COMPUTER VISION AND PATTERN RECOGNITION
КОМП'ЮТЕРНИЙ ЗІР ТА РОЗПІЗНАВАННЯ ОБРАЗІВ

https://doi.org/10.15407/intechsys.2026.02.006
UDC 004.896

D.I. RYABOKON, PhD (Engineering), Senior Researcher,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/0000-0002-9329-6706
d.ryabokon@gmail.com

SYNTHETIC DATA GENERATION FOR COMPUTER VISION TASKS IN SITL SYSTEMS WITH UNREAL ENGINE AND AIRSIM

Sustainable object detection, classification, and tracking are a critical path in robotics at perception phase. Deep learning models used for detection require vast and diverse datasets with accurate annotations. However, acquiring real-world data that captures rare, complex, or hazardous scenarios is both time-consuming and costly. To address this limitation, synthetic data generation using virtual environments has emerged as a promising alternative. This research investigates the application of synthetic datasets, created within high-fidelity 3D virtual reality environments, for training deep learning models in object detection tasks. The proposed method leverages tools such as Unreal Engine and procedural scripting to generate large volumes of realistic, annotated image data. These synthetic scenes are carefully constructed to include various object types, dynamic interactions, lighting variations, occlusions, and edge-case scenarios that are often missing from real-world datasets. The study employs the YOLO family of deep learning models, which are known for their high accuracy and low latency, making them suitable for real-time robotic applications. Experimental results demonstrate that models trained exclusively on synthetic data demonstrate partial transferability to real-world scenarios, while the addition of a small fraction of real data significantly improves performance. Key advantages include reduced development time, lower annotation costs, improved coverage of rare scenarios, and enhanced model generalization. The integration of synthetic data pipelines into the ML lifecycle also facilitates better experiment management and scalability. The research concludes that synthetic data is a powerful enabler for developing robust object detection systems in robotics and AI. Future work will focus on narrowing the domain gap between synthetic and real-world imagery, enhancing realism with photometric effects, and expanding the dataset to include additional object classes and behaviors for multi-task learning.

Cite: Ryabokon D.I. Synthetic Data Generation for Computer Vision Tasks in Sitl Systems with Unreal Engine and Airsim. *Information Technologies and Systems*. 2 (8). 2026. 6—17. https://doi.org/10.15407/intechsys.2026.02.006

© Publisher PH "Akademperiodyka" of the NAS of Ukraine, 2025. This is an Open Access article under the CC BY-NC-ND 4.0 license (https://creativecommons.org/licenses/by-nc-nd/4.0/

Keywords: *synthetic data, computer vision, deep learning, robotics, SITL, Unreal Engine, AirSim, YOLO.*

Introduction

Autonomous robotic systems are increasingly used in safety-critical and high-complexity domains such as aerial inspection, autonomous driving, disaster response, and smart manufacturing. A key enabler of such systems is the perception module, responsible for real-time object detection, classification, and tracking. Deep learning models, especially those from the YOLO family, have demonstrated state-of-the-art performance in these tasks, providing high-speed and high-accuracy inference on embedded platforms.

However, the performance of these models critically depends on the availability of large, diverse, and accurately annotated datasets. The collection of real-world data is time-consuming, costly, and often fails to capture rare, occluded, or hazardous scenarios, which are vital for robust model generalization. Moreover, the physical limitations and safety concerns involved in real-world testing severely constrain the development cycle of autonomous systems.

To overcome these challenges, recent studies have explored synthetic data generation using virtual environments. Notably, works from NVIDIA (SimNet, Omniverse), Intel (OpenBot + AirSim), and academic institutions (Stanford, MIT) have demonstrated that high-fidelity 3D simulation combined with domain randomization or domain adaptation significantly improves model robustness and transferability. Yet, many existing pipelines focus either on synthetic data creation or model deployment, but not the seamless end-to-end integration that connects perception, planning, and control in a unified development loop.

