarly examined deep-learning human detection in drone imagery, reinforcing YOLOv8's fit for aerial work while leaving recognition and alert dispatch unaddressed.

A review of Pixhawk-class autopilot design [7] frames GPS/IMU sensor fusion and waypoint autonomy as a solid foundation for UAV navigation, though it stays confined to the flight-control layer with no AI component. On the notification side, Ahmed et al. [8] built a closed-loop Firebase-based home-security system with real-time Android alerts, confirming that Firebase Realtime Database and Cloud Messaging work well for low-latency IoT security notifications — though for a fixed indoor setup rather than an airborne platform. A related smartphone-based face-recognition security system [9] paired Firebase, IoT connectivity, and OpenCV for identity-based access control, again built around a stationary camera rather than an autonomous UAV.

The YOLO family has itself been surveyed extensively: successive reviews spanning YOLOv1 through the YOLOv8–v11 generation trace a steady improvement in the real-time accuracy-latency balance and an expansion into instance segmentation, pose estimation, and embedded/edge deployment [10]–[12], while Caputo et al. [13] specifically confirmed YOLO-based human detection in drone footage for search-and-rescue work. On the security front, broader surveys of agentic and autonomous UAVs for defence and perimeter monitoring [15] and of UAV intrusion-detection and counter-drone methods [14], [18], [19] point to growing interest in autonomous aerial security, but still treat detection, navigation, and communication security mostly as separate research threads rather than one integrated, measured field system. A dedicated look at the Pixhawk autopilot [16] and the MAVLink protocol specification [17] further back up the navigation and telemetry layer adopted in Section VI-A.

Taken together, these works show that the individual pieces of aerial surveillance — GPS-guided navigation, human detection, face recognition, and cloud-based alerting — have each been studied on their own, but none combine all of them, with quantified detection/recognition performance and GPS-tagged mobile alerts, into one low-cost autonomous aerial platform. That gap is what the proposed system addresses.

III. Problem Identification

- Continuous surveillance of restricted and remote areas is still handled mainly through fixed CCTV networks and manual patrols.
- Stationary cameras only cover their mounted vantage points, leaving blind spots over large or irregular terrain.
- Watching multiple live feeds continuously is tiring and error-prone over time.
- Most existing systems have no built-in human detection or face-based authorization check.
- There is no single real-time alerting setup that ties together GPS tagging, cloud evidence storage, and instant mobile notification.
- Coverage that scales with the number of installed cameras also drives up installation and upkeep costs.

IV. Limitations of Existing Systems

A. Limited Spatial Coverage

Fixed CCTV units only see their mounted vantage points, so large, irregular, or hard-to-reach restricted areas end up with blind spots, and the cameras cannot be redirected to track an unfolding situation.

B. Continuous Manual Monitoring

Watching several camera feeds at once is labour-intensive; tired or inattentive observers can miss or delay spotting an intrusion.

C. No Automated Recognition

Most fixed-camera setups have no automated, AI-based human detection or face recognition, so telling authorized staff from intruders still comes down to a person's judgement.

D. No Unified Alerting Mechanism

Current solutions typically lack one pipeline that ties GPS-tagged evidence capture, cloud storage, and instant mobile notification together, which slows how quickly security staff can respond to a confirmed intrusion.

V. Proposed System

- Autonomous Aerial Platform — a Pixhawk/ArduPilot-controlled aircraft patrols a restricted zone along preset GPS waypoints with no continuous human piloting.
- Analog FPV Video Link — the onboard camera stream travels over a TS832 5.8 GHz video transmitter and is picked up at the ground station by an RC832H/RC832AV receiver, linking the aircraft to the AI-processing ground station.
- AI-Based Human Detection — the ground-control computer runs YOLOv8 on the incoming video feed to spot people in real time.
- Face Detection and Recognition — YuNet finds faces within a detected person's region, and ArcFace builds an embedding that is compared against an authorized-face database using cosine similarity.
- Intrusion Alerting with Cloud Storage — once someone is classified unauthorized, the system captures the frame, pulls GPS location and timestamp, and pushes the evidence to Firebase Storage/Firestore.
- Mobile Monitoring — a Flutter application streams live video, shows drone location and status, and pushes instant notifications of intrusion events to security staff.
- Safety and Recovery — the aircraft switches automatically to Return-to-Launch (RTL) once the mission is done, the battery runs low, or communication is lost.

