Street2Air: A Framework for Synthesizing Aerial Vehicle Views from Ground Images

Md Rubel Ahmed[1], Fazle Rahat[2], M Shifat Hossain[2], Sumit Kumar Jha[3], Rickard Ewetz[3]
[1]Louisiana Tech University, [2]University of Central Florida, [3]University of Florida
Email: mahmed@latech.edu[1],{fazle.rahat,mshifat.hossain}@ucf.edu[2],{sumit.jha,rewetz}@ufl.edu[3]

***Abstract*—Annotated aerial view images are often missing from fine-grained vehicle type classification datasets. This lack of data limits both the accuracy and robustness of models when applied to top-down views, which are essential for applications such as autonomous drones and aerial surveillance. Models trained only on street-level images often fail to generalize to aerial perspectives, requiring more time and multiple observations to recognize vehicles accurately. In contrast, models trained with both street-level and aerial views can perform more reliably and with faster inference in drone-based systems. However, collecting real aerial data at scale can be costly and logistically challenging. In this paper, we propose AVA (Automated Aerial View Augmentation), a framework for aerial data augmentation via 3D asset generation and contextual scene synthesis. Since standalone 3D vehicle models from 2D images are not directly usable for detection, we embed them in realistic backgrounds to enable learning of both object features and scene context. AVA first constructs 3D vehicle models from street-view images. To ensure data quality, we introduce a realism checker that discards incomplete or distorted assets. We then apply geometric transformations to generate aerial 2D views. The 2D views pass through a text-to-video generator that adds background context, mimicking typical drone imagery. We evaluate our data augmentation approach by fine-tuning several object detection backbones. Notably, the pretrained YOLOv11 model, when fine-tuned with AVA augmented data, achieves a significant mAP@0.5 improvement from 0.06 to 0.51 in classifying previously unseen vehicles from aerial perspectives.**

***Index Terms*—annotation, aerial views, augmentation, 3D, vehicle-detection, Synthetic-dataset**

I. Introduction

Advancements in autonomous drones and aerial surveillance have enabled wide-scale monitoring for applications such as traffic management, disaster response, and urban planning. Vision models are increasingly used in such systems for vehicle detection and type identification. However, most models are trained on datasets composed of street-level images, which limits their effectiveness to familiar ground-based perspectives. When applied to aerial views where object appearance, scale, and surroundings vary significantly, their performance often degrades in fine-grained classification tasks. A key barrier to improving aerial model performance is the lack of high-quality annotated datasets, particularly those with fine-grained vehicle labels [1]. Collecting aerial data using drones or satellites can be logistically complex, and affected by environmental factors such as lighting and weather [2], [3]. In some regions, local regulations also restrict such data acquisition. In addition, capturing and annotating a diverse set of views across locations, altitudes, and vehicle types remains a labor-intensive process, which limits the development of reliable vision models for aerial imagery [4], [5].

The VisDrone [6] and VAID [7] datasets are widely used for aerial vehicle detection but remain limited in terms of vehicle type variety. Both datasets contain only a small number of predefined classes, such as car, van, and truck, without distinguishing between finer categories like sedan, SUV, or sports car. As a result, models trained on them struggle to recognize vehicle types that do not appear in the training set. This creates a scalability challenge for vehicle classification in drone-based systems where new or uncommon vehicles may appear frequently. To overcome this limitation, we propose an alternative workflow to conventional aerial data collection by generating synthetic aerial views from 2D street-level images. Instead of relying on drone or satellite imagery, our approach leverages recent advances in 3D reconstruction [8], [9], generative models [10], vision-language models [11], and 3D modeling tools [12]–[14] to create realistic 3D vehicle assets. Aerial views are projected from 3D assets with visually coherent backgrounds to simulate real-world imagery. To ensure quality, a realism checker filters out incomplete or deformed reconstructions. By augmenting standard datasets with synthetic aerial views derived from annotated street-level images, we aim to improve object detection robustness for fine-grained vehicle classification in top-down views. We present the following key contributions:

- We propose AVA, a framework that synthesizes aerial views from 2D street-level images to overcome the scarcity of annotated aerial datasets for fine-grained vehicle classification.
- AVA constructs realistic 3D vehicle assets and renders them from top-down perspectives, embedding them into coherent backgrounds to simulate drone-captured scenes. These synthetic aerial views expand the visual diversity of training data.
- To address the lack of benchmark aerial view ground-truth test data, we introduce a dataset creation method that uses open-source 3D vehicle models to generate ground truth aerial images. This enables a direct evaluation of AVA 's effectiveness by measuring model generalization on verified aerial views.

