



DESIGN OF AN AUTONOMOUS UAV TRAFFIC ENFORCEMENT SYSTEM WITH PRIORITY-SCORED ANOMALY TRACKING AND EDGE-DRIVEN MQTT FLIGHT CONTROL

¹Manikandan S, ²Annapoorani M, ³Marikkannan M

¹M.E. Computer Science and Engineering (PG Scholar), ^{2,3}Assistant Professor (Senior), Department of
Computer Science and Engineering

¹Department of Computer Science and Engineering

¹Government College of Engineering, Erode, Tamil Nadu, India

Abstract: Conventional traffic surveillance infrastructure, predominantly comprising static closed-circuit camera networks, is inherently constrained by fixed spatial coverage, blind spots, and dependence on manual observation for enforcement action. This paper is proposed an Autonomous UAV Traffic Enforcement System (AUTES), a hardware-validated, closed-loop aerial surveillance platform that bridges the gap between passive video monitoring and active autonomous enforcement response. This system integrates a dual-stage YOLOv8 inference pipeline employing a Common Objects in Context (COCO)-pretrained YOLOv8n model for scene-level person and vehicle detection, followed by a custom-trained binary helmet classification model with an autonomous quadrotor flight controller built on a modified E88 Pro drone platform driven by an ESP32 microcontroller running ESPFC firmware. A composite violation priority scoring mechanism ranks simultaneously detected violations. Detected violation coordinates are translated into real-time Radio Control (RC) flight commands. In the absence of detected violations, the drone autonomously executes a bidirectional yaw-sweep patrol pattern to maximize surveillance coverage without GPS dependency. The custom helmet detection model, trained on a Roboflow dataset for 100 epochs using YOLOv8l as the base architecture, achieved a precision of 74.2%, recall of 63.7%, mAP@0.5 of 0.748, and mAP@0.5:0.95 of 0.450 on the test set. The end-to-end command latency from vision inference to motor actuation is maintained below 25 ms under standard operating conditions. AUTES demonstrates that a lightweight, edge-driven perception-to-control pipeline operating entirely over standard Internet of Things (IoT) communication protocols can deliver autonomous, adaptive traffic enforcement behaviour on a cost-effective embedded UAV platform, establishing a scalable foundation for multi-class violation detection in smart city surveillance deployments.

Keywords - autonomous UAV, traffic violation detection, YOLOv8, helmet detection, MQTT, ESP32, PD controller, visual servoing, anomaly scoring, smart surveillance, ESPFC, edge AI

I. INTRODUCTION

Rapid urbanization and escalating motor vehicle populations have intensified traffic law enforcement challenges globally. Among the most life-threatening violations is the non-use of protective helmets by motorcyclists, directly linked to severe head injuries and fatalities. Conventional fixed closed-circuit television (CCTV) surveillance suffers from inherent spatial limitations restricted fields of view, stationary blind spots, and inability to dynamically relocate in response to evolving violation patterns [18]. These static systems are fundamentally reactive, requiring significant human intervention for enforcement action and rendering them ineffective for real-time autonomous response [9]. Unmanned Aerial Vehicles (UAVs) have emerged as a transformative solution, offering aerial vantage points, mobility, and dynamic coverage ideally suited for intelligent traffic monitoring [17]. Recent advances in edge computing, lightweight deep learning architectures, and low-power embedded communication protocols have further enabled UAVs to perform real-time perception and autonomous decision-making with minimal centralized infrastructure dependency [8].

The YOLO (You Only Look Once) family of deep convolutional neural network models has established a benchmark for real-time detection accuracy and speed. YOLOv8, introduced by Ultralytics, delivers superior mean Average Precision (mAP) with reduced computational overhead, making it well-suited for resource-constrained aerial platforms [1]. Helmet compliance detection has been demonstrated at high accuracy using YOLOv8-based architectures, with F1 scores exceeding 0.95 reported on benchmark datasets [14]. Combining YOLOv8 with Deep Convolutional Generative Adversarial Networks (DCGANs) further improves detection robustness under class-imbalanced conditions [14], while edge-based deployment on embedded devices confirms the feasibility of on-board real-time inference [10]. In multi-target surveillance, rationally directing drone response toward the highest-risk violator among multiple simultaneous detections is a critical challenge. Prior research establishes that combining detection confidence with spatial and contextual weighting produces more reliable prioritization than confidence thresholds alone [6].

Autonomous quadrotor flight control requires precise regulation of attitude (roll, pitch, yaw) and altitude under aerodynamic disturbances. Proportional-Derivative (PD) and Proportional-Integral-Derivative (PID) controllers remain the predominant choice for attitude stabilization due to their interpretability and real-time suitability [7]. ESP32-based embedded PID implementations have confirmed reliable performance under standard flight conditions [5], while self-tuning fuzzy PID controllers demonstrate improved adaptability under dynamic loads [11]. The Message Queuing Telemetry Transport (MQTT) protocol, a publish-subscribe standard designed for constrained Internet of Things (IoT) environments, has been adopted in UAV communication frameworks for its minimal bandwidth consumption and real-time telemetry exchange support [4]. Security analyses highlight the necessity for encrypted topic management to prevent command injection [22]. The ESP32 System-on-Chip (SoC), with dual-core processing and integrated Wi-Fi, provides sufficient headroom for concurrent RC signal decoding, sensor fusion, and MQTT communication [3], [5], with sub-100ms command latency validated in UAV network coordination frameworks [4].

Despite this progress, a critical gap remains no prior work delivers a tightly integrated, hardware-validated pipeline coupling real-time vision-based anomaly detection directly to autonomous drone flight commands. Prior studies treat perception and control in isolation, validating detection on static benchmarks [1], [22] or evaluating flight control without live vision-in-the-loop feedback [7], [11]. Helmet violation systems remain confined to fixed-camera deployments [2], [21], without autonomous aerial relocation capability. The proposed Autonomous UAV Traffic Enforcement System (AUTES) addresses this gap through a closed-loop vision-to-flight pipeline operating at 40 Hz. A dual-stage YOLO stack detects persons

and vehicles using COCO-pretrained YOLOv8n, followed by a custom helmet classification model evaluating each person crop. Violations are ranked by a composite priority score combining detection confidence, normalized bounding-box area, and frame-center proximity.

II. RELATED WORKS

2.1 Deep Learning-Based Helmet and Traffic Violation Detection

Helmet violation detection has attracted considerable research attention as a critical component of automated traffic enforcement. [1] proposed an improved YOLOv8 architecture incorporating enhanced backbone features and multi-scale detection heads, achieving superior precision and recall on helmet detection benchmarks compared to prior YOLO variants. [2] further extended this to a complete real-time traffic monitoring pipeline, demonstrating that YOLOv8 deployed in a traffic surveillance context achieves reliable helmet violation identification with minimal false positives under varying illumination conditions. [10] validated edge deployment of a YOLOv9-based helmet detection model on a Raspberry Pi embedded platform, demonstrating practical feasibility of on-device inference at acceptable frame rates, directly motivating edge-driven detection in the proposed system. [21] proposed an Artificial Intelligence(AI)-based helmet violation detection system using Perspective Net, while helmet violation using DCGAN-augmented YOLOv8 training [14] demonstrated that synthetic data augmentation significantly improves detection under class-imbalanced conditions, achieving an F1 score of 0.96.

