Software in the Loop Perception and Engagement Validation for the EBABIL Autonomous Multi-UAV System

Department of Software Engineering,

Ankara Science University,

Ankara, Türkiye

Yakup Tekeli

Abstract

This paper presents the simulation-based development of EBABIL, a fully autonomous multi-UAV swarm system for target detection, threat evaluation, target assignment, and coordinated engagement. The system is designed to make mission decisions without real-time operator intervention. Isaac Sim was used to generate labeled synthetic images for four classes: tank, air-defense system, military personnel, and military vehicle. Eighteen controlled YOLO11 training runs were completed. The tests studied dataset composition, image augmentation, input resolution, model size, and limited real-data mixing. The strongest Isaac dataset configurations reached 0.995 mAP50, up to 0.9764 mAP50–95, approximately 0.998 precision, and 0.9987 recall in the recorded TensorBoard validation results. The trained detector was connected to a Gazebo software-in-the-loop engagement scenario with QGroundControl and MAVLink integration. A Cesium-based interface displayed the swarm state and V formation. QGroundControl and Cesium were used only for telemetry monitoring and visualization. The results show that simulation can support perception training, autonomous decision logic, and engagement tests before controlled hardware trials.

Keywords—autonomous UAV, Gazebo, Isaac Sim, object detection, QGroundControl, sim-to-real, software-in-the-loop, swarm intelligence, synthetic data, YOLO11.

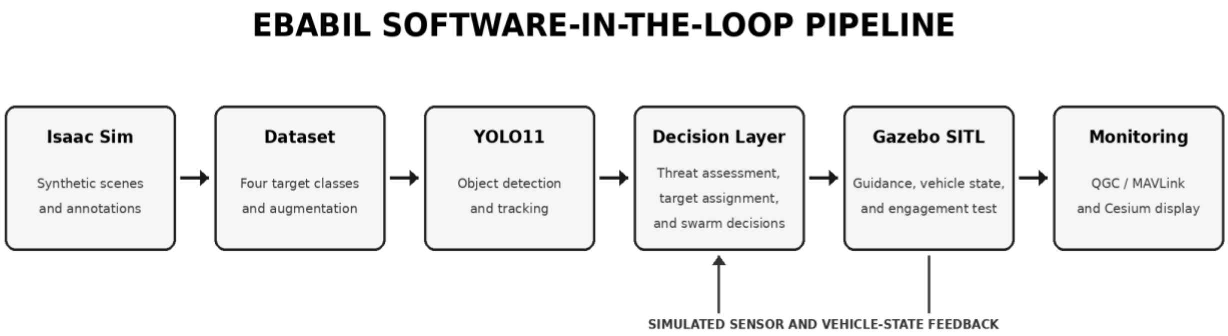

Fig. 1. Integrated software-in-the-loop perception, autonomous decision, engagement, and monitoring workflow used in the EBABIL study.

I. INTRODUCTION

Multi-UAV systems can provide wide-area observation and fast task execution. However, their development requires safe and repeatable tests. Real flight tests are costly and may not cover enough environmental variation. Simulation provides a controlled method for testing perception, guidance, communication, and autonomous decision making before field deployment.

Gazebo is widely used for robot and vehicle simulation because it provides physics, sensor models, plugins, and programmatic interfaces [1]. Isaac Sim provides photorealistic rendering and synthetic data tools for perception models [2]. Replicator can generate camera images and automatic annotations from semantically labeled scene objects [3]. These features make the two simulators useful for different parts of the same development pipeline.

The main goal of EBABIL is to support autonomous and coordinated UAV missions in dynamic and GPS-denied environments. The project combines deep-learning-based perception, threat assessment, target assignment, swarm coordination, and autonomous mission decisions. Ground-side interfaces are used for monitoring rather than command generation. This report focuses on two completed work packages. The first work package covers synthetic data generation and object detection. The second covers Gazebo software-in-the-loop engagement tests and swarm visualization.

The main contributions of this study are as follows. First, a labeled military-object dataset was generated in Isaac Sim. Second, eighteen controlled training experiments were completed with a fixed optimization recipe. Third, the training and validation behavior of the detector was compared through mAP50, mAP50–95, precision, and recall curves. Finally, the detector was connected to a Gazebo software-in-the-loop engagement test with QGroundControl, MAVLink, and a Cesium-based monitoring display.

