

TPD: Black Bee Drones (Brazil), Outdoors Category

Nicole Monteiro Silva¹, Eduardo José de Souza Castro¹, Abner Diniz¹, Enzo Andrade Junqueira Lopes¹, Samuel Assunção Cavallerie¹, Alexandre Carlos Brandão Ramos¹

¹Federal University of Itajubá, MG, Brazil (UNIFEL)

ABSTRACT

We present a practical, from-scratch configuration to execute the first two IMAV 2025 outdoor missions, mapping a disaster area and finding a person, using an accessible, reproducible embedded stack. The platform is centred on a Raspberry Pi 5 as the companion computer and a Pixhawk 2.4.8 as the flight-control [1]–[3]. We detail how to bring up the system end-to-end: installing ROS 2, integrating MAVROS, and establishing a robust MAVLink bridge between the Raspberry Pi and Pixhawk, with a dedicated telemetry channel maintained for supervision and safety [3]–[5]. This layout exposes vehicle state and control as ROS topics/services, allowing the autonomy nodes for mapping and person localisation to run in parallel without operator scripting.

Perception is supported by an OAK-D-class sensor accessed through DepthAI with standard OpenCV bindings, while the person-finding task leverages a single-class YOLOv11 detector; both are interfaced at the host level to simplify integration and logging [6], [7]. The emphasis of the paper is architectural: why these components are selected, how they are connected, and how power, timing, and communications are organised to maximise robustness, repeatability, and development speed.

While our practical implementation concentrates on Missions 1 and 2, we also discuss extensions towards the Transportation Challenge, the Load and Release mission, and the Surveillance Challenge, outlining how the same embedded framework can be adapted to cooperative transport, resource delivery, and endurance-oriented fixed-wing flight. By deliberately focusing on the first two missions while recognising the scalability to the others, the system demonstrates a compact and accessible template that aligns embedded vision and autonomous navigation with the broader objectives of IMAV 2025.

1 INTRODUCTION

The International Micro Air Vehicle (IMAV) 2025 competition adopts Search and Rescue as its central theme, highlighting the role of micro aerial vehicles (MAVs) in rapid situational awareness and victim assistance during disasters. The scenario motivates

autonomy, reliable perception, and safety under realistic constraints.

In the outdoor competition, teams operate in a wide open area and may employ any MAV type to address a set of four missions defined by the organisers. These missions emphasise mapping, localisation, cooperative transport, and delivery tasks characteristic of disaster response, with an additional surveillance challenge encouraging endurance-oriented designs.

Mission 1 – Mapping of a disaster area requires generating a 2D panorama/orthophoto or a 3D model of a GPS-bounded region supplied in advance, with additional credit for autonomous landing and optional real-time visualisation at the ground station. This mission tests coverage planning, data quality, and workflow robustness.

Mission 2 – Find a person provides a textual description and an approximate GPS location; the MAV must autonomously reach the target area, identify the described mannequin, and land within a 2 m radius as objective evidence of success. This mission stresses end-to-end autonomy, from perception to controlled landing near a human proxy.

Mission 3 – The Transportation Challenge requires the collaborative lift and carriage of a 3 kg dummy, simulating the transport of a baby. A minimum of three MAVs must operate in coordination using a suspended net, maintaining formation stability throughout the flight and landing within 2 m of the target GPS location. This mission tests distributed autonomy, synchronisation, and cooperative payload sharing.

Mission 4 – Load and Release involves autonomously dipping a container into a pool at one GPS location, carrying at least two thirds of its capacity, and releasing the water within 3 m of a second GPS waypoint. Teams may repeat the cycle to accumulate points, which demands reliable sequencing of dipping, transit, and release operations while observing safety and stability constraints.

Mission 5 – The Surveillance Challenge runs in parallel and evaluates the ability of a MAV to complete repeated laps over a 500 m by 250 m area for as long as possible, with scoring defined by endurance relative to wingspan.

This mission stimulates efficiency-driven design choices, particularly for fixed-wing configurations, and demonstrates the contribution of long-duration aerial monitoring to disaster response.