The application of synthetic data for training deep learning models has gained significant attention recently. Shah et al. [1] introduced AirSim, a platform combining Unreal Engine with physics-based simulation, enabling realistic data collection for autonomous vehicles. Similarly, Laux et al. [2] demonstrated the feasibility of synthetic aerial data for on-drone object detection tasks. Kiefer et al. [3] and Barisic et al. [4] further highlighted synthetic data advantages in UAV scenarios, emphasizing domain adaptation challenges. Maxey et al. [5] proposed neural-rendering techniques to narrow the synthetic-to-real domain gap. Unlike previous studies focusing primarily on either data generation or isolated model evaluation, our work provides a fully integrated SITL pipeline—covering scene generation, automated annotation, model training, perception, and real-time path planning validation—offering a complete, scalable framework for AI-driven robotics development. Collins et al. [6] present a scalable and modular synthetic data generation framework designed to support aerial autonomy, demonstrating how structured simulation pipelines can enhance dataset diversity and improve the development of perception systems for UAV applications.

The object of this research is a complete software-in-the-loop (SITL) simulation system that enables the generation, testing, and evaluation of AI-driven perception and navigation in robotics.

The aim of the study is to develop and validate an end-to-end pipeline that integrates synthetic data generation, perception that uses deep learning, and path planning for autonomous systems operating in simulation.

The objectives of the research include:

- Building 3D environments and generating labeled datasets;
- Emulating robotic agents and sensors using AirSim with SITL capabilities;
- Applying YOLO-based models for real-time object detection;
- Integrating system components via ROS for closed-loop testing.

This approach provides a reproducible, scalable, and cost-effective framework for training and evaluating AI components for autonomous robots in realistic, yet fully controllable, environments. It provides a quantitative sim-to-real data efficiency analysis, demonstrating that the introduction of 2% real-world data is sufficient to eliminate the majority of the domain gap.

System Architecture and Solution Design

The developed system is a modular Software-in-the-Loop (SITL) pipeline that enables end-to-end testing of robotic perception and navigation using synthetic data. It combines a high-fidelity 3D simulation environment built with Unreal Engine and AirSim, a PX4 or ArduPilot SITL flight controller, and a ROS2-based control layer for system coordination.

AirSim provides realistic rendering and physics-based simulation of sensors and flight dynamics, supporting the generation of labeled datasets and live testing. The ROS2 workspace manages control and telemetry through a command node that translates high-level goals into MAVLink messages. These are sent via MAVROS to the SITL controller, which emulates the drone's behavior and returns real-time feedback.

A web server running on a companion PC (e.g., NVIDIA Jetson Orin) provides a user interface for sending commands and visualizing telemetry and video. This interface communicates with the ROS2 system to control missions and monitor system state.

The architecture (see Fig. 1) supports future transition to Hardware-in-the-Loop (HIL) by replacing the SITL controller with a real flight computer and sensors. This flexible design enables reproducible testing of perception and planning algorithms in a safe, scalable, and controlled virtual environment.

Virtual Scene Design and Synthetic Dataset Generation

The virtual scene was created using Unreal Engine with the AirSim plugin to simulate realistic environments for synthetic data generation.

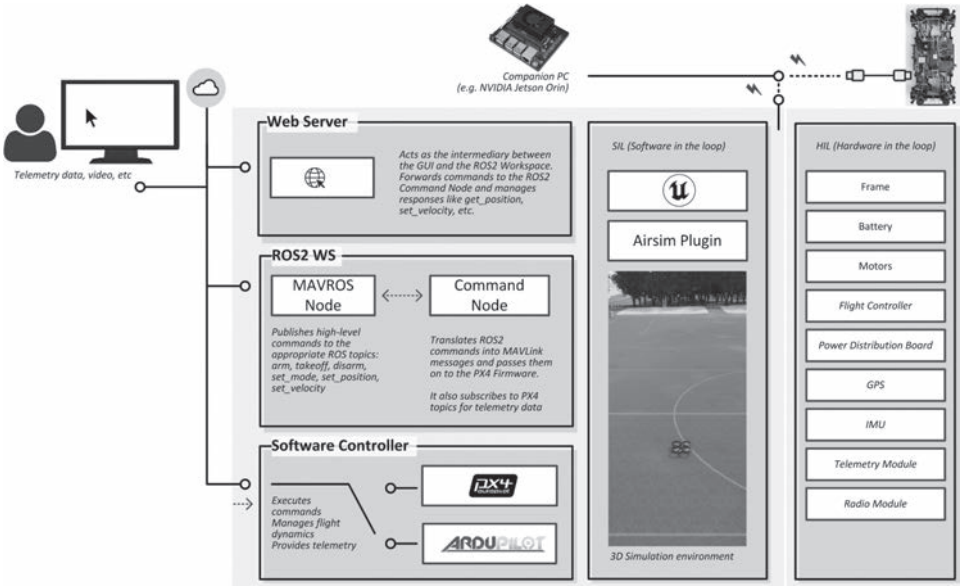


Fig. 1. Modular SITL architecture integrating Unreal Engine, AirSim, ROS2, and flight controllers

