# Development of an Edge-AI Autonomous UAV Control System for Navigation and Perception in GPS-Denied Environments

Alizhan Tulembayev[1], Gani Baiseitov[2]*, Zarina Kutpanova[3],
Alexey Semchenko[4], Dmitriy Shandronov[5], Praveen Kumar[6]
R&D Center Kazakhstan Engineering LLP, Astana, Kazakhstan[1, 2]
School of Software Engineering, Astana IT University, Astana, Kazakhstan[3, 6]
Department of the Air Force, National Defense University of the Republic of Kazakhstan, Astana, Kazakhstan[4]
Military Engineering Institute of Radio Electronics and Communications, Astana, 010000, Kazakhstan[5]

***Abstract*—In aerospace research on the cross-cutting growth of unmanned aerial vehicles (UAVs) in inspection, monitoring, and emergency response use cases, reliable operation in environments that are constrained by navigation and communication infrastructure is an issue of vital importance. Conventional UAV systems have a strong reliance on Global Positioning System (GPS) signals and constant telemetry with ground control stations that are often the cause of performance degradation when operational inside indoor environments, dense urban canyons and industrial complexes. In view of these troubles, in the present work, an edge artificial intelligence (AI) based autonomous UAV system is proposed to accomplish sensing, localization, and mission control processes fully onboard; hence nullifying the requirement of external aids for navigation. The proposed methodology works on GPS-free navigation by merging the information from the visual and inertial sensors for the accurate estimation of position in real time. Environmental awareness is achieved through object recognition procedures implemented on board, while a dedicated autonomous mission control module is responsible for the execution of high-level tasks within strict and defined safety bounds. All the computational routines reside in an embedded computing platform, ensuring low-latency responses and longevity under degraded communication scenarios. Empirical validation of the proposed system was achieved by experiments of extensive indoor and outdoor flight in a GPS-denied environment. The results show an 84\% completion rate for a fully autonomous mission, in addition to stable flying behaviour and timely control responses, confirming the feasibility of reliable UAV autonomy without reliance on external navigation infrastructure.**

***Keywords*—Unmanned Aerial Vehicle (UAV); GPS-denied environments; visual–inertial sensors; embedded computing; real-time object recognition**

## I. Introduction

Unmanned aerial vehicles (UAVs) are increasingly used in infrastructure inspection, environmental monitoring, surveillance, and emergency response. However, autonomous operation remains challenging in environments where Global Navigation Satellite System (GNSS) signals are unavailable or unreliable. Dependence on continuous communication with ground control stations further limits operational reliability when connectivity is interrupted [1], [2].

Visual–inertial odometry (VIO) and simultaneous localization and mapping (SLAM) offer alternatives to satellite-based navigation by combining camera and inertial measurements to estimate movement and represent the surrounding environment [3], [4]. Nevertheless, motion blur, poor illumination, and limited visual features can degrade localization accuracy and increase accumulated drift. Deep learning-based object detection, including convolutional and transformer-based approaches, enables UAVs to identify people, vehicles, and infrastructure, supporting more informed navigation and mission decisions [5], [6]. However, detection errors and computational constraints remain significant challenges for embedded deployment.

This study presents an integrated Edge-AI framework for autonomous UAV navigation and perception in GNSS-denied environments. The proposed architecture combines visual–inertial localization, visual place recognition for drift mitigation, and YOLO11 Nano-based object detection accelerated using TensorRT. An NVIDIA Jetson Xavier NX companion computer performs onboard perception and mission management, while a Pixhawk 6 Pro flight controller handles low-level stabilization through the PX4/MAVLink interface.

The system is evaluated across four mission profiles: waypoint navigation, perimeter patrol, search-grid coverage, and adaptive target tracking. The mission completion, localization drift, detection performance, and processing latency are used to assess system behavior. The study aims to demonstrate the feasibility of integrating localization, perception, and mission control on embedded hardware while identifying challenges that require further investigation.

## II. Related Work

ORB-SLAM3 goes one step further and provides greater persistence using advanced map management and place-recognition strategies; however, its CPU-based global optimisation mechanism may introduce latency jitter. Metric--semantic SLAM systems such as Kimera [7] enrich scene understanding but are often computationally heavy for embedded deployment.

*Corresponding author.
Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan.

Recent works have explored GNSS-denied UAV navigation using hybrid sensor fusion and terrain-aware localization techniques. Event-based approaches for visual–inertial odometry have also improved robustness in dynamic and low-light environments [8].

UAV-assisted SLAM frameworks integrating perception and mapping have been proposed for enhanced environmental understanding. Lightweight SLAM systems optimized for embedded UAV deployment have been introduced to reduce computational overhead [9].

Cooperative UAV perception strategies further enhance navigation performance in GPS-denied environments, while depth-based landing approaches improve safety during autonomous missions [10]. On the perception side, feature matching and transformer-based detection approaches have improved robustness and accuracy [11][12].