Guided by these requirements, our work focuses on an embedded and pragmatic approach that couples dependable onboard perception with flight-control integration. The system is designed to demonstrate consistent mapping outputs, reliable person localisation and landing, cooperative aerial transport, resource delivery through repeated load-and-release cycles, and fixed-wing endurance for persistent surveillance. The remainder of the paper details our system setup, methodology for the missions, expected outcomes, and considerations for future improvements.

2 SYSTEM SETUP

2.1 General Configuration for Autonomous Flight

Our configuration is engineered to meet the specific demands of the two outdoor missions, mapping a disaster area and locating a person, while maximising robustness, repeatability, and development speed. We prioritise the Raspberry Pi 5 as the companion computer because Mission 1 requires sustained image acquisition and buffering, and Mission 2 imposes low-latency perception-and-control loops that benefit from the RPi 5's higher CPU and I/O throughput.

A dedicated Pixhawk 2.4.8 remains the flight-control authority. This division of labour, high-level autonomy and perception on the Raspberry Pi, inner-loop stabilisation and failsafes on the Pixhawk, minimises control latency and isolates mission-software faults from the attitude loop. In practice, this ensures stable coverage tracks during mapping even when the companion is busy writing images, and guarantees controlled descent/precise loitering during person approach while colour/proximity cues are being validated before autonomous landing.

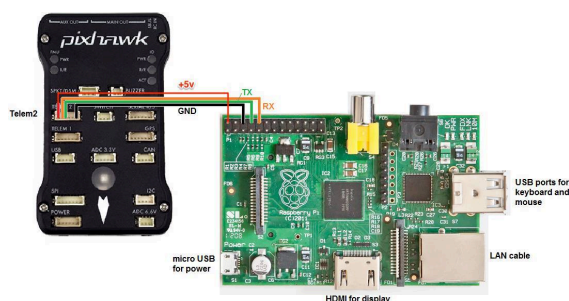


Figure 1: Pixhawk and raspberry connection.

For command and telemetry, we use ROS 2 with MAVROS to bridge MAVLink between the Raspberry Pi and Pixhawk. Vehicle state, parameters and control are exposed as ROS topics, services and actions, allowing the mapping and person-approach nodes to run in parallel without ad-hoc shell interaction. This also simplifies verification: we log synchronised state, images and setpoints for post-flight analysis of coverage (Mission 1) and approach geometry (Mission 2). MAVROS provides a single programmatic interface to change modes, arm/disarm and command land.

In sum, each design choice is driven by mission needs: the RPi 5 provides compute headroom for continuous imaging and real-time perception; the Pixhawk 2.4.8 guarantees deterministic flight control; ROS 2 + MAVROS offers a unified, testable API; and the power/link layout reduces failure modes that would degrade mapping coverage or safe landing near the described person. This foundation supports executing both outdoor missions with technical rigour and operational resilience.

2.2 Vision Hardware

The primary perception sensor is the OAK-D Lite, selected to meet the dual requirements of wide-area image acquisition (Mission 1) and range-aware person localisation (Mission 2). Its integrated stereo pair of monochrome cameras provides disparity-based depth, while a co-located RGB camera supplies colour imagery suitable for mapping and visual detection. This co-registration of colour and depth reduces calibration overhead and enables a single, compact payload to serve both missions with consistent geometry.

The OAK-D Lite connects to the companion computer, ensuring sufficient bandwidth for concurrent colour and depth streams without compromising frame rate or image quality. Factory calibration and hardware synchronisation facilitate the production of rectified stereo images and dense depth maps, which we use to estimate target distance and to gate approach manoeuvres during autonomous landing near the mannequin.



Figure 2: OAK-D Lite depth camera.

We interface with the device through the DepthAI software stack and expose its outputs to our autonomy pipeline via OpenCV. DepthAI is used to define the on-device processing graph, colour frames, rectified mono frames, disparity/depth, and to negotiate resolutions and frame rates appropriate to each task. OpenCV then consumes these streams as standard arrays, enabling uniform pre-processing and downstream integration with our ROS 2 nodes.

For Mission 1 (Mapping), the RGB stream supplies high-quality frames at regular intervals along the waypoint track; frames are timestamped and buffered on the Raspberry Pi for subsequent mosaicking in post-processing. For Mission 2 (Find a person), the RGB frames are fed to the YOLOv11 detector (restricted to the person class), after which OpenCV performs HSV-based colour verification on the detected region. The stereo depth associated with the detection is used to estimate range and to parameterise the descent profile, improving the safety and repeatability of the autonomous landing within the required radius. This sensor configuration therefore unifies mapping and proximity tasks under a single, USB-powered device with a well-supported software interface. The software stack that exposes these sensing capabilities is outlined next.