The paper is organized as follows: Section II presents a motivating case study; Section III reviews related work;

1946-0759/25/$31.00 ©2025 IEEE
DOI 10.1109/ICMLA66185.2025.00069

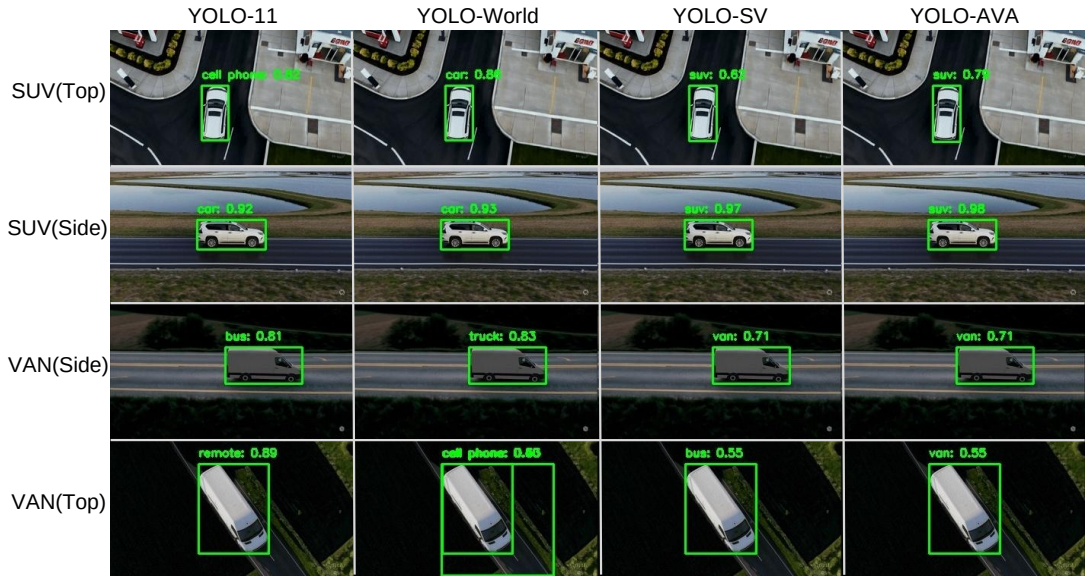

Fig. 1. From top-left YOLOv11 correctly detects the white SUV in street views but misclassifies its aerial view as a non-vehicle (cell phone) and similarly misidentifies the VAN as a bus. YOLO-World performs slightly better detecting the SUV but fails on the VAN top view. YOLO-SV (YOLOv11 fine-tuned on street views) still struggles with the aerial view of VAN. However, the model, YOLO-AVA fine-tuned with AVA's augmented synthetic aerial views correctly detects both the SUV and VAN from the aerial perspective, demonstrating that synthetic aerial views improves YOLOv11's generalization.

Section IV outlines our methodology; Section V reports experiments; Section VI discusses limitations; and Section VII concludes with future directions.

II. Case Study: Fine-Grained Vehicle Classification from Aerial Views

Most vehicle classification models are trained on datasets composed primarily of side or street-view (SV) images, which are commonly available. While this enables models to perform well on familiar perspectives, their accuracy drops significantly when applied to aerial views. This performance gap becomes especially evident in fine-grained classification tasks that require distinguishing between visually similar vehicle types, such as SUVs, vans, and sedans. To assess this limitation, we conducted a case study using YOLOv11 and YOLO-World [11] which are pretrained on MS COCO [15] and ImageNet [16]. Despite including a general "car" class, COCO lacks fine-grained distinctions, and ImageNet is not optimized for object detection. As shown in Fig. 1, both models perform reliably on street views but fail on aerial images misclassifying an SUV as a "cell phone" and a van as a "remote". YOLO-World offers slight improvement for SUVs but still fails to generalize.

To explore whether limited exposure to our dataset could help, we fine-tuned YOLOv11 using a small set of synthetic street-view images, resulting in YOLO-SV. While YOLO-SV improves performance on street-view images, it continues to struggle with aerial inputs, highlighting that viewpoint diversity is essential for generalization. We then fine-tuned YOLO-SV with synthetic aerial views generated by AVA to create YOLO-AVA. This model significantly improves performance on aerial views as shown in Fig. 1.

III. Related Works

Vehicle type classification has traditionally relied on large-scale datasets such as ImageNet [16] and MS COCO [15]. These datasets include general vehicle classes like car, truck, bus, motorcycle, and bicycle, but the images are mostly captured from street-level views. They lack aerial perspectives, which limits their use in drone-based surveillance or top-down monitoring applications. Large vision-language models (VLMs) [17] also suffer from limited angular coverage, which reduces their ability to identify vehicles from unfamiliar viewpoints. Some aerial vehicle detection datasets [18]–[20] do exist, but they are often small in size and mostly collected from stationary video feeds, such as traffic cameras or parked drones. As a result, they lack environmental diversity and sufficient coverage for fine-grained vehicle analysis.

Data augmentation using generative models has become an effective way to handle data scarcity. Methods such as GANs [21], Neural Radiance Fields [22], and Diffusion Models [23] have been used to synthesize object views in various conditions. Rendering tools like Blender [24] enable controlled simulation using 3D assets. CARLA [25] took this approach further by building an entire urban simulator for autonomous driving research. However, to the best of the authors knowledge, synthetic aerial view generation specifically for fine-grained vehicle classification remains underexplored. Our work addresses this gap by generating synthetic street and aerial views using modeling tool Blender and generation