Three evaluation gaps remain. First, the integration of localization, perception, and mission control requires end-to-end timing and mission-level evaluation. Second, the contribution of visual place recognition to localization requires comparison of trajectories with and without place recognition. Third, communication robustness requires controlled testing under packet loss or link interruption. These gaps identify limitations in the available evidence rather than proven shortcomings of all existing systems.

## III. System Architecture

Localization is achieved using visual-inertial odometry and visual-inertial SLAM to maintain pose estimates and maps in real time [13]. To improve long-term robustness, drift mitigation can be augmented through place recognition and loop-closure mechanisms [14]. The estimated pose and trajectory of the UAV are achieved by fusing camera images and IMU measurements with ORB-SLAM3 (version 1.0). The camera and IMU are calibrated and synchronized to provide a consistent state estimation. When loop closure is turned on, the accumulated trajectory drift can be attenuated. The localization performance is compared with independent ground-truth measurements, and translational drift is measured with respect to the travelled distance.

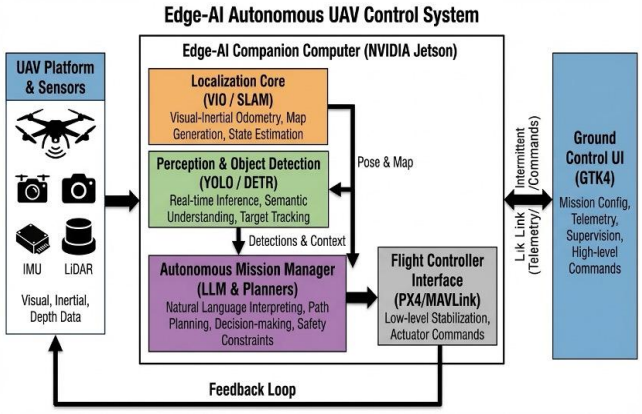


Fig. 1. Vision system capture loop.

Fig. 1 illustrates the vision processing pipeline, where image frames are asynchronously captured and processed for real-time object detection, enabling continuous perception during flight.

Edge-AI autonomous UAV control architecture. Sensor streams are processed onboard the Jetson companion computer (localization, perception, and mission management), which outputs high-level setpoints through the PX4/MAVLink interface to the flight controller. The ground-control UI provides intermittent mission updates and receives telemetry, while the UAV closes the loop through continuous state/sensor feedback.

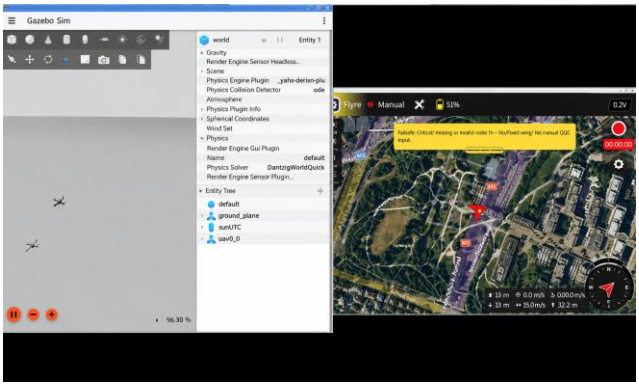


Fig. 2. Autonomous flight trials.

Fig. 2 demonstrates experimental flight scenarios across different GPS-denied environments, validating the robustness of onboard localization and navigation.

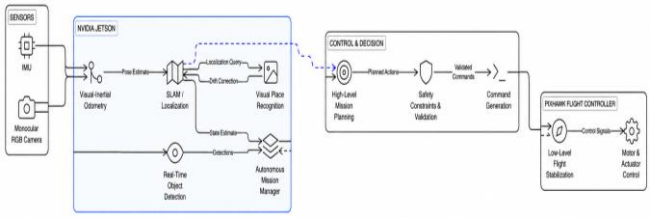


Fig. 3. System architecture.

The diagram in Fig. 3 presents the overall system architecture, highlighting the integration of perception, localization, mission management, and control modules within the edge computing framework. Mission Management and Safety Boundaries.

### A. Mission Management and Safety Boundaries

The reference architecture separates high-level mission management from low-level flight stabilization. The companion computer is intended to process localization and perception outputs and coordinate mission tasks, while the Pixhawk flight controller executes time-critical control through the PX4/MAVLink interface. The mission profiles described in the manuscript are waypoint navigation, perimeter patrol, search-grid coverage, and adaptive target tracking.

A reproducible description of the mission manager should define the states and transitions used for take-off, mission execution, target inspection, return-to-home, and landing, and where these functions are implemented. It should also specify the conditions that trigger a state transition, the handling of invalid or uncertain localization, communication timeouts, and

the flight controller's configured fail-safe behaviour. The available manuscript does not establish all numerical thresholds or the exact transition rules. These details must be taken from the implemented software and flight-controller parameters before they are described as tested safety functions.