2.3 DepthAI Platform and API

DepthAI is a spatial-AI platform built around the Intel Movidius Myriad-X VPU and provides a complete hardware–firmware–software stack exposed via depthai-python and depthai-core (C++). The API adopts a pipeline programming model: developers compose a graph of nodes, and messages flow along links between nodes on the device. The host (our Raspberry Pi 5) connects over USB3, deploys the pipeline to the OAK-D, and exchanges frames/metadata through XLink with minimal host overhead.

In this work, we use the pipeline abstraction to standardise acquisition for both missions. For Mission 1, the pipeline delivers timestamped RGB frames for mapping. For Mission 2, it streams RGB and depth; neural inference (YOLOv11) runs on the Raspberry Pi using Ultralytics, while DepthAI provides synchronised colour–depth data to the detection pipeline. Although Figure 3 illustrates an on-device NN node as an example, our configuration keeps NN off-device to prioritise flexibility and ease of integration with ROS 2/MAVROS.

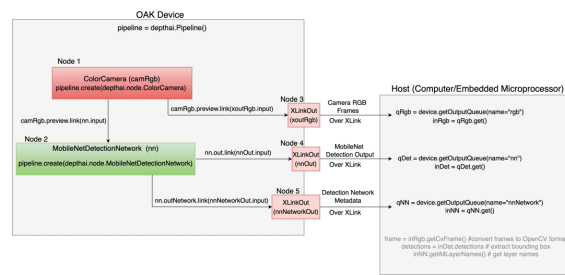


Figure 3: High-level DepthAI pipeline concept: device-side nodes (e.g., ColorCamera, optional NN) linked to the host over XLink. In our setup, RGB and depth are produced on the device, while YOLOv11 inference executes on the Raspberry Pi (host).

3 MISSION 1 - MAPPING OF A DISASTER AREA

3.1 Mapping Workflow

The mapping mission is executed as an autonomous waypoint flight. Before take-off, the set of GPS waypoints provided by the organisers is uploaded to the autopilot. During flight, the Pixhawk follows the route while the companion computer supervises mission state and logging.

Throughout the coverage, the OAK-D Lite acquires RGB frames continuously. Image capture is dynamic, performed while the vehicle traverses the designated area, so that the dataset reflects the actual flight path. No heavy processing is performed onboard during this stage to preserve flight stability and recording continuity.

All acquired images are stored on the Raspberry Pi with minimal pre-processing. Along with the frames, basic metadata is recorded to support downstream reconstruction. The resulting dataset is then available for offline mosaicking/orthophoto generation as described in the post-processing section.

3.2 Post-Processing

After completing the coverage track, the drone autonomously returns to the take-off zone and the imagery recorded on the Raspberry Pi is offloaded to a Windows workstation. The dataset, RGB frames with timestamps and mission status, undergoes offline reconstruction using Microsoft Image Composite Editor (ICE). In ICE, frames are imported and automatically aligned through feature matching; the software then performs global adjustment and selects an appropriate

projection (planar for orthophoto-like maps or spherical/cylindrical for panoramas), followed by seam optimisation, exposure compensation and multi-band blending. The final deliverable is exported either as a 2D orthophoto-style mosaic (planar projection) or as a panoramic mosaic of the surveyed area, depending on the mission's evaluation needs.

The choice for offline post-processing is deliberate. Mapping quality benefits from the workstation's superior CPU/GPU resources and larger memory, enabling more robust alignment, outlier rejection and high-resolution blending than would be feasible in flight. Equally important, removing heavy image stitching from the onboard workload reduces power draw and thermal/compute contention on the companion computer, protecting flight stability and logging continuity. This division of labour, capture in flight, reconstruct on the ground, provides higher-fidelity maps while preserving autonomy and safety during the mission.