As shown in the figure, the editor environment includes manually placed 3D assets such as vehicles (e.g., UAZ-452), traffic cones, and terrain details. Each object is inserted as a Static Mesh Actor and organized in the World Outliner, enabling precise control over scene composition, object types, and spatial arrangement. This environment allows the simulation of diverse visual conditions, including different lighting setups, occlusions, angles of view, and background clutter. To ensure reproducibility, the simulation environment was configured with the following technical parameters:

- rendering resolution: up to 2 MP (1920×1080),
- average rendering rate 30+ fps,
- dynamic shadows enabled,
- camera FOV set to 90°.

The AirSim plugin is used to control a virtual drone and capture sensor data, including RGB images, depth maps, and segmentation masks. Scene parameters such as weather, sun angle, and camera position (see Fig. 2) are varied programmatically or manually to enhance diversity.

By leveraging the photorealistic rendering and simulation capabilities of Unreal Engine combined with AirSim's automation and control interface, this approach produces large volumes of high-quality, labeled data suitable for training object detection models. The synthetic scenes form the foundation for developing and testing deep learning-based perception systems in a controlled and repeatable manner.



Fig. 2. Scene setup in Unreal Engine for synthetic data generation

Model Training and Evaluation

To assess the effectiveness of synthetic data for training object detection models, we implemented a full machine learning lifecycle (see Fig. 3) integrating data synthesis, annotation, model training, and evaluation. As illustrated in the provided CI/CT pipeline diagram, the dataset created in Unreal Engine + AirSim is stored along with its metadata, artifacts, and targets, which are versioned and managed in a dedicated model registry.

The synthetic dataset comprises three object classes (see Fig. 4): a geometric cube, a pyramid, and a miniature vehicle model (Bob), generated within the Unreal Engine and AirSim environment under varied lighting conditions, camera angles, and background configurations.

The class distribution in the training set is intentionally imbalanced, with 160 instances each for Cube and Pyramid and over 1000 instances for the Bob class, reflecting the relative importance of vehicle-type objects in the target application domain. The bounding box spatial distribution analysis confirms that objects are distributed across the full image frame, with Bob instances exhibiting a wider range of aspect ratios due to perspective variation during aerial capture. The YOLOv8n architecture was selected as the detection backbone, comprising 186 layers and 2.68M parameters with a computational cost of 6.8 GFLOPs, making it suitable for deployment on embedded platforms such as NVIDIA Jetson. Generalization performance was evaluated on a held-out real-world test set comprising 1494 images. Inference was performed on an NVIDIA GeForce RTX 3080 Laptop GPU, achieving a mean processing time of 2.0 ms per image (0.2 ms preprocessing + 1.8 ms inference), which confirms suitability for real-time operation.

Quantitative results are summarized in Table 1 and illustrated on Fig. 5—6. A temporary performance degradation is observed when only a

