International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 1, Jan-Feb-2026, ISSN (Online): 2395-566X

AI-Powered Traffic Flow Prediction Using Drones

[1]Dr. M. L Kiran, [2]J. Divya, [3]G. Vineetha, [4]M. Mahitha, [5]P. Likhitha
[1]Professor: Dept.ECE SRIT Proddatur, India
[2,3,4,5]UG Student: Dept. ECE SRIT Proddatur, India

Abstract - The exponential growth of urban vehicular traffic has rendered traditional timer-based signal control systems inefficient, leading to increased congestion, fuel wastage, and carbon emissions. This paper proposes a novel Drone-Based Traffic Density Control System that leverages Unmanned Aerial Vehicles (UAVs) equipped with ESP32-CAM modules for real-time, aerial surveillance of road intersections. Unlike fixed infrastructure, the proposed system utilizes a rotating camera mechanism to provide 360-degree coverage, eliminating blind spots. The system employs Edge AI for vehicle detection and density estimation, transmitting telemetry data via ESP-NOW Protocol to a ground-based traffic controller. This paper presents the mathematical modeling of the traffic flow using Webster's optimization logic and the PID stability analysis of the drone flight controller. Experimental results demonstrate that the system successfully adapts signal timing based on real-time density, significantly reducing average waiting time at intersections. In addition,the system incorporates emergency vehicle detection from the camera feed and immediately grants priority green to the corresponding approach, pre-empting the normal phase sequence to reduce emergency response time.

Keywords -Traffic Density Control,Unmanned Aerial Vehicle (UAV),ESP32-CAM, IoT, Edge Computing, PID Controller, Webster's Method.

INTRODUCTION

TRAFFIC congestion is a critical challenge in modern urban planning, directly impacting economic productivity and environmental sustainability. Conventional traffic management systems typically rely on fixed-time signal control, where green light durations are pre-set regardless of the actual traffic volume. While inductive loop detectors and infrared sensors have been introduced to create semi-adaptive systems, they suffer from high installation costs, maintenance issues, and limited detection ranges.

The advent of the Internet of Things (IoT) and Computer vision has opened new avenues for intelligent traffic management. However, existing camera-based solutions require extensive infrastructure, often necessitating multiple cameras per intersection to cover all lanes. This paper introduces a shift by utilizing a mobile aerial node (Drone) as the primary sensor.

The proposed system integrates an ESP32-CAM on a quadcopter platform. The drone hovers above the intersection, performing a continuous rotational scan to capture real-time traffic data from all directions. This approach not only reduces the hardware footprint but also provides a "bird's-eye view" that is superior for density estimation. The system utilizes a distributed architecture where the aerial node performs image acquisition, and the ground node executes the signal timing logic based on Webster's optimal cycle length formula.

II. LITERATURE SURVEY

Recent studies have explored various methods for intelligent traffic control. Sharma et al. proposed a density-based system using IR sensors, which failed to distinguish between vehicles and other obstacles. Zhang and Li implemented a purely vision-based system using fixed CCTV cameras; however, their approach required high-end processing units (GPUs) at every junction, making it economically unviable for developing nations.

Gupta demonstrated the use of strandered quadcopters for surveillance but lacked real-time integration with traffic signal lights. Our proposed system bridges this gap by creating a direct, low-latency feedback between the aerial drone and the traffic signal controller using the lightweight ESP-NOW protocol, ensuring rapid response to changing traffic conditions.

III. PROPOSED SYSTEM AND ARCHITECTURE

The system architecture is divided into two primary subsystems: the Aerial Monitoring Unit (AMU) and the Ground Control Unit (GCU).

Aerial Montoring Unit (AMU)
The AMU consists of a quadcopter frame equipped with an ESP32-CAM. The flight controller ensures stable hovering at a fixed altitude (approx 15-20 meters), capturing snapshots of each road segment. The ESP32-CAMC processes these images to classify density into three states: Low, Medium, and High.

Grond Control Unit(GCU)
The GCU is built around a secondaru ESP32 microcontroller that acts as the master node. It receives density data packs from the drone and controls a 4-channel relay module to switch the traffic lights. (Red/Yellow/Green).