2.2 Autonomous UAV Navigation and Control

Autonomous UAV navigation integrating computer vision perception has been extensively surveyed, systematically reviewed deep learning frameworks applied to UAV flight control, obstacle avoidance, and target tracking, identifying visual feedback loops as a key enabler of fully autonomous operation [8]. [17] provided a comprehensive taxonomy of autonomous UAV architectures, categorizing control strategies into reactive, deliberative, and hybrid classes, and identifying PID-based visual servoing as the most deployment-ready paradigm for real-time platforms. [9] proposed a robust finite-time model reference adaptive control scheme for quadrotor attitude stabilization under external disturbances, demonstrating fast convergence and disturbance rejection superior to conventional PID implementations. [7] developed and experimentally validated an adaptive PID control system for simultaneous attitude and altitude regulation of a quadcopter, providing tuning guidelines directly applicable to the proposed yaw and throttle control design. [11] introduced a self-tuning fuzzy PID controller for quadcopters, demonstrating improved tracking performance under dynamic load variations.

2.3 Visual Servoing and UAV Target Tracking

Visual servoing the paradigm of deriving flight control commands directly from image-plane error signals has been identified as the most suitable approach for GPS-free autonomous UAV target engagement. A comprehensive survey of AI-enhanced visual servoing for autonomous UAVs [15] reviewed image-based and position-based visual servoing architectures, concluding that image-based approaches offer greater robustness to depth estimation errors in aerial tracking tasks. Experimental implementation of UAV target tracking via visual servoing using PID controllers confirmed real-time tracking feasibility on physical quadrotor platforms, with pixel-space centroid error used as the feedback signal an approach adopted in the proposed yaw PD controller design [16].

2.4 MQTT Communication and ESP32-Based Drone Control

Reliable, low-latency communication is indispensable in closed-loop autonomous UAV systems. [4] demonstrated that MQTT-based communication frameworks for UAV networks achieve sub-100ms command delivery latency, validating MQTT as a viable real-time control channel for drone systems. Security considerations in MQTT-based drone communication [22] highlighted the importance of topic-level access control and payload encryption to prevent unauthorized command injection a consideration incorporated in the proposed system's broker configuration. [3] experimentally validated ESP32 Wi-Fi mesh networking for UAV swarm communication, confirming the ESP32 platform's capability for concurrent multi-task embedded processing at drone-relevant data rates. [5] conducted experimental PID attitude control of a quadcopter using an ESP32 microcontroller, reporting stable flight performance, directly validating the embedded hardware platform adopted in the proposed system. [12] reviewed wireless communication protocols for drones including Wi-Fi, Zigbee, and MQTT, concluding that MQTT offers the optimal balance of latency, reliability, and implementation simplicity for constrained UAV embedded systems.

2.5 Anomaly Detection and Smart Surveillance Systems

Anomaly detection in video surveillance has evolved from rule-based thresholding toward composite scoring methods that integrate multiple contextual features. [6] demonstrated real-time anomaly detection in surveillance video using YOLOv8, validating detection-confidence-based scoring as an effective proxy for anomaly severity. [19] evaluated unsupervised machine learning algorithms for anomaly detection, establishing that composite multi-feature scoring outperforms single-metric thresholding in terms of detection reliability and false alarm rate. [22] enhanced YOLOv8 with soft-Non-Maximum Suppression (NMS) and multi-scale feature fusion for crowd anomaly detection, achieving state-of-the-art performance in dense scenes supporting the adoption of YOLOv8n as the base detection backbone in this work. [18] reviewed smart surveillance system trends, identifying autonomous mobile platforms and AI-driven real-time detection as the two most impactful directions for next-generation surveillance infrastructure. An IoT-enabled UAV system for real-time traffic mobility analysis demonstrated the practical integration of aerial platforms with urban traffic monitoring, achieving 88% vehicle detection accuracy using YOLOv8n in a live deployment, closely paralleling the detection pipeline adopted in the proposed system [13].

2.6 Research Gap and Positioning of Proposed Work

A synthesis of the reviewed literature reveals that while significant progress has been achieved individually in helmet detection [1], [2], autonomous UAV control [7], [9], MQTT-based drone communication [4], [5], and video anomaly detection [6], [22], no prior work has delivered a tightly integrated, hardware-validated pipeline coupling real-time YOLOv8 violation detection directly to autonomous ESP32 drone flight commands via MQTT at operational frequency. Furthermore, the concept of priority-scored multi-target selection dynamically ranking detected violators by a composite confidence-size-position metric and directing drone maneuvers toward the highest-priority target has not been previously reported in the UAV traffic enforcement literature. The proposed system addresses this gap by delivering a complete, end-to-end operational autonomous UAV traffic enforcement platform validated on physical hardware.

III. PROPOSED SYSTEM

3.1 System Architecture

The proposed Autonomous UAV Traffic Enforcement System (AUTES) is a fully hardware-implemented, closed-loop surveillance platform that integrates real-time computer vision-based traffic violation detection with autonomous quadrotor flight control. The system operates as a continuous

perception-decision-action loop at 40 Hz, mapping detected violation positions in image space directly to RC flight commands executed on a custom-built ESP32-based quadcopter via the MQTT protocol. Unlike prior works that treat detection and control as independent subsystems [1], [7], AUTES fuses both into a single real-time pipeline where each processed surveillance frame drives an immediate, calibrated flight response. The system supports three autonomous operational modes: Violation Tracking (yaw + pitch + altitude pursuit of the highest-priority detected violator), Hover-and-Record (stabilized hovering when the drone is locked on target), and Patrol (bidirectional yaw sweep when no violation is detected), with seamless mode transitions governed by detection state and temporal logic. Figure 1 shows the complete closed-loop system across 6 subsystems: Vision captures and detects → Decision scores violations and selects mode → Flight Controller computes yaw/throttle/pitch corrections → MQTT transmits RC commands at 40Hz → Hardware (ESP32 → MPU6050 → MOSFETs) executes flight → Physical flight changes drone position → new frame captured → loop repeats. Table 1 Represents the Component summary of AUTES.

