



FlyAwareV2: A multimodal cross-domain UAV dataset for urban scene understanding[☆]

Francesco Barbato¹, Matteo Caligiuri¹, Pietro Zanuttigh^{1*}

Department of Information Engineering, University of Padova, Via Gradenigo, 6B 35131 Padova, Italy

ARTICLE INFO

Keywords:

Semantic segmentation
Aerial imaging
Depth data
Multimodal scene understanding
Unsupervised domain adaptation

ABSTRACT

The development of computer vision algorithms for Unmanned Aerial Vehicle (UAV) applications in urban environments heavily relies on the availability of large-scale datasets with accurate annotations. However, collecting and annotating real-world UAV data is extremely challenging and costly. To address this limitation, we present FlyAwareV2, a novel multimodal dataset encompassing both real and synthetic UAV imagery tailored for urban scene understanding tasks. Building upon the recently introduced SynDrone and FlyAware datasets, FlyAwareV2 introduces several new key contributions: (1) Multimodal data (RGB, depth, semantic labels) across diverse environmental conditions including varying weather and daytime; (2) Depth maps for real samples computed via state-of-the-art monocular depth estimation; (3) Benchmarks for RGB and multimodal semantic segmentation on standard architectures; (4) Studies on synthetic-to-real domain adaptation to assess the generalization capabilities of models trained on the synthetic data. With its rich set of annotations and environmental diversity, FlyAwareV2 provides a valuable resource for research on UAV-based 3D urban scene understanding.

Dataset link: https://medialab.dei.unipd.it/paper_data/FlyAwareV2

1. Introduction

The rapid diffusion of Unmanned Aerial Vehicles (UAVs) has revolutionized a wide range of applications, from surveillance and monitoring to precision agriculture and urban planning [1–3]. UAV technology has seen a rapid rise in popularity in recent years, driven by a growing range of applications that span from recreational uses to deep integration in critical industrial and agricultural operations [4]. Drones are now deployed across a variety of domains. For example, police forces use them for security purposes, allowing rapid and efficient monitoring of areas without deploying personnel, or providing a bird's-eye perspective to support ground operations [5]. In agriculture, UAVs are widely used for field evaluation and monitoring, as well as for the precise application of fertilizers [6,7]. They have also become widespread in cinematography and photography, offering unmatched flexibility for aerial shots. Beyond these applications, UAVs have demonstrated their effectiveness in numerous other scenarios [8].

As a result, UAVs are increasingly expected to meet higher standards in terms of performance, reliability, and operational capabilities. This growing expectation requires the integration of multimodal sensors

capable of capturing a comprehensive representation of the surrounding environment, as well as the development of advanced intelligent systems that allow UAVs to accurately interpret sensory data, make informed decisions in real time, reliably avoid obstacles, and perform fully autonomous operations when necessary [9]. All of this necessitates the use of advanced computer vision capabilities, typically achieved using powerful deep learning models. However, the development of robust vision algorithms based on machine learning for UAV imagery is limited by the scarcity of large-scale, accurately annotated datasets that capture the complexity and diversity of the real world.

Existing UAV datasets for tasks like object detection and tracking [10–12] provide valuable resources, but often lack the dense pixel-level annotations required for semantic scene understanding. On the other hand, datasets designed for the semantic segmentation of aerial views [13–15] tend to be limited in size, scene variability, sensor modalities, and the range of annotated classes. Moreover, most available datasets focus solely on clear daytime conditions, failing to represent the challenges posed by adverse weather scenarios that UAVs routinely encounter during operations.

[☆] This article is part of a Special issue entitled: 'IMAGE Immersive and Networked Media' published in Signal Processing: Image Communication.

* Corresponding author.

E-mail address: zanuttigh@dei.unipd.it (P. Zanuttigh).

¹ Authors contributed equally.

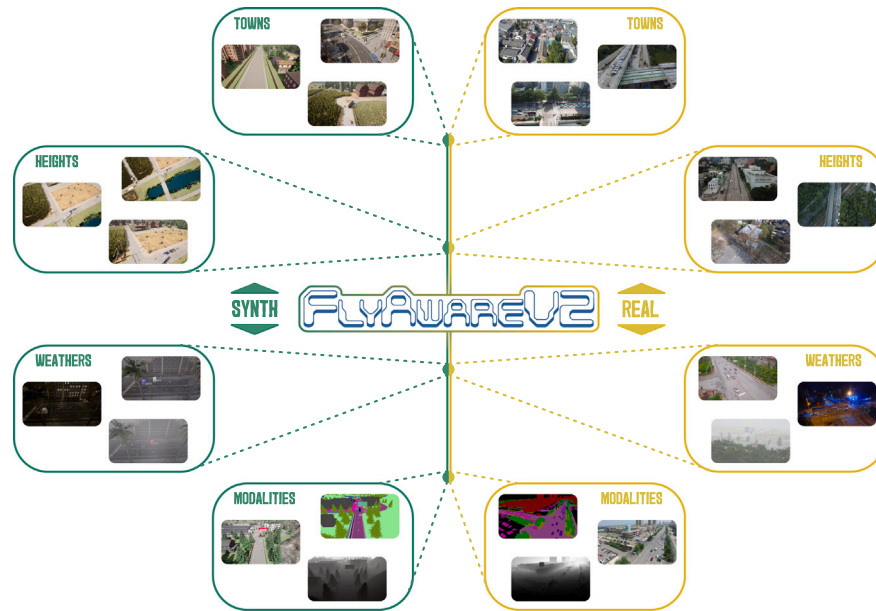


Fig. 1. We introduce FlyAwareV2, a mixed-reality multimodal dataset for UAV imagery. It contains synthetic and real samples in varying weather conditions, with ground-truth depth information as well as semantic segmentation labels.

To bridge this gap, we introduce FlyAwareV2, a novel multimodal dataset that includes synthetic and real data tailored to understanding urban scenes from aerial imagery under various environmental conditions (a visual example of the content is reported in Fig. 1). Building upon our previous efforts on synthetic data generation [16] and real-world adverse weather translation [17], FlyAwareV2 offers several unique contributions:

1. Multimodal data, consisting of color images, depth maps, and semantic labels, cover a wide range of time and weather conditions, such as daytime, nighttime, rain, and fog. This facilitates the development and evaluation of multimodal learning techniques for robust scene understanding.
2. Depth maps for real samples computed using the state-of-the-art Marigold monocular depth estimation model [18]. Leveraging this approach can alleviate the lack of accurate depth annotations in existing UAV datasets.
3. A unified dataset combining synthetic data from simulators derived from the autonomous driving field and real-world UAV images. This enables systematic studies on cross-domain adaptation and generalization from synthetic to real environments.
4. Extensive benchmarking for the multimodal semantic segmentation task using popular deep learning architectures, providing solid baselines for future research.
5. Comprehensive analysis of synthetic-to-real domain adaptation performance, assessing the generalization capabilities of models trained solely on synthetic data when tested on real UAV imagery under diverse conditions.

As already pointed out, the development of FlyAwareV2 is motivated by the need for large, diverse, and accurately annotated datasets to fuel advances in UAV perception, particularly in challenging real-world operating scenarios involving variable weather and illumination. Although simulated environments offer the ability to generate virtually unlimited amounts of annotated data [19,20], they often do not capture the nuances and complexities present in real-world observations. In contrast, manually annotating large real-world UAV datasets is extremely labor-intensive and costly. Our contribution aims to combine the complementary strengths of both synthetic and real data sources.