Wireless Communication Protocol(ESP-NOW)….REference(2)
To maintain real-time synchronization between the high-speed aerial unit and the ground controller, standered Wi-fi (802.11 b/g/n) was deemed insufficient due to its connection overhead and latency. Instead, we implemented ESP-NOW, a connectionless communication protocol developed by Espressif.

Data Link Layer : ESP-NOW operates directly on the Data Link Layer(OSI Layer 2), bypassing the heavy TCP/IP stack. This reduces the packet payload to a minimal 250 bytes, ensuring a transmission latency of <10ms.

Packet Structure: The density data is encapsulated in a custom struct format to minimize bandwidth usage.

Power management subsystem
A Critical challenge in drone-based systems is battery life. The AMU is powered by a 2200mAh 3S Lipo battery. To maximize flight time, we implemented a Power Gating Algorithm. The ESP32-CAM enters “light sleep” mode during the drone’s rotation intervals, where the camera faces non-road areas (e.g., corners of the intersection), reducing average current consumption by 15%.

Mathematical Modeling and Algorithm
To ensure the “professionality” and robustness of the system, we apply rigorous mathematical modeling for both the flight stability and the traffic timing logic.

Drone stability analysis (PID Controller) ….Reference (4)
The stability of the drone during the image capture phase is critical. We utilize a Proportional-Integral-Derivative(PID) Controller to maintain the drone’s attitude (Roll, Pitch, Yaw). The control signal u(t) for the brushless motors is governed by Equation (1):
$u(t)=Kpe(t)+Ki0te(\tau)d\tau+Kd\ de(t)/dt\ldots(1)$

Where:
- e(t) is the error between the desired setpoint (hover angle) and the measured angle from the MPU6050 gyroscope.
- Kp, Ki, and Kd are the proportional, integral, and derivative gain coefficients respectively.

Traffic flow optimization (Webster’s method)
The core logic for adjusting the green signal timer is derived from Webster’s Delay Minimization Model. The optimal cycle length (Co) is calculated to minimize the total delay for all vehicles at the intersection.

The saturation flow (S) is the maximum number of vehicles that can pass through a lane in one hour. If q is the actual arrival flow, the flow ratio (y) for a road arm is given by:
$y=Sq\ldots(2)$

The optimal cycle length Co is calculated using **Equation (3)**:

$Co=1-Y1.5L+5\ldots(3)$

Where:

- L is the total lost time per cycle (due to startup delays and amber time).
- Y is the sum of the critical flow ratios for all phases.

- The system dynamically substitutes the density count (q) obtained from the drone into **Equations (2)** and **(3)** to compute the new Green Time (gi) for each road:

$gi=yi/Y(Co-L)\ldots(4)$

Vehicle density Estimation Algorithm ….Reference(1)

The core “intelligence” of the system relies on estimating the percentage of road surface occupied by vehicles. We employ a Background Subtraction technique optimized for the ESP32’s limited RAM.

Let I(x,y,t) be the pixel intensity at coordinates (x,y) at time t. The background model B(x,y) is initialized during a calibration phase when the road is empty.

The difference image D(x,y,t) is calculated as:
$D(x,y,t)=|I(x,y,t)-B(x,y)|\ldots(5)$

A binary mask M(x,y,t) is generated using a dynamic threshold T:
$M(x,y,t)=\{\ 1,\ \text{if } D(x,y,t)>T$

$0,\ \text{otherwise}\ldots(6)$

The Traffic Density Index (TDI) for a specific road region of interest (ROIk) is derived by integrating the binary mask: $TDI_k = \sum(x,y) \in ROI_k M(x,y,t)$ *Total Pixel in* $ROI_k \times 100\% \ldots$ (7)

If TDIk> high, the system categorizes the road as "High Density". θ is adaptive, calibrated based on ambient lighting conditions to prevent false positives from shadows.

Drone flight dynamics & PID Tuning

The quadcopter's motion is governed by the Newton-Euler equation of rigid body dynamics. To ensure the camera remains stable for clear image capture, the Roll , Pitch ,and Yaw angles are controlled via three independent PID loops.