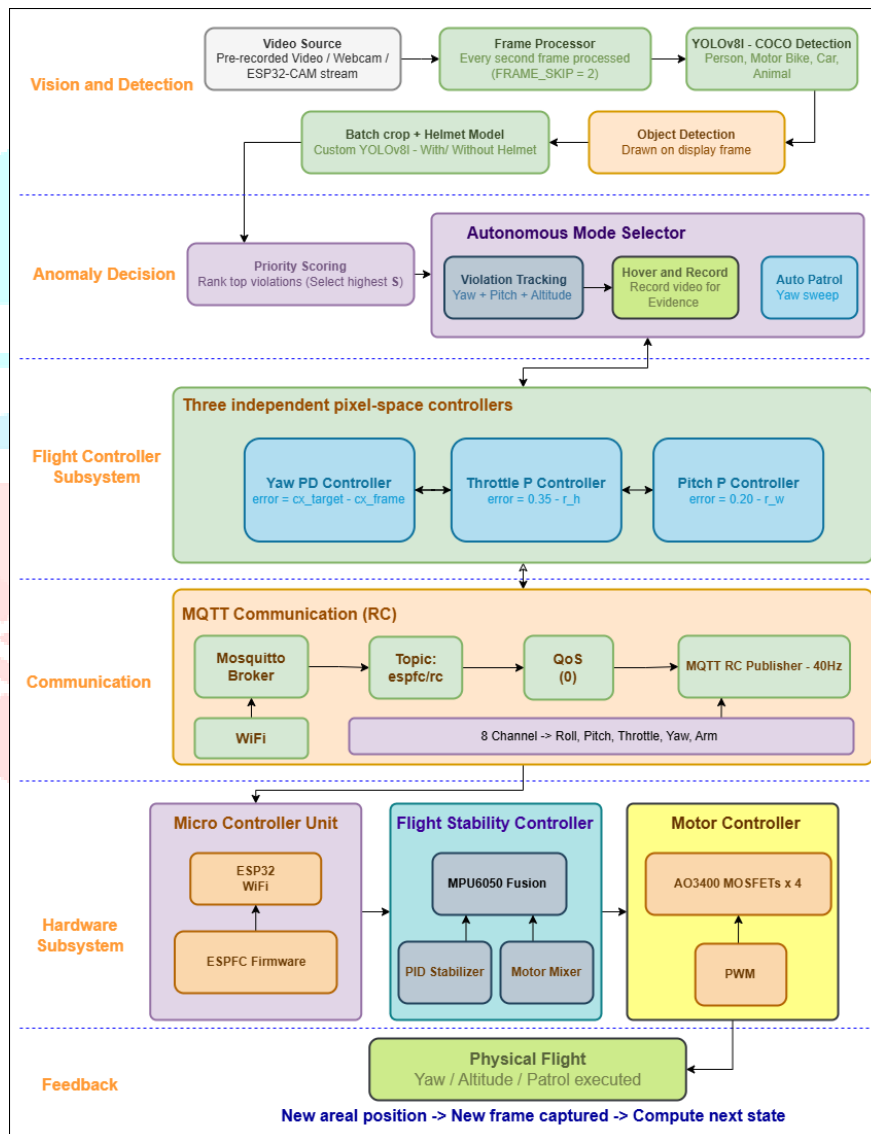


Fig.1. System Architecture of AUTES

Table 1 AUTES Component Summary

Component	Technology / Method	Function in System
Object Detection	YOLOv8n (COCO pretrained)	Detects persons, vehicles per frame
Helmet Classification	Custom YOLOv8 (binary)	With Helmet / Without Helmet per person crop
Anomaly Scoring	Composite weighted formula	Ranks violations by confidence, size, position
Yaw Control	PD controller + EMA smoothing	Rotates drone to face highest-priority violator
Altitude Control	Throttle-based P controller	Adjusts height via bounding-box height ratio
Approach Control	Pitch P controller	Moves drone forward when yaw is locked on
RC Communication	MQTT @ 40 Hz (paho-mqtt)	Publishes 8-channel RC commands to ESP32
Flight Controller	ESP32 + ESPFC firmware	Decodes RC channels, drives ESCs / motors
Patrol Behavior	Bidirectional yaw sweep	Autonomous scanning when no violation detected
Video Source	Video file / Camera stream	Frame-by-frame surveillance input at 40 Hz

3.2 Custom UAV Flight Model

3.2.1 ESP32 Flight Controller Hardware Platform

Figure 2 Shows the physical aerial platform selected for AUTES is the E88 Pro foldable quadcopter, a consumer-grade rotary-wing drone constructed from high-strength engineering plastic with a symmetric foldable arm design. The E88 Pro's lightweight frame structure, symmetric QuadX motor layout, and centrally recessed battery bay make it a suitable base platform for custom embedded hardware integration. The four motor mounts are arranged symmetrically around the central hub, with diagonally opposite motors sharing the same rotation direction - two clockwise (CW) and two counterclockwise (CCW) - which is the fundamental counter-rotational pairing necessary for aerodynamic torque neutralization during hover and controlled yaw maneuvers [5], [7]. The foldable arm hinges allow the drone to be compactly stored and deployed without disassembly, while the four propeller guard attachment points protect the rotors during low-altitude surveillance flight near obstacles. The frame's structural geometry ensures that roll and pitch moments of inertia are approximately equal, supporting symmetric attitude control response - a critical property for PD controller tuning stability. The 3.7V Li-Po battery is housed at the geometric center of the frame's underside, maintaining the Center of Gravity (CoG) at the drone's centroid and preventing static pitch or roll bias during hover.

Figure 3 Represents the replacement flight controller is built around the ESP32 microcontroller, a dual-core processor operating with Flash memory and integrated Wi-Fi. The ESP32 runs the open-source ESPFC (ESP Flight Controller) firmware, which provides a complete quadrotor stabilization stack including RC input decoding, Inertial Measurement Unit (IMU) sensor fusion, PID attitude control loops, and LEDC

Pulse Width Modulation (PWM) motor output generation - all executing concurrently on the dual cores. The ESP32's integrated Wi-Fi module subscribes to the MQTT topic via a local Mosquitto broker, receiving 8-channel RC command strings published by the host PC's AUTES control pipeline at 40 Hz. This Wi-Fi MQTT interface completely replaces the original RF receiver, transforming the drone into a network-controlled autonomous platform without requiring any additional communication hardware [3], [5]. The ESP32's LEDC peripheral generates four independent PWM channels assigned to the motor General Purpose Input/Output (GPIO) pins, modulating each motor's speed through the AO3400 MOSFET driver stage at the duty cycle computed by the ESPFC stabilization algorithm.

Each of the four brushed coreless motors is independently driven by a dedicated AO3400 N-channel MOSFET configured as a low-side switch between the motor's ground terminal and the circuit ground rail. The ESP32 LEDC PWM signal is applied to the MOSFET gate, turning the device on and off at the PWM frequency to regulate average current through the motor windings and thereby control rotational speed. The AO3400 is selected for its low on-state drain-source resistance and high continuous drain current capability, which ensure minimal conduction losses and thermal stability under the high-frequency PWM switching demands of brushed motor control [5]. Since the ESP32 GPIO outputs operate at 3.3V logic levels, the AO3400's logic-level gate threshold ensures full enhancement with no level-shifting circuitry. Gate resistors inserted in series between the ESP32 GPIO and each MOSFET gate limit gate charge inrush current and damp parasitic oscillations at the switching transitions. Decoupling capacitors placed across the motor power supply rails absorb the back-EMF voltage spikes generated when motor current is interrupted at PWM off-transitions, protecting the ESP32 and MOSFET gate oxide from overvoltage stress. Figure 6 Illustrates the circuit diagram of our proposed AUTES.