A key contribution of FlyAwareV2 is facilitating multimodal scene understanding by providing co-registered RGB, depth, and semantic

data streams. The combination of information across multiple sensor modalities has been shown to improve the robustness and accuracy of perception systems [21,22], a key requirement for autonomous UAV operations. However, existing UAV datasets lack coherent multi-sensor data, limiting their applicability for multimodal algorithms development and evaluation.

Another crucial aspect is representing the diverse environmental conditions faced by UAVs during deployments. Adverse weather phenomena such as rain, fog, and night operations can severely degrade the performance of vision algorithms tuned for clear daytime scenarios [23,24]. Although recent driving datasets have made strides in this direction [25,26], comparable resources for UAVs have been lacking. FlyAwareV2 aims to close this gap by providing adverse weather data for both synthetic and real aerial imagery, enabling a systematic study of domain adaptation and generalization across environmental conditions.

The importance of large and accurately annotated datasets cannot be overstated for developing data-driven computer vision solutions, especially in the context of safety-critical applications such as autonomous UAV navigation. Through this work, we strive to provide the research community with a valuable resource that can accelerate progress in this rapidly evolving field. In this paper, we present the details of the FlyAwareV2 dataset, extensive experimental evaluations, and insights gained from our analysis, paving the way for future advances in robust UAV perception in real-world conditions.

2. Related work

With the increasing use of unmanned aerial vehicles (UAVs) in various applications such as surveillance, monitoring, and mapping [1,2], there has been a growing need for robust computer vision algorithms tailored to aerial imagery. However, the development of such algorithms is hindered by the lack of large-scale annotated data sets that capture the diversity of real-world scenarios encountered by UAVs. This section reviews existing datasets and methods for UAV-based computer vision tasks, with a focus on semantic segmentation.

Datasets for UAV computer vision

Early datasets such as AerialScapes [13] and ICG Drone [27] pioneered the collection of aerial annotated images, but were limited in

Table 1
Coarse-to-fine class mapping.

Fine ID	Fine name	Coarse name	Coarse ID	Fine ID	Fine name	Coarse name	Coarse ID
0	Building			7	Vegetation		
1	Fence	Building	0	11	Ground	Vegetation	3
8	Wall			19	Terrain		
4	Road Line			20	Person		
5	Road	21	Rider				
6	Sidewalk	Road	1	-1	Unlabeled		
12	Bridge			2	Fence		
13	Rail Track			3	Pole		
22	Car			9	Traffic Sign		
23	Truck	Vehicle	2	10	Sky	Unlabeled	-1
24	Bus			14	Guard Rail		
25	Train			15	Traffic Light		
26	Motorcycle			16	Static		
27	Bicycle			17	Dynamic		
				18	Water		

scale and diversity. AerialScapes contains 141 video sequences with 11 semantic classes, while ICG Drone provides high-resolution residential scenes with 22 classes, but lacks common road objects. The UAVid dataset [14] offered video sequences from low-altitude UAVs with 300 labeled frames suitable for semantic segmentation. However, its small size and limited frame rate restrict its utility for training modern deep networks.

Recent efforts have aimed to create larger and more comprehensive UAV datasets. The Urban Drone Dataset (UDD) [15] focuses on 3D reconstruction from aerial data across four cities, but is constrained to just four semantic classes. WoodScape [28] provides a multi-task dataset with fisheye cameras and LiDAR from UAVs, enabling applications beyond semantic segmentation. However, it lacks adverse weather conditions that are critical for robust UAV operations.

A notable limitation of most existing real-world UAV datasets is their small size, lack of environmental diversity, and restricted set of annotated classes. This has motivated the use of synthetic data generation. Datasets like SynWoodScape [29] and OmniScape [30] leverage game engines to render synthetic aerial views, but are limited to clear daytime conditions. The IDDA [31] and SELMA [20] datasets provide various synthetic driving scenarios with adverse weather and annotations for autonomous driving tasks, although from a ground vehicle perspective.

A more recent line of work [32–35] studies the end-to-end learning of embodied agents, encompassing both 3D scene generation and rendering, or leveraging guidance from Vision-Language Models (VLMs). While such approaches can in some cases enhance the realism and diversity of the rendered data, they incur substantial computational overhead and sometimes fail to produce plausible real-world scenarios, especially when a large number of samples is required.

The SynDrone dataset [16] represents one of the first attempts to create a large-scale multimodal synthetic dataset specifically for UAV applications. It offers more than 72K images from drone viewpoints at multiple altitudes, with annotations for semantic segmentation and object detection. However, SynDrone only considers clear daytime conditions, limiting its applicability to real-world UAV deployment in variable weather.

Methods for UAV semantic segmentation

Given the scarcity of large real-world datasets, several works have explored the usage of unsupervised domain adaptation (UDA) to leverage synthetic data for training models that can be deployed in real imagery. Some approaches use adversarial learning [36,37] or self-training [38] to align features between synthetic and real domains. Others employ data translation to render synthetic data in real-world styles [39,40]. However, these methods primarily consider ground-level viewpoints and clear daytime conditions.

Only a few studies have specifically targeted UAV semantic segmentation. Nigam et al. [13] used ensemble knowledge transfer to adapt a model from the synthetic GTA-V data set to AerialScapes UAV data. Marcu et al. [41] proposed semi-supervised label propagation on aerial video sequences. However, these methods do not account for the challenges of adverse weather conditions faced by UAVs in real-world operations.

In summary, while significant progress has been made in UAV datasets and vision algorithms, there is a pressing need for data and methods that can handle the multimodal nature of UAV sensors and the diverse operating conditions encountered in practical deployments, including varying weather, lighting, and viewpoints. The FlyAwareV2 dataset presented in this work aims to fill this gap by providing a comprehensive multimodal benchmark with synthetic and real data under adverse environmental conditions.

3. The FlyAwareV2 dataset

The dataset proposed in this work extends and improves the original FlyAware dataset in two key ways: the former consists of adding high-resolution depth maps to all samples, both real and synthetic, while the latter introduces novel weather conditions to the synthetic data and refines the weather augmentation strategy for the real samples.

In this section, we present a detailed description of the proposed dataset, highlighting its organization, data sources, and the additional annotations that accompany it. The dataset contains approximately 290k frames, of which 288k are synthetically generated and 2k are real-world samples. All frames depict drone perspectives over diverse urban and rural environments, captured across multiple spatial and environmental conditions to maximize variability. A summary of the characteristics of the synthetic and real splits of the dataset is reported in Table 2.

3.1. Synthetic UAV data

The synthetic portion is constructed from 24k unique locations, each rendered under four different weather conditions and at three flight altitudes, resulting in a broad spectrum of environmental and illumination conditions.

Synthetic sequences were generated from 8 FullHD (1920 × 1080 px) video streams rendered at 25 Hz. Each sequence corresponds to a different simulated environment, resulting in a total of roughly 3k frames for each of the eight unique environments. To emulate realistic drone behavior, we adjusted flight altitude to three representative levels: 20 m, 50 m, and 80 m. In addition to changing height, we varied the camera tilt to simulate realistic observation angles: 30° at 20 m, 60° at 50 m, and 90° at 80 m. This procedure introduces substantial heterogeneity by combining both geometric (height) and viewpoint

Table 2
Characteristics of the two splits of the FlyAwareV2 datasets.