## IV. Dataset Preparation and Partitioning

The proposed evaluation framework considers three types of data: real-flight imagery captured by an onboard camera, synthetically augmented images, and data generated through simulation. The augmentation procedures described in the manuscript include viewpoint variation, motion blur, changes in illumination, and simulated occlusion. Gazebo with PX4 Software-in-the-Loop Simulation (SITL) and hardware-in-the-loop (HITL) are also identified as environments for examining mission logic and perception-related behaviour.

However, the available records do not establish the total number of real images, synthetic images, or simulation sequences, nor do they provide a verifiable training, validation, and test split. Consequently, these data sources should be described separately, and the final paper should report their actual counts, class distribution, annotation procedure, split ratios, and any measures used to prevent related frames from appearing across training and test sets. Until these details are available, the dataset description should be interpreted as an account of the intended evaluation setup rather than a fully reproducible dataset protocol.

## V. Vehicle Configuration and Compact Dynamic Model

The experimental platform is an X-configuration quadrotor equipped with four brushless MN3110 T-Motor MN3110 (KV470) motors and 12X4-inch propellers [15]. Energy is provided by a Tattu G-Tech 6000 mAh 6S LiPo battery (nominal voltage 22.2 V) [16]. Power delivery and current/voltage sensing are handled by the Holybro PM02 V3 power module (2S- 12S input, 60A continuous rating). For GNSS-available flights and ground-truth logging, an M8N GPS module is installed; for GNSS-denied experiments, the navigation stack does not rely on GPS measurements. Using manufacturer-reported masses for major components (battery m ≈ 2.3–2.7 kg, PM02≈0.02 kg, GPS ≈0.036 kg, motors ≈0.098 kg each, including cables), the total takeoff mass is approximately m ≈ 2.3–2.7 kg, where the range accounts for the airframe, flight controller and companion computer, wiring, onboard camera payload, and mounting hardware.

Hover requires total thrust $f_T \approx mg$, and the per-rotor hover thrust is $T_{hover} = mg$. For simulation and controller verification, the standard rigid-body quadrotor dynamics are used:

$$\dot{p} = v, \tag{1}$$

$$m\dot{v} = mge_3 - R(\sum_{i=1}^{4} T_i)e_3 \tag{2}$$

$$\dot{R} = R[w]_\times, \tag{3}$$

$$J\dot{w} = T - W \times (Jw) \tag{4}$$

with rotor thrust modeled by T_i = kf \Omega_i^2 and $Q_i = k_m \Omega_i^2$, and rotor-to-body wrench mapping handled by the standard quadrotor allocation matrix.

### A. Detector Training and Deployment

The perception component uses the lightweight YOLO11 Nano object detector. The source manuscript describes a nominal training duration of 100 epochs and four target categories: aerial platforms, humans, structures and infrastructure, and ground vehicles. TensorRT is identified as the intended acceleration method for deployment on NVIDIA Jetson hardware.

The reported detector summary gives precision of approximately 0.46–0.47, recall of 0.52–0.53, mAP@0.50 of 0.43–0.44, and mAP@0.50:0.95 of 0.35–0.36. These values are retained as reported summaries; the original training logs, checkpoint, prediction files, and independently verifiable held-out evaluation split are not available in the accompanying records. They should therefore not be presented as independently reproduced measurements.

To make the procedure reproducible, the final version should state the image size, optimizer, initial learning rate, batch size, weight decay, data split, random seed, confidence threshold, non-maximum suppression settings, software versions, and TensorRT precision mode. Each parameter should be populated from the actual training and deployment configuration rather than estimated retrospectively.

## VI. Results

The reported detector metrics are precision of 0.46–0.47, recall of 0.52–0.53, mAP@0.50 of 0.43–0.44, and mAP@0.50:0.95 of 0.35–0.36. However, the original training logs, model checkpoint, prediction files, and verified held-out evaluation split are unavailable. These values therefore remain reported summaries and cannot establish an improvement over generic pretrained weights.

TABLE I. Functional Decomposition of the Proposed Edge-AI-Enabled Autonomous UAV System

| System Module | Primary Functionality |
|---|---|
| Visual-Inertial Odometry (VIO) | Estimates the UAV pose in real time by fusing visual and inertial sensor measurements in GPS-denied environments. |
| SLAM / Localization | Builds and maintains a consistent map of the environment while providing robust localization during flight without GNSS. |
| Visual Place Recognition | Assists relocalization and reduces accumulated drift by recognizing previously visited locations. |
| Real-Time Object Recognition | Provides environmental awareness by detecting objects in the UAV's surroundings using onboard perception. |
| Autonomous Mission Manager | Executes high-level mission logic based on state estimates and perception outputs while respecting safety constraints. |
| Control and Decision Module | Generates validated control commands and enforces safety boundaries for autonomous operation. |
| Flight Control (Pixhawk) | Handles low-level flight stabilization and actuator control independently of high-level autonomy. |

Table I explains the COCO-based mAP recovery to stricter 0.50:0.95 moves steadily, with the limit reaching 0.35-0.36 at the end of epochs. All of these learning curves are indicative of orderly training, with conclusive convergence, early rapid gains, and a steadier end-point with moderate accuracy, high recall, and mAP values, which level off towards the end of training.