Fig.2. E88 Pro Base Drone

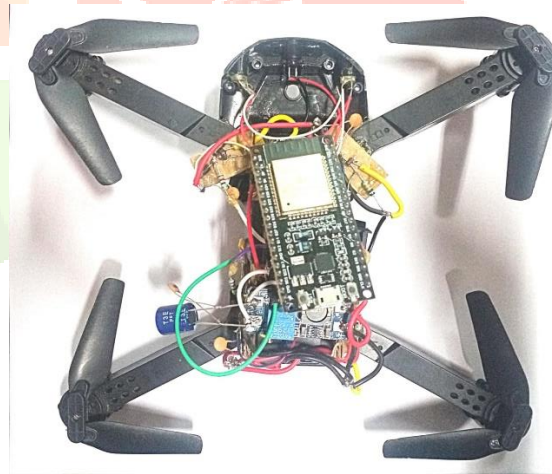


Fig.3. Customized AUTES Flight

Closed-loop attitude stabilization is enabled by the MPU6050 6-axis Inertial Measurement Unit, connected to the ESP32 via the I2C serial bus. The MPU6050 integrates a 3-axis MEMS gyroscope and a 3-axis MEMS accelerometer, delivering both angular velocity and linear acceleration measurements at a configurable output data rate. The ESPFC firmware performs a complementary filter fusion of accelerometer-derived tilt estimates and gyroscope-integrated angle increments, producing roll, pitch, and yaw angle estimates with significantly reduced drift compared to pure gyroscope integration [7]. These attitude estimates are fed into the ESPFC stabilization PID loops, which compute per-axis correction torques that are superimposed on the incoming RC channel values before the mixer stage distributes motor speed commands. The MPU6050's I2C interface requires only two GPIO pins (SDA and SCL) on the ESP32, preserving GPIO availability for the four PWM motor output channels. The physical IMU mounting position on the custom

Printed Circuit Board (PCB) is aligned with the drone's principal axes to minimize sensor-to-body misalignment errors in the attitude estimation.

The architectural transformation achieved through the hardware modification is most clearly understood by contrasting the stock and modified signal chains. In the AUTES-modified configuration, the pipeline becomes: *Surveillance Video* → *YOLOv8 Inference* → *Priority Scoring* → *PD Controllers* → *MQTT @ 40Hz* → *ESP32 (ESPFC)* → *AO3400 MOSFETs* → *Motors*, a fully software-configurable, network-driven, autonomous control architecture [4], [12]. This modification enables the drone's flight trajectory to be determined entirely by real-time computer vision output, with the PD controllers translating pixel-space violation positions into calibrated RC channel adjustments that drive physical motor speed changes. The host PC and ESP32 form a distributed embedded system connected over the local Wi-Fi network, with the MQTT broker acting as the lightweight asynchronous message bus that decouples the vision processing loop from the RC command delivery thread, ensuring that motor actuation continues at 40 Hz regardless of transient YOLO inference latency variations.

3.3 Computer Vision Pipeline

3.3.1 Custom YOLO Model Training

The AUTES helmet violation detection subsystem employs a custom-trained binary object detection model based on YOLOv8l, leveraging transfer learning from MS COCO pre-trained weights. The large-scale YOLOv8l architecture is selected for its deep Cross Stage Partial (CSP)-based backbone, which enables robust feature extraction under challenging conditions such as occlusions, low-resolution aerial crops, and visual similarity between helmets and hair. Compared to lightweight variants, YOLOv8l provides improved classification accuracy, while maintaining acceptable inference latency due to operation on cropped person regions. The training dataset, sourced from Roboflow, consists of annotated motorcyclist images labeled as *With Helmet* and *Without Helmet*. It is organized into train, validation, and test splits with a YOLOv8-compatible configuration. A class imbalance exists, with more compliant (helmeted) instances, influencing model performance across classes.

Training is conducted on Google Colab using an NVIDIA Tesla T4 GPU for 100 epochs at 640×640 resolution, employing stochastic gradient descent with cosine learning rate scheduling. The best-performing model, based on validation mAP@0.5, was saved and deployed for inference. To enhance accuracy, the model operated on cropped person bounding boxes detected by an upstream YOLOv8n model, reducing background interference and improving feature focus. The trained model is integrated into the AUTES pipeline in inference mode, ensuring efficient real-time processing and reliable helmet violation classification.

3.3.2 Dual-Model YOLO Detection Pipeline

The perception layer of AUTES employs a two-stage YOLO inference pipeline to achieve both scene-level object detection and instance-level violation classification. In the first stage, the COCO-pretrained YOLOv8n model processes each surveillance frame to generate bounding boxes for all detected persons (class 0) and vehicles — specifically motorcycles (class 3), cars (class 2), buses (class 5), and trucks (class 7). YOLOv8n is selected for its optimal balance of detection accuracy and inference speed on CPU-constrained edge hardware [1], [8]. In the second stage, each detected person bounding box is cropped from the original frame and independently passed to a custom-trained binary helmet classification model, which outputs one of two class labels: *With Helmet* or *Without Helmet*, along with a detection confidence score. This cascaded two-stage design full-frame detection followed by crop-level classification - significantly reduces false positives compared to single-stage approaches by constraining the helmet model's inference domain exclusively to confirmed person regions [6], [14]. To maintain real-time performance, the YOLO

inference pipeline processes every third frame (frame skip = 2), while RC command publication continues uninterrupted at 40 Hz on a dedicated thread.

3.3.3 Violation Priority Scoring Mechanism

A central contribution of AUTES is its composite violation priority scoring mechanism, which enables rational, autonomous selection of the highest-risk target when multiple simultaneous violations are detected in a single surveillance frame. For each detected *Without Helmet* instance, a scalar priority score S is computed as:

$$S = w_v \times C_h \times (A_{bb} / A_f) \times (1 - d_x / W_f)$$

where $w_v = 2.0$ is a fixed violation weight, C_h is the helmet model detection confidence $\in [0, 1]$, A_{bb} is the bounding-box area of the detected person in pixels (computed as bounding-box width \times bounding-box height), A_f is the total frame area in pixels (frame width \times frame height), and their ratio A_{bb} / A_f is the normalized bounding-box area representing the person's apparent size relative to the full surveillance, and $(1 - d_x / W_f)$ is the positional weight, where d_x is the horizontal distance of the person centroid from the frame center and W_f is the frame width. This formulation ensures that violations with higher detection confidence, larger apparent size (closer to the drone), and more central frame position are assigned higher priority scores [6], [19]. Up to five candidate violations are scored per frame, and the highest-scoring instance is designated as the active tracking target, with its centroid coordinates and bounding-box dimensions forwarded to the flight control subsystem. Figure 4 Represents the output of YOLO drone anomaly flight controller.

