DOT/FAA/TC-26/30

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Verification of the Learned Models in Machine Learning and Artificial Intelligence-based Aviation Systems

October 2026

Final report



U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The U.S. Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report. The findings and conclusions in this report are those of the author(s) and do not necessarily represent the views of the funding agency. This document does not constitute FAA policy. Consult the FAA sponsoring organization listed on the Technical Documentation page as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Technical Report Documentation Page

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/TC-26/30		

4. Title and Subtitle	5. Report Date
Verification of the Learned Models in Machine Learning and Artificial Intelligence-based Aviation Systems	October 2026
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Peng Wei, Can Vu	

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
George Washington University (GWU) Department of Mechanical and Aerospace Engineering Washington, DC 20052	11. Contract or Grant No. 692M15-24-T-00008

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Federal Aviation Administration William J. Hughes Technical Center Atlantic City International Airport, NJ 08405	Final Report
	14. Sponsoring Agency Code AIR-120

15. Supplementary Notes

16. Abstract

This report presents a comprehensive verification framework for machine learning (ML) and artificial intelligence (AI) models integrated into safety-critical avionic software stacks, with a specific focus on computer vision for autonomous Unmanned Aircraft System (UAS) package delivery and safe landing operations. Acknowledging that traditional static validation metrics (such as mean Average Precision (mAP) fail to capture temporal anomalies and continuous perception dropouts during flight, this research introduces a three-thrust approach to ensure deterministic safety and statistical reliability.

Thrust I establishes mathematical metrics for evaluating training data sufficiency, introducing Frame Interval Error (FIE) to quantify time-varying perception blackouts and utilizing an Optimal Transport (OT) framework with Wasserstein distances to measure statistical mismatch between training data and the Operational Design Domain (ODD). Thrust II implements conformal prediction to quantify runtime uncertainty, providing finite-sample confidence guarantees with bounded error rates; a comparative benchmarking between a Real-Time Detection Transformer (RT-DETR) and YOLOv8s highlights critical trade-offs between statistical calibration accuracy and onboard computational latency. Thrust III accelerates the verification pipeline by deploying a high-fidelity synthetic data generation suite using Microsoft AirSim and Unreal Engine to expose edge-case model vulnerabilities under extreme weather and lighting conditions. Collectively, these methodologies provide a mathematically rigorous toolkit for the Federal Aviation Administration (FAA) to certify learning-based perception systems in next-generation aviation.

17. Key Words	18. Distribution Statement
Aviation Safety Certification, Computer Vision, Autonomous UAS, Conformal Prediction, Optimal Transport, Synthetic Data Generation, AirSim	This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	33	

Contents

1 Introduction and project overview 1

2 Research objective and technical approach 2

2.1 Thrust 1: Sufficient conditions of training dataset 2

2.2 Thrust 2: Confidence bound of ML/AI model output 2

2.3 Thrust 3: Efficiency improvement of verification process 3

3 Technical core and methodology 3

3.1 Thrust 1: Sufficient conditions of training dataset given ML/AI module requirements. 3

3.1.1 Task 1.1: Impact of pre-training dataset selection on AI model performance 3

3.1.2 Task 1.2: Verifying complete ODD coverage of training dataset 8

3.2 Thrust 2: Providing confidence bound and uncertainty quantification of ML/AI model 14

3.2.1 Importance of uncertainty quantification 14

3.2.2 Task 2.1: Conformal prediction foundations in computer 14

3.2.3 Conformal prediction application to computer vision 15

3.2.4 Task 2.2: Verify confidence bound of classification uncertainty 16

3.3 Thrust 3: Speeding up verification by leveraging synthetic data generation 19

3.3.1 Task 3.1: Synthetic data generation for computer vision based autonomous landing 19

3.3.2 Task 3.2: Scaling flight simulation and data synthesis with generative 3D reconstruction 21

4 Dataset collected and generated 23

4.1 Real world dataset collection and ODD definition 23

4.2 Synthetically generated datasets 23

5 Conclusion 24

6 References 25

Figures

Figure 1. Technical framework overview 1
Figure 2. Pre-training and fine-tuning procedure to develop object detection models 4
Figure 3. Pre-training and fine-training mAP metrics on different datasets 5
Figure 4. Real-world flight test video with orange landing pad 6
Figure 5. Proposed FIE metrics for all YOLOv8s models evaluated on flight test dataset 8
Figure 6. Conceptual visualization of data distribution in latent space 9
Figure 7. OT between source and target distribution 9
Figure 8. U-Net architecture used as denoising backbone in diffusion model 10
Figure 9. Visualization of denoising process in DDPM 11
Figure 10. Synthetic images produced by diffusion model at 64 × 64 resolution 12
Figure 11. Sample images from diffusion generation process with detections 13
Figure 12. System block diagram of RT-DETR end-to-end object detection model 17
Figure 13. Estimated distributions of class prediction set sizes on VisDrone test dataset 18
Figure 14. Architecture and data flow of AirSim simulation framework 20
Figure 15. Rendering of front yard landing zone with weather API implementation 21
Figure 16. Synthetic data generation workflow 22
Figure 17. Generative mesh reconstruction pipeline using SAM 3D 22

Tables

Table 1. Validation criteria for object detection prediction errors 5
Table 2. Optimal fine-tuning performance across architecture variations 6
Table 3. Aggregate performance indicators on flight test dataset 7
Table 4. Wasserstein transport mismatch to target ODD distribution 11
Table 5. Downstream YOLO performance map on real ODD target frames 11
Table 6. Generative diffusion realism mapping across resolutions 13
Table 7. Wasserstein distances following synthetic ODD augmentation 14
Table 8. End-to-end processing speed breakdown per frame 18
Table 9. Conformal prediction set statistics 19
Table 10. Automated label refinement metrics by target asset class 23

Acronyms

Acronym	Definition
AI	Artificial Intelligence
API	Application Programming Interface
APS	Adaptive Prediction Sets
CLIP	Contrastive Language-Image Pre-Training
CNN	Convolutional Neural Network
COCO	Common Objects in Context
DDPM	Denoising Diffusion Probabilistic Model
DOTA	Dataset for Object Detection in Aerial Images
FAA	Federal Aviation Administration
FID	Fréchet Inception Distance
FIE	Frame Interval Error
GWHD	Global Wheat Head Detection
HITL	Hardware-in-the-Loop
IoU	Intersection-over-Union
mAP	mean Average Precision
ML	Machine Learning
NMS	Non-Maximum Suppression
ODD	Operational Design Domain
OT	Optimal Transport
SAM	Segment Anything Model
RT-DETR	Real-Time Detection Transformer
UAS	Unmanned Aircraft System
VLM	Vision-Language Model

Executive summary

The swift integration of machine learning (ML) and artificial intelligence (AI) components into safety-critical avionic software systems poses a fundamental challenge for traditional airworthiness certification paradigms. Deep Neural Networks (DNNs), while computationally robust at processing complex perception tasks like obstacle tracking and landing site localization, often act as uninterpretable black boxes that degrade unpredictably under structural distribution shifts. This research addresses these critical verification gaps by formulating a structured, multi-thrust validation framework specifically engineered for the safety assessment of computer vision algorithms in UAS autonomous landing environments.

This research task designed and mathematically evaluated three integrated technical thrusts:

1. An Optimal Transport (OT) framework utilizing 1-Wasserstein and 2-Wasserstein metrics to evaluate the statistical sufficiency and Operational Domain Design (ODD) coverage of training datasets.
2. A conformal prediction deployment providing rigorous finite-sample uncertainty quantification and risk control boundaries for model classification streams.
3. A high-fidelity synthetic data synthesis engine powered by Cosys-AirSim and Unreal Engine 5 to rapidly generate annotated training images and evaluate model failure boundaries under off-nominal atmospheric fields.

Quantitative evaluation benchmarks systematically contrast the lightweight Convolutional Neural Network (CNN)-based YOLOv8 architecture against the advanced transformer-based Real-Time Detection Transformer (RT-DETR) framework, exposing critical tradeoffs between processing latency, detection precision, and statistical self-calibration accuracy. The resulting procedures provide the Federal Aviation Administration (FAA) with practical, mathematically sound tools and structural insight required to define future certification baselines for learning-based avionic systems.

1 Introduction and project overview

The primary objective of this project is to design, implement, and formally evaluate systematic safety verification methodologies for learned models embedded within aviation systems. The research introduces a three-thrust approach to ensure deterministic safety and statistical reliability. Specifically, the scope focuses on the itemized verification of deep learning components within an ML- or AI-based avionic software stack. This structured approach integrates three novel core methodologies to resolve prominent verification challenges:

1. Examining and identifying the formal sufficient conditions of the training dataset given explicit ML/AI subsystem functional requirements.
2. Providing mathematically rigorous confidence bounds or localized uncertainty quantification metrics of the ML/AI model outputs during runtime and design-time optimization.
3. Speeding up and optimizing the verification timeline by leveraging automated high-fidelity synthetic data generation pipelines.