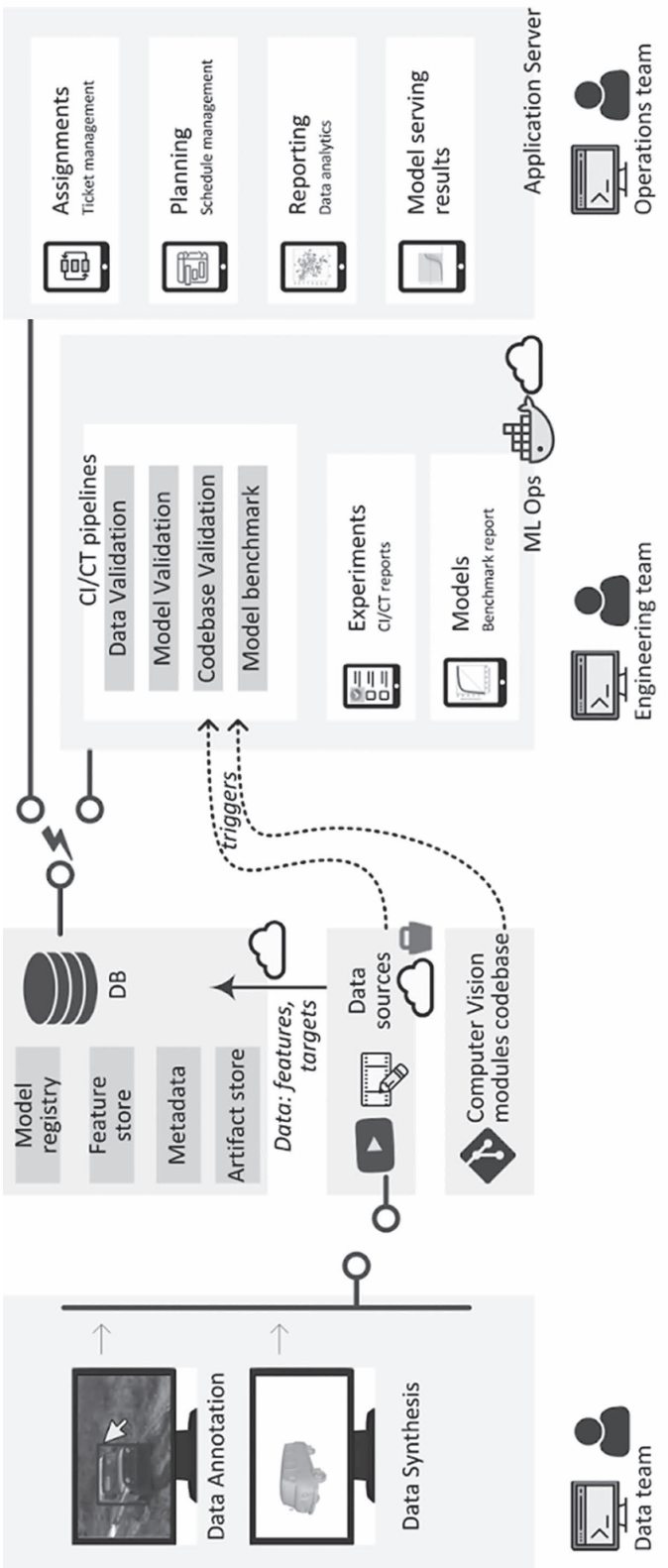


Fig. 3. ML lifecycle pipeline for synthetic data management and model validation

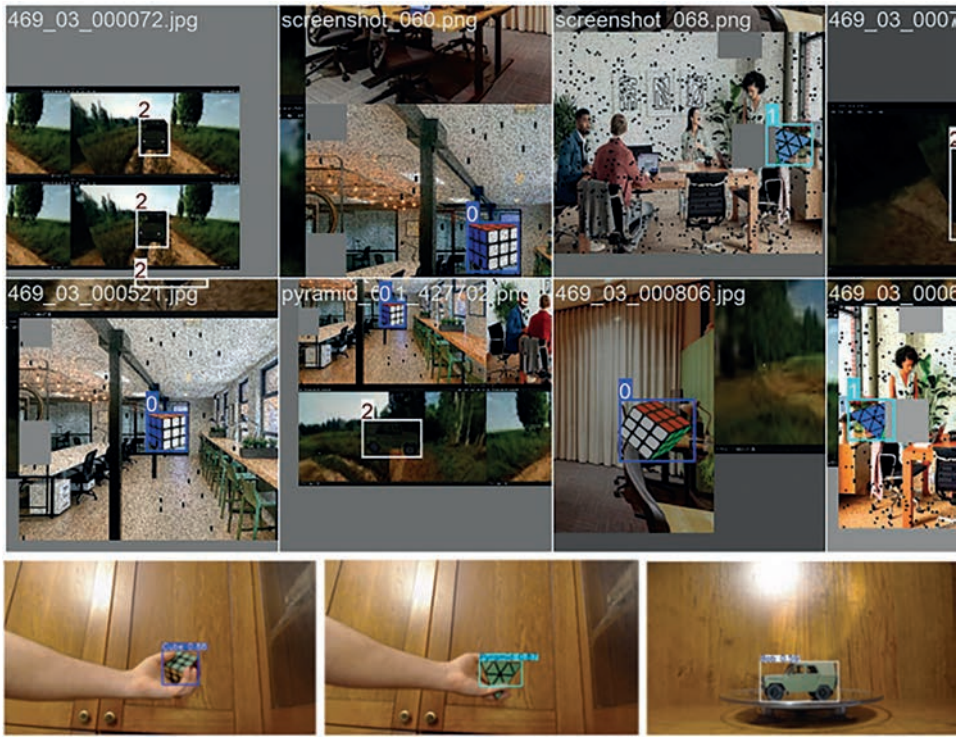


Fig. 4. Examples of synthetic data for training on top and real images on bottom

Table 1. **Model validation with 0%, 1%, 2%, and 5% of real data in train dataset**

Training data \ mAP@0.5	Cube	Pyr	Bob	All
Synthetic: 160 cubes, 160 Pyramids, 1000+ Bobs	0.90	0.91	0.30	0.71
Synthetic + Real: 6 cubes, 6 pyramids, and 10 Bobs	0.62	0.89	0.99	0.84
Synthetic + Real: 12 cubes, 12 pyramids, and 20 Bobs	0.98	0.95	0.99	0.97
Synthetic + Real: 30 cubes, 30 pyramids, and 50 Bobs	0.99	0.99	0.99	0.99