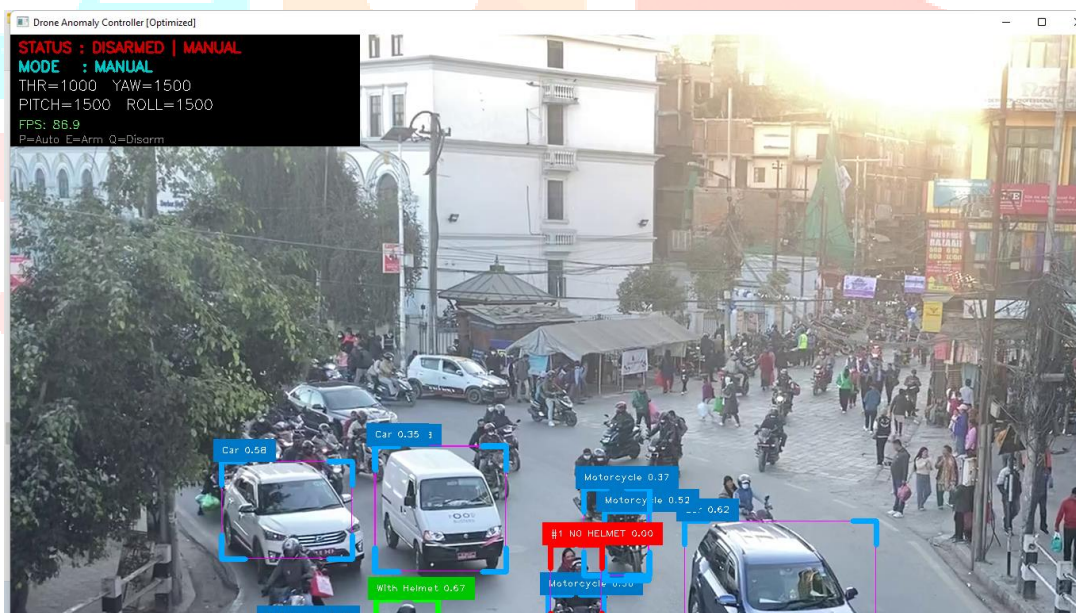


Fig.4. YOLO Drone Anomaly Controller

3.4 Autonomous Flight Control

3.4.1 Vision-Driven Autonomous Flight Control

The flight control subsystem of AUTES translates the pixel-space position and size of the highest-priority violation into three independent RC channel adjustments yaw, throttle, and pitch - each governed by a dedicated proportional controller with derivative smoothing. All RC channel values are expressed in standard PWM units in the range $[1000, 2000] \mu s$, centered at $1500 \mu s$, consistent with ESPFC firmware RC input specification [5], [3]. Figure 5 Shows the flight control framework directional control.

Yaw Control (Horizontal Target Centering): The horizontal pixel error between the target centroid and the frame center is used to generate a yaw command that rotates the drone to face the violator. The yaw error $\epsilon_x = cx_{target} - cx_{frame}$ is first filtered through an Exponential Moving Average (EMA) smoother with smoothing factor $\alpha = 0.80$ to suppress sensor noise and detection jitter [7]. A deadzone of ± 30 pixels suppresses micro-corrections near the center. The smoothed error is then scaled by proportional gain $K_p = 0.06$ and derivative gain $K_d = 0.01$, and clamped to a maximum step of ± 50 PWM units per cycle.

Altitude Control (Throttle Regulation): The drone's altitude is regulated based on the ratio of the detected person's bounding-box height to the total frame height ($r_h = h_{bb} / H_f$). The altitude error is defined as $\epsilon_{alt} = r_{target} - r_h$, where $r_{target} = 0.35$ is the desired target occupancy ratio, corresponding to the drone hovering at an optimal surveillance altitude. A proportional throttle adjustment of $K_{alt} = 80 \times \epsilon_{alt}$ PWM units is applied per frame, clamped within ± 150 units of the nominal hover throttle ($T_{hover} = 1550 \mu s$).

Approach Control (Forward Pitch): Once the drone is yaw-locked on the target, a forward pitch command is issued if the violator's bounding-box width ratio ($r_w = w_{bb} / W_f$) is below the target proximity threshold $r_{approach} = 0.20$, indicating that the target is still distant. The pitch offset is computed proportionally as $P_{offset} = K_{approach} \times (r_{approach} - r_w) \times PITCH_STEP$, where $K_{approach} = 0.4$ and $PITCH_STEP = 30$ PWM units. When $r_w \geq 0.20$, the pitch channel is returned to center ($1500 \mu s$), and the drone transitions to Hover-and-Record mode.