II. EBABIL SYSTEM OVERVIEW

A. Autonomous Operational Concept

EBABIL is designed as a fully autonomous mission architecture based on swarm-level decision making. A human operator is not part of the real-time decision loop. The software processes sensor inputs, evaluates threats, prioritizes targets, assigns UAVs, and manages engagement actions. The mission flow contains three main stages. The first stage is intelligence, surveillance, and reconnaissance. The second stage covers threat evaluation, target prioritization, and autonomous UAV assignment. The third stage covers coordinated target engagement.

The perception layer detects and classifies objects of interest. The decision layer evaluates the detected objects and creates target assignments. The engagement layer guides each selected UAV toward its assigned target. Swarm intelligence supports dynamic task sharing and coordinated movement. Ground-side software receives telemetry and displays system decisions. It does not select targets, assign UAVs, or generate engagement commands in the evaluated concept.

B. Simulation Architecture

The project uses separate tools for separate tasks. Isaac Sim is used for image generation and annotation. Gazebo is used as a software-in-the-loop environment for vehicle motion, camera feedback, guidance, and engagement tests. QGroundControl provides vehicle setup and mission-state monitoring through MAVLink [5], [6]. The Cesium interface provides a map-based view of the complete swarm.

This separation reduces development cost. Perception models can be trained without running a full vehicle simulation. Guidance and engagement logic can then be tested in Gazebo with a selected model. Simulated sensor data and vehicle state are returned to the autonomous software during the test. The same telemetry is sent one way to the monitoring interfaces. Fig. 1 shows the complete workflow used in this report.

III. SYNTHETIC DATA GENERATION IN ISAAC SIM

A. Scene Design and Automatic Annotation

Two Isaac Sim datasets were used. The first dataset contained 7,768 raw synthetic images. The second contained 7,188 images and added desert and snow ground conditions. Four object classes were defined: tank, air-defense system, military personnel, and military vehicle. Different camera positions, target positions, target scales, surfaces, and lighting conditions were used to increase scene diversity.

Scene objects were assigned semantic class labels. Isaac Sim Replicator used these labels to generate aligned RGB images and object annotations. This removed the need for manual bounding-box labeling. Automatic labels also reduced annotation inconsistency. Fig. 2 shows one synthetic military-vehicle scene from the generated environment.

Fig. 2. Example military-vehicle scene generated in Isaac Sim for synthetic data collection.

The scene design also included hard examples. These examples changed object size, background, contrast, and visibility. The goal was not to build one visually perfect environment. The goal was to expose the model to enough appearance variation. This follows the main idea of domain randomization, where simulation diversity is used to reduce the gap between rendered and real images [4].

B. Training and Augmentation Strategy

A total of eighteen training runs were completed. The study was designed as a controlled ablation study, not as a broad hyperparameter search. The optimizer, learning rate, training duration, warm-up period, and random seed remained fixed. One main variable was changed in each phase. The changed variables were dataset composition, augmentation policy, input resolution, model size, and real-data mixing.

TABLE I

FIXED TRAINING CONFIGURATION

Parameter	Value
Base model	YOLO11s; YOLO11m in selected runs
Initial weights	COCO-pretrained weights
Optimizer	AdamW
Learning rate	0.001
Weight decay	0.0005
Schedule	Cosine learning-rate decay
Warm-up	3 epochs
Training length	40 epochs
Random seed	42
Resolution	640 px; 1280 px ablation
Hardware	NVIDIA RTX 3060 Laptop GPU, 6 GB VRAM

Three augmentation policies were compared. The default policy used the standard Ultralytics transformations. The sim2real_v1 policy added motion blur, defocus blur, Gaussian blur, sensor noise, JPEG compression, downscaling, brightness and contrast changes, color jitter, and fog. The sim2real_v2 policy added grayscale conversion and CLAHE. These operations were applied during data loading. They did not change the stored dataset files.

IV. OBJECT DETECTION MODEL AND RESULTS

A. Training Behavior

YOLO11s was used as the baseline detector because it provides a practical balance between accuracy and computing cost [7]. YOLO11m was tested in the final phase to study the effect of model capacity. Most runs converged rapidly during the first ten to twenty epochs. The best Isaac dataset configurations reached about 0.995 mAP50, about 0.97 mAP50–95, and about 0.999 precision and recall on the development validation curves.