Data	#frames	Train fr.	Test fr.	Unique locs.	Modalities	Weathers	Altitudes	Cam. angles	Resolution	Frame rate	#classes
Synth.	288k	240k	48k	24k	3	4	3	3	1920 × 1080	25 Hz	28
Real	2.5k	2.3k	200	2.5k	2 (train) 3 (test)	4	Multi	Multi	1320 × 720 (train) 3840 × 2160 (test)	20–24 Hz ^a	8

^a Frame rate of the original video sequences from which images were sampled.

(orientation) changes, and allows for a precise knowledge of the height and orientation parameters for testing viewpoint domain adaptation strategies.

Each sample in the dataset is paired with depth information. The ground-truth depth maps have been obtained directly from the underlying 3D scene geometry through the rendering engine. Beyond depth, we include semantic segmentation labels. Both training and test splits are annotated with fine-grained labels considering 28 semantic classes, a detailed overview is reported in Table 1.

The synthetic data was generated using a customized CARLA 0.9.12 simulator [16,19,20]. CARLA, originally developed using Unreal Engine 4 (UE4), provides photorealistic rendering, realistic physics via NVIDIA PhysX, and basic Non-Player Character (NPC) logic to simulate vehicular and pedestrian behaviors. Our modified version extends CARLA with a larger and more diverse set of UE4 assets, encompassing static objects (e.g., buildings, vegetation, traffic signs) and dynamic ones (e.g., vehicles, cyclists, pedestrians), all modeled with consistent scale and realistic proportions. Note that, in our modified version, the semantic class taxonomy has been extended for better compatibility with established autonomous driving benchmarks [25,42].

To support this, additional vehicle categories such as trains, trams, buses, and trucks were introduced (see [20] for more details), further enriching the diversity of dynamic entities present in the dataset. More in detail, the base CARLA library includes 24 car models, 6 truck models, 4 motorcycle models, and 3 bicycle models, all customizable by color.² It also provides 41 pedestrian models that vary in ethnicity, body build, and clothing, allowing a diverse population simulation. The simulator also offers 8 detailed towns (Town01–07 and Town10HD), each with unique buildings, layouts, and landmarks, effectively creating 8 different simulation environments. Data collection in CARLA is managed through virtual sensors that can be precisely positioned, oriented, and attached to parent actors with rigid or spring-arm dynamics. Sensor outputs can be recorded at every step, and synchronous simulation ensures consistent timing across many high-resolution sensors.

3.2. Real world UAV data

The real-world portion of the dataset is built from 2.5k original frames that are further enhanced to increase environmental diversity.

The real samples, split into training and test sets, are derived from the VisDrone [11] and UAVid [14] datasets, respectively. These datasets include mainly scenes in daylight and clear weather conditions; therefore, we applied synthetic augmentation techniques to introduce variable weather conditions, as detailed in Section 4.3. This results in a set of frames that better matches the environmental variability of the synthetic data. The real samples are split into two resolutions: training images are provided in HD (1320 × 720px), whereas test images have a higher 4K resolution (3840 × 2160 px).

Each sample in the dataset is also paired with depth information. Since depth was not provided in the source datasets, we added estimated depth annotations using a state-of-the-art monocular depth predictor, as detailed in Section 4.1.

The semantic annotations are provided for the 200 test frames, thus allowing us to evaluate Unsupervised Domain Adaptation (UDA)

² The extended simulator with the extra 3D models is available at: <https://github.com/LTTM/SELMA>.

strategies. Compared to synthetic data, the label set is coarser, consisting of 8 semantic classes, which are further consolidated into 5 super-classes for evaluation purposes. Table 1 reports the mapping between the label sets of the synthetic and real data, thus allowing for synthetic-to-real adaptation, cross-domain training, and coarse-to-fine understanding strategies.

4. Data augmentation strategies

In order to build a complete and coherent dataset with multimodal data and all the weather conditions for all the settings we had to resort to some augmentation strategies. In particular, in this section, we detail how we obtained the depth data (Section 4.1) and how we simulated the various weather conditions for both synthetic and real data (Sections 4.2 and 4.3).

4.1. Depth estimation

As shown in Fig. 2, the proposed dataset provides the 3D information for each scene represented through a depth map aligned with the color view. For the synthetic samples, we simply extracted the ground truth depth information from the underlying 3D geometry of the scenes.

This was not possible for the real samples. An alternative would be to reconstruct the 3D geometry of real scenes using structure-from-motion techniques; unfortunately, the frame rate in most real-world datasets is not high enough to obtain reliable results with this strategy. Given these limitations, we had to resort to monocular depth estimation techniques to generate the depth information for the real samples.

Nowadays, state-of-the-art strategies for this task are based on deep learning [43]. For this dataset, we chose to employ the highly performing Marigold [18] monocular depth estimation model, which is based on the idea of training a diffusion model for color-to-depth domain translation. To align the monocular depths with those of the synthetic samples, we employed the 16-bit depth generation pipeline and re-normalized all depth samples (both synthetic and real) during processing.

For the training of our benchmark architectures, we normalized independently each depth sample. That is, we rescaled all depthmaps in the range [0, 1], regardless of the original maximum or minimum produced by Marigold. In this way, the absolute depth information available in the synthetic samples is destroyed, but the coherence between real and synthetic depth is increased, allowing for better domain transfer of models trained on synthetic data.

4.2. Weather conditions for synthetic data

The generation of various weather conditions for the synthetic dataset was achieved using the Unreal Engine (UE) [44] integrated within CARLA. By programmatically modifying the environmental configuration parameters, we produced photorealistic images under a wide range of adverse conditions. The physics-based rendering capabilities of UE ensure that these simulated weather effects closely approximate their real-world counterparts (see Fig. 3 for some visual examples).

To further enhance data realism and diversity, we customized the CARLA source code at multiple levels. First, we extended and refined the predefined environmental settings by adjusting the atmospheric scattering parameters, fog density, and solar elevation and azimuth

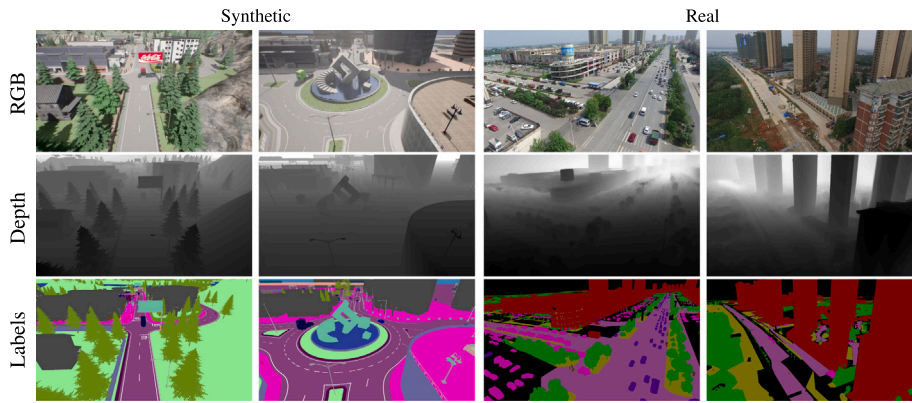


Fig. 2. The FlyAwareV2 dataset provides color information, depth data and semantic labels for each frame.