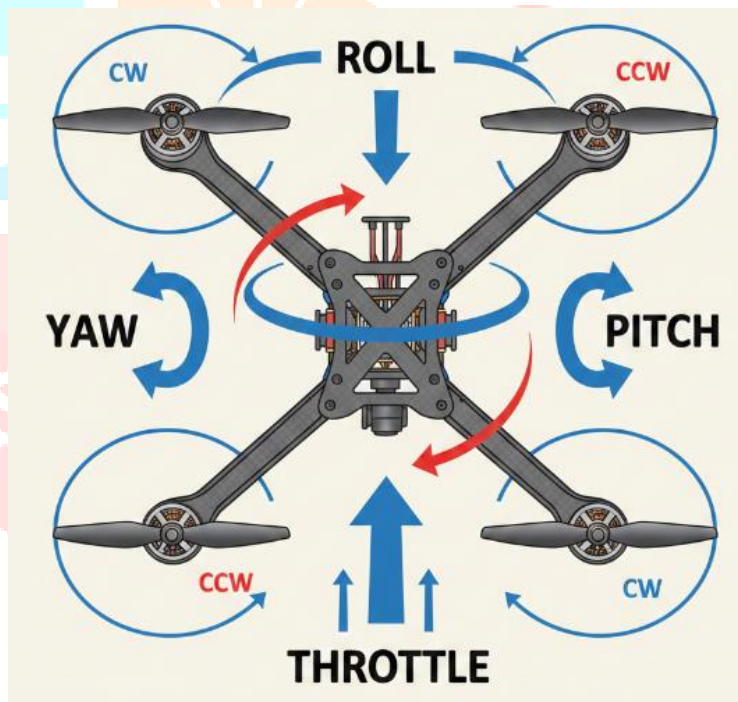


Fig.5. Flight Control Framework

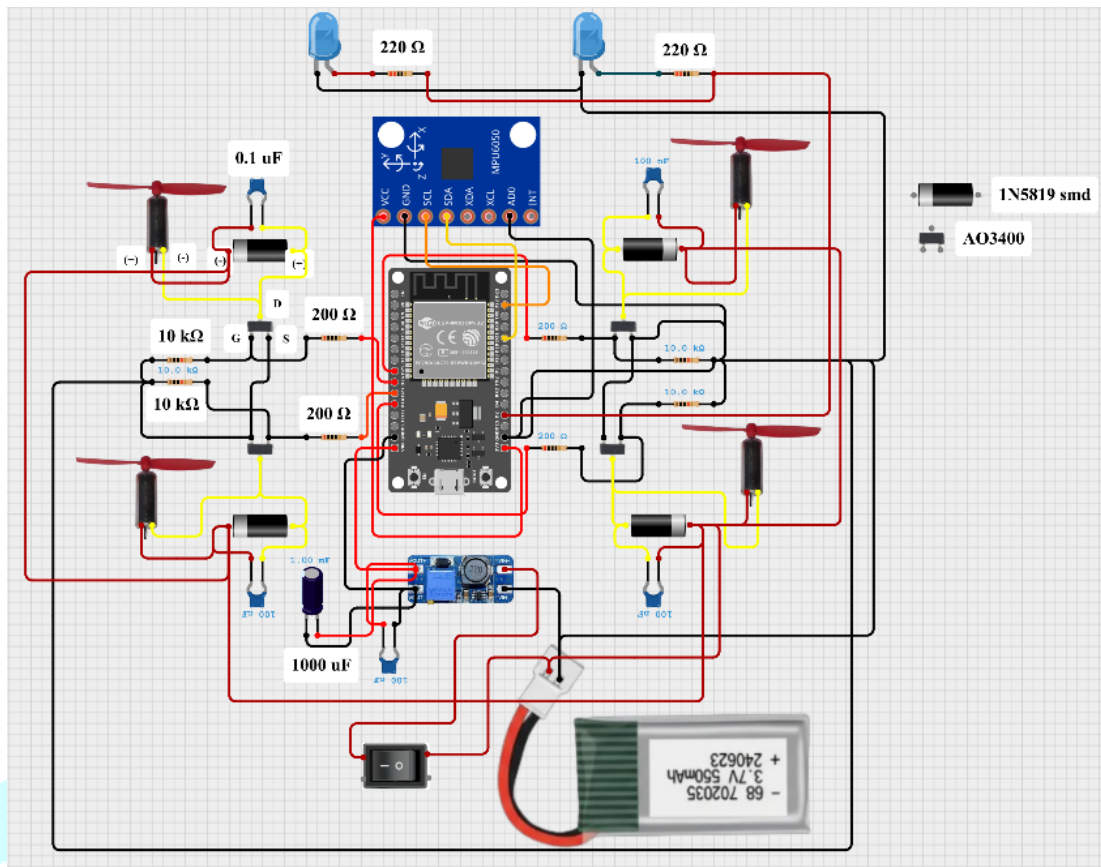


Fig.6. Circuit Diagram for AUTES Flight

3.4.2 MQTT-Based RC Communication Stack

All computed RC channel values - Roll (Ch1), Pitch (Ch2), Throttle (Ch3), Yaw (Ch4), and Arm/Disarm (Ch5) are packaged as a comma-separated 8-channel string and published to the MQTT topic via a local Mosquitto broker at 40 Hz on a dedicated background thread, independent of the vision processing loop. This decoupled threading architecture ensures that RC command delivery remains temporally consistent regardless of variable YOLO inference latency [4]. The ESP32 flight controller subscribes to the same MQTT topic, decodes the RC channel values, and maps them to motor PWM signals via the ESPFC firmware's RC input handler. The MQTT Quality of Service (QoS) level is set to 0 (fire-and-forget) to minimize latency overhead, with automatic broker reconnection configured at 1–5 second intervals to handle transient network interruptions [4], [22]. The complete end-to-end command latency from vision inference to motor actuation is maintained below 25 ms under standard operating conditions.

3.4.3 Autonomous Patrol and Mode Transition Logic

When no violation exceeding the minimum score threshold ($S_{\min} = 0.02$) is detected within the current surveillance frame, AUTES transitions to Autonomous Patrol Mode. In patrol mode, the drone executes a bidirectional yaw sweep pattern: the yaw channel is incrementally offset from center in one direction at a configurable patrol step rate until the sweep boundary is reached (± 100 PWM units from center), after which the direction is reversed following a brief hold period of 30 frames. Pitch and roll channels are held at center ($1500 \mu s$) during patrol, and throttle is maintained at the nominal hover value. Upon detection of a new violation, the system immediately exits patrol mode and re-initializes the PD yaw controller with a cold-start error, ensuring zero overshoot from accumulated integrator state. If a previously tracked violation disappears from the frame for longer than 5 seconds, the drone returns to patrol mode after resetting all control channels to their neutral values.

3.4.4 Safety Architecture and Manual Override

AUTES incorporates a layered safety architecture to prevent uncontrolled flight behaviour under detection failures or communication interruptions. The drone requires explicit arm confirmation (RC Ch5 = 2000 μ s) before any throttle command is honored by the ESPFC firmware, and arming is blocked if throttle exceeds 1100 μ s [5]. A keyboard-accessible manual override (keys E/Q/P) allows the operator to arm, disarm, or toggle between autonomous and manual control modes at any time. On system termination - whether via keyboard interrupt or exception a disarm burst of 15 consecutive packets with throttle = 1000 μ s and arm = 1000 μ s is transmitted at 50 ms intervals before MQTT disconnection, ensuring the drone safely de-arms even under abrupt shutdown conditions [3], [12]. The RC publish thread operates as a daemon thread, guaranteeing that RC stream cessation on system exit does not leave the drone in an undefined control state.

IV. RESULTS AND DISCUSSION

4.1. Helmet Detection Model Performance

The custom-trained YOLOv8 helmet detection model was evaluated using standard COCO metrics on a held-out test set. It achieved 74.2% precision and 63.7% recall, indicating reliable positive predictions with some missed violation instances. The overall mAP@0.5 performance reflects effective detection capability under real-world variability. Class-wise results show a clear imbalance: the *With Helmet* class achieved mAP@0.5 of 0.554, while the *Without Helmet* class reached 0.346. This disparity arises from dataset imbalance, where helmeted instances dominate training data, leading to weaker feature learning for violation cases. Despite this, the system remains operationally effective, as the priority scoring mechanism assigns higher weight to violation detections, compensating for lower confidence. Training over 100 epochs demonstrated stable convergence, consistent loss reduction, and no overfitting, confirming that the chosen YOLOv8 configuration is well-suited to the dataset and application. Figure 7-12 illustrates the output of trained YOLO model with evaluation metrics such as F1-Confidence curve, Precision-Confidence curve, Precision-Recall curve, Recall-Confidence curve, Confusion Matrix and Normalized Confusion Matrix. Figure 13 Shows the Bounding boxes labeled With Helmet / Without Helmet on traffic frames during the predictions.