The validation curves show that the strongest configurations reached a stable performance level within the forty-epoch training budget. Several runs converged during the first ten to twenty epochs. The high-performing runs used the new Isaac dataset alone or together with the old dataset and the sim2real augmentation policies. The recorded results are reported as final TensorBoard values, not as class-based real-image AP50 scores.

B. Training and Validation Results

The selected new-dataset runs reached 0.995 mAP50. Their final mAP50–95 values ranged from 0.9684 to 0.9764 in the representative configurations shown in Table II. Precision remained between 0.9976 and 0.9987, while recall reached 0.9987 for all listed runs. These values indicate stable convergence in the prepared validation setting.

Among the listed configurations, N0 produced the highest final mAP50–95 value of 0.9764. N1 and N3 followed with 0.9745 and 0.9724. The combined old-and-new synthetic configuration N2 reached 0.9712. Runs that included all data sources also preserved 0.995 mAP50 and precision above 0.997.

The results support the use of controlled data and augmentation comparisons. They also show that high validation scores must be separated from field performance claims. In this study, operational behavior is examined through the Gazebo software-in-the-loop engagement test, while real-world flight validation remains future work.

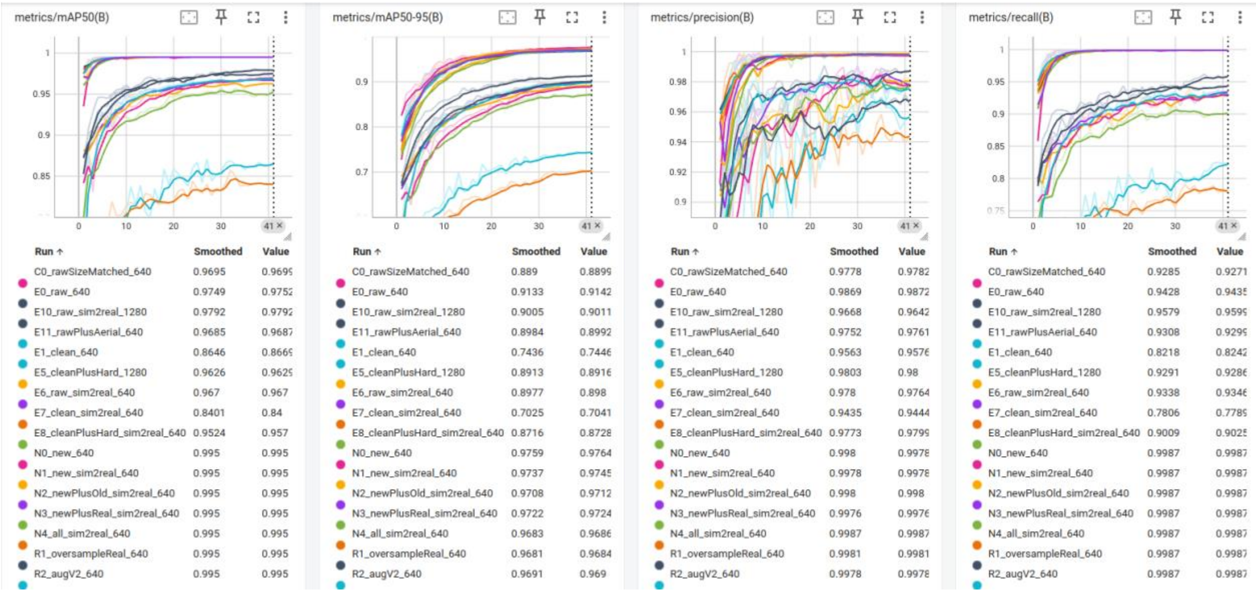

Fig. 3. TensorBoard curves for mAP50, mAP50–95, precision, and recall across the controlled training runs.