4 MISSION 2 - PERSON DETECTION AND RESCUE ASSISTANCE

4.1 Neural Network Setup

We employ YOLOv11 via the Ultralytics library as our real-time detector for the outdoor missions, configuring the network to operate exclusively on the "person" class. This single-class setting simplifies downstream logic, reduces class competition at inference time, and aligns the detector with the competition's success criterion in Mission 2. The choice of YOLOv11 reflects a balance between detection accuracy and latency on embedded platforms, while offering a mature API that integrates cleanly with our ROS 2-based pipeline.

At runtime, frames from the OAK-D RGB stream are pre-processed on the Raspberry Pi 5 (resize and normalisation) and passed to YOLOv11 for inference. In addition, we plan to deploy our trained YOLOv11 model in the optimised .blob format, enabling execution directly on the internal VPU (virtual processing unit) of the OAK-D. This approach takes advantage of the hardware acceleration and neural processing tools provided by the DepthAI framework, with the aim of offloading part of the inference workload from the Raspberry Pi 5 and improving overall latency and power efficiency. Figure 4 illustrates how the RGB stream from the ColorCamera node is forwarded with cropped preview configurations as an `ImgFrame`, which then serves as the input to the NeuralNetwork node.

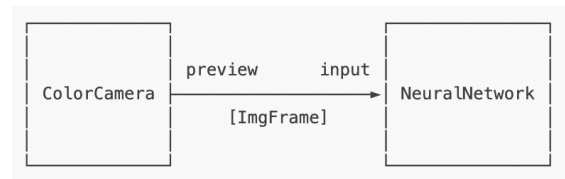


Figure 4: ColorCamera Node Output Linked to the NeuralNetwork Node Input.

Post-processing applies confidence thresholding and non-maximum suppression, and only validated "person" detections (bounding box and confidence) are published to the autonomy stack. The detected region is then forwarded to OpenCV colour verification stage to match the textual description, and the detection centroid is fused with stereo depth to estimate range and regulate the approach/landing controller.

Detection Pipeline

RGB frames are acquired from the OAK-D Lite and ingested on the companion computer through OpenCV, which standardises resolution, colour order, and normalisation. Each pre-processed frame is then forwarded to YOLOv11 (Ultralytics) for single-class inference (person), either on the Raspberry Pi or, when available, directly on the OAK-D VPU using the .blob-compiled model. The detector outputs a set of candidates composed of bounding boxes and associated confidence scores; non-maximum suppression is applied to remove duplicates and retain the most plausible hypotheses.

For every accepted detection, the corresponding bounding box is cropped from the original RGB frame and converted from RGB to the HSV colour space. This representation separates chromatic content from intensity, making the subsequent verification more robust to illumination changes. We then evaluate whether the hue and saturation ranges observed in the cropped region match the textual description provided by the organisers (jacket or helmet colour). If the colour constraint is satisfied, the detection is marked as validated and passed downstream to the autonomy stack; otherwise, it is discarded or flagged for re-evaluation on subsequent frames.

Operationally, the pipeline maintains fixed pre-processing and post-processing parameters across trials to ensure repeatability. Time stamps from acquisition through validation are preserved, allowing later correlation with vehicle state and facilitating analysis of detection latency and stability. Validated detections, location, confidence, and colour-match status are published as ROS 2 messages

for consumption by the approach and landing controller described in the following section.

4.2 Depth and Navigation

Stereo disparity from the OAK-D Lite's monochrome pair is converted to metric depth, yielding per-pixel range aligned with the RGB image. For each validated detection, we compute a robust estimate of the person's distance by taking the median depth inside the YOLOv11 bounding box and smoothing it with a short-horizon filter to suppress outliers. Using the camera intrinsics and the pixel coordinates of the bounding-box centroid, we project the line of sight to the world frame and obtain a lateral bearing and range relative to the vehicle.

Closed-loop visual servoing then regulates the approach. The guidance node publishes body-frame velocity (or position) setpoints via MAVROS so that the Pixhawk commands small, continuous corrections to centre the bounding box while reducing range. Lateral commands are proportional to the centroid error, with saturation to prevent aggressive manoeuvres; forward velocity is ramped down as the estimated range decreases, and vertical motion is inhibited until lateral error and yaw misalignment remain below thresholds for a fixed number of frames. If detection confidence or depth quality drops below limits, the controller enters a hold/loiter state and attempts re-acquisition before resuming the approach.