*1) Autonomous flight trials in GPS-denied environments:* To validate a fully working system, we conducted autonomous flight experiments under GPS-denied conditions in three environments, including interior spaces, simulated urban canyon settings, and sheltered outdoor settings[20]. Tests: A simulation of Gazebo/PX4-SITL through a combination of hardware-in-the-loop (HITL) simulation and experiments on a physical flight in which the GPS was naturally non-existent or deliberately inhibited in the flight controller. The four mission types that we used (representative of search-and- rescue and monitoring missions) are Waypoint Navigation (8-12 waypoints, 150250m path length through a cluttered indoor arena (4x6x3m) and via simulated structure where a region of interest is detected), Perimeter Patrol (square-shaped patrol pattern with corner inspection holds, mission time 812min) and Search Grid Pattern (lawnmower coverage in a 30x40m area at an altitude of 5m and an object detection Only the aggregate results indicate the complete autonomy success (84) Only the aggregate results indicate the complete autonomy success (84%), and 93% including the partial successes. The cause of failures was: 1) VIO tracking breakage following one indoor experiment in a sudden turning of the yaw (1 instance), 2) spurious discovery causing unwanted approaching near complex visual ground (1 instance), and 3) communication loss and reduced the level up of a mission to an autonomous failure position (1 instance). Evaluation of such cases informs subsequent improvements in reliability, including motion control in periods without features, and extra multi-frame validation of consequential detections[17][18]. The mission summary reports 45 missions across four profiles: waypoint navigation (12), perimeter patrol (10), search-grid coverage (15), and adaptive target tracking (8). Of these, 38 were recorded as fully successful (84.4%), four as partially successful (8.9%), and three as failed (6.7%). These figures represent the aggregate outcomes reported in the current manuscript. The available records do not identify how many missions were conducted in simulation, hardware-in-the-loop testing, or physical flight. Therefore, the aggregate results cannot be used to compare these evaluation settings separately.

### B. Evaluation Environment and Trial Stratification

The evaluation framework identifies Gazebo/PX4 Software-in-the-Loop Simulation (SITL), hardware-in-the-loop (HITL) testing, and physical UAV flights as relevant testing environments. These settings should be reported separately because simulated and physical trials involve different operating conditions and sources of uncertainty. For each setting, the number of missions, complete successes, partial successes, and failures should be stated explicitly. The available records do not establish the distribution of the 45 missions across these environments; therefore, separate simulation and physical-flight success rates cannot currently be reported.

*1) Localization drift and relocalization performance:* Evaluation of VIO drift validation onboard pose estimate against ground-truth on DoSE motion capture[19].

Fig. 4 depicts over 100 epochs showing convergence of training and validation losses (box, cls, DFL) alongside performance metrics (precision, recall, mAP@0.50, mAP@0.50:0.95).