TABLE II
REPRESENTATIVE FINAL TENSORBOARD VALIDATION METRICS

Run	Training data / augmentation	mAP50	mAP50–95	Precision	Recall
N0	New Isaac / default	0.9950	0.9764	0.9978	0.9987
N1	New Isaac / v1	0.9950	0.9745	0.9978	0.9987
N2	Old + new synthetic / v1	0.9950	0.9712	0.9980	0.9987
N3	New + real / v1	0.9950	0.9724	0.9976	0.9987
N4	Old + new + real / v1	0.9950	0.9686	0.9987	0.9987
R1	All data + real ×5 / v1	0.9950	0.9684	0.9981	0.9987
R2	Old + new + real / v2	0.9950	0.9690	0.9978	0.9987

V. GAZEBO SOFTWARE IN THE LOOP ENGAGEMENT TESTS

A. Test Setup

Gazebo was used to test the engagement workflow in a controlled outdoor environment. A tank model was placed in the scene. The UAV camera stream was processed by the trained detector. The detected bounding box and its center were used as the visual reference for the approach. The vehicle and mission state were observed through QGroundControl. MAVLink provided the communication layer between the simulated vehicle and the ground-side software. The guidance and engagement decisions remained inside the software pipeline.

The test focused on perception-guided approach rather than weapon effects. The system had to keep the target inside the camera view, estimate the image-space error, and reduce this error during the approach. The video recording also displayed a target marker, a relative direction vector, and a distance indicator. Fig. 4 shows a frame from the engagement sequence.

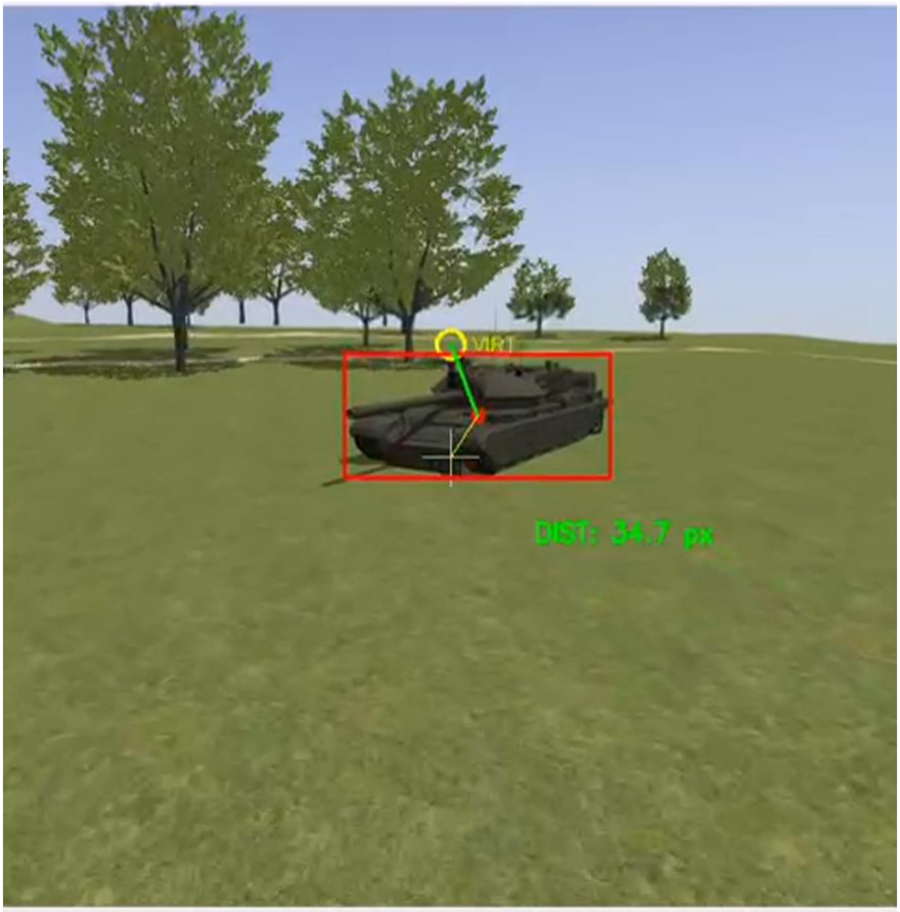

Fig. 4. Gazebo engagement test. The tank is detected and used as the visual guidance target during the approach.

B. Qualitative Result

The recorded test shows that the detector identified the tank during the approach. The target remained inside the guidance display while the camera moved closer. This result confirmed that the Isaac-trained perception model could be connected to the Gazebo software-in-the-loop engagement pipeline. It also confirmed basic data exchange with the ground monitoring layer.