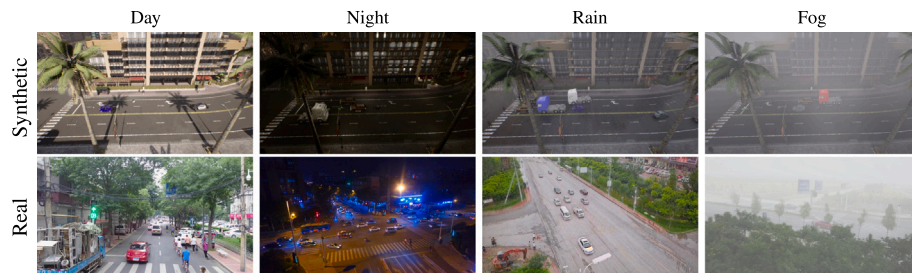


Fig. 3. The FlyAwareV2 dataset provides data in variable weather and daytime conditions.

Table 3

Weather configuration settings and strategies for synthetic and real data.

Weather	Synthetic config.	Real data aug. strategy
Day	ClearNoon	<i>Unchanged</i>
Night	ClearNight	<i>img2img-turbo</i>
Rain	HardRainNoon	<i>img2img-turbo + recolor</i>
Fog	MidFoggyNoon	<i>FoHIS + recolor</i>

angles. These modifications increased both the diversity and the realism of lighting and visibility conditions. We also introduced new configurations, including an additional nighttime setting and a new weather profile, hard fog. The configurations and augmentation strategies are summarized in Table 3.

4.3. Weather conditions for real data

Collecting real images under adverse weather conditions proved to be extremely challenging. Operating a drone during heavy rain or dense fog is very challenging and impractical due to reduced visibility and flight instability. Consequently, such imagery is largely unavailable in the literature. Nevertheless, achieving consistency between the real and synthetic datasets was crucial for our study, particularly in terms of the inclusion of adverse weather scenarios.

After evaluating multiple alternatives, we determined that augmenting real clear-weather images to simulate different weather conditions was the only feasible solution. Although the augmentation process varied slightly across weather types, in all cases we focused on preserving the structural realism and label alignment of the original images. A visual example of the final result is shown in the bottom row of Fig. 3.

Fog. Simulating realistic fog needs to take into account that natural fog exhibits complex interactions with scene depth and light scattering. We adopted an analytical approach that computes fog intensity per pixel using physics-based models. Specifically, we used the FoHIS algorithm [45] to generate depth-dependent fog effects.

To tailor the method to our use case, we updated the original implementation and extended it with color manipulation functionality. This addition allowed us to reproduce the cooler, desaturated tones typically associated with foggy and winter conditions. We name this operation *recoloring*, as it consists of darkening and desaturation operations applied to the colorspace. More in detail, we first shift the white-point of the image \mathbf{X} from the original warm-reddish color to a toneless one, then we desaturate the image by computing the weighted average with the grayscale counterpart with weight $r_d = 0.7$, i.e., $\mathbf{X}_{\text{desat}} = r_s \mathbf{X} + (1 - r_s) \mathbf{X}_{\text{gray}}$. Finally, we darken the scene by re-scaling the RGB values by $r_l = 0.8$, i.e., $\mathbf{X}' = r_l \mathbf{X}_{\text{desat}}$.

The algorithm takes in input the RGB image, its corresponding depth map, and scene-level parameters such as the camera position and the visibility range. Depth maps were obtained as described in Section 4.1, and environment-specific profiles were manually defined for each dataset.

Rain and night. Rain and night scenes cannot be reproduced analytically, as they are highly dependent on image-specific features such as light sources, reflections, and object materials. For example, generating a realistic night image requires simulating illuminated streetlights, active vehicle headlights, and lit windows in surrounding buildings.

Although diffusion-based generative models can produce visually convincing results, they often alter the structure of the scene, leading to inconsistencies between the generated images and ground-truth annotations. To avoid such distortions, we employed a U-Net [46]-based model, *img2img-turbo* [47], which enables pixel-level transformations while preserving spatial and content integrity. Using this approach, we generated realistic rainy and nighttime variants of daytime images at the same resolution, that preserve perfect alignment with their original labels. For rainy images only, we have coupled *img2img-turbo* with the recoloring step described above to achieve better fidelity with the real counterpart.

It is important to note that for the test set, night images were synthetically generated from daytime counterparts, whereas for the training set, we included real nighttime images from existing datasets whenever available.

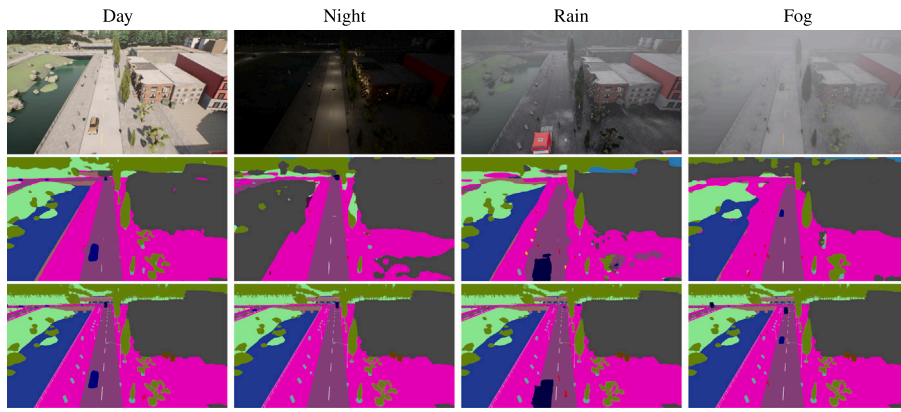


Fig. 4. Qualitative results: model trained on synthetic samples and tested on synthetic samples with varying weather conditions. First row: Input, Second row: Model Prediction, Third Row: Ground Truth.

Table 4

Training and testing on synthetic data with varying weather conditions in the train and test sets.

(a) mIoU on the full 28 classes set.						(b) mIoU on the coarse 5 classes set.							
Train	Test	Day	Night	Rain	Fog	All	Train	Test	Day	Night	Rain	Fog	All
		Day	57.9	1.7	11.8	1.9			17.0	Day	75.2	13.7	33.9
Night	2.2	30.6	1.7	0.7	8.4	Night	13.9	53.4	8.8	5.3	19.3		
Rain	9.3	1.7	53.5	9.7	17.1	Rain	32.3	9.5	72.3	27.8	34.1		
Fog	4.3	0.8	18.2	34.0	15.1	Fog	18.9	10.6	42.5	63.7	30.5		
All	51.7	29.8	49.9	39.8	42.5	All	72.1	55.0	70.6	61.6	64.5		

5. Experimental evaluation

We performed an extensive set of experiments using the FlyAwareV2 dataset to provide valuable insights into how it allows efficient training of deep learning models for multimodal semantic segmentation in urban environments tailored to UAV imagery. In this section, we start with the implementation details (Section 5.1), then we discuss the performances on synthetic data (Section 5.2). We continue analyzing how a model trained on the FlyAwareV2 synthetic data can perform on real-world data, firstly using it “as is” (Section 5.3) and then employing also Unsupervised Domain Adaptation (UDA) techniques (Section 5.4). Finally, we also discuss the performances of multimodal strategies that also exploit depth information (Section 5.5).