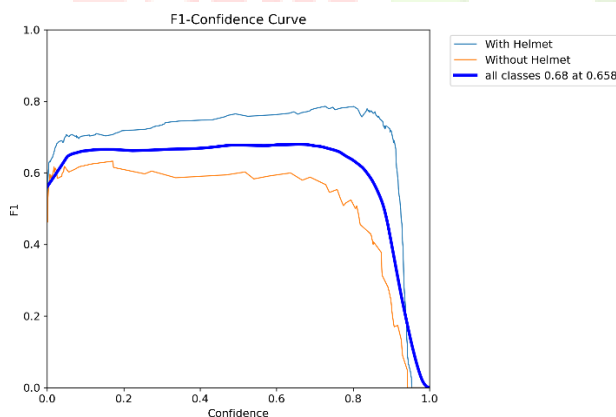


Fig.7. F1-Confidence Curve

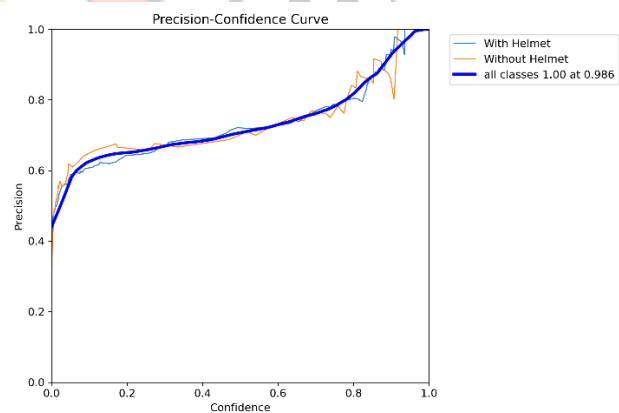


Fig.8. Precision-Confidence Curve

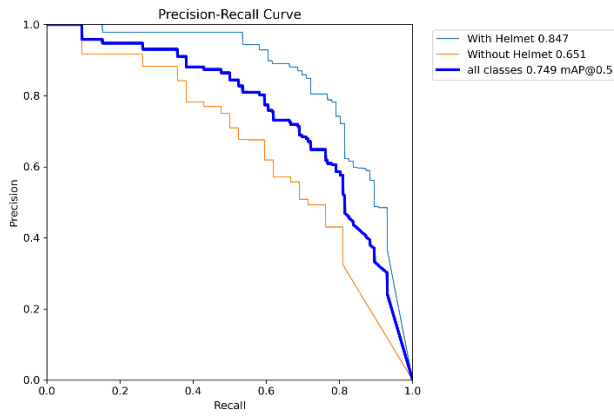


Fig.9. Precision-Recall Curve

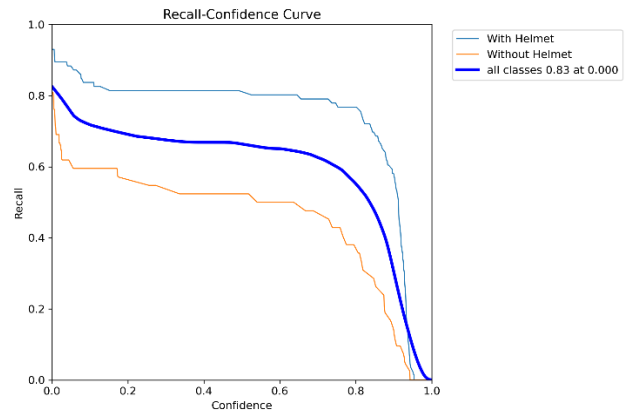


Fig.10. Recall-Confidence Curve

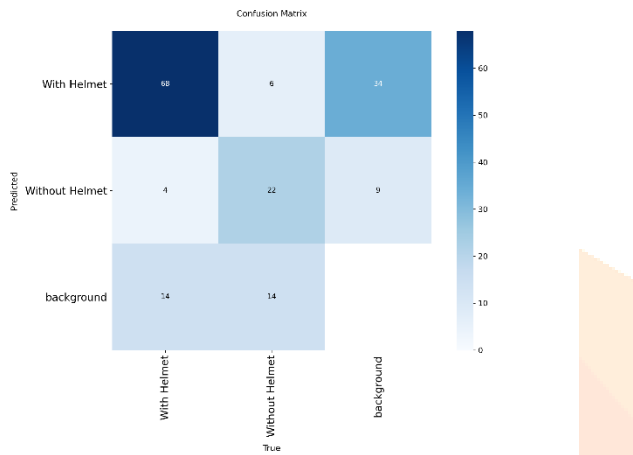


Fig.11. Confusion Matrix

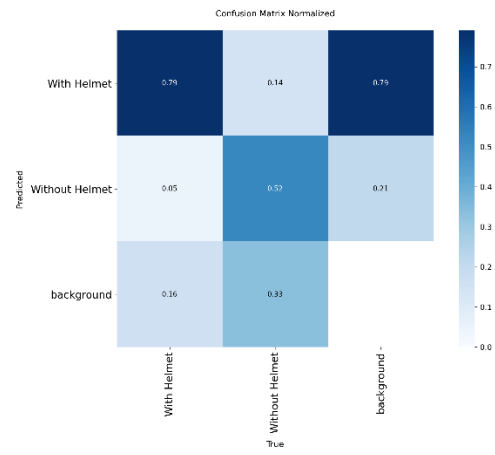


Fig.12. Normalized Confusion Matrix

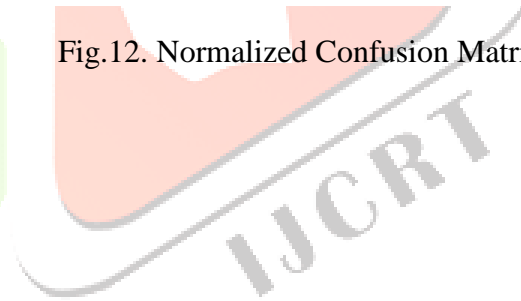
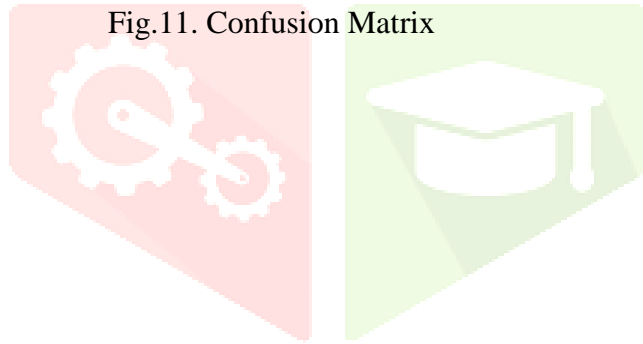




Fig.13. Bounding boxes labeled With Helmet / Without Helmet on traffic frames

4.2. Autonomous Drone Control and System Integration

The autonomous flight control subsystem of AUTES was validated through end-to-end testing on the modified E88 Pro drone. The RC command publisher, running at 40 Hz independently of the vision loop, ensured consistent command delivery with latency below 25 ms, meeting real-time UAV control requirements. The decoupled threading architecture effectively prevented delays caused by variable YOLO inference time. The yaw PD controller achieved stable target alignment with minimal oscillation, aided by smoothing and a ± 30 -pixel dead-zone. Target lock-on was consistently reached within 1.5–3 seconds, after which pitch-based approach control was activated. The altitude controller maintained stable hover with minimal deviation, while dead-zone tuning prevented oscillatory throttle behaviour. The anomaly hold mechanism ensured tracking continuity during brief detection losses. Mode transitions occurred seamlessly without control instability. MQTT communication demonstrated reliable transmission with no packet loss and quick reconnection. The system achieved robust real-time perception-to-control integration on embedded hardware. The complete system integration thus validated that the AUTES pipeline from raw video frame through dual-stage YOLO inference, priority scoring, PD control computation, MQTT transmission, and ESP32 RC decoding operates as a coherent, real-time autonomous enforcement platform on cost-effective embedded hardware.

4.3. Comparative Analysis and Discussion

AUTES is evaluated against existing traffic surveillance and UAV-based systems across detection, autonomy, hardware validation, and infrastructure independence. Conventional CCTV systems are limited to static monitoring without mobility or active response. Vision-based helmet detection on embedded platforms achieves high accuracy but remains passive, lacking enforcement capability. Manually piloted

UAVs provide mobility but depend on human intervention, limiting real-time responsiveness. This system overcomes these limitations by integrating real-time YOLOv8-based detection with a fully autonomous quadrotor capable of dynamic target tracking and approach without human control. The proposed composite priority scoring mechanism enables efficient multi-target selection using confidence, object scale, and spatial proximity. The MQTT-based communication framework ensures low-latency, software-defined control without proprietary hardware dependency. Current validation uses offline video input, results confirm successful real-time perception-to-action integration. AUTES establishes a practical foundation for scalable, autonomous UAV-based traffic enforcement and future expansion toward live streaming and multi-drone coordination.