When the detection remains stable, the bounding box exceeds a minimum apparent size, and the filtered range falls below a preset approach radius, the system initiates a controlled descent. Descent rate is limited and coupled to continuous re-centering, yielding a shallow, spiral-like final approach that prioritises stability. Touchdown is commanded once the range and optical flow indicate ground contact within the vicinity of the target. The mission declares success only if the autonomous landing occurs within 2 m of the mannequin, as required by the competition rules; otherwise, the vehicle aborts to a safe hover or return-to-launch and the pipeline restarts the search-approach cycle.

5 MISSION 3 - TRANSPORTATION CHALLENGE

5.1 Communication and Network Setup

To address the cooperative transport requirement, our solution deploys four homogeneous multi-rotor MAVs, each equipped with an onboard Raspberry Pi computer. One of the Raspberry Pis is designated as the Leader Node, responsible for establishing and maintaining a local Wi-Fi network. The remaining three Raspberry Pis act as Followers, connecting to the Leader over this ad hoc wireless network.

The system leverages ROS 2 middleware for distributed communication. The Leader runs publisher nodes that broadcast high-level commands (e.g., take-off, setpoint

updates, synchronisation triggers), while simultaneously running subscriber nodes to monitor the state feedback from all Followers. The Follower MAVs subscribe to the Leader's command topics, executing control actions in parallel and reporting status updates. This architecture ensures time-aligned manoeuvres across all four MAVs while preserving decentralised awareness of vehicle states.

5.2 Cooperative Transport Pipeline

Each MAV is mechanically identical, with balanced thrust capacity to share the load of the 3 kg dummy suspended in a cooperative net. Prior to take-off, the dummy is manually positioned inside the net, ensuring that all motors remain off during loading as per the rules. The suspension net, with a radius under 1 m, is rigidly shared among the four MAVs through equal tether points, ensuring symmetric tension distribution.

The operational pipeline follows four stages:

1. Take-off synchronisation: The Leader publishes a synchronised take-off command. All MAVs ascend to a predefined hover altitude, maintaining formation using local position setpoints.
2. Translational transport: The Leader streams waypoint commands toward the target GPS coordinate. Followers mirror these commands with negligible delay, yielding a cooperative trajectory. Lateral offset constraints keep each MAV in its designated quadrant relative to the net, maintaining load stability.
3. Monitoring and correction: Each MAV publishes IMU and tether force estimates, which are aggregated at the Leader for consistency checks. If deviations in thrust or formation exceed thresholds, corrective lateral setpoints are issued.
4. Approach and landing: As the formation nears the drop zone, descent velocity is reduced, and the MAVs collectively lower the payload within the 2 m target radius. Autonomous landing commands are synchronised, ensuring that the net remains taut until the dummy reaches the ground.

5.3 Navigation and Control

Closed-loop formation control is implemented at two layers. At the inner layer, each MAV tracks its individual position and velocity setpoints through PX4 autopilot stabilisation. At the outer layer, the Leader node computes formation geometry constraints based on GPS and onboard odometry, publishing corrections to minimise net deformation.

Formation spacing is regulated by proportional control over the centroid error between the MAV cluster and the

payload projection. Lateral corrections are saturated to avoid aggressive manoeuvres, and vertical descent is only allowed when inter-drone spacing and tether tensions remain within safe limits for multiple consecutive frames.

If communication with the Leader is interrupted, Followers enter a hold pattern, hovering at the last received setpoint until reconnection or until a safety timeout triggers return-to-launch. This guarantees resilience to network dropouts while prioritising payload safety.

Success for this mission is declared if the dummy is delivered autonomously within 2 m of the designated GPS coordinate and remains secured in the net during the entire transport process, satisfying IMAV 2025 rules.

6 MISSION 4 – LOAD AND RELEASE

The solution for this mission employs a single multi-rotor MAV equipped with a suspended transparent container of 0.5 litres, which is the minimum capacity required by the competition rules. The container is attached by a tether of at least two metres and is filled with blue-dyed water to facilitate visibility for the judges. Low-level flight control and GPS-based state estimation are provided by the Pixhawk autopilot, while a Raspberry Pi is used as the companion computer. Through the mavros2 bridge, which links ROS 2 nodes to the Pixhawk via MAVLink, the Raspberry Pi manages the higher-level mission logic, including waypoint navigation, dipping manoeuvres, and release actuation.