5.1. Implementation details

For the experimental evaluation, we employ an encoder–decoder architecture composed of a MobileNetV3+ [48] backbone integrated with a DeepLabV3 [49] decoder. This design choice, widely used in semantic segmentation literature, provides an effective balance between computational efficiency and segmentation accuracy.

Model training is conducted using a single NVIDIA L40s GPU with a batch size of 16 and full-HD images in input. Each training run spans 30k iterations. For experiments involving multi-modal architectures, we utilize two L40s GPUs to cope with the memory requirements arising from the increased model complexity and input dimensionality. Following previous works [16,17,20], optimization is carried out using the Adam algorithm, with an initial learning rate of 2.5×10^{-4} . The learning rate follows a cosine annealing schedule that decays to zero, preceded by a linear warm-up phase during the first 2000 iterations.

The data augmentation pipeline is designed to improve generalization by introducing controlled variations in image appearance and structure. Specifically, we apply random horizontal flipping with probability $p = 0.5$, as well as brightness, contrast, saturation, and hue jittering with a rate of $r = 0.5$, following the implementation in [50]. In addition, a Gaussian blur with $\sigma_b = 1.5$ and additive Gaussian noise with standard deviation $\sigma_n = 1.5$ are applied to simulate sensor

noise and slight defocus. These enhancements collectively improve the resilience to illumination changes, color variability, and moderate image degradation.

5.2. Synthetic data segmentation experiments

We start by training the network on the synthetic data in FlyAwareV2 and evaluating the performance of the network on the synthetic test set, *i.e.*, on the same domain used for training. The results are reported in Table 4 and Fig. 4. Table 4b reports the results on the reduced class set used by the synthetic-to-real experiments; details of the mapping are listed in Table 1.

As a starting point, Table 4a shows the performance on the full set of 28 classes. Using the model trained on all the training data (of all weathers) and evaluated on all testing data leads to a mIoU of 42.5%. From the table, it can be seen that the daytime data is easier (51.7%), while the night and fog settings proved to be more challenging, as expected (29.8% and 39.8%). Training only on a specific weather condition leads to overfitting that specific setting, with improved performance on the chosen setting at the price of strong degradation on the others (in many cases, the model simply does not work, leading to accuracies below 2%).

A similar situation is noticeable in Fig. 4, the daytime prediction is clearly of higher accuracy with respect to all other weather conditions, while fog and rain lag behind. The nighttime prediction is the worst, with the network confusing the water in the river for a building and completely missing the vegetation on the sidewalk.

Using the coarser class set (Table 4b), the task becomes easier and the accuracy improves. The all-weather experiment leads to an accuracy of 64.5%, but the general trend remains the same. The only difference is that, in settings where train and test data do not match, performance is low but not unacceptable as before. This is probably because *e.g.*, fog or rain makes it particularly difficult to recognize challenging small classes, while larger and simpler things like the road or a building can still be recognized.

Then, in Table 5 we analyze the impact of training or testing in different environments (in this case, the different Towns of the CARLA simulator). A model trained on all cities can generalize quite well to the

Table 5
mIoU varying training and testing towns. The results are on the full 28 classes set. Note how not all classes are present in all towns (the number of classes per town is in parentheses close to the town name).

		Test								
Train		Town01(25)	Town02(21)	Town03(26)	Town04(25)	Town05(27)	Town06(24)	Town07(24)	T.10HD(26)	All
		Town01	38.7	14.9	6.1	9.9	7.6	10.4	10.3	5.3
Town02	15.2	45.1	5.5	6.8	6.9	6.2	7.5	5.1	10.6	
Town03	8.8	8.8	50.7	12.1	14.5	13.8	7.3	10.0	<u>18.6</u>	
Town04	10.8	9.0	11.4	37.3	16.0	14.6	10.9	6.2	16.1	
Town05	7.5	7.5	14.7	14.1	37.8	12.1	8.9	8.5	14.9	
Town06	8.1	6.0	10.2	11.3	11.8	33.2	9.3	6.0	10.9	
Town07	7.3	5.2	7.4	10.0	9.8	12.6	40.1	3.4	10.3	
Town10HD	7.5	7.8	9.3	6.7	7.5	4.2	3.5	39.9	12.2	
All	<u>30.9</u>	<u>27.5</u>	<u>39.0</u>	<u>31.9</u>	<u>33.8</u>	<u>25.0</u>	<u>27.3</u>	<u>26.6</u>	42.5	
All ^a (town cl.)	34.6	36.7	42.0	35.7	35.1	29.2	31.9	28.6	42.5	

^a The mIoU computed only on the classes present in each town is reported in the last row.

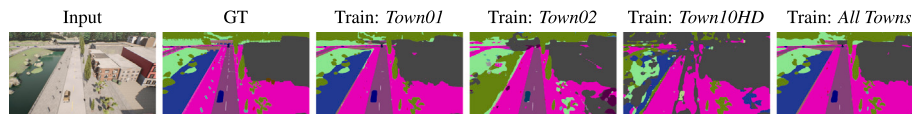


Fig. 5. Output segmentation maps for a sample frame from Town01 when using models trained on different towns (i.e., trained on Town01 itself, on Town02, Town10 and using the model trained on all towns).

Table 6
mIoU varying training and testing height (using synthetic data from all weathers and towns for both training and testing).

		Test			
Train		20 m	50 m	80 m	All
		20 m	48.6	21.5	10.7
50 m	21.6	44.4	31.0	<u>31.0</u>	
80 m	9.6	30.0	44.0	25.4	
All	<u>42.9</u>	<u>41.4</u>	<u>40.5</u>	42.5	

Table 7
Synthetic-to-real adaptation across weather conditions. Performances evaluated on real samples using the Coarse class-set.

		Test				
Train		Day	Night	Rain	Fog	All
		Day	<u>48.7</u>	21.3	37.6	12.9
Night	12.7	<u>22.7</u>	7.4	8.6	13.4	
Rain	38.5	17.7	<u>45.0</u>	18.4	31.4	
Fog	28.6	14.7	32.7	<u>19.7</u>	24.7	
All	<u>49.7</u>	<u>23.4</u>	<u>48.4</u>	<u>38.4</u>	42.3	

various environments corresponding to the different towns. Training on a single town instead leads to a model that is not able to generalize well due to the different appearance (some classes are unavailable in certain towns, making it impossible to train there). These findings are confirmed by the qualitative results reported in Fig. 5, where we show the performance of segmentation models trained on different synthetic towns and tested on a frame from Town01. As expected from quantitative results, the best accuracy is achieved by networks pretrained on the same town or on all towns together, while using other domains incurs in significant degradation.

Finally, we focus on the impact of flying height. Table 6 shows how a model trained on drones flying at different heights is able to generalize well to this aspect, obtaining a stable accuracy ranging between around 40% and 43% (as expected, higher heights are slightly more challenging since objects appear smaller). However, training using only data at a specific height leads to weaker models that are not able to generalize well to a change of viewpoint. This is consistent with the results found for weather and towns, confirming the validity of heterogeneous data.