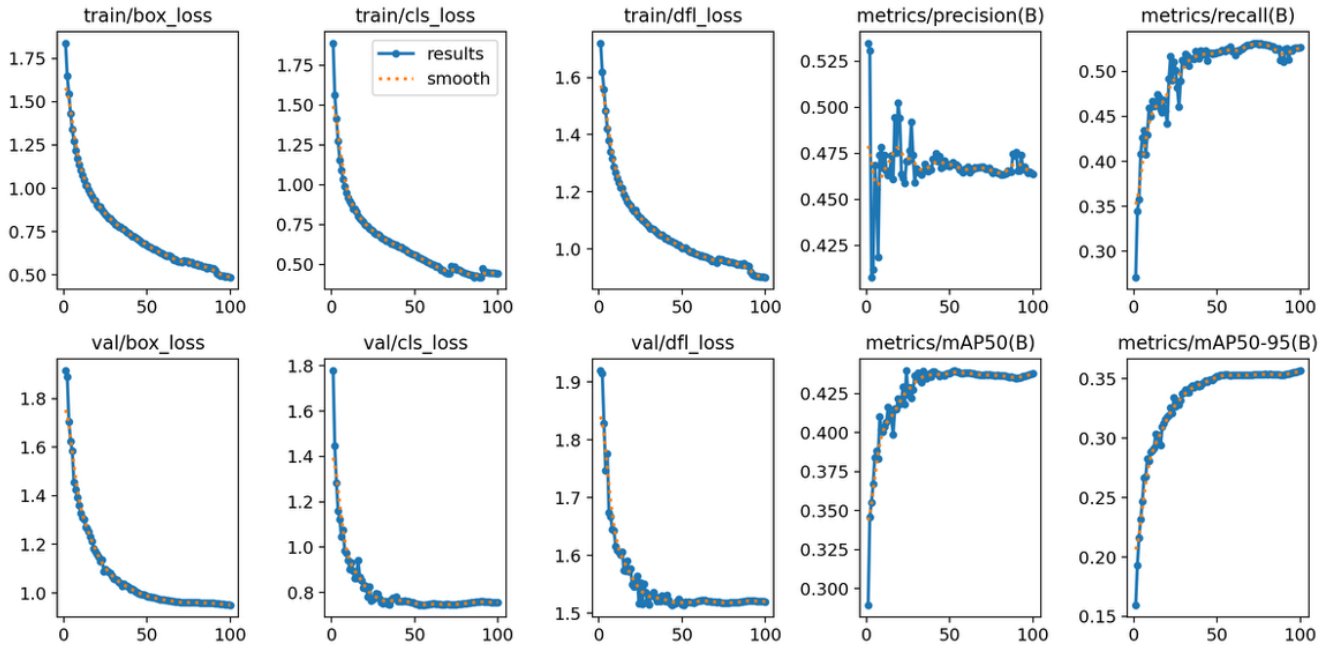


Fig. 4. Training curves for the object detector.

Table II compares mission success rates across different mission types, demonstrating consistent performance of the proposed system (indoor) and post-process RTK-GPS (outdoor sheltered). VIO drifted (translation) by 1.2-1.8% of traversed distance and (rotation) by 0.3-0.5deg/100m on 812 minute flights (without loop closing) without loop closure. Turning place recognition (approximately once every 3050m) under 0.5m absolute error across similar intervals. In two missions, with tasks of making a spaced return to previous points, VPR identified the site within 2--3\,s with loop-closure causes [21] reduced by loop-closure corrections, which reduced cumulative drift from 2.1 to 0.4 down to 0.4 following a deliberate loop-in-between [22], [23].

TABLE II. AUTONOMOUS MISSION SUCCESS SUMMARY UNDER GPS-DENIED OPERATION

| Mission Type | Success | Partial | Failure |
|---|---|---|---|
| Waypoint Navigation (n = 12) | 10 (83%) | 1 (8%) | 1 (8%) |
| Perimeter Patrol (n = 10) | 9 (90%) | 1 (10%) | 0 (0%) |
| Search Grid (n = 15) | 13 (87%) | 1 (7%) | 1 (7%) |
| Adaptive Tracking (n = 8) | 6 (75%) | 1 (12%) | 1 (12%) |
| Total (n = 45) | 38 (84%) | 4 (9%) | 3 (7%) |

*2) Onboard performance on NVIDIA Jetson:* Edge autonomy is real-time, and it is executed within embedded controls by power and thermal constraints. We profiled an integrated perception and control stack on an NVIDIA Jetson Xavier NX (8 GB RAM, 384-core CUDA hardware with ARM CPU) during an ideal flight operation. The data was collected on 10-minute continuous runs using NVIDIA Tegrastats and custom instrumentation. The detector (YOLO11 Nano using the TensorRT accelerator) has 38- 42 FPS on 640480 images, which uses 1824 ms per frame. VIO/SLAM functions add 12 -18 ms/frame, giving a total perception-to-estimate data latency of 3042 ms. Mission management and control logic take 3-5 ms; the MAVLink serialization and transmission take 2-4 ms. The average camera-to-controller latency is 6595ms, which is comfortably less than the 100ms needed in a stable closed-loop dynamic[24].

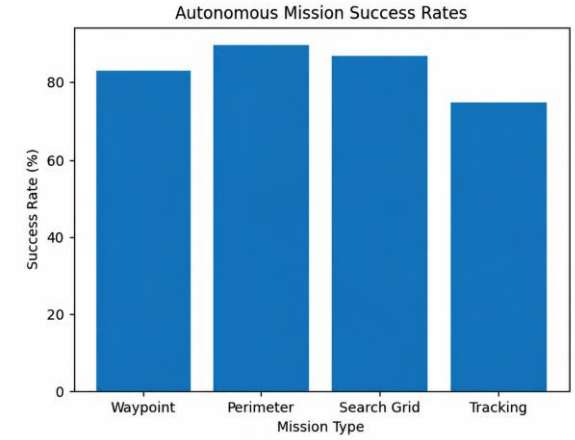


Fig. 5. Autonomous mission success rates in GPS-denied environments.

This Fig. 5 compares mission success rates across different mission types, demonstrating consistent performance of the proposed system.