At a system-wide level, these verification protocols offer feasible paths toward the safety validation of learning-based perception and computer vision models. The framework in Figure 1 supplies the FAA with candidate software toolboxes and statistical frameworks necessary to robustly certify next-generation ML/AI-driven avionic configurations.

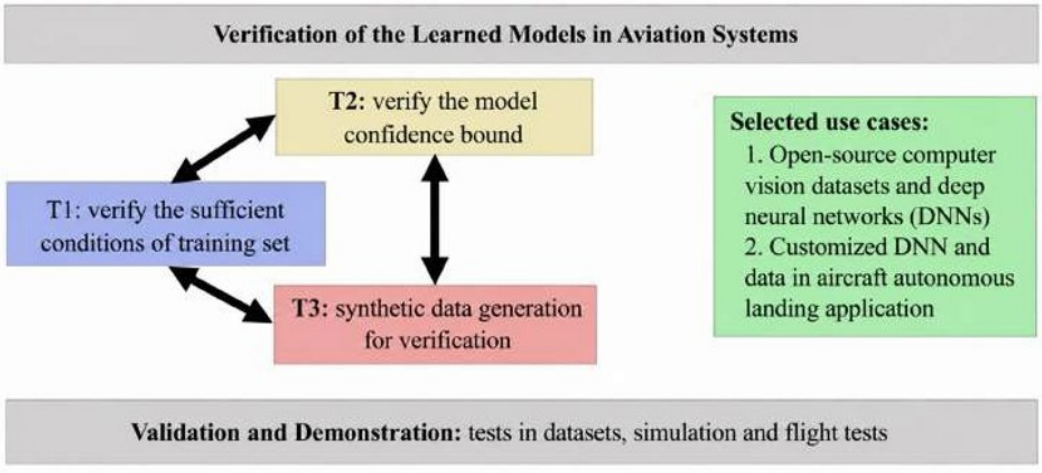


Figure 1. Technical framework overview

2 Research objective and technical approach

This project formulates verification procedures and gathers concrete numerical evidence targeting a primary generalizable ML/AI avionic use case: computer vision and deep learning-based autonomous vehicle landing under nominal and off-nominal conditions. Off-nominal perturbations include structural reconfigurations near the delivery site, such as the unexpected proximity of pedestrians, pets, and passenger vehicles, as well as rapid atmospheric changes due to shifting weather and lighting fields. From a methodological perspective, the research systematically addresses three central questions:

- **Research Question 1:** How can we mathematically verify if a target training dataset is sufficient to satisfy high-level avionic functional design requirements?
- **Research Question 2:** How can we precisely quantify the statistical uncertainty levels and confidence bounds of ML/AI model outputs during both design-time calibration and active runtime deployment?
- **Research Question 3:** How can the ML/AI verification lifecycle be rendered more efficient by interpolating real-world imagery with automated synthetic data generation engines?

2.1 Thrust 1: Sufficient conditions of training dataset

The required structural information is identified and isolated within a training dataset to successfully meet baseline model performance criteria. First, empirical studies illustrate how selecting distinct initial training datasets directly impacts downstream target detection capability. Second, the rigorous mathematical framework of OT used to compute the statistical coverage or mismatch between multi-dimensional training dataset distributions and the defined ODD is integrated. How this statistical mismatch directly translates into model accuracy degradation is then established.

2.2 Thrust 2: Confidence bound of ML/AI model output

An advanced statistical framework known as conformal prediction, which constructs finite-sample confidence intervals for any black-box model and dataset with low computational overhead, is leveraged. The CP is applied across both deep CNNs (the YOLOv8 architecture family) and vision transformers (the RT-DETR) to calibrate and quantify visual classification uncertainty under operational strain.

2.3 Thrust 3: Efficiency improvement of verification process

High-fidelity generative and physics-based data generation pipelines to supplement training records and stress-test computer vision algorithms are developed. This includes setting up automated pipelines to structure and annotate real-world and simulated flight videos, establishing a parametric toolbox to render flights using AirSim and Unreal Engine 4/5, and mitigating the sim-to-real gap by injecting generative 3D asset reconstructions from standalone 2D images directly into virtual evaluation worlds.

3 Technical core and methodology

3.1 Thrust 1: Sufficient conditions of training dataset given ML/AI module requirements

3.1.1 Task 1.1: Impact of pre-training dataset selection on AI model performance

During the final descent phase of an autonomous UAS package delivery sequence, the onboard perception stack must accurately localize and isolate ground hazards such as pedestrians, cyclists, and moving vehicles to satisfy rigorous safety profiles. To overcome the scarcity of specialized operational flight imagery, engineers frequently rely on transfer learning configurations, initializing deep network weights on expansive image repositories before performing localized fine-tuning on domain-specific aerial data. However, the explicit downstream impact of selecting different initial pre-training data distributions remains highly underexplored when evaluated using traditional computer vision metrics.

Standard validation paradigms depend heavily on static, aggregate indicators like mean Average Precision (mAP), computed at localized Intersection-over-Union (IoU) thresholds (mAP@50 and mAP@50-95). However, mAP scores exhibit severe verification limitations: they are highly sensitive to software implementation changes and processing latencies, and blend disparate indicators (such as localized spatial precision, classification capability, and confidence calibration) into a single score. To bridge these verification gaps, this study analyzes pre-training dynamics across four diverse source benchmarks utilizing the Ultralytics YOLOv8s architecture, combining a CSPDarknet backbone and a PANet multi-scale feature neck. Additionally, the Frame Interval Error (FIE) metric suite to track perception failure parameters during continuous time-varying flight operations is formulated. The pre-training and fine-tuning procedures used to develop object detection models is presented in Figure 2.

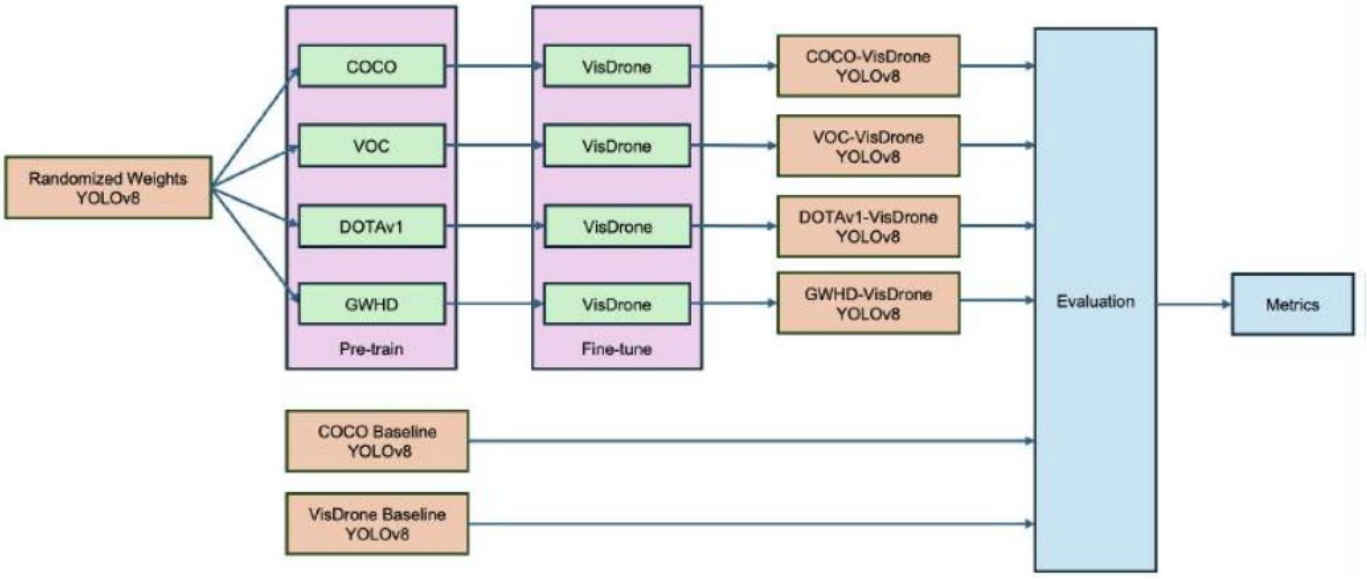
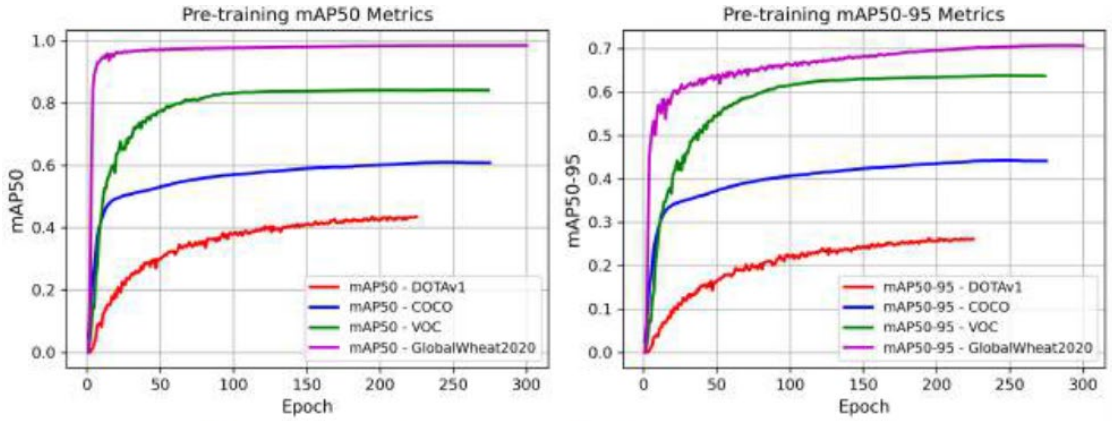