5.3. Real world evaluation of models trained on synthetic data

The next step is to deploy the models trained on the synthetic FlyAwareV2 data in the real world. We start by showing the results achieved by training on synthetic data and testing on real-world data across varying weather conditions in Table 7 and Fig. 6. Note that, for the synthetic pretraining, we trained the DeepLabV3/MobileNetV3-large segmenter using only color data and the fine-level class-set. At test time, we map the fine-level class-set into the corresponding

coarse-level one (refer to Table 1 for more details). As expected, when considering all the data, it can be observed that there is a significant drop in accuracy compared to synthetic test data (on the coarse set, the accuracy was 64.5%), but a reasonably good accuracy of 42.3% can be achieved.

Working in clear daytime conditions is, again, easier with an accuracy that reaches almost 50%, however, models trained on clear weather data struggle to generalize when tested on real-world nighttime and foggy weather scenarios: using only clear weather data for training, the overall mIoU drops significantly from 48.7% on daytime test data to 21.3% for nighttime and 12.9% for fog, while rain is slightly better at 37.4%. Training in a mixture of all weather conditions leads to much better performances, especially in rainy and foggy data (48.4% and 38.4%), while the night setting remains the most challenging at 23.4%. This highlights the importance of incorporating diverse environmental variations during training to enhance the robustness of UAV perception systems.

Like before, the qualitative results reported in Fig. 6 support the quantitative experiments. For this discussion, we focus on the predictions of the second row; the ones in the third will be discussed in the Unsupervised Domain Adaptation section. The daytime prediction offers the highest accuracy, especially on small segments like *person*. Rain and fog, again, offer suboptimal but acceptable results, given the relative similarity to the daytime conditions. On the other hand, the nighttime environment results in highly degraded performance, with significant confusion between classes (e.g., *Building* vs. *Vegetation*).

The impact of varying UAV flight height during training on synthetic-to-real adaptation performance is investigated instead in

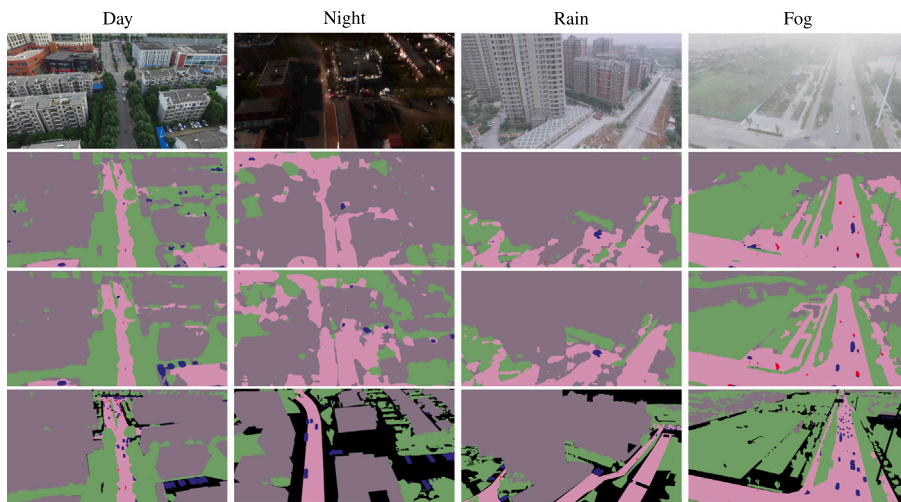


Fig. 6. Qualitative experiments: model trained on synthetic samples and tested on real samples with varying weather conditions. First row: Input, Second row: Model Prediction (no adaptation), Third Row: Model Prediction (UDA), Fourth Row: Ground Truth.

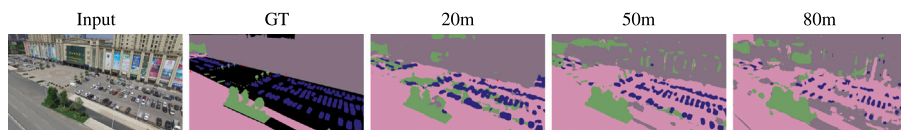


Fig. 7. Qualitative experiments, Model trained at different heights and tested on real samples.

Table 8 Synthetic-to-real adaptation varying training height, RGB only, Coarse class-set.

Height	Real mIoU	Synth mIoU
20 m	44.5	50.3
50 m	37.2	56.7
80 m	29.4	50.5
All	42.3	64.5

Table 8 and Fig. 7 (note that, in this case, the test samples have been taken at different heights and height data is not provided in the source real-world datasets).

As expected, models trained at lower altitudes (20 m) achieve better generalization, with a mIoU of 44.5%, compared to 37.2% at heights of 50 m and 29.4% at heights of 80 m. This can be attributed to the increased level of detail and resolution available in lower-altitude images, which aids in learning more discriminative features for semantic segmentation. In the Figure, we show how the architectures trained at different heights predict the input sample. We can observe a strong correlation between altitude and performance degradation, highlighting the differences in PoV between real and synthetic samples. This, of course, depends on the particular height of our real samples; different environments or applications may be closer to other synthetic configurations, highlighting the importance of data heterogeneity.

The influence of the specific synthetic urban environment used for training is analyzed in Table 9. Although there are variations in performance when employing different virtual towns, the general trend suggests that models trained on diverse synthetic environments can generalize reasonably well to real-world data, with mIoU scores ranging from 26.0% to 36.1%. However, training with all towns together leads to a score of 42.3%, much better than using each town alone. This underscores the importance of leveraging diverse synthetic data environments to improve generalization capabilities.

Table 9 Synthetic-to-real adaptation varying training town, RGB only, Coarse class-set.

Town	Real mIoU	Synth mIoU
Town01	34.9	41.9
Town02	36.1	41.7
Town03	35.1	44.9
Town04	32.4	43.2
Town05	34.3	42.2
Town06	26.0	38.8
Town07	27.9	33.9
Town10HD	34.3	36.1
All	42.3	64.5

5.4. Synthetic-to-real unsupervised domain adaptation results

As discussed in Section 5.3, the domain shift between the synthetic and real data causes a degradation of performance. Since FlyAwareV2 also provides a large set of unlabeled real-world samples, a viable solution is to use them to apply Unsupervised Domain Adaptation (UDA) strategies. Taking inspiration from the wide literature in the field [51,52], we tested 2 different classic UDA approaches, which can be considered as benchmarks:

A min-entropy UDA strategy (MaxSquareIW, MSIW [53]): in this case, the fine-tuning starts from the architectures pre-trained on the synthetic samples.

A Min-entropy and multi-batch normalization combined strategy: here, we mix the MSIW approach with a common strategy for test-time domain adaptation, *i.e.*, using different sets of batch normalization layers, one for each domain. In essence, we add a set of batch-norms (initialized with the original values) and fine-tune them to estimate the real samples distribution (the supervision comes only from the MSIW loss). The hypothesis behind these methods is that convolutional layers generalize across domains, while the BNs are domain-specific. After training, we obtain a network with two sets of normalization layers

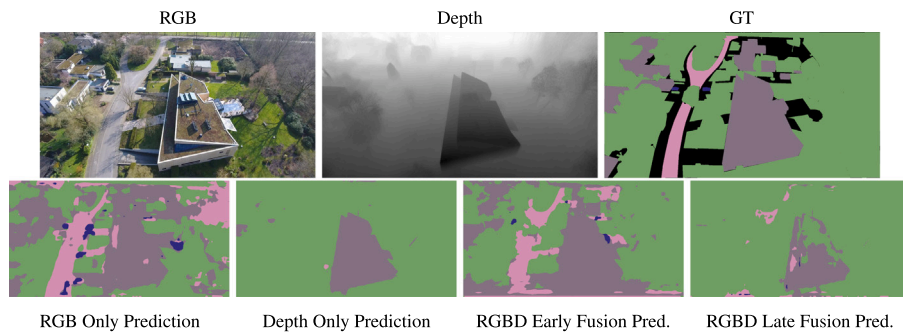


Fig. 8. Qualitative experiments: the second row shows the predictions from models trained with color alone, depth alone, and multimodal data and tested on real samples.