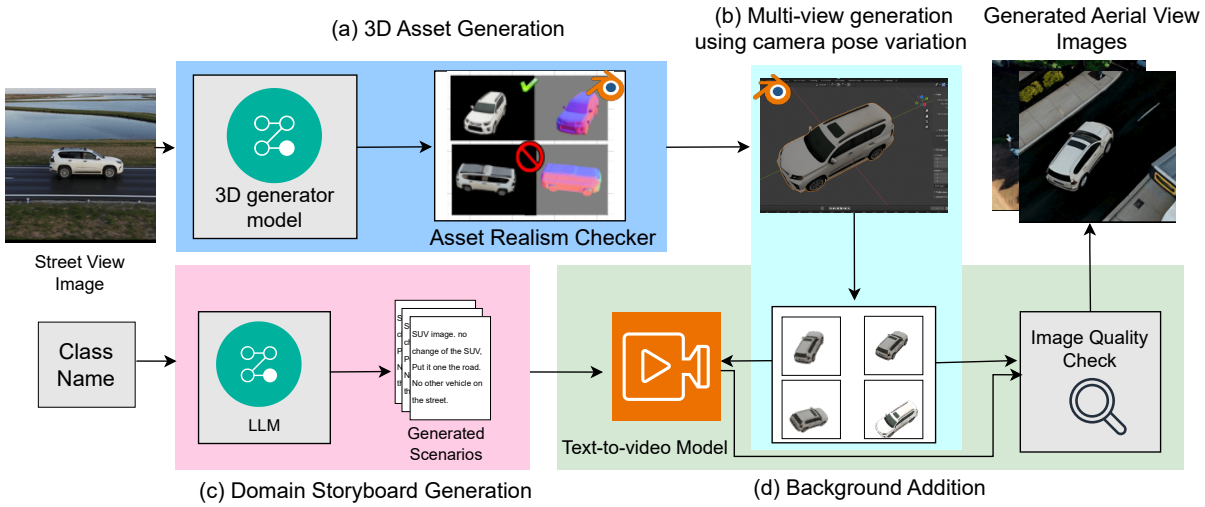

Fig. 2. Overview of the aerial view augmentation process using AVA. The method takes images from street view datasets and leverages Blender and 3D asset generation to synthesize corresponding backgrounds are applied to the images, and a realism checker verifies that the augmented aerial view images contain the correct vehicle within the frame.

tool TRELLIS [26], allowing models to learn from realistic drone-view perspectives. Besides, few works directly tackle vehicle classification from aerial views. Wang et al. [27] explored drone-based classification using RGB and LiDAR, while Chen et al. [28] studied satellite-based vehicle detection. Both identified major challenges, such as lack of labeled aerial data, altitude variance, and scale inconsistencies. AVA aims to tackle these challenges by generating synthetic aerial views leveraging 3D rendering tools and generative AI models.

IV. Automated Aerial View Augmentation

We present a scalable method for vehicle type detection dataset augmentation by synthesizing aerial views from street view images. We also introduce a mechanism for generating ground truth test dataset using open 3D assets to verify the generalization of models trained on the augmented datasets. The ground truths are essential since existing datasets, such as the Kaggle vehicle [29], lack annotated aerial views. The complete aerial view augmentation method is illustrated in Fig. 2. It starts by taking SV images as input to generate 3D assets through a realism-validated pipeline. The method incorporates domain storyboard generation using a LLM to define scenarios and integrates camera pose variations for aerial rendering. The output dataset comes in MS COCO format, consists of high-quality, consistent aerial view images validated through quality checks to ensure realism and correctness, enabling robust augmentation for fine-grained vehicle type classification tasks. A functional overview of AVA as follows.

a) 3D Asset Generation: The first stage of the AVA pipeline involves generating 3D vehicle assets from 2D street-view images. We use TRELLIS [26], an open-source 3D reconstruction tool available via the Hugging Face API, to convert single side-view images into textured 3D models. TRELLIS is chosen for its efficiency, simplicity, and ability to produce models in under one second, making it well-suited for large-scale data augmentation. The resulting 3D assets are exported in GLB format, which is compatible with common 3D processing tools such as Blender.

3D Pre-construction. For reliable reconstruction, input images should contain well-centered vehicles viewed from a horizontal side perspective. TRELLIS performs best when the object is in focus and not clipped at the image boundaries. While TRELLIS can automatically detect the main object in the image, we find that careful curation of input images improves reconstruction quality. Incorrect positioning or oblique angles often result in incomplete or distorted models, which cannot be used for generating realistic aerial views.

Realism Check. To filter out low-quality or structurally flawed assets, we introduce a two-step realism verification process. First, the GLB file is imported into Blender, and a synthetic street-view render of the model is generated using a fixed camera positioned at side view. The original input image is background-subtracted using a vision-language model [11], and both the original and rendered images are compared using the Structural Similarity Index Measure (SSIM). Given the grayscale versions of the original image I and the rendered image $\hat{I}_\theta$ for a rotation angle θ, the SSIM score is computed as:

$$\text{SSIM}(I, \hat{I}_\theta) = \frac{(2\mu_I\mu_{\hat{I}_\theta} + c_1)(2\sigma_{I\hat{I}_\theta} + c_2)}{(\mu_I^2 + \mu_{\hat{I}_\theta}^2 + c_1)(\sigma_I^2 + \sigma_{\hat{I}_\theta}^2 + c_2)},$$

where μ and σ represent the mean and standard deviation of pixel intensities, and c_1, c_2 are constants for numerical stability. We evaluate the asset across $N = 100$ evenly spaced rotations along the Z-axis:

$$\text{SSIM}_{\max} = \max_{\theta \in [0, 2\pi)} \text{SSIM}(I, \hat{I}_\theta).$$

If $\text{SSIM}_{\max} < \tau$, where τ is a predefined similarity threshold (e.g., 0.85), the asset is rejected. This filtering step ensures that the generated 3D asset closely resembles the input image in structure and texture before proceeding to aerial rendering.

