electronics

MDPI

Article

KoSim-GL: A Large-Scale Simulation-Based Dataset for UAV Cross-View Geo-Localization in Korean Urban Environments

Heejin Ahn [1], Changhwan Lee [1], Sangwook Lee [1], HyeonJoong Wi [2,3], Insung Jang [2] and Dong-Geol Choi [1,*]

[1] Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Republic of Korea; heejin.ahn@edu.hanbat.ac.kr (H.A.); changhwan.lee@edu.hanbat.ac.kr (C.L.); sangwook.lee@edu.hanbat.ac.kr (S.L.)
[2] Mobility Infra Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea; wjh4877@etri.re.kr (H.W.) ; e4dol2@etri.re.kr (I.J.)
[3] ICT (Computer Software), University of Science and Technology, Daejeon 34113, Republic of Korea
* Correspondence: dgchoi@hanbat.ac.kr

Abstract

We propose KoSim-GL, a large-scale vision-based geo-localization dataset for drone positioning in GPS-denied environments. Geo-localization estimates a drone's location by matching drone-view imagery against a geo-referenced satellite image database, offering a reliable alternative to GPS under conditions such as signal jamming, spoofing, or degradation in dense urban canyons. Although this task is challenging due to the domain gap between drone-view and satellite-view imagery, existing benchmarks are built predominantly around urban environments in the United States and China, leaving South Korea largely unrepresented, despite its distinctive landscape in which mountainous terrain coexists with dense high-rise districts and low-rise residential neighborhoods. To address this gap, we introduce KoSim-GL, constructed from drone-view images captured via an AirSim- and ROS-based flight simulator and satellite images collected through the Google Maps Tile API, covering the urban area of Daejeon, South Korea. Its key feature is a multi-view configuration that simultaneously captures five views, one nadir and four oblique, at each flight position across altitudes from 100 m to 600 m, enabling robust localization even in feature-sparse environments where nadir-only matching is prone to fail. In total, KoSim-GL comprises 2,450,315 drone images and 1704 satellite images. We further provide systematic comparisons against five existing benchmarks and baseline evaluations of ten representative geo-localization models under single- and multi-view settings. Experimental results show that the multi-view configuration substantially improves localization performance; for example, FSRA improves Recall@1 from 44.08% (single-view) to 65.37% (multi-view), a gain of 21.29 percentage points. The dataset is publicly available.

Keywords: multi drone; geo-localization; image matching; image retrieval; benchmark

Check for updates

Academic Editor: Kefeng Ji

Received: 15 May 2026
Revised: 10 June 2026
Accepted: 17 June 2026
Published: 19 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Geo-localization is a vision-based positioning technique that estimates a drone's location by matching drone-view images against a geo-referenced satellite image database, without relying on external signal infrastructure. This makes it a promising alternative in environments where GNSS signals become degraded or unreliable, such as dense urban canyons formed by high-rise buildings, and in scenarios affected by jamming or spoofing attacks [1–3]. As UAV applications continue to expand across diverse domains such as

precision agriculture, emergency rescue, and infrastructure inspection [4,5], UAVs are increasingly deployed in complex communication scenarios including relay-assisted networks, aerial base stations, and secure transmission systems [6,7]. In such communication-intensive deployments, UAVs must operate reliably in environments where GNSS signals are unavailable or compromised, making the importance of GNSS-independent positioning technology more critical than ever.

Previous works have proposed a variety of benchmark datasets for evaluating cross-view geo-localization algorithms. Most previous geo-localization datasets such as University-1652 [8] and SUES-200 [9] focused primarily on target-centric matching around specific buildings but were limited by narrow altitude ranges and a lack of support for free-flight trajectory scenarios, making them poorly suited for real-world deployment. Subsequent datasets including DenseUAV [10] and UAV-VisLoc [11] attempted to reflect autonomous flight scenarios more closely yet remained constrained by single-altitude, single-scene settings or narrow altitude coverage. More recently, simulation-based large-scale datasets such as GTA-UAV [12] have gained attention for their broad coverage of altitudes, scenes, and flight conditions. However, these datasets are grounded in Western virtual environments, limiting their generalizability to East Asian urban contexts such as South Korea. Furthermore, existing benchmarks rarely provide synchronized nadir and multi-directional oblique views from the same flight position to compensate for matching failures in feature-sparse regions such as mountains, rivers, and coastal areas, where nadir-only matching is prone to fail.

