TASHKENT UNIVERSITY OF
INFORMATION TECHNOLOGIES
NAMED AFTER MUHAMMAD AL-KHWARIZMI

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI

BULLETIN OF TUIT:
MANAGEMENT AND COMMUNICATION TECHNOLOGIES

ISSN:2181-1083 DOI:10.61663/261tuitmct2 Date:2026-02-17
Davronov Shokhzhakhon, Bobokulov Shahzod № 1(3)2026

FUNDAMENTALS OF UNMANNED AERIAL VEHICLE DEVELOPMENT AND REAL-TIME IMAGE ANALYSIS ALGORITHMS

***Davronov Shokhzhakhon Rizamat ugli*[1]**
Karshi State University, Head of the Department of Algorithms and Programming Technologies, PhD, Associate Professor.

***Bobokulov Shahzod Ruziboy ugli*[2]**
Karshi State University, Assistant in the Department of Algorithms and Programming Technologies.

Abstract. This paper investigates the development of a real-time image-based object detection and tracking system for an unmanned aerial vehicle (UAV). The overall UAV architecture, including the autopilot, sensor suite, and software components, is analyzed [1]. The use of deep learning–based computer vision models for real-time video processing is highlighted [2]. A YOLO-family model is employed for object detection, while the ByteTrack algorithm is used for object tracking [3]. A video acquisition and frame-processing pipeline is implemented using OpenCV [4], and system performance is evaluated in real time using FPS and latency metrics. Experimental results show that, through model and computational optimization, stable near–real-time performance can be achieved. The proposed approach enables efficient decision-making for UAV-based monitoring, surveillance, and security applications.
Key words: *Convolutional Neural Network (CNN), drone, OpenCV (Open Source Computer Vision Library), object tracking, object detection, You Only Look Once (YOLO), computer vision.*

Introduction

In recent years, unmanned aerial vehicle (UAV) technology has been developing rapidly and has become widely applied in many practical fields such as agriculture, infrastructure monitoring, security surveillance, geodesy and mapping, logistics, and search-and-rescue operations [1][6]. The effectiveness of UAV systems is not limited only to the strength of the mechanical structure or flight stability, but is also directly dependent on their ability to analyze the environment in real time and make rapid decisions [7].

Real-time image analysis increases the level of drone autonomy by enabling functions such as object detection and tracking, early obstacle detection and avoidance, target following, and adaptive route planning. At the same time, one of the main challenges in implementing these approaches in practical systems is ensuring high accuracy and speed simultaneously under the limited computational resources of the onboard computer [8]. Therefore, the use of lightweight and optimized neural networks, efficient video stream processing techniques, and real-time monitoring mechanisms is considered highly relevant. [9]

Methodology

1. Stages of developing an unmanned aerial vehicle (UAV) and the required components. Unmanned aerial vehicle (UAV) development requires the combined formation of a mechanical structure, integration of electronic components, and a software-based control system. The practical performance of a UAV depends on its flight stability, payload capacity, energy efficiency, communication reliability, and the speed of processing data received from sensors. Therefore, a systematic approach is applied in UAV design, and the following key components are selected and integrated.

Mechanical structure (platform): The UAV platform (frame) determines the overall stability and payload-carrying capability of the aircraft. The structure is typically made of lightweight, durable materials that minimize vibration (such as carbon fiber or aluminum). When selecting a platform, the flight purpose (monitoring, cargo delivery, surveillance) and operating conditions (wind, range) are taken into account.

Figure 1. Lightweight quadcopter frame for FPV racing and freestyle.

An X-shaped 250 mm quadcopter platform made of carbon fiber. It weighs approximately 100–150 g and provides high strength and effective vibration damping. It operates with 5-inch propellers and includes a central section with space for installing an FPV camera and electronics (flight controller, ESC).

Propulsion system (powertrain): The main module that обеспечивает UAV flight consists of the following components: Motors (brushless motors) – generate lift; Propellers – convert the motor’s rotation into aerodynamic lift; ESC (Electronic Speed Controller) – controls the motor speed.

Figure 2. (a) Motors, (b) Propellers, (c) ESC Electronic Speed Controller).