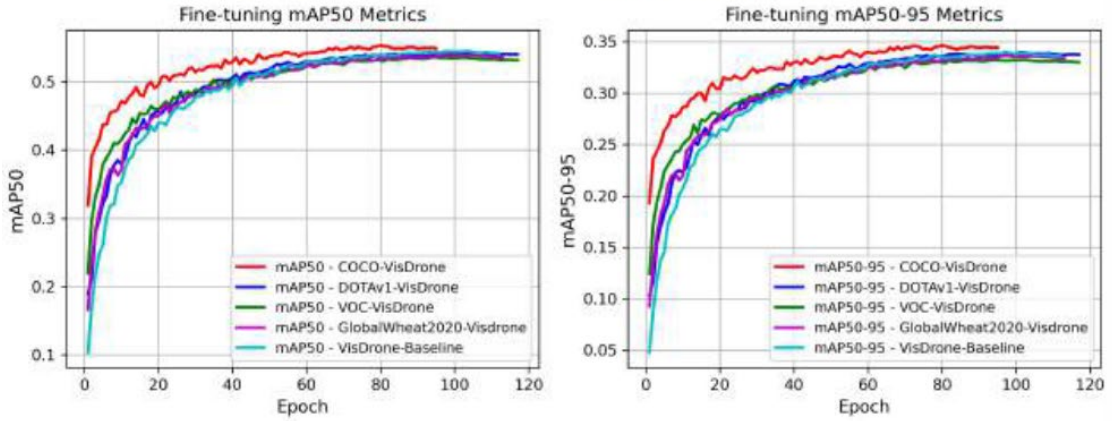
Figure 2. Pre-training and fine-tuning procedure to develop object detection models

Initial pre-training parameters were executed at a resolution of 736 × 736 pixels for 300 epochs utilizing Adam optimizer paired with a cosine learning rate scheduler. Downstream fine-tuning on the target VisDrone (Zhu, et al., 2022) database was conducted at an elevated layout of 1280 × 1280 pixels for 150 epochs. The empirical training histories manifested classic transfer learning contours. Specifically, domain adaptation prompted a sharp, immediate drop in composite training loss within the first 25 epochs, yielding an asymptotic stabilization of mAP metrics through epoch 150. Figure 3 shows pre-training and fine-training mAP metrics on different datasets.

To expose fine-grained performance drift, the proposed FIE framework partitions a continuous operational video stream into non-overlapping temporal windows (fixed at 30 frames). Spatial and categorical tracking errors are logged frame-by-frame based on formal threshold criteria, IoU, as specified in Table 1.



(a) Pre-training mAP metrics across different datasets over 300 epochs.



(b) Fine-tuning mAP metrics on the VisDrone dataset using different pre-training sources.

Figure 3. Pre-training and fine-training mAP metrics on different datasets

Table 1. Validation criteria for object detection prediction errors

IOU $\geq \tau_{iou}$	**Class Match**	**Outcome**
Yes	Yes	Correct Detection
Yes	No	Prediction Error - misclass
No	No	Prediction Error – low IoU
No	Yes	Prediction Error – low IoU

Summing these frame errors inside each discrete interval yields three independent, normalized tracking indicators: the Missed Detection Rate, the Spatial Localization Error Rate, and the Categorical Misclassification Rate. The operational validation utilized real-world vertical descent imagery extracted from a DJI Mini 3 drone descending over a suburban site is shown in Figure 4. This example annotated frame from a real-world flight test video contains a high-contrast orange

landing pad populated by four pedestrians and a vehicle (600 frames total at 30 frames per second). Under static, macro-level benchmarking, the variants converged within a narrow 1.3% variance margin, with the Common Objects in Context (COCO) pre-trained model securing the highest aggregate score (mAP@50 of 0.98356), as detailed in Table 2 and Table 3.



Figure 4. Real-world flight test video with orange landing pad

Table 2. Optimal fine-tuning performance across architecture variations

Model Architecture Variant	Speed (ms)	mAP@0.50	mAP@0.50:0.95
COCO-Baseline (No Fine-Tune)	4.8	0.50456	0.35167
VisDrone-Baseline (From Scratch)	4.4	0.95881	0.76121
COCO-VisDrone	4.7	0.98356	0.78956
VOC-VisDrone*	4.6	0.98137	0.77161
DOTAv1-VisDrone*	4.3	0.97096	0.78492
GlobalWheat2020-VisDrone	4.6	0.96977	0.77572

**Visual Object Class (VOC)*
** Dataset for Object Detection in Aerial Images (DOTA)*

Table 3. Aggregate performance indicators on flight test dataset

Model Architecture Variant	Test Set mAP@0.50	Test Set mAP@0.50:0.95
VisDrone-Baseline	0.54510	0.34024
COCO-VisDrone	0.55311	0.34628
VOC-VisDrone	0.53603	0.33248
DOTAv1-VisDrone	0.54288	0.33963
GlobalWheat2020-VisDrone	0.53955	0.33656

Analyzing the descent sequence via the FIE metrics exposes performance anomalies that typical accuracy metrics do not capture. Models initialized on domain-restricted data, such as the satellite-based DOTAv1 (Xia, et al., 2018) or agricultural Global Wheat Head Detection (GWHD) (David, et al., 2020), exhibited severe spikes in continuous missed detections and low-IoU localization dropouts as the UAS descended. Pre-training on hyper-dense, low-context, or strictly top-down imagery restricts the network's internal feature representation space, degrading its capacity to track rapid scale and perspective shifts. Conversely, models initialized on high-diversity repositories (Microsoft COCO (Lin, et al., 2014) and Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes (PASCAL VOC) (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010) demonstrated improved temporal resilience, confirming that broad initial background clutter is vital to formulate robust regression features. Figure 5 represents the evaluation or FIE metric across all YOLOv8s models.

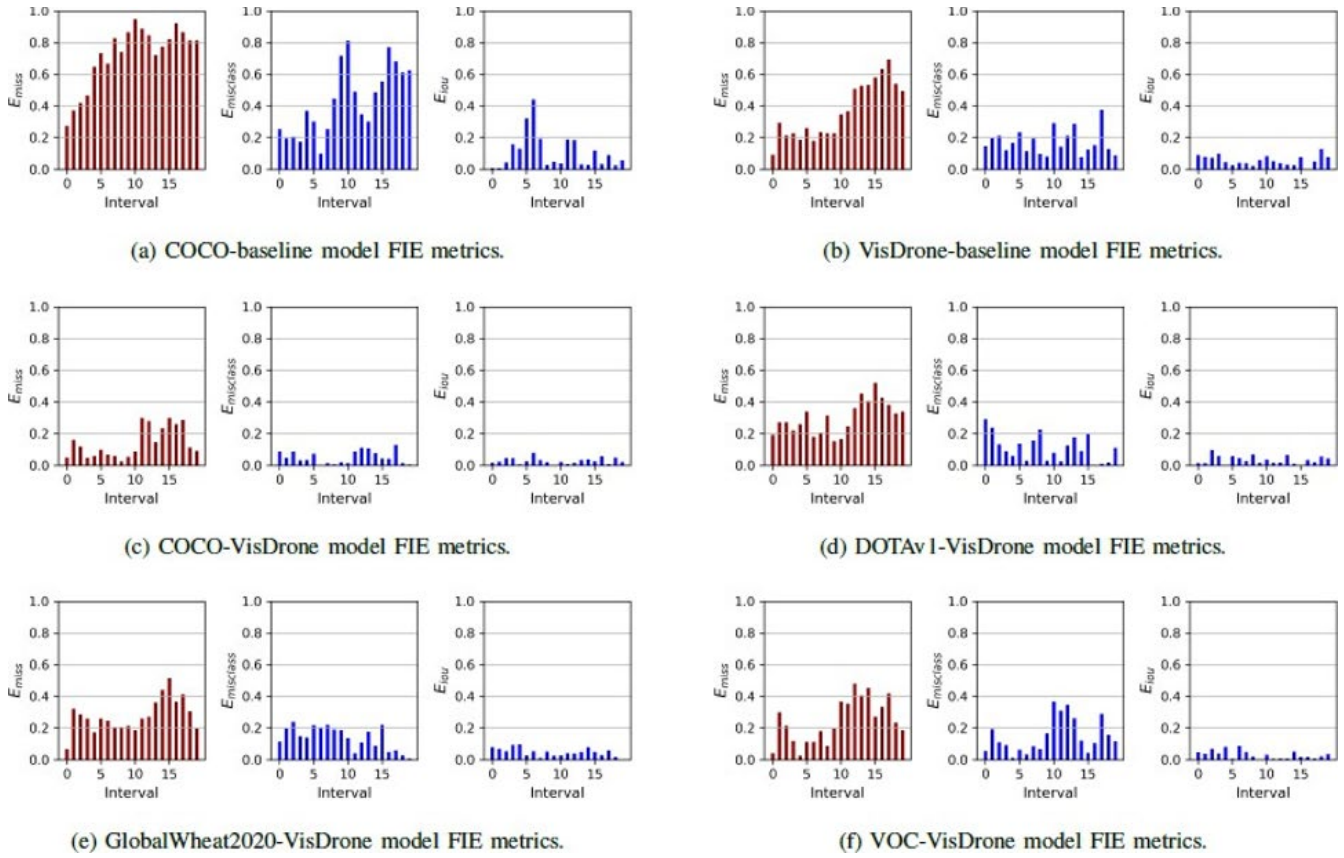


Figure 5. Proposed FIE metrics for all YOLOv8s models evaluated on flight test dataset