Algorithm 1 Multi-View Image Generation for a 3D Model

Require: 3D object model $\mathcal{O}$, camera $\mathcal{C}$, number of views N, radius range $[r_{\min}, r_{\max}]$, height variation Δh, horizontal variation Δd

Ensure: A set of N images capturing $\mathcal{O}$ from different perspectives

Initialize scene with 3D object $\mathcal{O}$ and camera $\mathcal{C}$
Compute the center of the object $c_{\mathcal{O}}$
Calculate the initial distance $d_{\mathcal{C},\mathcal{O}}$ between $\mathcal{C}$ and $c_{\mathcal{O}}$
Set the orbit radius $r = \text{Clamp}(d_{\mathcal{C},\mathcal{O}}, r_{\min}, r_{\max})$
for $i = 0$ to $N - 1$ **do**
    Compute the base angle $\theta_i = \frac{2\pi i}{N}$
    Add random angle offset $\delta\theta \sim \mathcal{U}(-0.1\pi, 0.1\pi)$
    Add random radius offset $\delta r \sim \mathcal{U}(-\Delta d, \Delta d)$
    Add random height offset $\delta z \sim \mathcal{U}(-\Delta h, \Delta h)$
    Compute the camera position:

$$x_i = c_{\mathcal{O}}.x + (r + \delta r) \cdot \cos(\theta_i + \delta\theta)$$
$$y_i = c_{\mathcal{O}}.y + (r + \delta r) \cdot \sin(\theta_i + \delta\theta)$$
$$z_i = \mathcal{C}.z + \delta z$$

    Position camera $\mathcal{C}$ at (x_i, y_i, z_i)
    Orient $\mathcal{C}$ to point towards $c_{\mathcal{O}}$
    Update scene
    Render image $\mathcal{I}_i$ from camera $\mathcal{C}$
    Save $\mathcal{I}_i$ with filename indicating view number i
end for
return Set of rendered images $\{\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_{N-1}\}$

b) Multi-View Generation: After passing the realism check, the 3D vehicle models generated by TRELLIS are used to synthesize diverse aerial views at scale. This step addresses the lack of annotated aerial data in existing vehicle classification datasets. 3D models are imported into Blender, and multiple views are rendered using the `bpy` Python module, which provides programmatic access to Blender's 3D rendering engine. The multi-view generation follows a systematic process outlined in Algorithm 1. The scene is initialized with the 3D object and a camera. The object's center is calculated to determine the orbit radius, which sets the average distance between the camera and the object. The camera is placed at varying azimuth angles and altitudes around the object to simulate top-down drone views.

To introduce variation and realism, we add small random offsets to the camera's azimuth ($\delta\theta$), orbit radius (δr), and height (δz). The angle offset $\delta\theta$ is sampled uniformly from $[-0.1\pi, 0.1\pi]$, which allows realistic angular deviation while maintaining the aerial character of the view. Larger variations would result in views that are too oblique and closer to ground-level perspectives, which is not the objective. The radius and height offsets δr and δz are drawn from uniform ranges defined by parameters Δd and Δh, respectively. In our experiments, we set Δd and Δh based on the object's bounding box size to maintain a consistent object scale across views, while introducing enough variation to prevent overfitting to a single camera setup.

The camera is always oriented to point toward the center of the object, ensuring that the vehicle remains centered in each view. This process yields N aerial images per object, with each image representing a slightly different aerial perspective. At this stage, the rendered images contain only the 3D model on a transparent background. Background synthesis is handled in the subsequent step to ensure contextual realism. We utilize text-to-video generation tool SORA [10] for generating realistic background keeping the vehicle as the subject. However, SORA prompt must be composed with certain domain knowledge and avoid words to make the video useful.

c) Story Generation: To guide the background generation process, we first construct textual prompts using an automated storyboard generator. Each prompt is based on the known vehicle class (e.g., SUV, VAN) and includes both a specified terrain and constraints to reduce background clutter. The prompts follow a structured format:

Put the {vehicle class} on a {terrain} road with {constraints}.

For example, we generate prompts such as "Put the SUV on an asphalt road with no other vehicles, not moving," or "Put the van on a dirt road with no pedestrians and no traffic." Another example would be, "Put the sports car in a parking lot with no shadows, not moving." To ensure prompt consistency and effectiveness, we apply several simple rules. Prompts are written using unambiguous language and explicitly define the scene type, such as "asphalt road" or "urban street." Constraints like "no other vehicles" or "not moving" are used to maintain subject isolation and visual clarity. These structured prompts are class-specific and are used as direct inputs to the video generation model.

d) Background Rendering with SORA: Once the prompts are created, we use a high-quality text-to-video model such as SORA to generate short drone-like scenes with realistic background contexts. SORA is chosen for its strong visual fidelity and consistency, though other open-source video generation tools can also be used. The previously rendered object-only aerial view is used as a visual anchor, and the prompt is used to guide the scene generation around it. Each prompt is rendered into a 5–10 second video. These clips typically yield 300–500 usable frames. The video is saved in GIF format and split into individual image frames. Only single-object scenes are retained, as our framework targets isolated vehicle classification tasks.