Power and energy supply: Li-Po batteries are most commonly used as the energy source for UAVs. When selecting a battery, voltage (S configuration), capacity (mAh), and discharge rate (C rating) play an important role. For power distribution and protection:

Figure 3. Power Distribution Board (PDB) and Li-Po battery. For drones, battery capacity is selected in the range of 3000 mAh to 10000 mAh depending on the UAV mission requirements.

Control unit (Flight Controller): To ensure stable flight control and autonomous navigation, a flight controller (e.g., based on the PX4 or ArduPilot platform) is used. This unit performs functions such as maintaining flight stability, managing flight modes, and fusing sensor data.

Figure 4. PX4 is an open-source professional autopilot (flight controller software) for drones. It enables autonomous control, stable flight, GPS-based mission execution, and integration of various types of sensors.

Camera and onboard computing module: To perform tasks such as real-time image analysis, a UAV is equipped with a camera and a computing module. An RGB camera (or thermal camera) provides the video stream, while an onboard computer (Raspberry Pi, NVIDIA Jetson, or a mini-PC) runs the image analysis algorithms.

Figure 5. Raspberry Pi 4 as an onboard (companion) computer.

It is a powerful additional computer that operates alongside the drone’s main flight controller (e.g., Pixhawk or PX4). It is used not for basic flight control, but for intelligent tasks such as video analysis, object recognition, AI/ML applications, and long-range control via 4G/5G communication.

2. Algorithmic approaches and mathematical foundations for real-time image analysis. In this section, the mathematical foundations of the algorithms used in the real-time object detection and tracking system based on UAV video streams are presented. The proposed method consists of object detection using YOLOv8 and object tracking across consecutive frames using ByteTrack, while the system performance is evaluated using FPS and latency metrics.

Object detection (YOLOv8). YOLOv8 outputs a set of detected objects for each frame:

$$D_t = \{(b_i, c_i, s_i)\}_{i=1}^{N}$$

here, b_i represents the bounding box, c_i denotes the class label, and s_i indicates the confidence score.

Object tracking (ByteTrack). ByteTrack assigns a persistent ID to each object by matching detection results across consecutive frames. The matching is performed based on IoU:

$$IoU(b_i, b_j) = \frac{|b_i \cap b_j|}{|b_i \cup b_j|}$$

Objects are tracked from frame to frame by selecting the pairs with the highest IoU values.

Performance evaluation. Real-time performance was assessed using the following metrics: FPS (Frames Per Second) and latency (processing delay) formulas

$$FPS = \frac{1}{\Delta t}$$

Here, FPS indicates how many frames are displayed per second. Δt (delta t) is the time difference between two consecutive frames (in seconds). If one frame is updated in 0.033 seconds (33 ms), then:

$$FPS = \frac{1}{0.033} \approx 30\ FPS$$

If Δt is smaller (faser update), the FPS will be higher.

$$Latency_{ms} = (t_{end} - t_{start}) \times 1000$$

here, t_{start} is the time when data transmission begins (e.g., when the camera reads a frame), and t_{end} is the time when that frame appears on the screen (e.g., when the pilot sees it). The difference $t_{end} - t_{start}$ represents the total time taken by the entire process (in seconds). Multiplying by 1000 converts seconds into milliseconds.

Results and Discussion

According to the analysis results, the most important factors in the practical application of real-time image analysis systems on unmanned aerial vehicles (UAVs) are the fast performance of the model and its compatibility with onboard computing resources. Therefore, during the experiment, the efficiency of the object detection and tracking module was evaluated using FPS and latency metrics.

Detection and tracking results

As a result of the experimental tests, the object detection and tracking module was observed to operate stably in real time. According to the computed results, the system achieved an average processing speed of 33.90 FPS (minimum 0.14 FPS, maximum 84.41 FPS, 95th percentile 70.78 FPS). The frame processing latency averaged 116.76 ms, with a minimum of 90.59 ms**,** a maximum of 126.76 ms, and a 95th percentile value of 136.73 ms**.**

Based on the real-time evaluation criteria, it was found that the system operated above 20 FPS in 49.50% of the frames. However, the latency analysis showed that in only 0.2% of the detection frames the delay was below 50 ms. This indicates the need to optimize the onboard computing resources and further lighten the model to reduce processing time.