To address these limitations, we propose KoSim-GL, a large-scale cross-view geo-localization dataset targeting Korean urban environments. KoSim-GL covers the urban area of Daejeon, South Korea, and provides drone images across multiple altitudes (100–600 m), multiple scenes, and multiple satellite image scales. Crucially, five simultaneous viewing angles are captured at each flight position, comprising one nadir view and four oblique views. To evaluate matching performance across these views, we introduce an Any-Hit Recall@K metric that considers localization successful if at least one of the five views achieves a correct satellite image match, enabling robust evaluation even in environments where nadir-only matching is unreliable.

The main contributions of this paper are as follows.

- First, we propose KoSim-GL, the first cross-view geo-localization dataset targeting Korean urban environments.
- Second, KoSim-GL provides multi-altitude, multi-scene, multi-view, and multi-scale satellite imagery.
- Third, we introduce a Hit Rate metric based on multi-directional oblique views to compensate for nadir-view matching failures, offering an evaluation framework better aligned with real-world flight scenarios.
- Fourth, we conduct baseline evaluations on representative geo-localization models, establishing an experimental foundation for future research.

2. Related Work

2.1. Geo-Localization

Vision-based cross-view geo-localization has evolved through various deep learning techniques to overcome the severe domain gap between drone-view and satellite-view imagery, caused by differences in viewpoint, scale, and environmental conditions.

Early approaches focused on structural methods to effectively capture not only the image center but also surrounding context and geometric information. LPN [13] introduced a square-ring feature partition based on the distance from the image center, enabling robust matching against rotational variations. The transformer-based FSRA [14] automat-

ically segmented feature map regions based on heat distribution and aligned them in a one-to-one manner to resolve position and scale variation problems. CAMP [15] further combined a Position-aware Partition Branch (PPB) with intra-platform contrastive learning to simultaneously enhance spatial information and discriminative power within a scene.

Subsequent research shifted focus toward improving generalization in cross-area scenarios and under environmental variations encountered during real-world drone operations. MuSe-Net [16] dynamically adjusted domain shifts caused by extreme weather conditions such as rain and fog through multi-environment style extraction and self-adaptive feature extraction. MFRGN [17] prevented cross-area performance degradation by leveraging multi-scale features and a global-prompt attention mechanism without complex preprocessing. MCCG [18] utilized ConvNeXt [19] to minimize information loss through cross-dimension interaction and extracted multiple feature representations to improve robustness. DAC [20] effectively overcame appearance variations across domains via fine-grained feature domain alignment and a cross-batch scene consistency strategy.

More recently, the field has seen a growing trend toward incorporating large-scale Vision Foundation Models alongside advanced regularization and sampling techniques. DWDR [21] proposed a decorrelation regularization technique that minimizes inter-channel correlations, enabling diverse visual pattern extraction even within constrained feature dimensions. Sample4Geo [22] achieved state-of-the-art performance using hard negative sampling based on geographic proximity and visual similarity combined with a symmetric InfoNCE loss, without introducing complex additional modules. Notably, CV-Cities [23] integrated DINOv2 [24] into cross-view geo-localization and constructed a large-scale global dataset, demonstrating strong generalization performance at a global scale.