e) Aerial View Annotation: The validated image frames are annotated using YOLO-World [11]. YOLO-World is effective at localizing general vehicle classes such as "car," but not fine-grained subtypes. Since we already know the precise class name from the input prompt (e.g., SUV, van), we query YOLO-World to detect and localize the "car" in the image. The resulting bounding box is used, and the label is replaced with

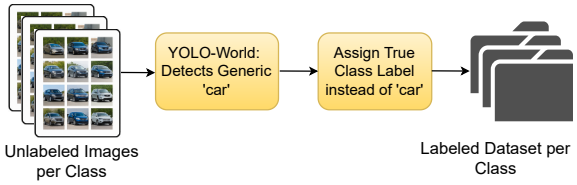

Fig. 3. Class-specific annotation pipeline using YOLO-World to relabel generic detections with true vehicle classes.

the known fine-grained class name. This process is illustrated in Fig. 3. Each image and its corresponding annotation are then saved in MS COCO format, producing a clean, labeled dataset for training and evaluation.

V. Experiments

In this section, we present experiments that demonstrate the effectiveness of aerial view augmentation using AVA and evaluate its impact on vehicle classification performance. All experiments were conducted on a system equipped with an NVIDIA RTX A4000 GPU (16 GB VRAM), using CUDA 12.8 and PyTorch as the primary deep learning framework. We begin by outlining the motivation for using 3D models in dataset and test benchmark creation, followed by a detailed evaluation of how AVA enhances fine-grained vehicle classification through synthetic aerial view augmentation. The project repository is available at AVA.

A. Experiment Setup

In our experiments, we aim to validate the effectiveness of 3D generation methods as tools for data augmentation in fine-grained vehicle classification from aerial views. We argue that reconstructed 3D assets enable scalable synthetic data generation from arbitrary viewpoints, making them highly suitable for data augmentation. However, a major challenge is the lack of benchmark datasets with fine-grained aerial annotations, making direct evaluation difficult.

To address this, we adopt an alternative dataset creation strategy. Rather than collecting real-world aerial and street-view images, we use publicly available 3D vehicle models to generate both street-view and aerial-view images in Blender. This allows us to obtain ground truth aerial views by directly rendering them from the 3D assets, thereby offering a reliable test set for evaluating how well models fine-tuned with synthetic aerial views generalize. The overall dataset generation pipeline is illustrated in Fig. 4. A maximum SSIM score is computed across 100 Z-axis rotations, and assets with no configuration exceeding a threshold of 0.85 are discarded. This threshold is empirically selected to balance realism and diversity, ensuring that only structurally accurate models are used for synthetic aerial rendering.

We curate six vehicle classes: bus, sedan, sport, SUV, truck, and van, with multiple 3D instances per class 6 models each for bus, sedan, and sport; 8 for SUV; 5 for truck; and 7 for van. Using the multi-view generation strategy from Algorithm 1, we render both top-down (aerial) and side (street) views of each 3D model in Blender. Realistic backgrounds are added

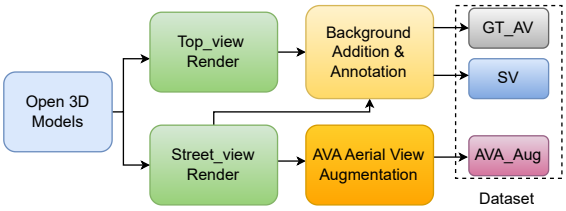

Fig. 4. Dataset creation for the experiments. The input is the open 3D assets of different vehicles and the output is the data for fine-tuning and validation and ground truths.

using SORA via structured prompts, and annotations are generated automatically through the process shown in Fig. 3.

The rendered street-view images represent typical data found in traditional vehicle classification datasets. These images serve two roles: they act as a baseline training set, and they are also used as inputs to AVA to generate synthetic aerial views for augmentation. This dual use allows us to compare models trained with and without aerial view augmentation. In contrast, the aerial views used for evaluation are rendered separately from a disjoint set of 3D vehicle models that are not used in training. This separation is critical as it allows us to construct a controlled ground truth benchmark for fine-grained aerial vehicle classification, where models must generalize to both street views and aerial viewpoints. These ground truth aerial views form the core of our evaluation and are key to validating the real-world utility of synthetic aerial augmentation using 3D reconstruction tools.

We evaluate several YOLO variants to assess detection performance. Each model is tested in two phases: first, on the street-view dataset to measure baseline performance, and then on the ground truth aerial-view dataset to assess generalization to unseen aerial perspectives. The goal is to demonstrate that synthetic aerial view augmentation via AVA significantly improves model robustness and fine-grained classification performance in aerial perspectives.

a) Dataset Description: Table I summarizes the datasets used in our study. The data is carefully partitioned to ensure that models are never exposed to the ground truth aerial-view instances during training.

TABLE I
Overview of curated datasets used in our experiments across six vehicle classes.

Dataset	Description	Images
AVA_Aug	Synthetic aerial views generated by AVA	3057
GT_AV	Ground truth aerial views similar to the top views in Fig. 1 rendered from unique 3D vehicle instances.	1016
SV	Street views of vehicles used for aerial view augmentation. Simulates traditional vehicle classification datasets.	5054

Ground Truth Aerial Views (GT_AV): Rendered directly from unique 3D vehicle models not used in training. This dataset serves as a validation benchmark to evaluate model generalization in fine-grained aerial classification.
Street Views (SV): Rendered from 3D vehicle assets using street perspectives. This dataset simulates commonly available street-view datasets and also serves as input to the AVA augmentation process.
Augmented Aerial Views (AVA_Aug): Generated by applying AVA to the SV dataset. This forms the main synthetic training set with aerial-view content.