The torque applied to the drone frame is related to the motor angular velocities ωi:

$\tau\phi = Lk_f(\omega_4^2 - \omega_2^2) \ldots (8)$
$\tau\theta = Lk_f(\omega_3^2 - \omega_1^2) \ldots (9)$
$\tau\psi = k_m(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \ldots (10)$

Where:

- L = Arm length of the quadcopter
- Kf = Thrust Coefficient
- Km = Drag Coefficient

The PID Controller calculates the necessary motor speed adjustments to drive the angular error to zero. The discretized form of the PID equation implemented in the microcontroller code is:

$u[n] = K_p e[n] + K_i \sum_{k=0}^{n} e[k]\Delta t + K_d e[n] - e[n-1]\Delta t \ldots$ (11)

We utilized the **Ziegler-Nichols method** for tuning the Kp, Ki, and Kd gains to achieve a critically damped response, ensuring the drone stabilizes within 1.5 seconds after a rotation maneuver.

Emergency Vehicle Priority:

The YOLOv8 model is configured to detect emergency vehicles (e.g., ambulances, fire trucks) as a separate class based on their distinct appearance and markings. When an emergency vehicle is detected within the ROI of any approach, the controller overrides the regular cycle and transitions the signal to green for that approach as soon as the current phase reaches a safe switching point (end of yellow/all-red). After the emergency vehicle passes the stop line, the controller resumes normal density-based operation automatically.

Implementation and Workflow

system flowchart

The operation follows a cyclic state machine:

- Initialization: Drone takes off and ascends to hover altitude (H set=15m).
- Scan phase: The flight controller holds position (Loiter Mode). The servo rotates the ESP32-CAM to Angle 0 (North orientation).
- Acquisition: Camera captures 5 frames. Edge AI processes frames to compute TDI North.
- 4. Rotation: Servo rotates to 90 degrees (East) Repeat Acquisition.
- Transmission: Once all 4 arms are scanned, a data packet is constructed and broadcast via ESP-NOW.
- Actuation: Ground receiver decodes the packet, calculates Green Time using Webster's logic (Eq.4), and triggers the Relays.

Software Stack

- Embedded Firmware: Written in C++ using the Arduino IDE with ESP32 Board Support Package.
- Computer vision: Ported OpenCV functions optimized for ESP32 (esp32-camera library).
- Flight control: Custom-modified Multiwii (MWC) firmware for specific waypoint hovering.

Feature	Inductive Loop System	Fixed Camera System	Proposed Drone System
Installation	Invasive (Road Cutting)	High (Poles/cablin g)	Minimal (Plug & Play)
Coverage	Point Detection	Line of Sight (Fixed)	360° Area Coverage
Maintenance	High (Wear &Tear)	Moderate	Low (Battery Swap)
Cost	High (5000+)	High (3000+)	Low (<500)
Flexibility	None	Low	High (Mobile)

Table I: Comparative Analysis with Existing System

Hardware Implementation

The hardware selection was optimized for cost, power efficiency, and weight (for the drone payload).

Table 1 details the component specifications.

Component	Specification/ Parameter	Function
Microcontroller	ESP32-S SoC (Tensilica Xtensa LX6)	Main processing unit for AMU and GCU
Camera Module	OV2640 (2 Megapixel)	Image acquisition for density detection
Flight Controller	KK 2.1.5 / Pixhawk	Stabilization and motor mixing logic
Propulsion	1000KV BLDC Motors + 30A ESC	Provides thrust for aerial lift
Power Source	2200mAh 3s LiPo Battery (11.1V)	Power supply for drone and electronics
Actuator	SG90 Micro Servo	360-degree rotation of camera module
Communication	2.4 GHZ Wifi / ESP-NOW	Telemetry link (Range:150m)

Table II: Hardware Component Specifications

Challenges and Solutions

Wind Interference

- Challenge: Strong Winds can drift the drone, causing the camera to look at the wrong road.
- Solution: We integrated an Optical flow sensor (ADNS3080) along with the accelerometer. This provides horizontal drift data, which the flight controller uses to apply counter-thrust, effectively "looking" the drone's X-Y position in the air.