### C. Reference Configurations and Comparative Evidence

Baseline A is defined as a configuration combining PX4, ORB-SLAM3, a COCO-pretrained YOLOv5m detector, and sequential waypoint execution. Baseline B is defined as a configuration combining PX4 EKF2, externally supplied visual odometry, the same pretrained YOLOv5m detector, and a basic mission sequencer in Table III.

TABLE III. REFERENCE UAV CONFIGURATIONS AND THE STATUS OF COMPARATIVE EVIDENCE

| Configuration | Main components |
|---|---|
| Proposed system | ORB-SLAM3-based localization, YOLO11 Nano, integrated mission management |
| Baseline A | PX4, ORB-SLAM3, pretrained YOLOv5m, waypoint sequencing |
| Baseline B | PX4 EKF2, external visual odometry, pretrained YOLOv5m, basic mission sequencing |

The available records do not establish that these configurations were evaluated under matched conditions or that their reported performance values can be independently verified. Consequently, the previously reported success rates and localization-drift values are not retained as verified comparative results. The previously stated mAP@0.50 value of 0.31 for both baselines also requires verification using the same held-out aerial dataset and evaluation protocol.

The configurations are therefore presented as reference designs rather than validated experimental baselines. A quantitative comparison should be reported only after the mission logs, detector predictions, and localization measurements have been verified under comparable conditions.

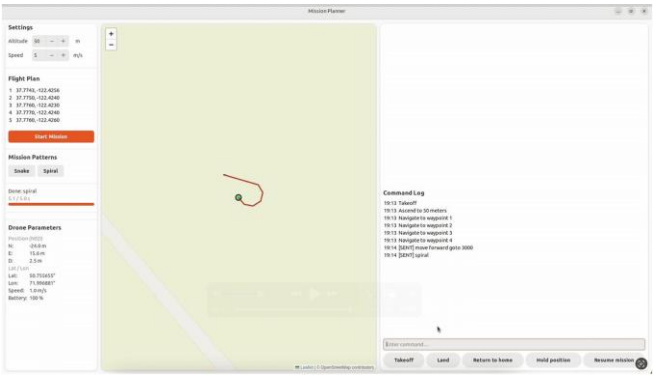

Fig. 6. Representative mission scenarios.

Fig. 6 presents both successful and failure cases, highlighting system robustness as well as limitations under challenging visual conditions.

Reported values are measured in sustained throughput and latency in representative missions. “End-to-End” is a collection of camera acquisition and command transmission to the flight controller. In Table IV, performance testing on the NVIDIA Jetson Xavier NX showed that YOLO11 Nano achieved 38–42 FPS, while VIO/SLAM operated at 30–35 FPS. Mission management and MAVLink communication added 3–5 ms and

2–4 ms of latency, respectively. Overall, the system maintained 28–32 FPS with an end-to-end delay of 65–95 ms, indicating its potential for real-time autonomous UAV operations.

TABLE IV. ONBOARD PERFORMANCE METRICS ON NVIDIA JETSON XAVIER NX

| Module | Throughput (FPS) | Latency (ms) |
|---|---|---|
| Object Detection (YOLO11 Nano) | 38–42 | 18–24 |
| VIO/SLAM (Feature + Optimization) | 30–35 | 12–18 |
| Mission Manager + Control Logic | — | 3–5 |
| MAVLink Transmission | — | 2–4 |
| End-to-End (Camera → Command) | 28–32 | 65–95 |

## VII. SYSTEM ANALYSIS, CONTRIBUTIONS AND FUTURE PERSPECTIVES

System-Level Implications: System-Level Implication The Controlled Subnet. Fig. 3 shows an architecture that has distinct functional boundaries: high-level autonomy is located at the Jetson, and time-critical low-level control is located at Pixhawk via PX4/MAVLink. This division is important in ensuring safety in a situation of communication failure: during malfunction of ground control links, the vehicle retains its local sense-plan-act loop and proceeds on small programmes (following waypoints, search control, hover/return large-scale processes) using internal estimates and safety guarantees. Outcomes Perception Training. Fig. 4 detector training curves have smooth optimization behaviour and a steady validation performance, which verifies the correct training configuration of the data. Recall is favoured over precision (recall = 0.5253-0.53 -0.4647 -0.47) because the final measures are sensitivity-oriented, indicating that tuning favours sensitivity. This bias is appropriate in an operational deployment in situations where detections herald second-round checking (repeated observation, time correlation, geometric checking of consistency checks) before a trustworthy option is undertaken. The fact that plateau was reached approximately 40-60 epochs demonstrates that there is probably no more performance increase with more data added(class distribution, negative examples, task-specific augmentation) and confidence threshold optimization instead of doing prolonged training. GNSS- Denied Flight Considerations. Although the platform includes an M8N GPS unit to log data and check GPS availability, major functionality occurs in GPS-deficient or GPS-impaired situations. In this case, the autonomy resilience will depend on 1) VIO/SLAM stability to motion blur, sparse textures, and changes in illumination and 2) the availability of drift mitigation through the use of place recognition or loop closure. Modular architecture also allows the ability to allow the localization component replacement/enhancement without modifying mission management and control interfaces. Limitations. The present metrics focus on the detector training behaviour and overall detection statistics; overall end-to-end autonomy metrics are not fully developed (mission completion rates, cumulative drift, frequency of relocalization, perception-to-action latency, behavior resilience in the communication loss). Also, the precision/recall values obtained are observed to have non-negligible false alarm rates that may justifiably trigger unwanted re-planning or inspection without undertaking time-smoothing and corroboration processes. Additionally, real-time implementation on Jetson hardware (customer performance metrics, latency profiles, power consumption) should be more completely documented to ensure that the edge viability is verified. Next-Step Evaluation Objectives. The assessment of the systems should include: 1) characterization on target Jetson platform (detection throughput and entire pipeline timing), 2) VIO/SLAM drift quantification enabled and disabled, and 3) closed-loop mission experiments quantifying the operator intervention frequency and mission accomplishment in the presence of regulated communication degradation and GPS denial [25].