3.1.2 Task 1.2: Verifying complete ODD coverage of training dataset

To mathematically quantify the statistical mismatch between training data distributions and ODDs, this study establishes a rigorous framework rooted in OT theory. When evaluating high-dimensional empirical distributions, datasets are projected into semantic embeddings generated by a pre-trained Contrastive Language-Image Pre-Training (CLIP) (ViT-B/32) encoder, mapping images into a latent space aligned with human-interpretable concepts. In Figure 6, conceptual visualization of data distributions in latent space is represented. Each point represents an image sample projected using a CLIP (ViT-B/32) encoder. The red cluster corresponds to the training set A (COCO), the orange cluster to the training set B, and the blue cluster to the ODD samples. The spatial proximity between VisDrone (orange) and ODD (blue) indicates a lower distributional mismatch compared to COCO (red), which has improved performance of models trained on VisDrone when evaluated on ODD data.

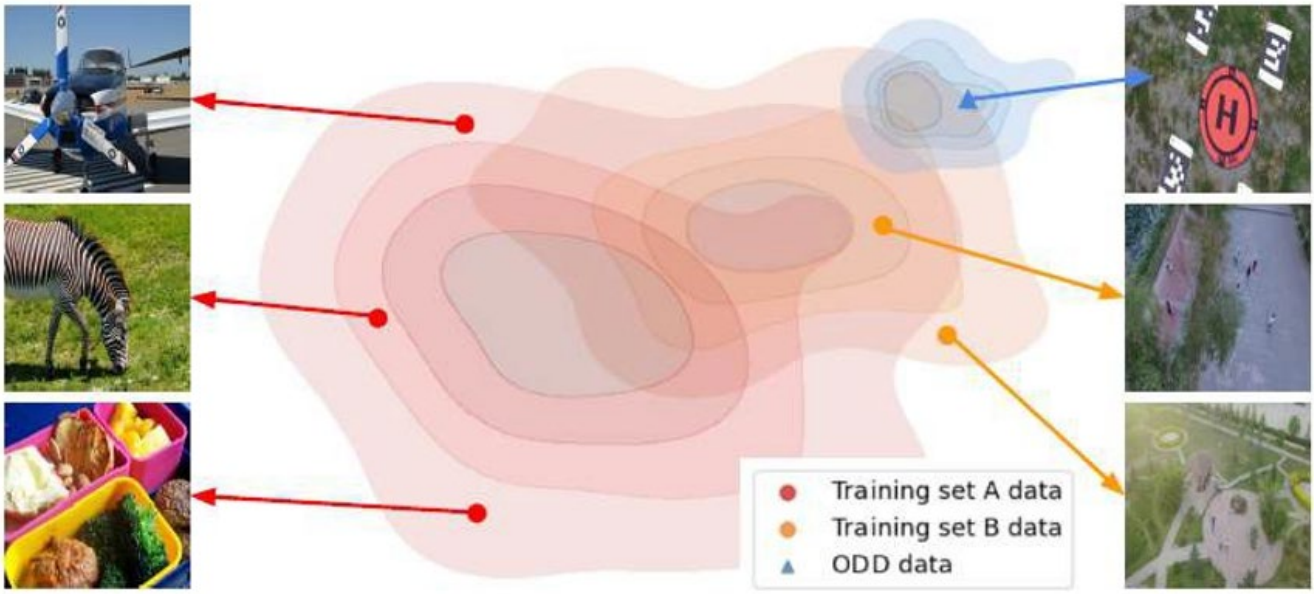


Figure 6. Conceptual visualization of data distribution in latent space

When the cost function is defined by spatial displacement, it leads to the p-Wasserstein distance. The 1-Wasserstein distance is robust to small perturbations and preserves mass movement over long distances, making it suitable for applications emphasizing spatial shifts. The 2-Wasserstein distance penalizes larger displacements more heavily and emphasizes variance-like effects. The source distribution (in red) and the target distribution (in blue) are visualized in Figure 7.

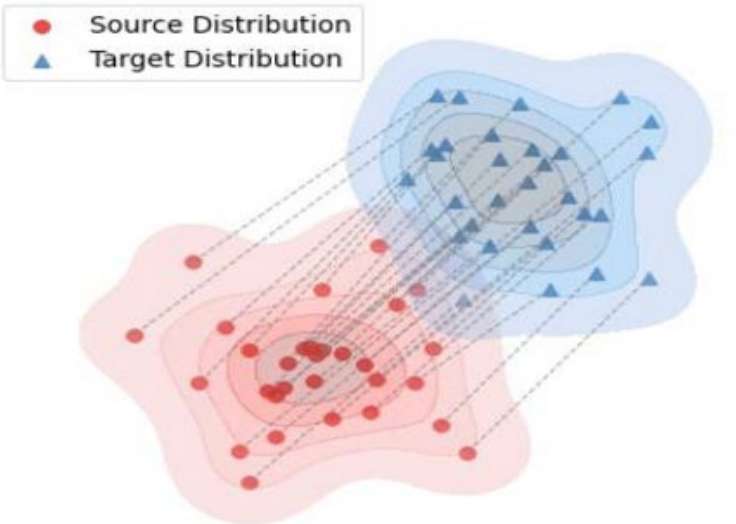


Figure 7. OT between source and target distribution

Dashed gray lines represent the optimal couplings that minimize the transport cost between individual samples from the source to the target distribution. We compute the p-Wasserstein distance between the source training distribution and the target ODD tracking arrays. The 1-Wasserstein distance is highly robust to localized mass shifts, whereas the 2-Wasserstein metric penalizes large outliers severely, capturing variance-like transformations. To model edge-case samples when physical operational data is scarce, the pipeline deploys Denoising Diffusion Probabilistic Models (DDPMs) leveraging a U-Net denoising backbone, as depicted in Figure 8, with explicit skip connections. Figure 9 shows the visualization of the denoising process in a DDPM. The generative model synthesizes high-fidelity ODD imagery from pure Gaussian noise through a learned reverse Markov chain. Image synthesis quality and structural diversity are rigorously validated against real drone flight captures using the Fréchet Inception Distance (FID) score, which maps the 2-Wasserstein distance across deep feature spaces.

The 1-Wasserstein (W_1) and 2-Wasserstein (W_2) distances and subsequent model precision boundaries computed across the Microsoft COCO and VisDrone source setups are documented in Table 4 and Table 5.

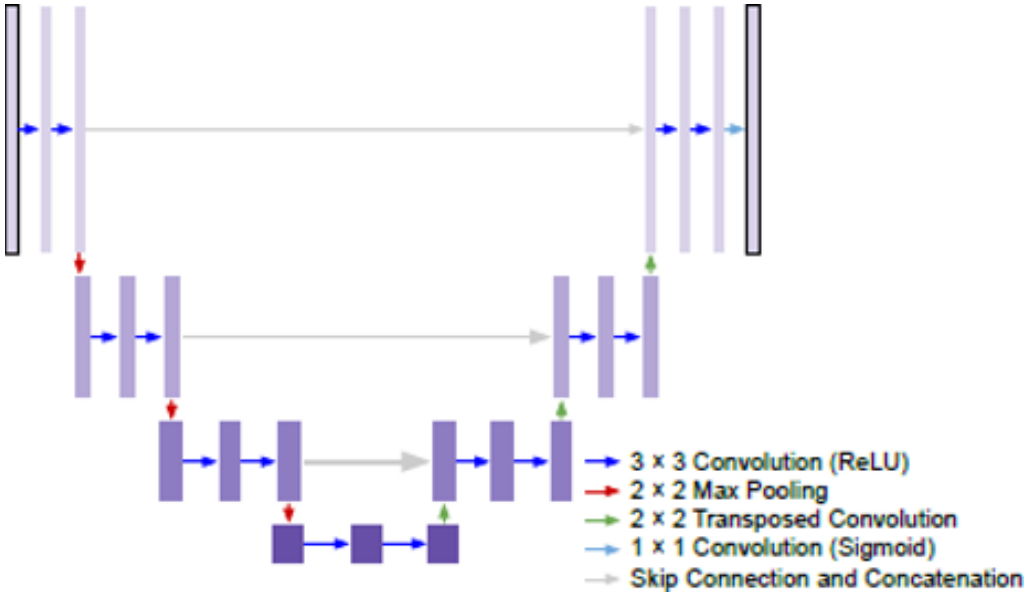


Figure 8. U-Net architecture used as denoising backbone in diffusion model



Figure 9. Visualization of denoising process in DDPM

Table 4. Wasserstein transport mismatch to target ODD distribution

Representational Format	**Source Training Dataset**	**1-Wasserstein (W_1) Distance**	**2-Wasserstein (W_2) Distance**
Raw Pixel Arrays	COCO	26.585	27.774
Raw Pixel Arrays	VisDrone	20.583	22.794
CLIP Latent Embedding	COCO	0.904	0.853
CLIP Latent Embedding	VisDrone	0.761	0.625