minimal number of real samples is introduced. This behavior can be explained by unstable domain alignment and catastrophic forgetting effects during fine-tuning. To mitigate this instability, future work will explore partial backbone freezing and lower learning rates during the fine-tuning stage.

Experiments with perception and 3D Path Planning in SITL

The developed interface integrates real-time perception, bird's-eye view (BEV) mapping, and 3D path planning (see Fig. 7) within a unified Software-in-the-Loop (SITL) environment. The top visualization shows re-

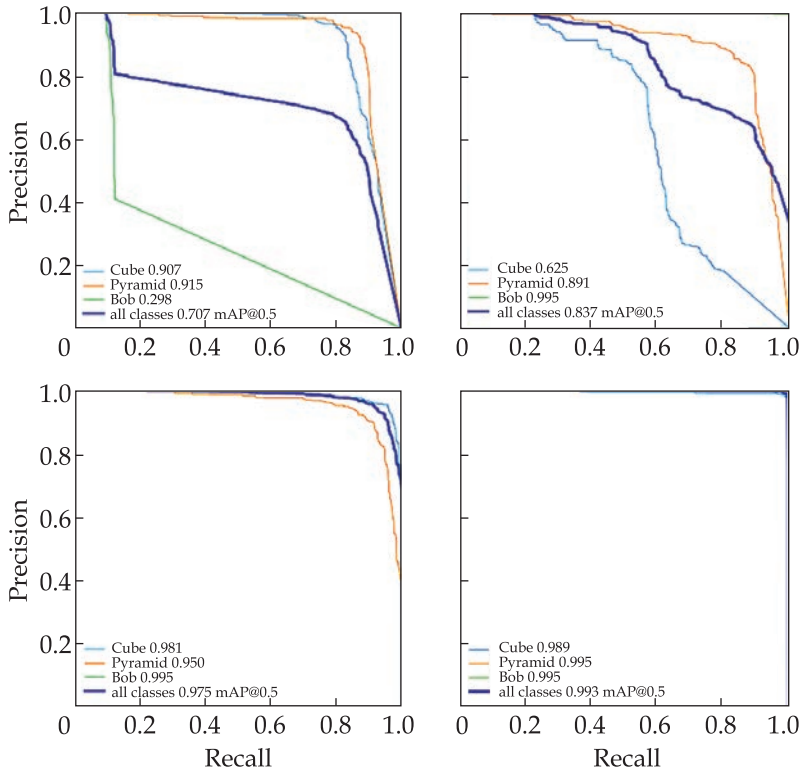


Fig. 5. PR-curves for model validation with 0%, 1%, 2%, and 5% (from left to right) of real data in train dataset