The mission is executed in a fully autonomous fashion. After take-off, the drone flies to the pool GPS coordinates communicated beforehand. At this point, the guidance node regulates a stable hover and executes a controlled descent until the suspended container touches the water. A dwell period of several seconds ensures that at least two thirds of the container volume are filled. Once loading is complete, the drone climbs to a safe altitude and proceeds to the release zone, which is defined by a second GPS coordinate at a distance of no less than 150 metres from the pool. At the target location, the vehicle enters a low-speed loiter pattern, correcting for wind and maintaining stable heading. A servo mechanism then actuates the release of the water, which must fall within a three metre radius of the waypoint to be considered valid.

Immediately after the release, the MAV reorients towards the pool and repeats the sequence. According to the rules, the same drone is allowed to perform several loading and releasing cycles without landing, and in our design the aircraft is programmed to carry out up to three such repetitions in order to maximise scoring. The 0.5 litre container is reused for all cycles, and between releases the MAV executes only a short transit back to the pool and subsequent return to the drop zone. If the mission must be terminated early, a manual landing is permitted either within the flight area or at the original take-off zone.

The navigation and control strategy relies on the Pixhawk for inner-loop stabilisation of position and velocity setpoints, while the Raspberry Pi handles mission-level decisions and trajectory sequencing. Horizontal speeds are restricted during dipping phases to avoid pendulum effects in the suspended payload, and vertical motion is limited by immersion and clearance checks to prevent excessive tether swing. The final release phase is performed in a stabilised loiter with corrections for environmental disturbances, ensuring that the water is delivered within the required radius. In case of degraded GPS accuracy or sensor inconsistencies, the vehicle enters a hold state and retries the manoeuvre, or returns to launch if conditions persist.

Compliance with the IMAV 2025 rules is fully observed. The pool and release positions are pre-defined GPS coordinates. The container meets the minimum requirements of 0.5 litres, is transparent for visual inspection, and is attached via a tether of more than two metres. Water loading and release are completely autonomous, and the distance between pool and release is greater than 150 metres. The scoring scheme grants 0.25 times the container capacity for successful loading and the same for successful release, resulting in 0.25 points per half-litre for each step. By performing up to three cycles, the proposed solution ensures both compliance and efficiency in the Load and Release mission.

7 MISSION 5 – SURVEILLANCE CHALLENGE

For the surveillance challenge, we propose the use of a fixed-wing platform based on the Ugly Stick kit, with a wingspan of 110 cm, fuselage length of 72 cm, and an all-up weight of approximately 600 g including the 3S 2200 mAh battery. The aircraft is powered by an Emax 2822 motor rated at 1200 kv, paired with a 9x5 propeller, a 30 A ESC, and four 9 g servos. This lightweight configuration, combined with the presence of a dihedral wing for passive stability and glass-fibre reinforcement rods already integrated into the structure, yields a stable, efficient, and easily controllable airframe suitable for long-duration flight.

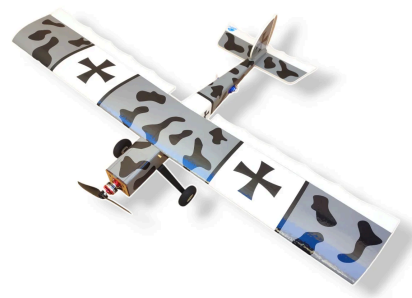


Figure 5: Ugly Stick aeromodel.

The mission plan is programmed directly through the ground station software, where waypoints are uploaded to define the 500 m by 250 m rectangular area prescribed by the competition rules. After manual take-off, the Ugly Stick transitions into autonomous flight and executes continuous laps around the rectangle at altitudes up to 100 m AGL, with vertical variation limited to 25 m as required. The autopilot follows the rectangular pattern with shallow turns designed to minimise induced drag and energy expenditure, while telemetry to the ground station provides real-time monitoring of battery status, position, and lap count. Landing is performed manually, as permitted by the challenge regulations.

With its compact wingspan of only 1.10 m, the scoring metric of endurance divided by wingspan favours this configuration, since each minute of flight time yields a higher ratio than in larger fixed-wing aircraft. Based on the propulsion system and battery capacity, we expect a flight endurance in the range of 15 to 20 minutes under cruise conditions optimised for best lift-to-drag ratio. This translates into an estimated score between 13.6 and 18.1, which is competitive within the context of the mission.