The system is organized around five cooperating processes — Autonomous Navigation (P1), AI Surveillance/Detection (P2), Face Recognition and Intrusion Decision (P3), Cloud Synchronization (P4), and Mobile Monitoring (P5) — backed by data stores for flight telemetry, the YOLOv8 model, the authorized-face database, and Firebase-hosted evidence and alert logs, shown in Fig. 1. Live video moves from the drone to the ground-control station over the TS832/RC832AV analog link, while control commands travel over a separate Pixhawk/MAVLink telemetry radio; recognition results and GPS-tagged alerts computed at the ground station go to Firebase, and the mobile application pulls images, logs, and drone status from there for real-time viewing.

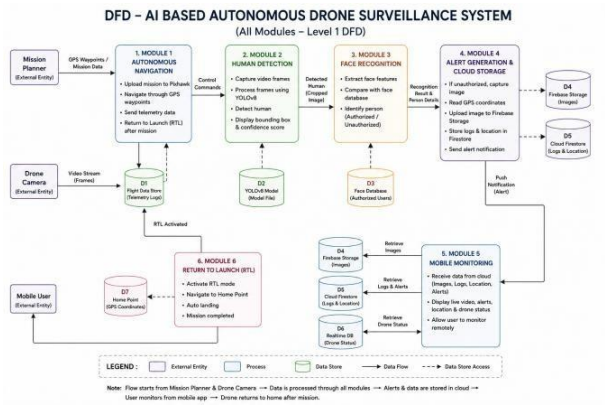

Fig. 1. System-level data flow diagram showing autonomous navigation, AI-based detection and recognition, cloud synchronization, and mobile monitoring across the five cooperating modules.

VI. System Modules and Methodology

A. Autonomous Navigation and Waypoint Control

The drone tracks GPS waypoints in Pixhawk/ArduPilot's AUTO mode. An operator lays out GPS waypoints in Mission Planner and uploads them to Pixhawk over MAVLink; the aircraft then arms itself, runs pre-flight checks (battery, GPS lock, compass, IMU), takes off, and flies the mission on its own. The great-circle distance to the next waypoint comes from the Haversine formula, where φ is latitude, λ is longitude, and r ≈ 6371 km is Earth's radius:

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

$$d = 2r \cdot \arcsin(\sqrt{a}) \quad (2)$$

Once d drops under an acceptance radius (1.5 m here), the drone moves to the next waypoint, and switches to Return-to-Launch (RTL) once the mission queue empties, battery voltage falls below a safe threshold, or telemetry drops. The Pixhawk-class autopilot combines IMU, magnetometer, barometer, and GPS sensor fusion for attitude and position estimation [7], [16], exchanging mission uploads and telemetry with Mission Planner over the MAVLink protocol [17]. Hardware used: Pixhawk 4 flight controller, GPS module, and Mission Planner ground-control software.

B. Video Downlink and Human Detection (YOLOv8)

Rather than process video onboard, the aircraft sends its camera feed live over a TS832 5.8 GHz analog transmitter to a paired RC832H/RC832AV receiver at the ground station. The analog signal is digitized there and fed to a YOLOv8 Nano model running on the ground-control computer, which detects people in the live stream in real time. Each frame is preprocessed and run through the network, which outputs bounding boxes along with an objectness score and class probability. A predicted box B_p only counts if its Intersection-over-Union (IoU) with the matching ground-

truth box B^{gt} clears a threshold and its detection confidence exceeds 0.5:

$$IoU = Area(B_p \cap B^{gt}) / Area(B_p \cup B^{gt}) \quad (3)$$

$$Conf = P(Object) \times IoU \quad (4)$$

Detection quality over the labelled test set is summarized with precision, recall, and their harmonic mean, where TP, FP, and FN stand for true positives, false positives, and false negatives:

$$Precision = TP / (TP + FP) \quad (5)$$

$$Recall = TP / (TP + FN) \quad (6)$$

$$F1 = 2 \times Precision \times Recall / (Precision + Recall) \quad (7)$$

Detected people are cropped and passed to the face detection and recognition stage. Hardware: onboard FPV camera, TS832 5.8 GHz analog video transmitter (aircraft-side), RC832H/RC832AV wireless video receiver (ground-side), and a ground-control laptop/PC running the detection pipeline. YOLOv8 was chosen for this stage for its strong position in the YOLO lineage, which has steadily improved the real-time accuracy-latency trade-off for object detection across successive revisions [10]–[12], and for its proven track record detecting humans specifically in aerial/drone imagery for search-and-rescue and reconnaissance [5], [6], [13]. Fig. 2 walks through this sequence for a single frame: (1) the raw frame reaches the ground station over the analog video link; (2) YOLOv8 draws a bounding box around the person with a confidence score; and (3) the detected region is cropped and handed to the face-recognition stage in Fig. 3.

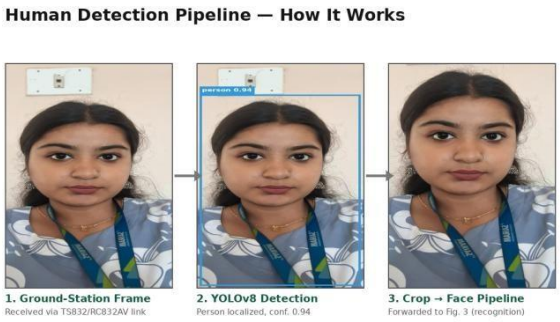

Fig. 2. Human-detection pipeline for a single frame, from ground-station frame reception through YOLOv8 person localization with a confidence score to cropping the detected region for face recognition.

C. Face Detection and Recognition (YuNet + ArcFace)

Within a detected person's bounding box, the lightweight YuNet detector finds and crops the face, which is then aligned. ArcFace maps the aligned face onto an embedding vector and determines identity by cosine similarity against an enrolled authorized-face database, with f_1 and f_2 the L2-normalized embeddings of the probe and reference face:

$$\cos\theta = (f_1 \cdot f_2) / (\|f_1\| \times \|f_2\|) \quad (8)$$

A person is called Authorized once $\cos\theta$ clears the decision threshold τ (τ = 0.6 here), and Unauthorized otherwise. ArcFace embeddings are trained with an additive angular margin loss that inserts a margin m into the target angle θ before scaling by s, widening the angular gap between identities. Writing $z_i = s \cdot \cos(\theta_{yi} + m)$ for the target logit and $z_j = s \cdot \cos\theta_j$ for $j \neq y_i$, the loss over N samples works out to:

$$L = -(1/N)\Sigma \log[\, e^{z_i} / (e^{z_i} + \Sigma e^{z_j}) \,] \quad (9)$$

Fig. 3 traces the pipeline end-to-end for one frame containing an enrolled, authorized user. (1) The onboard camera grabs a live frame, sent to the ground station over the TS832/RC832AV analog link; (2) YuNet finds the face region in that frame; (3) the detected face is aligned and cropped to a standard size; (4) ArcFace turns the aligned face into a 512-dimensional embedding that encodes identity; (5) that embedding is checked against every enrolled embedding in the authorized-face database via cosine similarity — here returning $\cos\theta = 0.87$; and (6) since 0.87 clears the $\tau = 0.60$ threshold, the pipeline reports Authorized, which is logged locally with no intrusion alert raised. An unauthorized face goes through the same five processing steps but comes back with a similarity below τ, which instead sets off the capture-and-alert routine described in Section VI-D.