Table 5. Downstream YOLO performance map on real ODD target frames

Source Training Dataset	**Operational ODD mAP@0.50**	**Operational ODD mAP@0.50:0.95**
Microsoft COCO	50.456%	35.167%
VisDrone Aerial Set	95.881%	76.121%

The VisDrone source dataset secured significantly tighter Wasserstein distances to the operational envelope than COCO, directly correlating with a massive boost in detection accuracy (95.881% vs 50.456% mAP@50). This validates the core hypothesis that a higher statistical transport mismatch directly forecasts severe performance degradation upon deployment.

Figure 10 presents synthetic images produced by our diffusion model at a resolution of 64 × 64. The generated samples resemble aerial views featuring helipad markings and surrounding terrain,

closely aligning with the visual characteristics of the ODD dataset. This indicates that the model has successfully captured the underlying data distribution.



Figure 10. Synthetic images produced by diffusion model at 64 × 64 resolution

The down-sampled and generated synthetic images are directly utilized to validate downstream perception models. The trained object detection model successfully detects critical objects, such as landing pad markings and dynamic obstacles, within these synthetic images. The successful detection, as depicted in Figure 11, shows the utility of diffusion-generated data as a viable mechanism for testing edge-case deployment scenarios without risking physical hardware in the field.

An illustration of the interpolation effect of synthetic data in the ODD domain is shown on the right of Figure 11. Solid and hollow blue triangles represent real and synthetic data points, respectively, linked to their corresponding images. To evaluate spatial resolution boundaries, the diffusion engine synthesized data blocks across three distinct layouts, tracking their visual realism via FID metrics as outlined in Table 6.



Figure 11. Sample images from diffusion generation process with detections

Table 6. Generative diffusion realism mapping across resolutions

Synthetic Image Resolution (Pixels)	Fréchet Inception Distance (FID) Score
64 × 64	180.532
128 × 128	139.729
256 × 256	57.294

Expanding the target spatial scale strongly optimizes output quality, with the 256 × 256 configuration achieving an FID score of 57.294. Augmenting the sparse real flight paths with 3,000 high-fidelity diffusion-generated frames bridges representation gaps, yielding lower 1-Wasserstein (W_1) and 2-Wasserstein (W_2) distance values across both raw pixel formats and semantic CLIP embeddings compared to the unaugmented baseline. The resulting Wasserstein distances of the augmented datasets are given in Table 7.

Table 7. Wasserstein distances following synthetic ODD augmentation

Representational Format	Source Training Dataset	Augmented W_1 Distance	Augmented W_2 Distance
Raw Pixel Arrays	COCO	23.332	25.654
Raw Pixel Arrays	VisDrone	18.458	20.643
CLIP Latent Embedding	COCO	0.847	0.813
CLIP Latent Embedding	VisDrone	0.641	0.532

3.2 Thrust 2: Providing confidence bound and uncertainty quantification of ML/AI model

3.2.1 Importance of uncertainty quantification

Modern computer vision systems achieve impressive performance on image classification, object detection, segmentation, and multimodal reasoning tasks. However, predictive accuracy alone is insufficient in safety-critical applications. Autonomous vehicles, aviation systems, medical imaging platforms, and surveillance systems must also understand when their predictions may be unreliable. Traditional approaches include Bayesian inference, Gaussian processes, confidence intervals, variance estimates, and ensemble methods. Conformal prediction complements these approaches by converting heuristic uncertainty measures into statistically valid guarantees.

3.2.2 Task 2.1: Conformal prediction foundations in computer

Conformal prediction is a mathematically rigorous, model-agnostic meta-algorithm that converts heuristic uncertainty estimates into distribution-free, finite-sample prediction regions. Given a user-specified error tolerance $\alpha \in [0,1]$, CP constructs a dynamic prediction set $T(X)$ satisfying the strict coverage guarantee:

$$P\big(Y \in T(X)\big) \geq 1 - \alpha \qquad 1$$

For example, α = 0.10 corresponds to 90% coverage. The central idea is to compute nonconformity scores that measure how unusual an example appears relative to the calibration set.

The original classification CP Algorithm is calculated in three steps as follows:

- **Step 1:** Compute nonconformity scores on a calibration set using $s_i = 1 - \hat{\pi}_{y_i}(x_i)$ where $\hat{\pi}_{y_i}(x_i)$ is the softmax probability assigned to the true class
- **Step 2:** Compute the empirical quantile $\hat{q} = Quantile_{(\lceil (n+1)(1-\alpha) \rceil / n)}(s_1, \ldots, s_n)$
- **Step 3:** For a new image x_{test} form the prediction set $C(x_{test}) = \{ y: \hat{\pi}_y(x_{test}) \geq 1 - \hat{q} \}$

The resulting prediction set may contain one or multiple labels. Ambiguous images produce larger sets, while easy images often produce singleton sets.

A major strength of conformal prediction is its exact finite-sample guarantee:

$$1 - \alpha \leq P\big(Y \in T(X)\big) \leq 1 - \alpha + 1/(n+1) \qquad 2$$

This result holds regardless of model architecture, dataset complexity, or calibration set size, assuming exchangeability between calibration and test samples. Unlike asymptotic confidence intervals, CP remains valid even for relatively small datasets.

3.2.3 Conformal prediction application to computer vision

Conformal prediction is best viewed as a meta-algorithm. Any uncertainty score can be transformed into a valid uncertainty estimate through calibration. The generalized framework consists of:

1. Define a nonconformity score $s(x, y)$.
2. Compute calibration scores.
3. Estimate a quantile threshold $\hat{q}$.
4. Construct prediction sets satisfying $s(x, y) \leq \hat{q}$.

This flexibility has led to numerous variants including Adaptive Prediction Sets (APS), Regularized Adaptive Prediction Sets (RAPS), Conformalized Quantile Regression (CQR), Risk-Controlling Prediction Sets (RCPS), and Distribution-Shift-Aware CP.

3.2.3.1 Image classification

CP is most mature in classification tasks. Instead of predicting a single class, the model outputs a set of plausible classes. This significantly improves reliability under ambiguity and class overlap.

3.2.3.2 Object detection

Conformal methods can provide uncertainty-aware bounding boxes, confidence regions, and object-level guarantees. Recent approaches calibrate localization errors and detection confidence scores.

3.2.3.3 Semantic segmentation

Pixel-wise predictions can be calibrated to produce guaranteed coverage maps. These methods help identify uncertain regions in medical images, satellite imagery, and autonomous driving scenes.

3.2.3.4 Instance segmentation

The slide deck highlights conformal risk control for instance segmentation. Key metrics include:

- Recall Risk
- IoU Risk
- Coverage Risk

Rather than guaranteeing only classification accuracy, these approaches guarantee performance metrics directly relevant to segmentation quality.

3.2.3.5 Vision-language models (VLMs)

Recent work extends CP to multimodal foundation models such as CLIP, GPT-4V, and other vision-language architectures. Prediction sets can quantify uncertainty in image captioning, visual question answering, and retrieval systems.

3.2.4 Task 2.2: Verify confidence bound of classification uncertainty

This technical task evaluates and compares the lightweight, anchor-free CNN-based YOLOv8s architecture (Figure 12) against the transformer-based RT-DETR. While YOLOv8s balances bounding box regression using a PANet neck configuration, it encounters structural calibration limitations under high occlusion and relies heavily on Non-Maximum Suppression (NMS) blocks. RT-DETR resolves the NMS latency bottleneck by applying an Efficient Hybrid Encoder composed of an Attention-based Intra-scale Feature Interaction (AIFI) module and a Cross-scale Feature Fusion (CSFF) layer, paired with an Uncertainty-Minimal Query Selection design.