The expected outcome is a stable and efficient surveillance flight, with smooth tracking of the designated area and multiple completed laps before battery depletion. The Ugly Stick platform demonstrates that even a low-cost, small-scale fixed-wing aircraft can be adapted to deliver competitive endurance-to-span performance, aligning with the IMAV 2025 objective of encouraging innovative and pragmatic designs for extended aerial surveillance in disaster-response scenarios.

8 EXPECTED RESULTS

For Mission 1 (Mapping), we expect to obtain a consistent 2D orthophoto-style mosaic (or panoramic mosaic) of the designated area using offline reconstruction. By decoupling capture from stitching, the flight can prioritise stable coverage and complete image logging, while the workstation performs robust alignment and blending afterward. The anticipated outcome is a visually coherent map that reflects the flown track with minimal seams and exposure discontinuities, suitable for qualitative situational awareness and subsequent analysis.

For Mission 2 (Person Detection), we expect real-time identification of the target mannequin using a single-class YOLOv11 detector, followed by HSV-based colour confirmation. Coupled with stereo depth, the system should regulate approach by continuously re-centring the bounding box and reducing range until the vehicle can initiate a controlled descent. The intended result is an autonomous landing in the vicinity of the target, meeting

the competition requirement of proximity within two metres.

For Mission 3 (Transportation Challenge), we expect coordinated lift and transport of the 3 kg dummy using a four-MAV formation controlled through ROS 2 communication. The leader-follower structure, established by a Raspberry Pi acting as the network master, will ensure synchronised take-off, translation, and landing, while the shared net distributes the payload evenly among the vehicles. By maintaining formation geometry and tension symmetry during flight, the system should achieve a smooth delivery of the dummy within two metres of the target coordinate, demonstrating the viability of distributed onboard computing for cooperative aerial transport.

For Mission 4 (Load and Release), we expect the single MAV equipped with a 0.5 L suspended container to autonomously perform multiple loading and release cycles between the designated pool and drop zone. Through Pixhawk-Raspberry Pi integration and mission sequencing in ROS 2, the drone will immerse the container for sufficient dwell to guarantee filling, transit to the target waypoint, and actuate the servo release within the prescribed three-metre radius. By repeating this cycle up to three times without intermediate landing, the system should accumulate valid points and illustrate the reliability of combining low-level autopilot control with high-level task logic for repetitive autonomous operations.

For the surveillance challenge, we expect the 110 cm Ugly Stick airframe to complete multiple laps within the 500 m by 250 m designated rectangle, sustaining a stable and efficient flight for 15 to 20 minutes on a 3S 2200 mAh battery. The dihedral wing geometry and light weight of 600 g contribute to natural stability, reducing the control effort and conserving energy during extended operations. With a wingspan of only 1.10 m, the scoring rule based on endurance divided by span is particularly favourable, yielding an expected score between 13.6 and 18.1 depending on the actual endurance achieved. The anticipated result is a robust demonstration that a compact, low-cost fixed-wing aircraft can be adapted into a competitive endurance platform, capable of providing continuous aerial surveillance under the IMAV 2025 requirements.

Overall, we expect the platform to demonstrate the feasibility of a low-cost, embedded solution that combines onboard perception, depth sensing, and flight-control integration. The chosen architecture (Raspberry Pi + Pixhawk + OAK-D, bridged by ROS 2/MAVROS) is designed to balance computational load,

operational safety, and development agility, supporting reliable mapping and person localisation under the constraints of the outdoor missions.

9 CONCLUSION

This paper presented an end-to-end pipeline tailored to the first two outdoor missions of IMAV 2025: mapping a disaster area and finding a person. The system couples onboard perception and autonomy, image capture, detection, colour verification, depth-aware ranging, and setpoint generation, with reliable flight control, yielding a cohesive loop from sensing to action. The mapping workflow prioritises stable data acquisition in flight and defers stitching to ground-based post-processing, while the person-finding workflow closes the perception-control loop to enable a controlled landing within the required proximity.