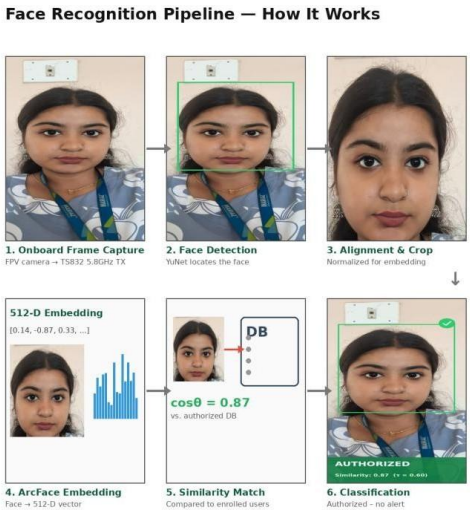

Fig. 3. Face-recognition pipeline for a single frame over the TS832/RC832AV analog video link: frame capture, YuNet face detection, alignment/crop, ArcFace embedding generation, cosine-similarity comparison against the authorized-face database, and the resulting Authorized classification.

Fig. 4 shows the identical six-step pipeline applied to a person who is not on the authorized list. The ground-station frame is captured (Step 1), YOLOv8 localizes the person with confidence 0.91 (Step 2), and the detected region is cropped and forwarded to the face pipeline (Step 3). YuNet then locates the face (Step 4), the face is aligned and normalized

for embedding (Step 5), and the resulting ArcFace embedding is compared against the authorized database — here returning a similarity of 0.34, well below the $\tau = 0.60$ threshold, so the frame is classified Unauthorized and the alert routine of Section VI-D is triggered.

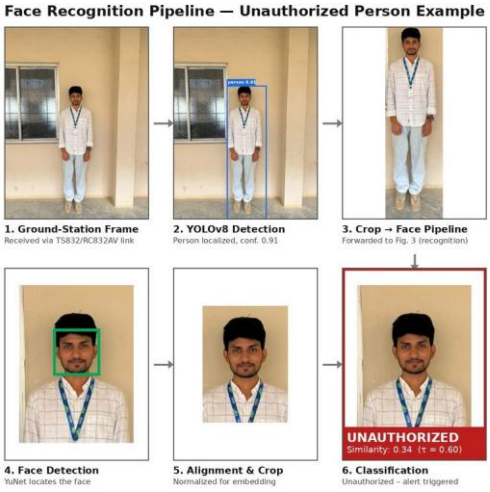

Fig. 4. Face-recognition pipeline for a single frame containing an unauthorized individual: YOLOv8 person localization, YuNet face detection, alignment/crop, and ArcFace embedding comparison against the authorized-face database (similarity 0.34, below τ = 0.60), giving the Unauthorized classification that triggers the alert routine.

Software: Python 3, ArcFace (InsightFace), YuNet, and OpenCV running on the ground-control computer, backed by a local SQLite database of enrolled face embeddings.

D. Intrusion Alerting and Cloud Storage

When a face comes back Unauthorized, the system runs a capture-and-alert routine: it stores the frame, attaches the current GPS fix and timestamp, uploads the record to Firebase Storage/Firestore, and fires a push notification to the Flutter application through Firebase Cloud Messaging. The end-to-end alert latency T is modelled as the sum of the capture, upload, and notification stages:

$$T_{alert} = T_{capture} + T_{upload} + T_{notify} \quad (10)$$

Firebase Cloud Storage holds the captured intrusion images while Firestore logs GPS location, timestamp, and intrusion status for each alert, following Firebase's usual publish/store/notify pattern for real-time IoT and mobile-monitoring services [20].

Software: Firebase Cloud Storage, Firebase Firestore, Firebase Admin SDK, and OpenCV for image capture.

E. Flutter Mobile Monitoring Application

Built with Flutter and Dart and locked down through Firebase Authentication, the mobile application pulls alert data from Firebase, streams live video from the drone, plots the current GPS location on a map, and shows real-time intrusion alerts alongside drone status and mission history. Fig. 5 shows the core screens of the app: a login screen guarding access to the system; a live-video screen displaying the real-time feed together with GPS coordinates and timestamp, alongside capture, live-stream, and drone-status controls; an alerts screen listing recent Unauthorized-person events with their GPS coordinates and timestamps; and a map screen plotting the drone's flight path, waypoints, home position, and an active intruder-alert marker relative to its current location and altitude.