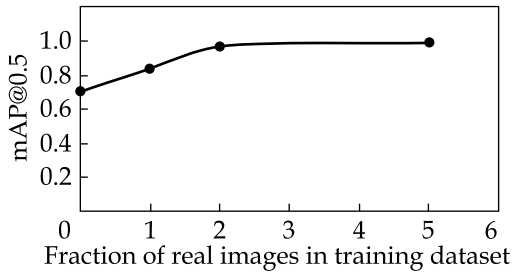


Fig. 6. Efficiency curve(for held-out real test set): saturation of mAP@0.5 over fraction of real images in training dataset

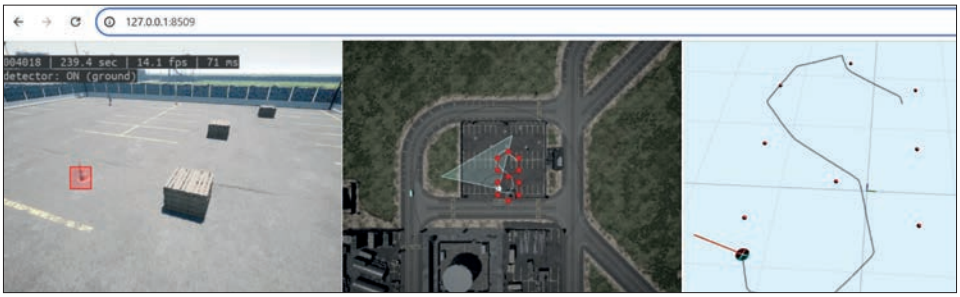


Fig. 7. Object detection, bird-eye-view visualization, and 3D visualization for path planning

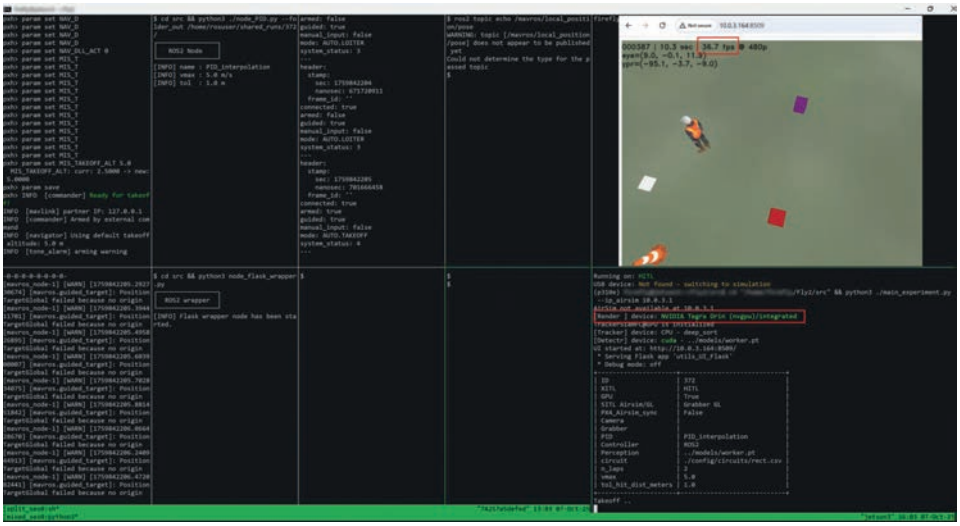


Fig. 8. Example of the simulation environment with flight controller and messaging

al-time object detection output (YOLOv8 model) from the drone's RGB camera stream, validating perception in dynamic scenes generated via Unreal Engine and AirSim.

The bottom-left BEV projection provides intuitive monitoring of object positions, waypoints, and drone orientation, with detected objects marked in red and the drone's field of view indicated by a blue triangle. The 3D visualization on the right illustrates the drone's trajectory planned based on perception outputs and mission goals. This integrated visual framework enables comprehensive validation of perception and navigation algorithms without risk to real hardware.

Simulation Performance and Technical Parameters

The SITL setup was run in a multi-pane tmux session (see Fig. 8) that included PX4 SITL, MAVProxy, ROS2 nodes, and a small Flask server for sending commands. This allowed all system messages, ROS2 topics, and perception outputs to be viewed at the same time.