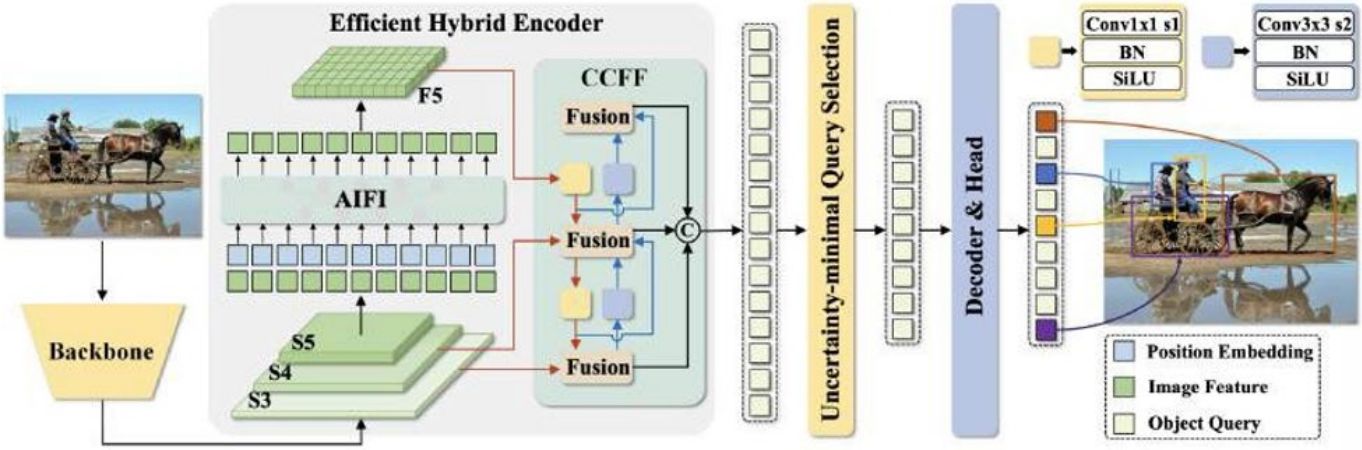


Figure 12. System block diagram of RT-DETR end-to-end object detection model

Aerial imagery captured by UAS often contains multiple, densely packed, and variably scaled objects, making accurate classification under uncertainty a critical task. Conformal prediction resolves this by generating statistically valid prediction sets, rather than single point predictions of typical Neural Networks detecting object classes. If the model is confident, the set shrinks to a single label; if uncertain, the set expands to include multiple candidate classes, ensuring that the true label is captured within a user-specified marginal coverage probability (or confidence) of $1 - \alpha$ (α is also referred to as the error tolerance or error level).

In this work, CP is executed as follows. During the initial calibration phase, the validation split of the VisDrone dataset acts as the calibration set. For each prediction, a softmax-based nonconformity score is calculated using the formula $s_i = 1 - \hat{f}(X_i)_{Y_i}$ which measures the model's lack of confidence in the true ground-truth label. This is followed by threshold calculation, where, based on a predefined error tolerance $\alpha \in [0,1]$, an empirical quantile threshold $\hat{q}$ is computed from the calibration nonconformity scores at the $\lceil (n+1)(1-)/n \rceil$ level. Finally, during inference on new testing data, a candidate class label is retained in the final prediction set only if its predicted score satisfies the conformal constraint $\hat{f}(x_{test})_y \geq 1 - \hat{q}$. This process yields conformal prediction sets with a theoretical marginal coverage of at least $1 - \alpha$. Estimated distributions of the class prediction set sizes on the VisDrone test dataset are presented in Figure 13.

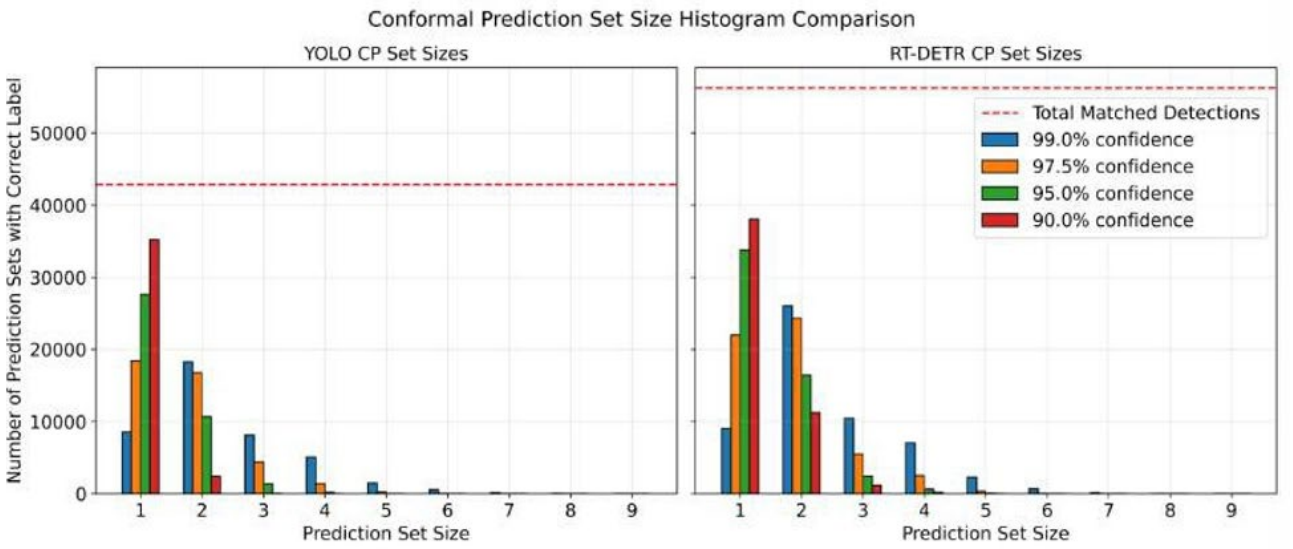


Figure 13. Estimated distributions of class prediction set sizes on VisDrone test dataset

Four values of α were specified to observe the changes to the prediction set size distribution for each model. As α decreases, the confidence level increases, resulting in larger prediction sets to maintain the specified coverage. Both models were trained using the VisDrone2019-DET dataset, consisting of 2.6 million bounding boxes across ten urban object classes captured from high-altitude aerial vantage points. To ensure fair comparative benchmarking, models were initialized with COCO pre-trained weights and fine-tuned under standardized hyperparameters. Input images were standardized to a resolution of 720 × 1280 pixels. Optimization was driven by the AdamW algorithm with an initial learning rate of 0.0001, combined with a cosine learning rate scheduler. To prevent overfitting, early stopping was enforced with a 15-epoch patience threshold. Processing speed metrics and real-time inference latency breakdowns are compiled in Table 8.

Table 8. End-to-end processing speed breakdown per frame

Model Type	**Pre-processing (ms)**	**Inference Latency (ms)**	**Post-processing (ms)**	**Total Runtime (ms)**
YOLOv8s (CNN)	9.2	186.4	5.8	201.4
RT-DETR (Transformer)	9.5	1252.3	0.5	1262.3

Although RT-DETR achieved a superior standalone precision rating (mAP@50 of 0.5270 vs 0.4914 for YOLOv8s), its complex global self-attention layers heavily penalized computational efficiency, inducing an inference lag of 1262.3 ms per frame. However, conformal calibration routines demonstrated that the transformer architecture generates exceptionally well-calibrated and reliable internal estimates of its own uncertainty compared to the CNN, as documented in Table 9 (total detections: RT-DETR=56,253; YOLO=42,850).

Table 9. Conformal prediction set statistics

Target Error Level (α)	RT-DETR Total Sets	RT-DETR Mean Set Size	RT-DETR Coverage (%)	YOLOv8s Total Sets	YOLOv8s Mean Set Size	YOLOv8s Coverage (%)
0.10 (90% Confidence)	50,675	1.28	90.1%	37,657	1.07	87.8%
0.05 (95% Confidence)	53,282	1.44	94.7%	39,930	1.35	93.2%
0.025 (97.5% Conf.)	54,708	1.81	97.3%	41,284	1.75	96.3%
0.01 (99% Confidence)	55,648	2.47	98.9%	42,220	1.41	98.5%

At the 90% target coverage limit ($\alpha = 0.1$), RT-DETR achieved a precise 90.1% empirical coverage with a mean set size of 1.28 classes. Conversely, YOLOv8s failed to reach the safety bound, outputting an under-calibrated empirical coverage of 87.8% due to systematically overconfident softmax layers. This establishes that while YOLOv8s remains the optimal option for edge platforms lacking hardware acceleration, RT-DETR delivers superior statistical reliability for safety-critical aviation loops.

3.3 Thrust 3: Speeding up verification by leveraging synthetic data generation

3.3.1 Task 3.1: Synthetic data generation for computer vision based autonomous landing

Beyond Visual Line of Sight (BVLOS) operations authorized by the FAA emphasize the critical need for robust autonomous validation setups. Physical flight tracking loops introduce operational bottlenecks due to geographical restrictions, weather variability, and the

impossibility of safely capturing extreme edge-case states like dense dust, low-light fields, or severe air pollution. To bypass these limitations, this task evaluates a parameterized synthetic data pipeline utilizing Microsoft AirSim inside a customized Unreal Engine 4.27 simulation framework as depicted in Figure 14. The structural environment model maps kinematics, sensor logs, and lighting directly through programmatic Python Application Programming Interface (API) connections and localized Remote Procedure Calls (RPCs).