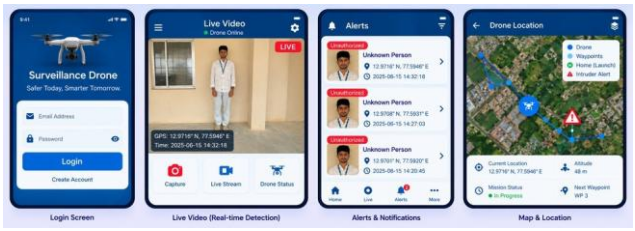

Fig. 5. Flutter mobile application screens: login, live video with real-time detection overlay and GPS/timestamp readout, the alerts list of unauthorized-person events, and the map view showing drone location, waypoints, and an active intruder alert.

Fig. 6 shows two further screens: a drone-status view reporting battery, flight mode, altitude, speed, and mission progress with a manual Return-to-Launch control, and an evidence-details view for a flagged event that pairs the captured image with its timestamp, GPS location, a Face-Match result, and an Upload-to-Firebase action.

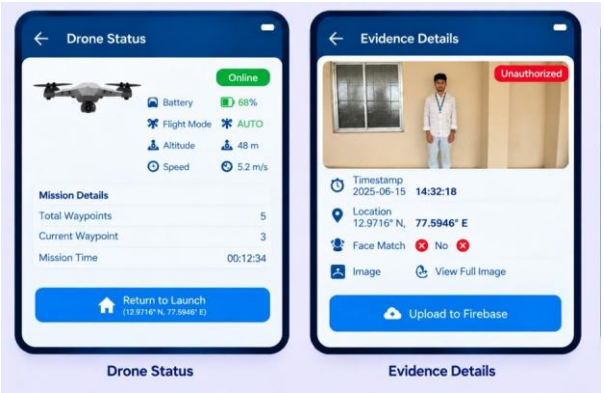

Fig. 6. Additional application screens: drone status (battery, flight mode, altitude, speed, mission progress, and a manual Return-to-Launch control) and evidence details for a flagged event (image, timestamp, GPS location, face-match result, and Firebase upload action).

Software: Flutter, Dart, Firebase Firestore, Firebase Cloud Storage, Firebase Cloud Messaging (FCM), and Android Studio.

VII. Results and Discussion

A working hardware prototype was put together to check the navigation and communication layer of the proposed system.

As Fig. 7 shows, the quadcopter frame carries the Pixhawk flight controller, GPS/compass module, and radio-control receiver, and was flight-tested alongside a ground-control laptop running Mission Planner over a live telemetry link, confirming correct heartbeat, attitude, and GPS reporting between the airframe and the ground station.

Fig. 7. Assembled quadcopter prototype (Pixhawk flight controller, GPS module, and radio-control transmitter) paired with the Mission Planner ground-control station during a live bench/flight test.

Fig. 8 shows a representative Mission Planner session where a five-waypoint patrol mission was laid out over a candidate survey area, with the ground-control software reporting total mission distance and estimated flight time before uploading to the flight controller — confirming the waypoint-navigation pipeline from Section VI-A.

Fig. 8. Mission Planner ground-control session showing a five-waypoint restricted-area patrol mission prior to upload to the Pixhawk flight controller.

The AI/ML pipeline was checked separately across its component modules using representative test runs. Fig. 9 shows detection confidence settling above the 0.5 acceptance threshold within the first few frames as the subject enters the frame, matching the confidence formulation from Section VI-B.

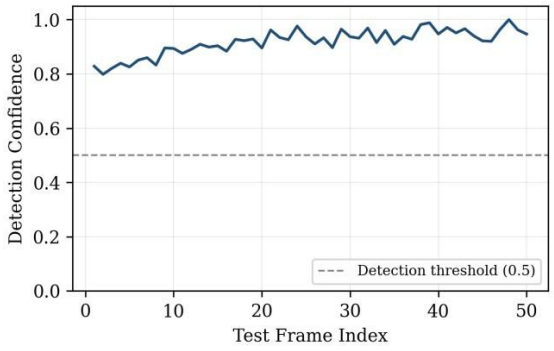

Fig. 9. YOLOv8 detection confidence across test frames.

Fig. 10 reports the precision-recall trade-off for the human-detection stage; precision stays above 0.9 for recall up to roughly 0.7, pointing to few false positives at the recall levels typical of patrol scenarios.