The available test material did not include a complete numeric engagement log. Therefore, this report does not claim a measured hit rate, mean engagement time, or guidance error. The current result is a functional and qualitative integration result. Future tests should record target-lock duration, pixel error, command latency, miss distance, and mission success rate.

VI. GROUND MONITORING INTERFACE AND SWARM DISPLAY

A Cesium-based ground monitoring interface was used to display the swarm on a geographic map. Each UAV was represented by a labeled marker. The interface made it possible to observe the position of each vehicle and the shape of the formation. It did not participate in target selection, UAV assignment, path planning, or engagement decisions. Fig. 5 shows eleven UAVs in a V formation.

Fig. 5. Cesium-based ground monitoring display showing the eleven-UAV swarm in a V formation.

The formation view supports system monitoring and mission review. It can also be extended with target markers, system-generated assignment lines, vehicle health, communication state, route history, and engagement status. In a complete EBABIL mission, this interface will present perception results, autonomous target assignments, and vehicle telemetry on one screen.

VII. DISCUSSION

The study shows the value of using more than one simulator. Isaac Sim was effective for data generation and automatic labeling. Gazebo was effective for software-in-the-loop vehicle integration and engagement testing. QGroundControl and Cesium provided two monitoring views. Neither interface was part of the autonomous decision loop. This modular design reduced the need to force all project functions into a single tool.

The perception experiments produced high and stable TensorBoard validation metrics for the strongest Isaac dataset configurations. However, these metrics do not by themselves prove field performance. Viewpoint, texture, lighting, sensor noise, and target scale can still change after deployment. More controlled real-image and hardware tests are therefore required before the detector can be evaluated outside simulation.

The next Isaac dataset should include more aerial camera angles, smaller targets, realistic sky backgrounds, stronger texture variation, and more balanced class frequencies. The engagement tests should also move from qualitative video review to automated metrics. Repeated scenarios with different target positions, lighting conditions, vehicle speeds, and communication delays will provide stronger evidence.

The current tests remain simulation tests and do not prove field performance. The present stage is software-in-the-loop. However, it provides a useful integration baseline for the autonomous perception and engagement pipeline. The same architecture can later support hardware-in-the-loop and controlled flight testing.

VIII. CONCLUSION

This paper presented the synthetic-data and engagement-test stages of the EBABIL project. Isaac Sim was used to generate labeled images for four military-object classes. Eighteen YOLO11 training experiments were completed with controlled changes. The strongest recorded configurations reached 0.995 mAP50, up to 0.9764 mAP50–95, and approximately 0.998 precision and recall in TensorBoard validation.

The detector was connected to a Gazebo software-in-the-loop engagement scenario with QGroundControl and MAVLink. The system detected a tank and supported a perception-guided approach. A Cesium interface displayed the eleven-UAV swarm in V formation. Together, these results form an initial end-to-end simulation pipeline for perception, autonomous decision making, engagement, and ground-side monitoring.

Future work will add balanced real images, improved synthetic scenes, repeated engagement measurements, multi-target assignment, and stronger swarm-intelligence methods. These steps will make the evaluation more complete and prepare the system for controlled hardware tests.

REFERENCES

[1] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Sendai, Japan, 2004, pp. 2149–2154.

[2] NVIDIA, “Isaac Sim Documentation,” 2026. [Online]. Available: https://docs.isaacsim.omniverse.nvidia.com/

[3] NVIDIA, “Perception Data Generation (Replicator),” Isaac Sim Documentation, 2026. [Online]. Available: https://docs.isaacsim.omniverse.nvidia.com/6.0.0/replicator_tutorials/

[4] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2017, pp. 23–30.

[5] QGroundControl Project, “QGroundControl User Guide,” 2026. [Online]. Available: https://docs.qgroundcontrol.com/

[6] MAVLink, “Mission Protocol,” MAVLink Developer Guide, 2026. [Online]. Available: https://mavlink.io/en/services/mission.html

[7] Ultralytics, “Ultralytics YOLO11,” 2024. [Online]. Available: https://docs.ultralytics.com/models/yolo11/

[8] EBABIL Project Team, “Machine-learning-based automatic threat identification, assessment, prioritization, target assignment, and optimized UAV engagement,” internal project document, 2026.