The experiments were executed on an NVIDIA Jetson platform equipped with a 6-core Arm Cortex-A78AE CPU, an integrated Orin GPU running CUDA 12.6, 7.4 GB of RAM, Jetson Linux (L4T 36.4.4), and the standard deep-learning stack consisting of CUDA 12.6, cuDNN 9.3, and TensorRT 10.3.

The Jetson platform was selected because of well-established industrial standard for embedded AI combining low-power operation and affordable price of high performance. It gives great GPU-accelerated inference suitable for real-time deep-learning perception.

PX4 (upper 1) and MAVProxy (lower 1) operated with their default simulation. ROS2 nodes (upper 2,3, and 4) published telemetry and detec-

tion results at 10 Hz. The YOLOv8 perception module ran (lower 5) on an NVIDIA GPU and processed each frame with 30+ fps, which is fast enough for real-time use. The front end app (upper right) backed by flask server (lower 2) demonstrates 30+ fps for processing synthetic images of 640×480.

System flexibility is achieved by making the individual components (such as the flight controller, simulation environment, ROS2 middleware, or perception model) modifiable or replaceable. For example, the flight stack can choose between PX4 and ArduPilot whereas the simulation and perception are the same. ROS2 messaging can be adapted by adding or renaming topics to support alternative telemetry and control (publishing and listening various topics). Finally user interface can be implemented through different web servers (Flask/FastAPI/Node.js) without affecting the whole SITL execution pipeline.

Scene layout, lighting conditions, sensor configurations, and data-capture rates remain explicitly defined through configuration files and therefore stable across runs. With that we achieve a safe, and controlled environment to reproduce the experiments in a scale.

Limitations and Challenges

Although the proposed approach demonstrates strong performance and promising transferability from synthetic to real-world scenarios, several limitations exist. Primarily, a noticeable domain gap between simulated and real images persists, particularly regarding subtle variations in texture, material properties, and complex photometric effects. Additionally, while YOLO-based models offer high inference speed suitable for real-time applications, they may exhibit accuracy trade-offs in densely occluded or highly cluttered scenes. Future research should focus on advanced domain adaptation techniques, improving simulation realism, and exploring hybrid training strategies that combine synthetic and limited real-world data to enhance robustness further.

Conclusion

This research confirms the effectiveness of synthetic data as a core enabler for developing and testing AI-based perception and navigation systems in robotic applications. By integrating Unreal Engine and AirSim in a modular Software-in-the-Loop (SITL) pipeline, we demonstrated a reproducible and scalable framework for training deep learning models, evaluating them under diverse scenarios, and validating their performance across synthetic and real-world domains.

The approach significantly reduces the cost and effort associated with data collection and annotation, while improving model robustness by including rare and hazardous scenarios in the training loop. Experiments show that models trained exclusively on synthetic data demonstrate par-

tial transferability, while small amounts of real data significantly improve performance.

Moreover, the developed pipeline combines perception, situational awareness (BEV), and path planning modules, offering a complete testbed for real-time, closed-loop validation. The system's architecture is compatible with both SITL and future Hardware-in-the-Loop (HIL) extensions, making it suitable for continuous AI development and deployment.

Future work will address domain adaptation to narrow the synthetic-to-real gap, enrich scene realism, expand object class diversity, and explore applications beyond visual perception, such as multi-sensor fusion and autonomous decision-making. Overall, the proposed solution accelerates AI model development in robotics, supports safety and repeatability, and lays a solid foundation for next-generation autonomous systems.

DECLARATION

Declaration of Competing Interest. No potential conflict of interest is reported by the author.

***Funding*.** The author declare that no funds, grants, or other support were received during the preparation of this manuscript.

Use of AI. The author declare that AI-assisted tools were used solely for language editing and spell checking. No AI tools were used for scientific analysis, interpretation of results, or generation of original scientific content.