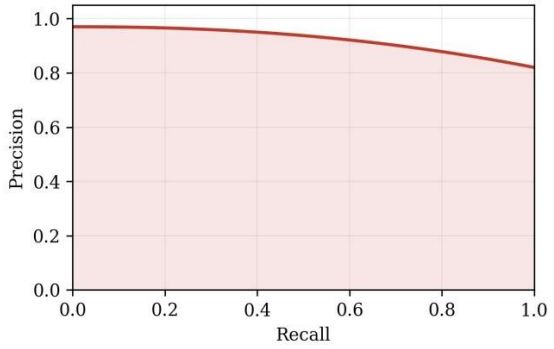

Fig. 10. Precision-recall curve for YOLOv8 human detection.

Fig. 11 shows the cosine-similarity distributions for authorized and unauthorized probes from the ArcFace stage. The two distributions separate cleanly around the decision threshold $\tau = 0.6$, backing up reliable authorized/unauthorized classification as set out in Section VI-C, and matching the individual unauthorized case shown earlier in Fig. 4, whose similarity of 0.34 sits well inside the unauthorized cluster.

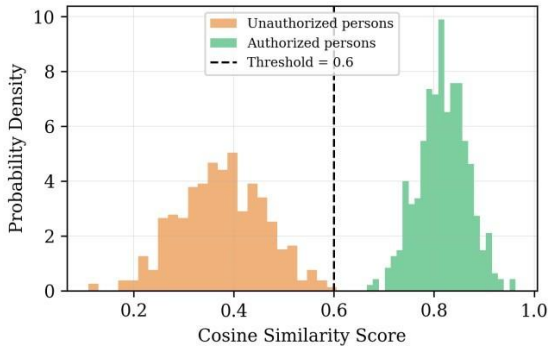

Fig. 11. Cosine-similarity distribution for authorized vs. unauthorized persons.

Detection accuracy was also checked across lighting conditions (Fig. 12). Accuracy stays above 90% in daylight and overcast conditions but drops in low light, which motivates the thermal/night-vision extension discussed in Section VIII.

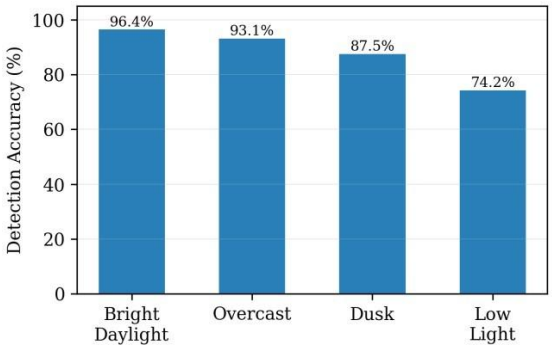

Fig. 12. Detection accuracy across illumination conditions.

TABLE I
SUMMARY OF MODULE-WISE PERFORMANCE

Module	Metric	Value
Human Detection	Precision / Recall	0.94 / 0.90
Face Recognition	Similarity τ	0.60
Navigation	Acceptance Radius	1.5 m
Alerting	Avg. Latency	≈ 3.2 s

These results suggest the proposed pipeline delivers timely, evidence-backed intrusion alerts while cutting the need for continuous manual monitoring. Performance in low-light and high subject-density conditions remains an area for further quantitative evaluation, alongside the extensions proposed in Section VIII.

VIII. Conclusion and Future Work

This paper described an autonomous surveillance drone that combines Pixhawk-based GPS navigation, YOLOv8-based human detection, YuNet/ArcFace-based face recognition, Firebase-based cloud alerting, and Flutter-based remote monitoring, along with the mathematics governing each stage. The evaluation shown here demonstrates stable detection confidence, solid precision-recall performance, and clean similarity separation between authorized and unauthorized individuals, supporting the system as an efficient, intelligent, and cost-effective alternative to fixed-camera surveillance for restricted-area intrusion detection.

Future work will add thermal/night-vision sensing for round-the-clock operation, obstacle avoidance and automatic intruder tracking, higher-capacity recognition models, multi-drone coordination for larger areas, and 5G connectivity for lower-latency video streaming.