Our design deliberately combines accessible hardware (Raspberry Pi, Pixhawk, and OAK-D Lite) with recent computer-vision techniques. This division of responsibilities leverages the strengths of each component: deterministic low-level control on the autopilot, flexible high-level logic on the companion computer, and co-registered colour-depth sensing for both missions. The result is a pragmatic architecture that balances computational load, operational safety, and reproducibility.

Beyond the first two tasks, the approach extends to cooperative transport and load-and-release operations. The leader-follower framework developed in ROS 2 enables multiple MAVs to share payload responsibility while maintaining formation stability, demonstrating the feasibility of distributed aerial transport. Similarly, the integration of mission sequencing with autonomous dipping and release illustrates the ability to execute repetitive, resource-oriented operations. Together, these outcomes highlight the adaptability of the platform to a wide spectrum of rescue-inspired challenges, from situational awareness to physical intervention.

Finally, the addition of the surveillance challenge showcases the use of a compact fixed-wing platform optimised for endurance-to-span performance. By selecting a 1.10 m Ugly Stick airframe, the system exploits the scoring metric to deliver competitive endurance values within a reduced wingspan, while remaining stable, lightweight, and reliable. This final mission demonstrates that low-cost, small-scale fixed wings can complement multirotor systems, extending the operational envelope of micro air vehicles.

In summary, the proposed solutions across all missions illustrate how embedded platforms, when carefully integrated across perception, communication, and control layers, can yield robust and versatile autonomy. The lessons drawn from these implementations provide not only a valid competition entry but also a step towards practical micro-aerial solutions for real disaster-response scenarios.

ACKNOWLEDGEMENTS

We would like to thank the VisCap Laboratory (UNIFEI) for their technical support and infrastructure, as well as the IMAV 2025 organising committee for encouraging innovation in MAV-based disaster response research.

REFERENCES

- [1] J. Martinez-Carranza (Ed.), "IMAV 2025 Competition Rules for Indoor and Outdoor Competitions, and Special Challenges," 16th International Micro Air Vehicle Conference and Competition, Puebla, Mexico, 2025.
- [2] Raspberry Pi Ltd., "Raspberry Pi 5 — Product Documentation," Online: <https://www.raspberrypi.com/documentation/>
- [3] PX4 Autopilot Dev Team, "PX4 User Guide," Online: <https://docs.px4.io/>
- [4] V. Ermakov et al., "MAVROS: MAVLink extendable communication node for ROS," Online: <https://wiki.ros.org/mavros>
- [5] Luxonis, "DepthAI Documentation," Online: <https://docs.luxonis.com/>
- [6] Ultralytics, "Ultralytics YOLO (YOLOv8/YOLOv11) Documentation," Online: <https://docs.ultralytics.com/>
- [7] Open Robotics, "ROS 2 Documentation (Jazzy)," 2024. [Online]. Available: <https://docs.ros.org/en/jazzy/>
- [8] MAVROS Developers, "MAVROS package for ROS 2," 2024. [Online]. Available: <https://docs.ros.org/en/iron/p/mavros/>
- [9] Mavlink Development Team, "MAVROS GitHub Repository," GitHub, 2024. [Online]. Available: <https://github.com/mavlink/mavros>
- [10] Mavlink Team, "MAVLink Developer Guide," 2024. [Online]. Available: <https://mavlink.io/en/>

[11] E. Renard, *ROS 2 From Scratch*. Barnes & Noble, 2024. [Online]. Available: <https://www.barnesandnoble.com/w/ros-2-from-scratch-edouard-renard/1146494960>

[12] SIG Mfg. Co., Inc., “Seagull Models Classic Ugly Stick – Specifications,” 2025. [Online]. Available: <https://sigmfg.com/products/seagull-models-classic-ugly-stick> (Wingspan: 180 cm; Flying weight: 3.5 kg; Fuselage length: 146.8 cm)

[13] RC-builder, “Ugly Stick – Specifications (Regular CNC short kit),” 2025. [Online]. Available: <https://www.rc-builder.com/products/ugly-stick> (Wingspan: 1670 mm; Wing area: 50 dm²; Flight weight from 1400 g)

[14] Gator-RC, “Classic Ugly Stick Master Scale Kit – Specifications,” 2025. [Online]. Available: <https://gator-rc.com/classic-ugly-stick-master-scale-kit-seagull-model> (Wingspan: 180 cm; Weight: 3.5 kg)