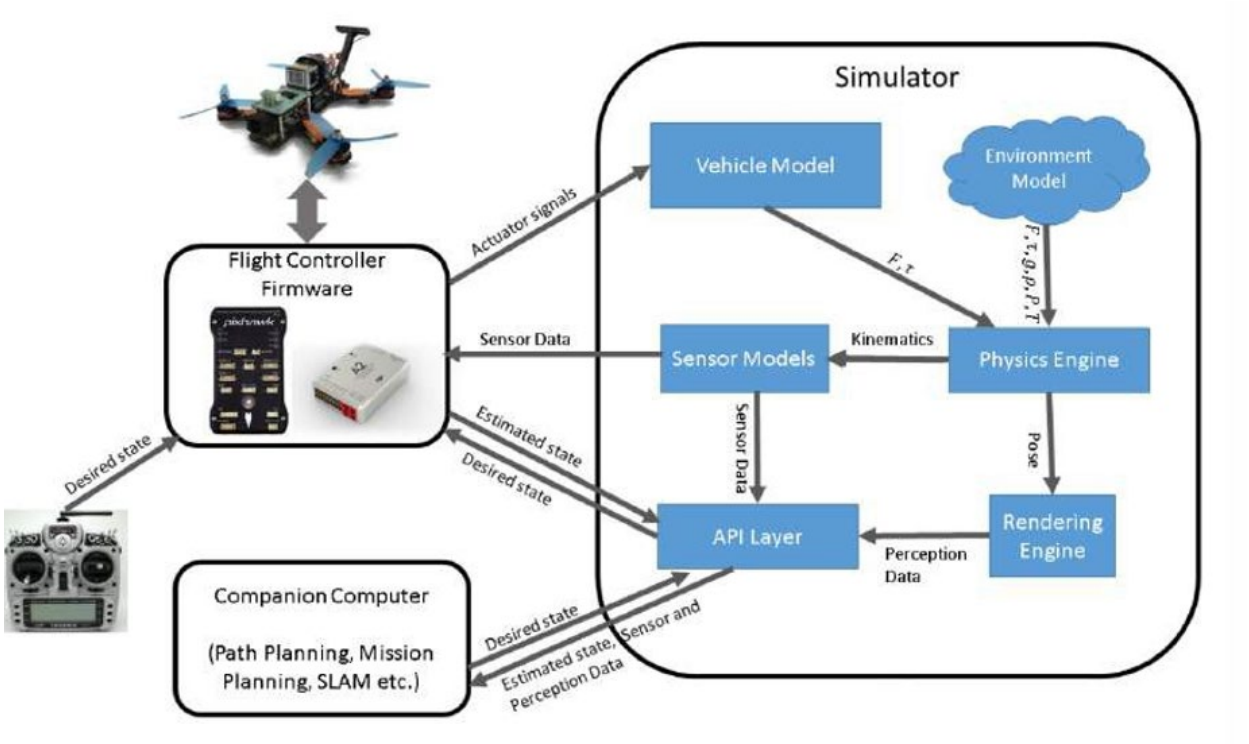


Figure 14. Architecture and data flow of AirSim simulation framework

Qualitative evaluation of the fine-tuned YOLOv8s model across clear daytime synthetic trials revealed excellent identification of vehicular assets and ArUco pads. However, critical edge-case failure modes were exposed: severe contrast-based tracking degradation occurred under foggy weather parameters (rendering shown in Figure 15), and structural pedestrian tracking collapsed during high-altitude nighttime park operations. As detailed by the precision-recall indicators, the "car" class maintained a precision score of 0.936, while the "pedestrian" tracking precision dropped to an exceptionally weak 0.197. Across the entire compiled synthetic data framework, the model achieved an overall mAP@50 of 0.567 and an aggregate mAP@50-95 of 0.390, underscoring the critical utility of synthetic generation in exposing hidden vision vulnerabilities before deployment.

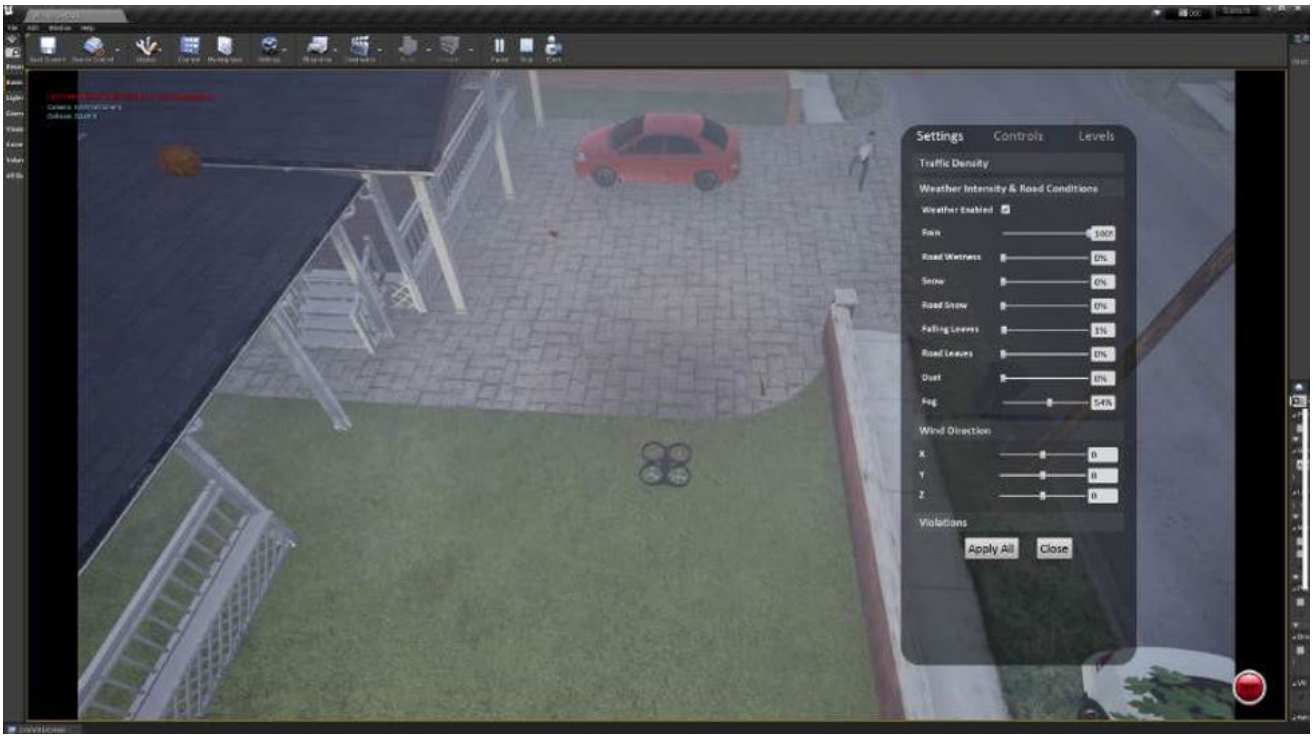


Figure 15. Rendering of front yard landing zone with weather API implementation

3.3.2 Task 3.2: Scaling flight simulation and data synthesis with generative 3D reconstruction

To automate environment creation and remove the bottleneck of manual 3D asset modeling, this task implements an end-to-end scalable data synthesis pipeline integrating generative AI directly within an Unreal Engine 5 simulation loop via Cosys-AirSim. The architecture extracts precise 2D object segmentations from real-world datasets –including vehicular variants sourced from the Stanford Cars Dataset (Krause, Deng, Stark, & Fei-Fei, 2013), architectural distributions from the SoCal Houses Dataset (Kaggle, 2019), and diverse target instances from the Open Images Dataset (Kuznetsova, et al., 2020)—utilizing the Segment Anything Model 3 (SAM 3) foundation model to yield clean Red, Green, Blue, Alpha (RGBA) masks. These targets are processed through a generative 3D reconstruction pipeline to synthesize structurally accurate 3D mesh assets with detailed texturing from a single viewpoint. Computational throughput and mesh scaling metrics tracking dataset size are illustrated in Figure 16 and Figure 17. Figure 16 details how synthetic data generation workflow integrates generative AI with high-fidelity simulation to automate the production of labeled datasets.