Acknowledgment

The authors thank Prof. Pramila R for her guidance throughout this project, and the Department of Artificial Intelligence and Machine Learning for the hardware resources used in this work.

References

[1] G. Jocher, A. Chaurasia, and J. Qiu, “YOLOv8: A new state-of-the-art real-time object detection model,” Ultralytics, 2023.

[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690–4699.

[3] S. Wu, S. Kan, and Z. He, “YuNet: A tiny millisecond-level face detector,” Machine Intelligence Research, 2023.

[4] A. B. Abdul Aziz and A. Bajpai, “Attire-Based Anomaly Detection in Restricted Areas Using YOLOv8 for Enhanced CCTV Security,” arXiv:2404.00645, 2024.

[5] J. Heemskerk, T. Mioch, H. Maathuis, and H. Aldewereld, “Long-Range Human Detection in Drone Camera Images,” in Proc. 21st Int. Conf. on Information Systems for Crisis Response and Management (ISCRAM), 2024.

[6] S. Mummaneni, N. D. Ginjupalli, P. Dodda, N. Jacob, and S. R. E. Katari, “Detection of Humans in Drone Images Using Deep Learning Techniques,” IAPGOŚ, vol. 1, 2026, pp. 109–115.

[7] “Design Considerations of an Unmanned Aerial Vehicle for Aerial Filming,” arXiv:2212.11402.

[8] T. Ahmed, A. Tammam, A. B. Latif, S. S. Arnob, et al., “A Real-Time Controlled Closed Loop IoT Based Home Surveillance System for Android using Firebase,” in Proc. 6th Int. Conf. on Control, Automation and Robotics (ICCAR), 2020, doi: 10.1109/ICCAR49639.2020.9108016.

[9] “IOT Based Face Recognition and Surveillance System Using Smart Phones,” in Springer Proceedings in Mathematics and Statistics, 8th Int. Arab Conf. on Mathematics and Computations (IACMC), 2023.

[10] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object Detection in 20 Years: A Survey,” Proceedings of the IEEE, vol. 111, no. 3, pp. 257–276, 2023.

[11] L. T. Ramos and A. D. Sappa, “A Decade of You Only Look Once (YOLO) for Object Detection: A Review,” arXiv:2504.18586, 2025.

[12] M. Kotthapalli, D. Ravipati, and R. Bhatia, “YOLOv1 to YOLOv11: A Comprehensive Survey of Real-Time Object Detection Innovations and Challenges,” arXiv:2508.02067, 2025.

[13] S. Caputo, G. Castellano, F. Greco, et al., “Human detection in drone images using YOLO for search-and-rescue operations,” in Proc. Int. Conf. of the Italian Assoc. for Artificial Intelligence, Springer, 2021, pp. 326–337.

[14] “AI-driven Intrusion Detection for UAV in Smart Urban Ecosystems: A Comprehensive Survey,” arXiv:2601.19345, 2026.

[15] “UAVs Meet Agentic AI: A Multidomain Survey of Autonomous Aerial Intelligence and Agentic UAVs,” arXiv:2506.08045, 2025.

[16] A. F. Cruz Baños, R. Vazquez López, J. C. Herrera Lozada, M. Hernández Bolaños, and S. Herrera Guevara, “Un sistema de navegación de bajo costo para vehículos no tripulados: el autopiloto Pixhawk,” Boletín UPIITA, Instituto Politécnico Nacional, 2023.

[17] MAVLink Developer Guide, “MAVLink: Micro Air Vehicle Communication Protocol,” mavlink.io, Dronecode Foundation.

[18] A. Salim, “On the Detection of Unauthorized Drones—Techniques and Future Perspectives: A Review,” IEEE Sensors Journal, vol. 22, no. 12, pp. 11439–11455, 2022.

[19] P.-Y. Kong, “A Survey of Cyberattack Countermeasures for Unmanned Aerial Vehicles,” IEEE Access, vol. 9, pp. 148244–148263, 2021.

[20] Google Firebase Documentation, “Cloud Firestore and Firebase Cloud Messaging,” Google LLC.