It is important to note that there is no overlap between the vehicle models used for GT_AV and those used for training data generation. This ensures that any performance on GT_AV reflects true generalization to unseen vehicles and perspectives.

b) Classification Model Variants: For fine-grained vehicle type classification, we adopt the YOLOv11 architecture due to its strong balance between inference speed and detection accuracy. Its lightweight design makes it ideal for real-time aerial surveillance applications such as drone-based monitoring. Although there are newer variants of YOLO, YOLOv11 provides a mature open source foundation with broad compatibility, which simplifies integration with our synthetic data generation pipeline. We develop and evaluate four model variants, all based on the YOLOv11 backbone:

- **YOLOv11:** The unmodified baseline model was pre-trained on general-purpose datasets such as MS COCO. It serves as a zero-shot/baseline reference with no exposure to our synthetic data.
- **YOLOV-SV:** Fine-tuned on synthetic street-view(SV) images to adapt to our domain. This model serves as the primary side view baseline.
- **YOLOV-TR:** Fine-tuned solely on synthetic aerial views generated by AVA (AVA_Aug), capturing the top-down viewpoint but without any side-view exposure.
- **YOLO-AVA:** Fine-tuned on a combined dataset of SV and AVA_Aug to improve robustness across both perspectives.

c) Training Procedure and Evaluation: Training is conducted in two stages. In the first stage, each model (except the zero-shot YOLOv11 baseline) is fine-tuned for 50 epochs using only the SV dataset to build competence in standard street-view detection. As summarized in Table II, the second column outlines the training data used for each model variant. To prevent overfitting, only the detection heads are updated while backbone layers are frozen. Standard augmentations such random flipping, scaling, and rotation are applied to promote generalization. Early stopping is used to halt training if validation performance stagnates. In the second stage, additional fine-tuning is performed using a combination of SV and AVA_Aug datasets for 50 more epochs. This stage focuses on learning cross-view representations, especially for aerial views.

Each model is evaluated on both street-view validation set SV(val) and GT_AV datasets after each stage. The dual-evaluation strategy quantifies how well the model generalizes from side-view data to aerial perspectives particularly when using synthetic views generated using AVA .

B. Vehicle Classification Results

We evaluate all models on the GT_AV dataset. This setup tests the model's ability to generalize to unseen instances and top-down views which a challenging scenario for fine-grained classification. The results are summarized in Table II. The baseline YOLOv11 model, which was not fine-tuned with our synthetic datasets, achieves only 0.05 mAP@0.5:0.95 and 0.06 mAP@0.5 on GT_AV. This confirms its limited capability to handle aerial views, as it was trained solely on general-purpose datasets with limited angular diversity.

Fine-tuning with side-view images using YOLOV-SV improves performance substantially. On GT_AV, it achieves 0.28 mAP@0.5:0.95 and 0.33 mAP@0.5. However, the recall of 0.35 suggests it still misses many true positives, indicating that side views alone are insufficient for learning top-down representations. The best-performing model, YOLO-AVA, is fine-tuned on both SV and AVA_Aug data. It achieves 0.43 mAP@0.5:0.95 and 0.51 mAP@0.5 on GT_AV, with a balanced precision and recall (0.47 and 0.51). This highlights the importance of aerial augmentation for generalization, especially in the absence of real aerial datasets. YOLO-TR, trained only on AVA_Aug, performs comparably to YOLO-AVA in terms of mAP@0.5:0.95 (0.44) but slightly underperforms in mAP@0.5. This suggests that while aerial augmentation is crucial, combining it with side-view data strengthens model robustness.

As seen in the table, multiple models reach 0.99 mAP on the SV(val) set. This is expected, as SV(val) images share the same rendering pipeline and visual characteristics as the training SV set. No images or 3D assets from GT_AV are used in training, eliminating the possibility of data leakage.

Vehicle Type Detection Analysis. Table III presents the model performance metrics for different vehicle classes. The results indicate varying levels of detection accuracy across categories. The model achieves high performance on the sport class, with a precision of 1.000 and an mAP@0.5 of 0.841. Similarly, the SUV and truck classes show strong detection capabilities, with mAP@0.5 values of 0.709 and 0.648, respectively. Conversely, the model struggles with certain vehicle types, particularly vans and sedans. The van class exhibits the lowest performance, with a precision of 0.046 and an mAP@0.5 of 0.053, highlighting challenges in detecting and classifying this category accurately. The sedan class also demonstrates relatively poor detection.

Our findings suggest that while the model performs well for distinct and well-represented vehicle categories, its accuracy declines for classes with higher visual ambiguity or fewer training instances. Future improvements could focus on refining detection for underperforming vehicle types through additional augmentation or class-specific fine-tuning.

Fig. 5 shows the effect of adding synthetic aerial view data on mAP@0.5 for YOLOv11, YOLO-TR, and YOLO-AVA.

TABLE II
DETECTION PERFORMANCE OF YOLOV11 VARIANTS ON STREET-VIEW AND AERIAL-VIEW DATASETS. EACH MODEL IS EVALUATED ON BOTH SV(VAL) AND GT_AV. HIGH MAP@0.5 VALUES ON SV(VAL) RESULT FROM MATCHING RENDERING PIPELINES. GT_AV REFLECTS GENERALIZATION TO UNSEEN AERIAL VIEWS.