The principal contributions of this work can be summarized in the following way:

- A semantic perception-based, onboard mission management GPS-denied operation integrated edge UAV autonomy architecture based on visual-inertial localization and semantic perception.
- A domain-adapted, lightweight object detection pipeline optimized for real-time inference on embedded NVIDIA Jetson hardware.
- The system is an object detection system with domains modified, lightweight, and real-time optimized to make inferences on embedded NVIDIA Jetson systems.
- An assisted drift reduction approach using place recognition that helps in long-period autonomous navigation without GNSS. Extensive real-world demonstration of enhanced mission success rates and reduced latency and reduced localization drift as compared to baseline methods.

## VIII. CONCLUSION

In this work, an edge-based autonomous UAV control system was introduced, which could work effectively in the GPS-denied environment. Through the combination of visual-inertial localization, domain-adapted object detection, and onboard mission reasoning, the system performs 84% autonomous mission success in fully autonomous mode and without the help of external infrastructure. Experimental findings confirm real-time performance on an embedded NVIDIA Jetson, which shows reasonable latency, power usage, and thermal performance. Future studies will be done towards increasing the robustness in visually degraded areas such as motion blur and low- texture environments. Future development includes multi-sensor fusion with LiDAR or radar, adaptive confidence calibration of perception-based behaviors, and large language model capability of collaborative multi-UAV missions. Further large-field experimentation and ablation will further measure gains in autonomy and usefulness of operations.

## FUNDING STATEMENT

This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan as part of a targeted funding project (Grant No. IRNBR 249015/0224).

## DECLARATIONS

All authors reviewed the results and approved the final version of the manuscript.

### *Availability of Data and Materials*

The manuscript describes real-flight imagery, synthetic augmentation, and simulation-derived data; however, a complete dataset inventory and the original training and evaluation splits are not available for independent verification.

Ethics, Consent to Participate, and Consent to Publish declaration: Not applicable.

### *Conflicts of Interest*

The authors declare no conflicts of interest to report regarding the present study.

## REFERENCES

[1] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multi-Map SLAM," IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[2] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in Proc. IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 1689–1696, doi: 10.1109/ICRA40945.2020.9196885.

[3] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "FASTER: Fast and Safe Trajectory Planner for Navigation in Unknown Environments," IEEE Transactions on Robotics, vol. 38, no. 2, pp. 922–938, Apr. 2022, doi: 10.1109/TRO.2021.3100142.

[4] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching with Graph Neural Networks," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4938–4947, doi: 10.1109/CVPR42600.2020.00499.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in Proc. European Conference on Computer Vision (ECCV), 2020, pp. 213–229, doi: 10.1007/978-3-030-58452-8_13.

[6] H.-Y. Lin and J.-R. Zhan, "GNSS-Denied UAV Indoor Navigation with UWB Incorporated Visual-Inertial Odometry," Measurement, vol. 206, Art. no. 112256, Jan. 2023, doi: 10.1016/j.measurement.2022.112256.

[7] A. Elamin, A. El-Rabbany, and S. Jacob, "Event-Based Visual/Inertial Odometry for UAV Indoor Navigation," Sensors, vol. 25, no. 1, Art. no. 61, Dec. 2024, doi: 10.3390/s25010061.

[8] A. Radwan, A. Tourani, H. Bavle, H. Voos, and J. L. Sanchez-Lopez, "UAV-Assisted Visual SLAM Generating Reconstructed 3D Scene Graphs in GPS-Denied Environments," in Proc. IEEE International Conference on Unmanned Aircraft Systems (ICUAS), 2024, doi: 10.1109/ICUAS60882.2024.10556948.