V. CONCLUSION AND FUTURE WORK

This paper provides a complete perception-to-control aerial enforcement pipeline by integrating a dual-stage YOLOv8-based helmet violation detector with an ESP32-driven quadrotor platform. This system enables real-time autonomous behaviours yaw alignment, altitude stabilization, and target pursuit using PD-based visual servoing and MQTT-based low-latency communication. A novel composite priority scoring mechanism, combining detection confidence, bounding-box scale, and spatial proximity, enables efficient multi-target selection. The custom helmet detection model achieves mAP@0.5 of 0.748, validating operational feasibility. AUTES proves that intelligent UAV-based traffic enforcement can be implemented on cost-effective embedded hardware without reliance on proprietary frameworks.

Future enhancements include improving the “Without Helmet” class performance using focal loss and data augmentation. The detection pipeline will be extended to multi-class violations. Integration of onboard camera streaming would enable fully untethered operation. GPS-assisted waypoint navigation would expand patrol coverage, while multi-UAV swarm coordination via MQTT will enable scalable enforcement. Real-world outdoor validation with robust flight stabilization and resilient communication strategies would be essential for reliable smart city deployment.

REFERENCES

- [1] B. Lin, "Safety Helmet Detection Based on Improved YOLOv8," in *IEEE Access*, vol. 12, pp. 28260-28272, 2024, doi: 10.1109/ACCESS.2024.3368161.
- [2] V. Rajamani and R. C. Koirala, "Real-time Traffic Monitoring and Helmet Violation Detection Using YOLOv8: A Deep Learning Approach for Automated Enforcement," 2025 International Conference on Software, Knowledge, Information Management & Applications (SKIMA), Paisley, United Kingdom, pp. 1-6, 2025, doi: 10.1109/SKIMA66621.2025.11155698.
- [3] Viswanadhapalli, Jaswanth & Devulapalli, Praveen & Ansari, Md & Reshwanth, "Efficient Communication in UAV Swarm Network Using ESP-32 Wi-Fi Mesh". pp.1-6, 2025, 10.1109/ICITEICS64870.2025.11340832.
- [4] Dohyun An, Hyeontae Joo, Hwangnam Kim, "Enabling low-latency digital twins for large-scale UAV networks using MQTT-based communication framework", *ICT Express*, vol. 11, Issue 2, pp. 264-269, ISSN 2405-9595, 2025, <https://doi.org/10.1016/j.ict.2025.01.001>.
- [5] Moussaoui, Ahmed & Habbab, Mohamed & Hazzab, Abdeldjebar & Slimane, Hireche & Chandra, Amrisha & Gouabi, Hicham & Alhachemi, Moulay Abdellah, "Experimental study of PID for attitude control of a quadcopter using an ESP32". *International Journal of Power Electronics and Drive Systems (IJPEDS)*. 15. 1426. 2024, doi.10.11591/ijpeds.v15.i3.pp1426-1434.

- [6] N. Praveen and Dr. Sandeep, "Anomaly Detection in Video Surveillance Using YOLOv8," IJRASET, vol. 13, pp. 970–976, 2025, doi: 10.22214/ijraset.2025.73680.
- [7] L. Cedro, K. Wieczorkowski, and A. Szcześniak, "An Adaptive PID Control System for the Attitude and Altitude Control of a Quadcopter," Acta Mechanica et Automatica, vol. 18, 2024, doi: 10.2478/ama-2024-0004.
- [8] Aditya Vardhan Reddy Katkuri et al., "Autonomous UAV navigation using deep learning-based computer vision frameworks: A systematic literature review", Array, vol. 23, 100361, ISSN 2590-0056, 2024, <https://doi.org/10.1016/j.array.2024.100361>.
- [9] Syed Awais Ali Shah et al., "Robust finite-time model reference adaptive control for attitude control of four-rotor unmanned aerial vehicle with disturbances", Ain Shams Engineering Journal, vol. 16, Issue 11, 103639, ISSN 2090-4479, 2025, <https://doi.org/10.1016/j.asej.2025.103639>.
- [10] N. Tahilramani et al., "Edge-based AI solution for enhancing urban safety: helmet compliance monitoring with YOLOv9 on Raspberry Pi," Discover Internet Things, vol. 5, p. 25, 2025, doi: 10.1007/s43926-025-00113-9.
- [11] A. Baharuddin and A. Basri, "Self-Tuning PID Controller for Quadcopter using Fuzzy Logic," Int. J. Robotics and Control Systems, vol. 3, pp. 728–748, 2023, doi: 10.31763/ijrcs.v3i4.1127.
- [12] P. Sodha and A. Kumar, "An Overview of Wireless Communication Protocols and Techniques for Drones," in Advances in Design and Automation, Lecture Notes in Mechanical Engineering, Springer, Singapore, 2025, doi: 10.1007/978-981-97-5621-6_29.
- [13] Murat Bakirci, "Internet of Things-enabled unmanned aerial vehicles for real-time traffic mobility analysis in smart cities", Computers and Electrical Engineering, vol. 123, Part D, 110313, ISSN 0045-7906, 2025, <https://doi.org/10.1016/j.compeleceng.2025.110313>.
- [14] Shoman, M et al., "Enforcing Traffic Safety: A Deep Learning Approach for Detecting Motorcyclists' Helmet Violations Using YOLOv8 and Deep Convolutional Generative Adversarial Network-Generated Images", Algorithms, 17, 202, 2024, <https://doi.org/10.3390/a17050202>.
- [15] Muaz Al Radi et al., "Progress in artificial intelligence-based visual servoing of autonomous unmanned aerial vehicles (UAVs)", International Journal of Thermofluids, vol. 21, 100590, ISSN 2666-2027, 2024, <https://doi.org/10.1016/j.ijft.2024.100590>.
- [16] I. A. Djafara, M. I. Moctar and D. Abdourahimoun, "Target Tracking with an Unmanned Aerial Vehicle Using Visual Servoing," 2024 12th International Conference on Systems and Control (ICSC), Batna, Algeria, pp. 20-26, 2024, doi: 10.1109/ICSC63929.2024.10928827.
- [17] T. Elmokadem and A. V. Savkin, "Towards Fully Autonomous UAVs: A Survey," Sensors, vol. 21, no. 18, p. 6223, 2021, doi: 10.3390/s21186223.
- [18] G. Moepi et al., "Smart Surveillance Systems: Trends, Challenges and Future Directions," The Indonesian J. Computer Science, vol. 14, 2025, doi: 10.33022/ijcs.v14i2.4802.
- [19] E. F. Agyemang, "Anomaly detection using unsupervised machine learning algorithms: A simulation study," Scientific African, vol. 26, 2024, doi: 10.1016/j.sciaf.2024.e02386.
- [20] Sushma, Polasi & Vajjha, Haragopal, "Secure Drone Communications using MQTT protocol", International Journal of Computational and Experimental Science and Engineering, 2024, doi:10.10.22399/ijcesen.685.

- [21] Y. Said et al., "AI-Based Helmet Violation Detection for Traffic Management System," CMES, vol. 141, no. 1, pp. 733–749, 2024, doi: 10.32604/cmcs.2024.052369.
- [22] R. Nasir et al., "An Enhanced Framework for Real-Time Dense Crowd Abnormal Behaviour Detection Using YOLOv8," Artificial Intelligence Review, vol. 58, 2025, doi: 10.1007/s10462-025-11206-w.