Model Name	Train Set	Val Set	mAP@0.5:0.95	mAP@0.5	mAP@0.75	Precision	Recall
YOLOv11	-	SV(val)	0.21	0.21	0.0	0.13	0.26
YOLOV-SV	SV	SV(val)	0.99	0.99	0.99	0.99	0.99
YOLO-AVA	SV+AVA_Aug	SV(val)	0.99	0.99	0.99	0.98	0.99
YOLO-AVA	SV+AVA_Aug	SV(val) +AVA_Aug(val)	0.98	0.98	0.98	0.98	0.98
YOLOv11	-	GT_AV	0.05	0.06	0.04	0.21	0.10
YOLOV-SV	SV	GT_AV	0.28	0.33	0.24	0.42	0.35
YOLO-AVA	SV+AVA_Aug	GT_AV	**0.43**	**0.51**	**0.48**	**0.47**	**0.51**
YOLO-TR	AVA_Aug	GT_AV	0.44	0.47	0.47	0.47	0.51

TABLE III
PER-CLASS DETECTION PERFORMANCE ON GT_AV

Vehicle Class	Precision	Recall	F1 Score	mAP@0.5
Bus	0.530	0.636	0.457	0.230
Truck	0.508	0.715	0.723	0.648
Sedan	0.143	0.225	0.150	0.149
SUV	0.621	0.790	0.710	0.709
Van	0.046	0.087	0.055	0.053
Sport	1.000	0.624	0.991	0.841

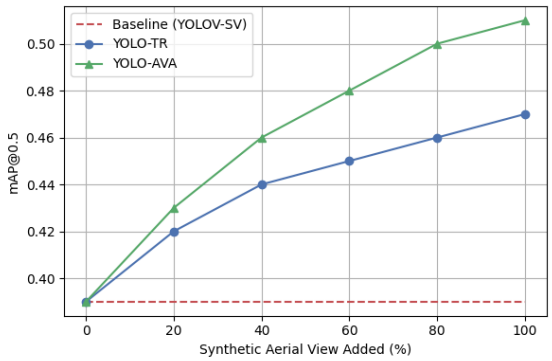

Fig. 5. Impact of aerial view augmentation on mAP@0.5 for YOLOV-SV, YOLO-TR, and YOLO-AVA.

The baseline YOLOv11 model remains at 0.39 mAP@0.50, highlighting its limited capability for aerial perspectives. YOLO-TR, fine-tuned only with aerial augmentations, improves steadily with more data, reaching 0.47 mAP@0.50. YOLO-AVA, trained with both side-view and aerial augmentations, achieves the highest mAP@0.5 of 0.51, demonstrating superior generalization across perspectives. This highlights the importance of combining diverse augmentations to enhance performance for drone-based vehicle classification tasks.

VI. DISCUSSION & FUTURE DIRECTIONS

While 3D modeling introduces challenges for vehicle classification data augmentation such as inconsistent texture mapping, scale mismatches, and occasional geometry defects, it remains a scalable and flexible approach for synthetic dataset generation. Our pipeline AVA includes automated realism checking and structured scene synthesis, significantly improves detection accuracy and generalization for aerial vehicle views. However, further refinement is needed to enhance realism and diversity, particularly under varied environmental conditions.

One of the key limitations of the current system is the absence of explicit modeling for occlusions, lighting variation, and dynamic scene complexity. These are common in real-world aerial footage and affect model robustness. Our synthetic pipeline currently assumes clean, well-lit, isolated views. Future work will incorporate domain randomization and physics-based rendering to simulate occlusions, lighting changes, and multi-vehicle scenarios, enabling models trained on synthetic data to better handle in-the-wild aerial imagery.

Another area for improvement lies in annotation quality. Although YOLO-World provides a practical zero-shot method to generate bounding boxes for general vehicle types, it is limited to coarse labels. In our framework, these annotations are replaced with known fine-grained labels derived from generation prompts. While this process is effective, the reliability of YOLO-World's bounding boxes under aerial perspectives varies depending on background clutter and object deformation. Additional post-processing, such as IoU-based filtering and validation using ensemble models, will be considered in future iterations to further improve annotation fidelity.

The current framework evaluates the benefit of synthetic aerial augmentation specifically on YOLOv11. We intentionally limited the scope to isolate the effects of data augmentation. However, we acknowledge that a full comparative analysis with state-of-the-art object detection models (e.g., YOLOv8, DINO-DETR, RTMDet) is needed to establish broader effectiveness. This benchmarking will be pursued once the synthetic dataset is scaled and released for open evaluation.

As our experiments rely on open-source 3D repositories like RigModels and BlenderKit, biases in the source datasets such as overrepresentation of certain vehicle types may propagate into the training data. We plan to mitigate this by curating a more diverse and balanced 3D asset library in future work. While our results show promising gains in classification accuracy, precision, and recall across unseen aerial perspectives, true generalization in real-world aerial settings remains an open challenge. Expanding the dataset to include uncontrolled variables, urban density, and complex vehicle arrangements will be necessary to close this gap and move closer to deployment-ready models.