REFERENCES

1. Shah S., Dey D., Lovett C., Kapoor A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. *Field and Service Robotics*, 2018, Vol. 5, 621–635. https://doi.org/10.1007/978-3-319-67361-5_40
2. Laux L., Schirmer S., Schopferer S., Dauer J.C. Build Your Own Training Data – Synthetic Data for Object Detection in Aerial Images. In: *Software Engineering 2022 Workshops,* Bonn, 2022, 182–190. https://doi.org/10.18420/se2022-ws-18
3. Kiefer B., Ott D., Zell A. Leveraging Synthetic Data in Object Detection on Unmanned Aerial Vehicles. 2*6th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada, 2022, 3564–3571. https://doi.org/10.1109/ICPR56361.2022.9956710
4. Barisic A., Petric F., Bogdan S. Sim2Air – Synthetic Aerial Dataset for UAV Monitoring. *Sensors*, 2022 , Vol. 22 (14), Article 5074. https://doi.org/10.3390/s22145074
5. Maxey C., Choi J., Lee H., Manocha D., Kwon H. UAV-Sim: NeRF-based Synthetic Data Generation for UAV-based Perception. *arXiv preprint,* 2023, arXiv:2310.16255. https://doi.org/10.48550/arXiv.2310.16255
6. Collins T., Waggoner J., Stanciulescu B., et all. Scalable Modular Synthetic Data Generation for Advancing Aerial Autonomy. *arXiv preprint,* 2022, arXiv:2211.05335. https://doi.org/10.48550/arXiv.2211.05335

Received 19.09.2025
Accepted 16.04.2026
Published 01.06.2026

Д.І. РЯБОКОНЬ, канд. техн. наук, старш. наук. співроб.,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, Київ, 03187, Україна
https://orcid.org/0000-0002-9329-6706
d.ryabokon@gmail.com

ГЕНЕРАЦІЯ СИНТЕТИЧНИХ ДАНИХ
ДЛЯ ЗАДАЧ КОМП'ЮТЕРНОГО ЗОРУ
В *SITL*-СИСТЕМАХ НА ОСНОВІ *UNREAL ENGINE* ТА *AIRSIM*

Вступ. Розвиток автономних робототехнічних систем напряму залежить від ефективності модулів сприйняття, які забезпечують виявлення, класифікацію та супровід об'єктів у реальному часі. Проте збирання та розмітка великих обсягів реальних даних є тривалим, дороговартісним процесом і рідко охоплює рідкісні або небезпечні сценарії. Це створює потребу у використанні синтетичних даних, отриманих у високодеталізованих і високоякісних віртуальних середовищах, які дають змогу відтворювати різноманітні умови та автоматизувати процес анотування.

Метою роботи є розроблення та апробація інтегрованого програмного контуру *Software-in-the-Loop (SITL)*, що поєднує генерацію синтетичних даних в *Unreal Engine* + *AirSim*, тренування моделей глибинного навчання *(YOLOv*8*)* та перевірку їх ефективності у задачах виявлення та супроводу об'єктів, а також перевірку працездатності в системі планування траєкторій.

Методи. У дослідженні застосовано побудову фотореалістичних віртуальних сцен в *Unreal Engine* із використанням бібліотек 3*D*-об'єктів та процедурних сценаріїв; генерацію синтетичних *RGB*-зображень, карт глибини та сегментаційних масок через *AirSim*; автоматизовану анотацію об'єктів; тренування моделей *YOLOv*8 на синтетичних наборах даних; оцінювання продуктивності на синтетичних і реальних даних; інтеграцію *perception*-модуля, *bird's-eye view* візуалізації та 3*D*-планування у *SITL*-середовищі.

Результати. Запропонований підхід забезпечив створення масштабованого набору даних з різними класами об'єктів та умовами освітлення, оклюзіями і фоновими ускладненнями; успішне тренування моделей *YOLOv8* виключно на синтетичних даних; отримання високої точності на реальних тестах; інтеграцію *perception* і *path planning* у єдиній *SITL*-системі з можливістю подальшого переходу до *Hardware-in-the-Loop*.

Висновки. Результати роботи свідчать про те, що синтетичні дані є ефективним інструментом для підготовки моделей комп'ютерного зору в робототехнічних системах. Використання *Unreal Engine* та *AirSim* у складі *SITL*-контуру дає змогу зменшити витрати на збирання та розмітку реальних даних, забезпечити відтворюваність експериментів і покращити генералізацію моделей на реальних сценаріях. Подальші дослідження призначені зменшити розрив між синтетичними та реальними даними, розширити класи об'єктів і застосування багатосенсорних стратегій.

Ключові слова: *синтетичні дані, комп'ютерний зір, глибоке навчання, робототехніка, SITL, Unreal Engine, AirSim, YOLO.*