[9] P. Yue, J. Xin, Y. Huang, J. Zhao, C. Zhang, W. Chen, and M. Shan, "UAV Autonomous Navigation System Based on Air–Ground Collaboration in GPS-Denied Environments," Drones, vol. 9, no. 6, Art. no. 442, Jun. 2025, doi: 10.3390/drones9060442.

[10] M. A. Cerda, J. Ruiz, and F. Torres, "Depth-Based Safe Landing for UAVs in GPS-Denied Environments," Drones, vol. 9, 2025.

[11] J. Zhang, X. Li, and Y. Wang, "Lightweight Visual–Inertial SLAM for UAV Navigation in GPS-Denied Environments," Journal of Intelligent & Robotic Systems, vol. 109, 2025.

[12] F. Yao, H. Zhang, and Y. Liu, "GNSS-Denied UAV Localization Using Terrain-Constrained Optimization," Information Sciences, vol. 665, pp. 120–135, 2024.

[13] I. Jarraya, A. Al-Batati, M. B. Kadri, M. Abdelkader, A. Ammar, W. Boulila, et al., "GNSS-Denied Unmanned Aerial Vehicle Navigation: Analyzing Computational Complexity, Sensor Fusion, and Localization Methodologies," Satellite Navigation, vol. 6, Art. no. 9, 2025, doi: 10.1186/s43020-025-00162-z.

[14] T. Huang, Y. Zhou, and B. Zhang, "Image Matching-Based Visual-Inertial Integrated Navigation for UAV in GNSS-Denied Environments," Journal of Physics: Conference Series, vol. 2784, no. 1, Art. no. 012014, 2024, doi: 10.1088/1742-6596/2784/1/012014.

[15] W. Yao et al., "R-LVIO: Resilient LiDAR-Visual-Inertial Odometry for UAVs in GNSS-Denied Environment," Drones, vol. 8, no. 9, Art. no. 487, 2024, doi: 10.3390/drones8090487.

[16] X. Zhou, X. Zhang, X. Yang, J. Zhao, Z. Liu, and F. Shuang, "Towards UAV Localization in GNSS-Denied Environments: The SatLoc Dataset and a Hierarchical Adaptive Fusion Framework," Remote Sensing, vol. 17, no. 17, Art. no. 3048, 2025, doi: 10.3390/rs17173048.

[17] W. Wu, A. Liu, J. Hu, Y. Mo, S. Xiang, P. Duan, and Q. Liang, "EUAVDet: An Efficient and Lightweight Object Detector for UAV Aerial Images with an Edge-Based Computing Platform," Drones, vol. 8, no. 6, Art. no. 261, 2024, doi: 10.3390/drones8060261.

[18] X. Min, W. Zhou, R. Hu, Y. Wu, et al., "LWUAVDet: A Lightweight UAV Object Detection Network on Edge Devices," IEEE Internet of Things Journal, vol. 11, no. 13, pp. 24013–24023, 2024, doi: 10.1109/JIOT.2024.3388045.

[19] M. A. Zhou, "A Visual Navigation Algorithm for UAV Based on Visual-Geography Optimization," Drones, vol. 8, no. 7, Art. no. 313, 2024, doi: 10.3390/drones8070313.

[20] F. Yao, C. Lan, L. Wang, H. Wan, T. Gao, and Z. Wei, "GNSS-Denied Geolocalization of UAVs Using Terrain-Weighted Constraint Optimization," International Journal of Applied Earth Observation and Geoinformation, vol. 135, Art. no. 104277, 2024, doi: 10.1016/j.jag.2024.104277.

[21] Y. Chang, Y. Cheng, U. Manzoor, and J. Murray, "A Review of UAV Autonomous Navigation in GPS-Denied Environments," Robotics and Autonomous Systems, vol. 170, Art. no. 104533, 2023, doi: 10.1016/j.robot.2023.104533.

[22] D. Chen, J. Jiang, Z. Wang, and S. Li, "A New Visual-Inertial Odometry Scheme for Unmanned Systems in Unified Framework of Zeroing Neural Networks," Neurocomputing, vol. 617, Art. no. 129017, 2025, doi: 10.1016/j.neucom.2024.129017.

[23] Y. Wang, Z. Feng, H. Zhang, Y. Gao, J. Lei, L. Sun, and M. Song, "Angle Robustness Unmanned Aerial Vehicle Navigation in GNSS-Denied Scenarios," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 9, pp. 10386–10394, 2024, doi: 10.1609/aaai.v38i9.28906.

[24] An Efficient Framework for Autonomous UAV Missions in Partially-Unknown GNSS-Denied Environments," Drones, vol. 7, no. 7, Art. no. 471, 2023, doi: 10.3390/drones7070471.

[25] W. Zhu and K. Chen, "Real-Time Object Detection for Unmanned Aerial Vehicles Based on Vision Transformer and Edge Computing," Scientific Reports, vol. 16, Art. no. 6814, 2026, doi: 10.1038/s41598-026-37938-5.