Table 10

Results after adapting on real (unlabeled) training FlyAwareV2 data. (Row-wise comparisons).

Weather	no-UDA	UDA	UDA-BN	
			BN-S	BN-T
Day	49.7	53.2	54.2	<u>53.5</u>
Night	23.4	29.1	<u>28.5</u>	27.5
Rain	48.4	48.0	55.3	<u>48.5</u>
Fog	38.4	36.1	43.8	<u>38.5</u>
All	42.3	44.0	47.1	<u>44.3</u>

Table 11

Multimodal experiments. Real results in coarse class-set.

Modality	Synth		Real
	Fine	Coarse	
RGB	42.5	64.5	<u>42.3</u>
D	<u>67.1</u>	<u>82.3</u>	17.0
RGBD early	63.5	80.0	47.8
RGBD late	68.4	82.8	25.6

(we denote the ones adapted to the source and target domains as BN-S and BN-T, respectively), leaving us with a choice between them at evaluation time. For completeness, we evaluate on the real samples using both, confirming an improvement over single-BNs architectures in all cases.

The results are shown in Table 10: UDA strategies consistently improve performance across all weather conditions compared to the no-adaptation baseline.

The best overall mIoU of 47.1% is achieved using the UDA-BN-S method, demonstrating the effectiveness of domain adaptation in bridging the synthetic-to-real gap for UAV scene understanding tasks. A visual example is also shown in the third row of Fig. 6, which highlights the effectiveness of UDA strategies. Note how in the daytime sample all objects are much better defined, how in nighttime and rain conditions the vegetation is restored, and in foggy environments the vehicles are better identified.

5.5. Multimodal segmentation experiments

Following [16], we evaluated our multimodal data using two benchmark architectures, one for early fusion and one for late (output-level) fusion.

They represent standard baseline approaches to multimodal fusion, highlighting the data quality and generalizability, rather than focusing on the achievements possible with highly complex state-of-the-art multimodal schemes. In the first, we simply concatenate RGB data and (normalized) depth at the input level, obtaining a 4-channel-input architecture. In the second, we merge the multimodal information at the output level: we duplicate the segmentation architecture, computing a

prediction from RGB and depth independently, before concatenating them together and merging them into a single prediction using a 1×1 Convolution without bias that maps the two outputs to a single one with the same number of channels.

The benefits of multimodal fusion are evident in Table 11, which compares the performance of models trained on RGB, depth (D), and their combination using early or late fusion. Although the depth-only model performs poorly (mIoU of 17.0%), incorporating depth information through early fusion with RGB significantly boosts performance to 47.8% mIoU. However, late fusion of RGB and depth modalities yields suboptimal results (25.6% mIoU), highlighting the importance of early multimodal integration for effective feature learning. These findings are confirmed by the qualitative results reported in Fig. 8, where one can appreciate how, compared to the second-best (RGB), the early fusion strategy leads to a much better segmentation of the Y bend in the road, as well as no confusion of the vegetation on the right side of the scene.

6. Conclusions and future work

In this paper we introduced FlyAwareV2, a novel large-scale dataset for UAV computer vision applications encompassing synthetic and real-world multimodal information in varying weather conditions. We provide experimental benchmarks showing how the large amount of provided synthetic data can be used to train segmentation models, achieving effective performances on real-world imagery. We also explored the domain transfer capabilities of segmentation models across different weathers, daytimes, flying heights, and environments. Finally we also showed how performances can be improved by exploiting multimodal data. Further extensions of the dataset will consider other computer vision tasks, such as object detection or panoptic segmentation, and the acquisition of a larger collection of real-world data.

CRedit authorship contribution statement

Francesco Barbato: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Matteo Caligiuri:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Pietro Zanuttigh:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Matteo Caligiuri, Pietro Zanuttigh reports financial support was provided by European Commission. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP C93C22005250001, partnership on “Telecommunications of the Future” (PE00000001- program “RESTART”).

Data availability

The paper provides also the link to the data.