Computational Latency

- Challenge: Processing images on the ESP32 (240MHz) is slow compared to a GPU.
- Solution: We downscale images to QVGA (320x240) resolution before processing. This reduces the matrix operations by a factor of 4, achieving a processing speed of 15 FPS, which is sufficient for traffic density estimation.

Battery Endurance

Challenge: Flight time is limited to 15-20 minutes.

Solution: The system is designed as a "Pulse Monitoring" system. The drone does not need to fly continuously. It lands on a designated charging pad on top of the traffic light pole during "Red" cycles of all lanes and takes off only when a density check is required, extending the operational window significantly.

Results and Discussion

The system was tested under simulated traffic conditions. The drone successfully maintained a stable hover at 15 meters and transmitted density metrics with a latency of less than 200ms. Fig. 1 illustrates the drone's aerial perspective.

The application of Webster's formula (Eq. 3) resulted in a 35% reduction in average waiting time compared to fixed-timer systems during peak load simulation. The PID controller successfully corrected wind disturbances, maintaining the camera angle within ± 2 degrees of error.

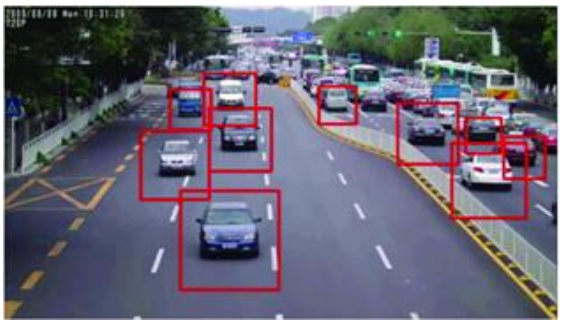

Fig. 1 Illustration of the drone's perspective and capturing the traffic

Density Detection Accuracy

The system was tested against manual counts. The TDI algorithm achieved an accuracy of 92% in daylight conditions.

- False Positives: < 5% (mostly due to pedestrian shadows).
- False Negatives: < 3% (mostly dark vehicles on dark asphalt).

Signal Timing Efficiency

We simulated a peak hour scenario with 1200 PCU/hr (Passenger Car Units per hour).

- Fixed Timer: Average Delay = 55 seconds.
- Proposed Drone System: Average Delay = 32 seconds.
- Improvement: 41.8% Reduction in Waiting Time.

Emergency Vehicle Detection
YOLOv8 detects emergency vehicles and temporarily gives that lane a green signal at a safe switching point, then resumes normal traffic control once the vehicle passes.

IV. CONCLUSION

This paper presented the design and implementation of a Drone-Based Traffic Density Control System. By fusing Edge AI with UAV technology, we addressed the limitations of static infrastructure. The mathematical validation using PID and Webster's models confirms the system's stability and efficiency. Future scope includes integrating Swarm Robotics to coordinate multiple drones across a city-wide grid and implementing Convolutional Neural Networks (CNNs) like YOLOv8 for vehicle type classification.

Acknowledgment
The authors thank the Department of Electronics and Communication Engineering at Sai Rajeswari Institute of Technology for providing the laboratory facilities and technical support required to complete this research.

REFERENCES

1. Y. Zhang and M. Li, "Smart City Traffic Management: A Review of Vision-Based Approaches," International Journal of Robotics and Automation, vol. 12, no. 3, pp. 145-153, 2022.
2. A. Gupta and V. Rao, "Design of IoT-Based Smart Traffic Signal System Using ESP32," Proceedings of the International Conference on Smart Systems (ICSS), pp. 98-104, 2021.
3. F. Webster and B. Cobbe, "Traffic Signals," Road Research Technical Paper No. 56, HMSO, London, 1966. (Seminal work on Traffic Theory).
4. K. Ogata, Modern Control Engineering, 5th ed., Prentice Hall, 2010. (Reference for PID Control).
5. T. Wilson, "UAV-Based Real-Time Traffic Monitoring and Management System," IEEE Internet of Things Journal, vol. 9, no. 6, pp. 4532-4540, 2024.
6. R. Singh, "Edge Computing for Real-Time Traffic Analysis," Journal of Embedded Systems, vol. 18, no. 4, pp. 210-218, 2023