VII. Conclusion

This work explores the use of 3D generation as a scalable method for creating synthetic training data for fine-grained vehicle classification from aerial perspectives, with a focus on drone-based applications. We proposed AVA, a pipeline that synthesizes aerial perspectives from street-level images using 3D asset generation, Blender-based rendering, and high-quality background synthesis. We show that fine-tuning YOLOv11 with AVA -augmented data significantly improves generalization to unseen aerial perspectives. Although our experiments focus on aerial view augmentation, the broader implication is that 3D reconstruction can be leveraged to improve detection and classification performance across a wide range of viewpoints and scenarios. A comprehensive library of fine-grained 3D assets would allow researchers to generate diverse, high-fidelity datasets for complex detection tasks without the overhead of real-world data collection. For future work, integrating the Model Context Protocol (MCP) with sensor fusion on UAVs could enable rendering engines such as Blender to operate online, supporting real-time data augmentation and adaptive model training under dynamic conditions including lighting variations, occlusions, and evolving terrains.

Acknowledgment: This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement No. FA8750-23-2-0501 and by a gift from the College of Engineering and Science at Louisiana Tech University. The views, opinions, and conclusions expressed in this document are those of the authors and do not necessarily reflect the official policies or endorsements, either expressed or implied, of the sponsoring organizations.

References

[1] T. Moranduzzo and F. Melgani, "Detecting cars in uav images with a catalog-based approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6356–6367, 2014.

[2] G. U. Sai Theja, M. S. Murari, M. F. Singha *et al.*, "A survey on surveillance using drones," in *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing*, ser. IC3-2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 250–257. [Online]. Available: https://doi.org/10.1145/3549206.3549253

[3] S. Hong, S. Kang, and D. Cho, "Patch-level augmentation for object detection in aerial images," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 127–134.

[4] Y. Tan, Y. Xu, S. Das *et al.*, "Vehicle detection and classification in aerial imagery," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 86–90.

[5] K. Corona, K. Osterdahl, R. Collins *et al.*, "Meva: A large-scale multiview, multimodal video dataset for activity detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1060–1068.

[6] D. Du, P. Zhu, L. Wen *et al.*, "Visdrone-det2019: The vision meets drone object detection in image challenge results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 213–226.

[7] H.-Y. Lin, K.-C. Tu, and C.-Y. Li, "Vaid: An aerial image dataset for vehicle detection and classification," *IEEE Access*, vol. 8, pp. 212 209–212 219, 2020.

[8] R. Liu, R. Wu, B. Van Hoorick *et al.*, "Zero-1-to-3: Zero-shot one image to 3d object," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 9264–9275.

[9] M. Liu, C. Xu, H. Jin *et al.*, "One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] Y. Liu, K. Zhang, Y. Li *et al.*, "Sora: A review on background, technology, limitations, and opportunities of large vision models," *arXiv preprint arXiv:2402.17177*, 2024.

[11] T. Cheng, L. Song, Y. Ge *et al.*, "Yolo-world: Real-time open-vocabulary object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.

[12] R. Hess, *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Focal Press, 2010.

[13] RigModels, "Rigmodels: Free and premium 3d models," https://rigmodels.com/, accessed: 2024-09-09.

[14] BlenderKit, "Blenderkit: 3d model and asset library," https://www.blenderkit.com/, accessed: 2024-09-09.

[15] T.-Y. Lin, M. Maire, S. Belongie *et al.*, "Microsoft coco: Common objects in context," 2015.

[16] J. Deng, W. Dong, R. Socher *et al.*, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[17] Z. Chen, J. Wu, W. Wang *et al.*, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," *arXiv preprint arXiv:2312.14238*, 2023.

[18] F. Nekouei, "Top-view vehicle detection image dataset." [Online]. Available: https://www.kaggle.com/datasets/farzadnekouei/top-view-vehicle-detection-image-dataset

[19] Y. Kharuzhy, "jekhor/aerial-cars-dataset," Aug. 2024. [Online]. Available: https://github.com/jekhor/aerial-cars-dataset

[20] A. Ammar, A. Koubaa, and B. Benjdira, "Aerial images of cars," 2023. [Online]. Available: https://www.kaggle.com/dsv/6382993

[21] C.-S. Hu, S.-W. Tseng, X.-Y. Fan *et al.*, "Vehicle view synthesis by generative adversarial network," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, p. 1–5.

[22] T. Qin, C. Li, H. Ye *et al.*, "Crowd-sourced nerf: Collecting data from production vehicles for 3d street view reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, p. 1–12, 2024.

[23] D. Watson, W. Chan, R. Martin-Brualla *et al.*, "Novel view synthesis with diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2210.04628

[24] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[25] A. Dosovitskiy, G. Ros, F. Codevilla *et al.*, "Carla: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, p. 1–16.

[26] J. Xiang, Z. Lv, S. Xu *et al.*, "Structured 3d latents for scalable and versatile 3d generation," *arXiv preprint arXiv:2412.01506*, 2024.

[27] L. Wang and Y. Huang, "Fast vehicle detection based on colored point cloud with bird's eye view representation," *Scientific Reports*, vol. 13, no. 1, p. 7447, May 2023.

[28] X. Chen, S. Xiang, C.-L. Liu *et al.*, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, p. 1797–1801, 2014.

[29] Marquis03, "Vehicle Classification Dataset," https://www.kaggle.com/datasets/marquis03/vehicle-classification, 2021, accessed: September 12, 2024.