References

- [1] N.H. Motlagh, M. Bagaa, T. Taleb, UAV-based IoT platform: A crowd surveillance use case, *IEEE Commun. Mag.* 55 (2) (2017) 128–134.
- [2] H. Kim, L. Mokdad, J. Ben-Othman, Designing UAV surveillance frameworks for smart city and extensive ocean with differential perspectives, *IEEE Commun. Mag.* 56 (4) (2018) 98–104.
- [3] S. Bhatnagar, S. Puliti, B. Talbot, J.B. Heppelmann, J. Breidenbach, R. Astrup, Mapping wheel-ruts from timber harvesting operations using deep learning techniques in drone imagery, *Forestry* 95 (5) (2022) 698–710.
- [4] M. Hassanalian, A. Abdelkefi, Classifications, applications, and design challenges of drones: A review, *Prog. Aerosp. Sci.* 91 (2017) 99–131.
- [5] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto, A. Sciarone, RF/WiFi-based UAV surveillance systems: A systematic literature review, *Internet Things* 26 (2024) 101201.
- [6] J. Liu, J. Xiang, Y. Jin, R. Liu, J. Yan, L. Wang, Boost precision agriculture with unmanned aerial vehicle remote sensing and edge intelligence: A survey, *Remote. Sens.* 13 (21) (2021) 4387.
- [7] M.F. Aslan, A. Durdu, K. Sabanci, E. Ropelewska, S.S. Gültekin, A comprehensive survey of the recent studies with UAV for precision agriculture in open fields and greenhouses, *Appl. Sci.* 12 (3) (2022) 1047.
- [8] C. Xu, Q. Li, Q. Zhou, S. Zhang, D. Yu, Y. Ma, Power line-guided automatic electric transmission line inspection system, *IEEE Trans. Instrum. Meas.* 71 (2022) 1–18.
- [9] H. Liu, Q. Long, B. Yi, W. Jiang, A survey of sensors based autonomous unmanned aerial vehicle (UAV) localization techniques, *Complex Intell. Syst.* 11 (8) (2025) 1–24.
- [10] U. Benchmark, A benchmark and simulator for uav tracking, in: *European Conference on Computer Vision*, Vol. 7, 2016.
- [11] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, H. Ling, Detection and tracking meet drones challenge, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (11) (2021) 7380–7399.
- [12] N. Jiang, K. Wang, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, J. Zhao, G. Guo, Z. Han, Anti-UAV: A large multi-modal benchmark for UAV tracking, 2021, arXiv preprint arXiv:2101.08466.
- [13] I. Nigam, C. Huang, D. Ramanan, Ensemble knowledge transfer for semantic segmentation, in: *2018 IEEE Winter Conference on Applications of Computer Vision*, WACV, IEEE, 2018, pp. 1499–1508.
- [14] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, M.Y. Yang, UAVid: A semantic segmentation dataset for UAV imagery, *ISPRS J. Photogramm. Remote Sens.* 165 (2020) 108–119.
- [15] Y. Chen, Y. Wang, P. Lu, Y. Chen, G. Wang, Large-scale structure from motion with semantic constraints of aerial images, in: *Chinese Conference on Pattern Recognition and Computer Vision*, PRCV, Springer, 2018, pp. 347–359.
- [16] G. Rizzoli, F. Barbato, M. Caligiuri, P. Zanuttigh, Syndrome-multi-modal uav dataset for urban scenarios, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2210–2220.
- [17] G. Rizzoli, M. Caligiuri, D. Shenaj, F. Barbato, P. Zanuttigh, When cars meet drones: Hyperbolic federated learning for source-free domain adaptation in adverse weather, in: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, WACV, IEEE, 2025, pp. 1587–1596.
- [18] B. Ke, K. Qu, T. Wang, N. Metzger, S. Huang, B. Li, A. Obukhov, K. Schindler, Marigold: Affordable adaptation of Diffusion-Based image generators for image analysis, 2025, arXiv:2505.09358 URL <https://arxiv.org/abs/2505.09358>.
- [19] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: *Conference on Robot Learning*, PMLR, 2017, pp. 1–16.
- [20] P. Testolina, F. Barbato, U. Michieli, M. Giordani, P. Zanuttigh, M. Zorzi, Selma: Semantic large-scale multimodal acquisitions in variable weather, daytime and viewpoints, *IEEE Trans. Intell. Transp. Syst.* 24 (7) (2023) 7012–7024.
- [21] D. Lahat, T. Adali, C. Jutten, Multimodal data fusion: an overview of methods, challenges, and prospects, *Proc. IEEE* 103 (9) (2015) 1449–1477.
- [22] G. Rizzoli, F. Barbato, P. Zanuttigh, Multimodal semantic segmentation in autonomous driving: A review of current approaches and future perspectives, *Technologies* 10 (4) (2022) 90.
- [23] R. Xia, C. Zhao, M. Zheng, Z. Wu, Q. Sun, Y. Tang, Cnda: Cross-modality domain adaptation for nighttime semantic segmentation, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21572–21581.
- [24] D. Brüggemann, C. Sakaridis, P. Truong, L. Van Gool, Refign: Align and refine for adaptation of semantic segmentation to adverse conditions, in: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 3174–3184.
- [25] C. Sakaridis, D. Dai, L. Van Gool, ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10765–10775.
- [26] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, T. Darrell, et al., Bdd100k: A diverse driving video database with scalable annotation tooling, 2018, p. 6, arXiv preprint arXiv:1805.04687, Vol. 2, no. 5.
- [27] G.U. of Technology, ICG Drone Dataset.
- [28] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O’Dea, M. Uricár, S. Milz, M. Simon, K. Amende, et al., Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9308–9318.
- [29] A.R. Sekkat, Y. Dupuis, V.R. Kumar, H. Rashed, S. Yogamani, P. Vasseur, P. Honeine, SynWoodScape: Synthetic surround-view fisheye camera dataset for autonomous driving, *IEEE Robot. Autom. Lett.* 7 (3) (2022) 8502–8509.
- [30] A.R. Sekkat, Y. Dupuis, P. Vasseur, P. Honeine, The omniscene dataset, in: *2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE*, 2020, pp. 1603–1608.
- [31] E. Alberti, A. Tavera, C. Masone, B. Caputo, IDDA: A large-scale multi-domain dataset for autonomous driving, *IEEE Robot. Autom. Lett.* 5 (4) (2020) 5526–5533.
- [32] R. Della Pietra, N. Garau, Render, Encode, Plan: A simple pipeline for hybrid RL-DL learning inside Unreal Engine, *Comput. Graph.* (2025).
- [33] S. Wang, S. Li, Y. Zhang, S. Yu, S. Yuan, R. She, Q. Guo, J. Zheng, O.K. Howe, L. Chandra, et al., UAVScenes: A Multi-Modal dataset for UAVs, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 28946–28958.
- [34] M.U. Zia, W. Xiang, T. Huang, J. Ahmad, J.N. Chattha, I.H. Naqvi, F.A. Butt, Unifying ground and air: a comprehensive review of deep learning-enabled CAVs and UAVs, *Artif. Intell. Rev.* 59 (1) (2026) 19.
- [35] L. Yao, F. Liu, S. Xu, C. Zhang, S. Di, X. Ma, J. Jiang, Z. Wang, J. Zhou, UEMM-Air: Enable UAVs to undertake more Multi-modal tasks, in: *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 12792–12798.
- [36] U. Michieli, M. Biasetton, G. Agresti, P. Zanuttigh, Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation, *IEEE Trans. Intell. Veh.* 5 (3) (2020) 508–518.
- [37] Y. Luo, L. Zheng, T. Guan, J. Yu, Y. Yang, Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2507–2516.
- [38] Y. Zou, Z. Yu, B. Kumar, J. Wang, Unsupervised domain adaptation for semantic segmentation via class-balanced self-training, in: *Proceedings of the European Conference on Computer Vision, ECCV*, 2018, pp. 289–305.
- [39] N. Araslanov, S. Roth, Self-supervised augmentation consistency for adapting semantic segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15384–15394.
- [40] J. Choi, T. Kim, C. Kim, Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6830–6840.
- [41] A. Marcu, V. Licaret, D. Costea, M. Leordeanu, Semantics through time: Semi-supervised segmentation of aerial videos with iterative label propagation, in: *Proceedings of the Asian Conference on Computer Vision*, 2020.
- [42] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [43] U. Rajapaksha, F. Sohel, H. Laga, D. Diepeveen, M. Bennamoun, Deep learning-based depth estimation methods from monocular image and videos: A comprehensive survey, *ACM Comput. Surv.* 56 (12) (2024) 1–51.
- [44] E. Games, Unreal engine 4, *Softw.* (2019) Available Online, <https://www.unrealengine.com>.
- [45] N. Zhang, L. Zhang, Z. Cheng, Towards simulating foggy and hazy images and evaluating their authenticity, in: *International Conference on Neural Information Processing*, Springer, 2017, pp. 405–415.
- [46] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.
- [47] G. Parmar, T. Park, S. Narasimhan, J.-Y. Zhu, One-step image translation with text-to-image models, 2024, arXiv preprint arXiv:2403.12036.
- [48] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

- [49] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017, arXiv preprint [arXiv:1706.05587](https://arxiv.org/abs/1706.05587).
- [50] TorchVision maintainers and contributors, TorchVision: PyTorch's computer vision library, GitHub Repos. (2016) <https://github.com/pytorch/vision>.
- [51] M. Schwonberg, J. Niemeijer, J.-A. Termöhlen, J.P. Schäfer, N.M. Schmidt, H. Gottschalk, T. Fingscheidt, Survey on unsupervised domain adaptation for semantic segmentation for visual perception in automated driving, *IEEE Access* 11 (2023) 54296–54336.
- [52] M. Toldo, A. Maracani, U. Michieli, P. Zanuttigh, Unsupervised domain adaptation in semantic segmentation: A review, *Technologies* (2020).
- [53] M. Chen, H. Xue, D. Cai, Domain adaptation for semantic segmentation with maximum squares loss, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2090–2099.