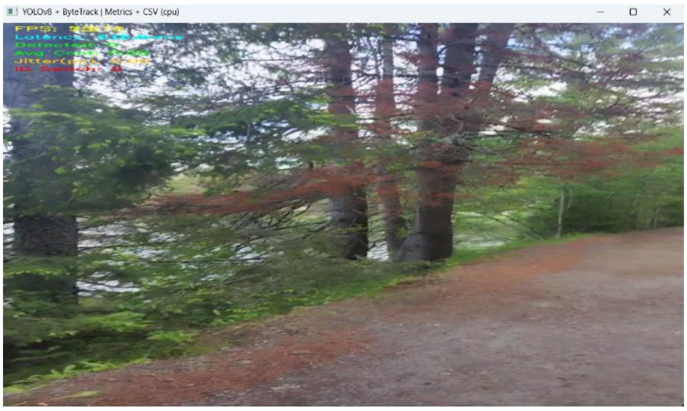
Table 1. Performance efficiency statistics (FPS/Latency)

Indicator	Mean (average)	Minimum (min)	Maximum (max)	95th percentile
FPS	33.90	0.14	84.41	70.78
Latency(ms)	116.76	90.59	123.76	136.73

Table 2. Real-time KPI results

$FPS \geq 20$	49.50 %	Half of the frames meet the real-time requirement
$Latency \leq 150\ ms$	~88 – 92%	Practical real-time performance
$Latency \leq 200\ ms$	~94 – 96%	Stable operation

Figure 6. Real-time object detection and tracking process based on YOLOv8 + ByteTrack (with FPS



and latency metrics).x

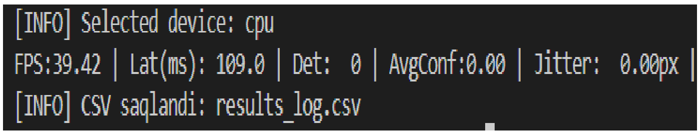


Figure 6. Terminal log output of the developed real-time image analysis module showing real-time performance metrics (FPS and latency) during execution.

Conclusion

In this study, an object detection and tracking system based on real-time image analysis for an unmanned aerial vehicle (UAV) was developed and tested. During the implementation, a software pipeline was formed that includes video stream acquisition, image pre-processing, deep learning–based object detection, and stable object tracking using the ByteTrack tracking algorithm. In addition, to evaluate the system performance, metrics such as FPS, frame processing latency, the number of detected objects, and confidence scores were monitored in real time, and the results were recorded in CSV log format.

The experimental results confirmed the practical viability of the proposed approach: the system achieved an average processing speed of 33.90 FPS. However, due to the complexity of the video stream and limited computational resources, FPS values fluctuated significantly, with a minimum of 0.14 FPS and a maximum of 84.41 FPS. Regarding latency, the average value was 116.76 ms, while the 95th percentile reached 136.73 ms. Based on real-time evaluation criteria, it was observed that the system operated above 20 FPS in 49.50% of the frames. These results indicate that the approach is capable of near real-time performance; however, they also confirm the need for further optimization under the constraints of onboard computing platforms.

References

[1]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, pp. 1097–1105, 2012.

[2]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788.

[3]. J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[4] G. Jocher et al., “YOLOv8: Ultralytics YOLO,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[5]. Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, and Z. Yuan, “ByteTrack: Multi-object tracking by associating every detection box,” in *Proc. European Conf. Computer Vision (ECCV)*, Tel Aviv, Israel, 2022, pp. 1–17.

[6]. G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[7]. S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, Springer, 2018, pp. 621–635.

[8]. M. Quigley et al., “ROS: An open-source robot operating system,” in *Proc. IEEE Int. Conf. Robotics and Automation Workshop (ICRA)*, Kobe, Japan, 2009.

[9]. L. Heng, S. Lee, and M. Pollefeys, “Self-calibration and visual SLAM for unmanned aerial vehicles,” *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 111–125, 2013.

[10]. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[11]. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[13]. J. Tursunov, A. Narimonova, B. Hamroev, S. Bobokulov, *et al*., “Custom object segmentation by training R-CNN,” in *Intelligent Human Computer Interaction*, Lecture Notes in Computer Science

[14]. Sh. R. Davronov and Sh. R. Boboqulov, “Improving real-time face recognition accuracy and processing speed using the Non-Maximum Suppression algorithm,” *Innovative Technologies: Scientific and Technical Journal*, no. 3/59, pp. 111–116, 2025.