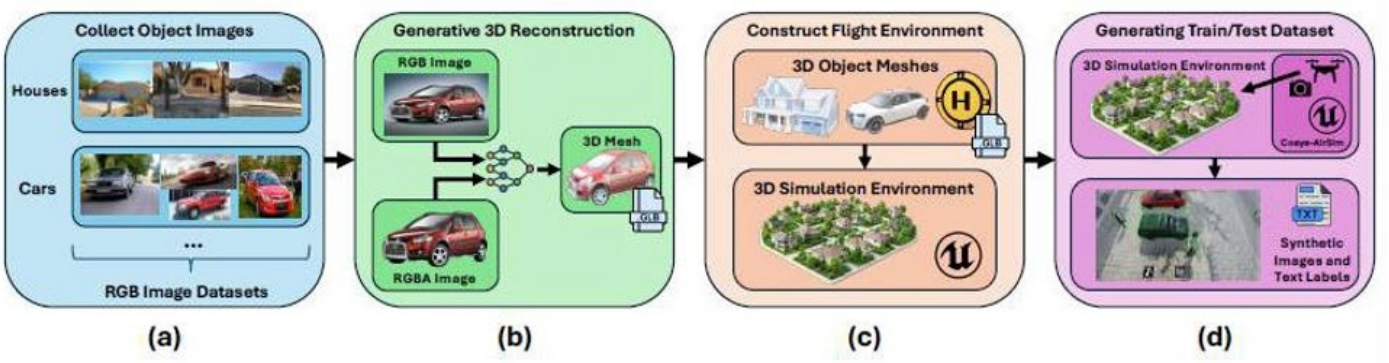


Figure 16. Synthetic data generation workflow

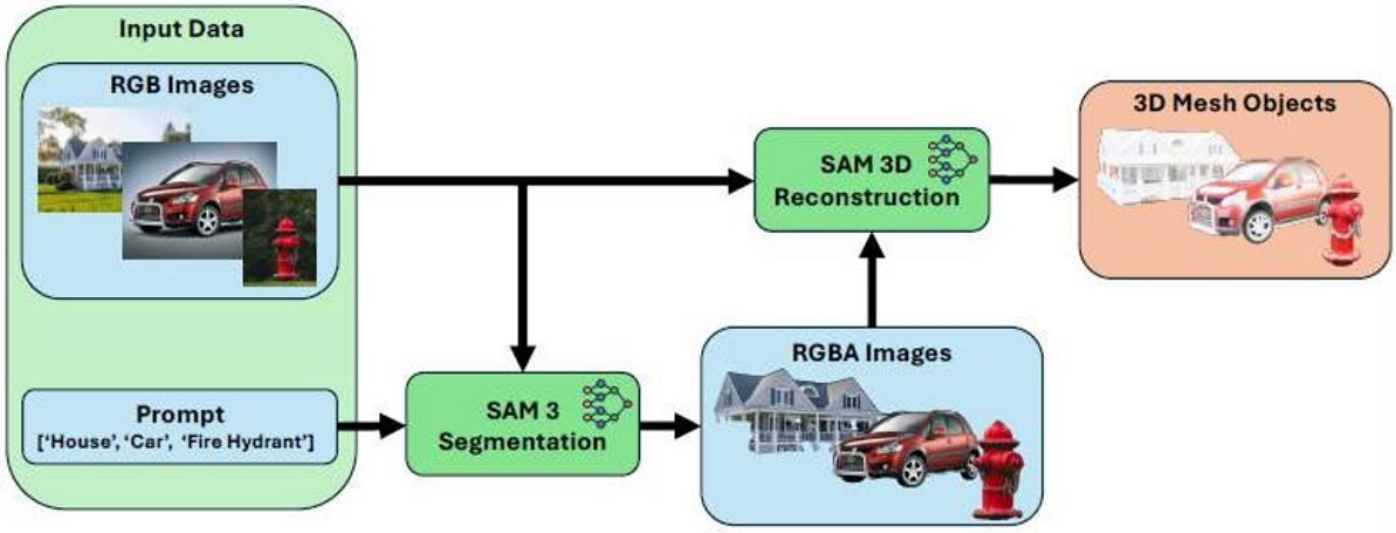


Figure 17. Generative mesh reconstruction pipeline using SAM 3D

To eliminate bounding box drift caused by standard 3D-to-2D projection overlays, an automated two-stage segmentation-based label refinement layer was integrated. The algorithm processes initial simulator bounding box outputs, queries instance segmentation tracking channels, and applies precise color masking to filter out highly occluded pixels. Quantitative performance of this label refinement module across 10,000 generated flight tracking images is compiled in Table 10.

Table 10. Automated label refinement metrics by target asset class

Object Class Name	Label Deletion Rate % (Count)	Internal Pixel Density Change (Unrefined → Refined)	Pixel Containment Mapping (Unrefined → Refined)
people	2.1% (1084)	0.24 → 0.48	0.93 → 0.99
car	2.7% (950)	0.55 → 0.74	0.99 → 1.00
truck	1.8% (137)	0.54 → 0.79	0.98 → 1.00
bus	1.5% (63)	0.47 → 0.64	0.97 → 1.00
trash bin	0.4% (133)	0.60 → 0.80	1.00 → 0.99
landing pad	0.1% (6)	0.74 → 0.78	1.00 → 1.00

The refinement layer successfully optimized background noise reduction without truncating true silhouette boundaries. For the "people" class, internal target pixel density effectively doubled (from 0.24 to 0.48) while purging 1,084 fully occluded label boxes, establishing a highly accurate automated training loop.

4 Dataset collected and generated

4.1 Real world dataset collection and ODD definition

To robustly evaluate perception systems in representative configurations, the team structured a comprehensive data curation pipeline using the Roboflow platform. Real-world flight data streams were captured via a DJI Mini 3 drone executing vertical descent trajectories over residential and outdoor topographies in Virginia, logging varied roof layouts, landscaping, dynamic shadows, and obstacles from multiple altitudes. Redundancy was mitigated using an OpenCV sampling script to extract every 10th video frame. Manual bounding box tracks were drawn around key elements, followed by auto-orientation filters and resolution standardization to a uniform 640 × 640-pixel canvas. To prevent model overfitting, three unique augmented variations were programmatically generated per frame using constrained horizontal/vertical flips and controlled rotation sweeps bounded between −15° and +15°.

4.2 Synthetically generated datasets

Large-scale synthetic datasets were compiled using Microsoft AirSim integrated with Unreal Engine 4.27 and the updated Cosys-AirSim fork on Unreal Engine 5. Environments focused on

suburban residential front yards mimicking driveway landing zones, alongside high-density neighborhood parks featuring complex vehicular trajectories and walking splines. Atmospheric sliders enabled continuous density modulation across fog, snow, rain, and solar exposure limits, providing an expansive edge-case training matrix.

5 Conclusion

Upon completing this 24-month research initiative, the prominent project contributions and empirical findings are summarized as follows:

- Pre-training dataset selection heavily shifts downstream computer vision capabilities, proving that perception validation profiles must evaluate both the training distribution data and the model architecture as a combined system.
- Integrating OT distances across semantic CLIP latent spaces provides a highly accurate mechanism to forecast model accuracy degradation prior to active physical deployment.
- Conformal prediction successfully delivers statistically valid uncertainty quantification bounds across both CNN and transformer stacks, exposing critical internal calibration differences.
- Parametric simulation via AirSim and Unreal Engine 5 remains vastly superior to existing end-to-end generative video models (e.g., Sora or Movie Gen) for synthesizing physically consistent, controllable flight tracking paths.

Based on these insights, we recommend the following future research actions:

1. Continue exploring the formulation of advanced time-varying tracking metrics, alongside isolating the rigorous correlation boundaries mapping OT statistical mismatch to targeted model degradation coefficients.
2. Expand evaluation scopes to analyze multi-modal Vision-Language Models (VLMs) acting as core perception and decision modules under active operational strain.
3. Explore agentic AI and generative AI to scale the creation of parametric, controllable, and realistic Software-in-the-Loop (SITL)/Hardware-in-the-Loop (HITL) flight video generation and annotation.

6 References

David, E., Madec, S., Sadeghi-Tehran, P., Aasen, H., Zheng, B., Liu, S., . . . Guo, W. (2020). Global Wheat Head Detection (GWHD) Dataset: A Large and Diverse Dataset of High-Resolution RGB-Labelled Images to Develop and Benchmark Wheat Head Detection Methods. *Plant Phenomics, 2020*(2), 1-12. doi:10.34133/2020/3521852

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision, 22*(2), 303–338. Retrieved from https://doi.org/10.1007/s11263-009-0275-4

Kaggle. (2019). *House Prices and Images - SoCal*. Retrieved from Socal Houses Dataset: ted8080: https://www.kaggle.com/datasets/ted8080/house-prices-and-images-socal

Krause, J., Deng, J., Stark, M., & Fei-Fei, L. (2013). *Stanford Cars Dataset.* Retrieved from "Collecting a Large-scale Dataset of Fine-grained Cars": https://api.semanticscholar.org/CorpusID:16632981

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J. R., Krasin, I., Pont-Tuset, J., . . . Ferrari, V. (2020). The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision, 128*(4), 1956–1981. Retrieved from https://doi.org/10.1007/s11263-020-01316-z

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. (2014). Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Ed.), *Computer Vision – ECCV 2014* (pp. 740–755). Zurich, Switzerland: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-10602-1_48

Xia, G.-S., Bai, X., Zhang, L., Belongie, S., Luo, J., Datcu, M., . . . Ding, J. (2018). DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 3974-3983). Salt Lake City, UT, USA. doi:10.1109/CVPR.2018.00418

Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., & Ling, H. (2022). Detection and Tracking Meet Drones Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(11), 7380-7399. doi:10.1109/TPAMI.2021.3119563