Several studies have also explored UAV geo-localization from different perspectives. Ding et al. [25] proposed a location classification-based cross-view matching method between UAV and satellite imagery. Zhuang et al. [26] introduced a multi-scale block attention network for faster and more effective UAV–satellite cross-view matching. Zhu et al. [27] incorporated UAV flight status information such as altitude into a fusion representation for improved geo-localization. Cui et al. [28] proposed a single-stage cross-view training method with a cross-view consistent attention module. Sun et al. [29] leveraged contrastive learning for multi-view scene matching between UAV and satellite images. Li et al. [30] designed a single-stream pyramid transformer for cross-view geo-localization of UAV and satellite images. Wang et al. [31] proposed a weight-adaptive multi-feature fusion network for UAV localization, and Chen et al. [32] developed a coarse-to-fine one-stream network for UAV geo-localization. Zhang et al. [33] introduced a cross-view matching method built on a digital twin platform for UAV geo-localization.

Studies published in Electronics have also investigated UAV localization and simulation-based approaches, including a review of indoor UAV positioning systems with machine learning [34], quantification of the simulation–reality gap for deep learning-based drone detection using AirSim [35], and deep reinforcement learning-based adaptive UAV navigation [36].

Transformer-based architectures have also contributed to this field, including the layer-to-layer transformer L2LTR [37], joint representation learning with keypoint detection [38], TransGeo [39], and UAV–satellite view synthesis [40]. Backbone networks such as ViT [41] and ResNet [42] underpin many of these approaches.

Despite these advances, benchmark datasets that sufficiently reflect the complexity of real-world flight environments remain essential for comprehensive model evaluation. This is discussed in detail in the following subsection.

2.2. Geo-Localization Benchmark

To evaluate cross-view geo-localization algorithms, various benchmark datasets have been proposed, and their main characteristics are summarized in Table 1. Most previous geo-localization datasets focused primarily on matching centered around specific objects or buildings. University-1652 [8] collected synthetic drone, satellite, and ground camera images targeting 1652 university buildings worldwide at multiple altitudes ranging from 121.5 m to 256 m; however, its target-centric design makes it unsuitable for evaluating localization along real-world free-path flight trajectories. SUES-200 [9] provided images at four altitude levels ranging from 150 m to 300 m for the same target, but its lack of contiguous flight areas creates a significant gap with real-world operational scenarios. DenseUAV emerged to reflect autonomous flight scenarios by densely sampling drone images in low-altitude (80–100 m) urban environments; however, its single-altitude, single-scene constraint limits its applicability to diverse real-world urban flight scenarios. UAV-VisLoc [11], covering 11 regions across China, proposed a coordinate estimation task on large-scale satellite maps but focused exclusively on high-altitude flights from 400 m to 2000 m, failing to reflect diverse altitude variations within a single scene.

Table 1. Comparison of UAV geo-localization benchmark datasets. Checkmarks (✓) indicate supported features, × indicates unsupported features, and - indicates not applicable. In the "Drone-view GPS locations" row, "Aligned" denotes datasets in which drone positions are anchored to predefined target centers under a one-to-one, center-aligned matching assumption, whereas "Arbitrary" denotes datasets in which drone positions are sampled freely along flight trajectories. In KoSim-GL, a drone image is paired with a satellite patch when the ground center point (latitude, longitude) of its nadir center-camera view falls within the geographic boundary of that patch. KoSim-GL is the only dataset to jointly satisfy all listed criteria.

	University-1652	SUES-200	DenseUAV	UAV-VisLoc	GTA-UAV	KoSim-GL (Ours)
Drone images	37,854	24,210	18,198	6742	33,763	2,450,315
Drone-view GPS locations	Aligned	Aligned	Aligned	-	Arbitrary	Arbitrary
Altitude range	121.5–256 m	150–300 m	80–100 m	400–2000 m	80–650 m	100–600 m
Contiguous area	×	×	✓	✓	✓	✓
Multiple altitudes	✓	×	×	×	✓	✓
Multiple scenes	×	×	×	✓	✓	✓
Multiple scales satellite images	×	×	×	-	✓	✓
Multi-view support	1 oblique	1 oblique	1 nadir	1 nadir	1 nadir	5 views (1 nadir + 4 oblique)

Recently, large-scale simulation-based datasets utilizing game engines have gained significant attention as a means to overcome the high costs and regulatory constraints of real-world data collection. GTA-UAV [12] covers diverse flight altitudes (80–650 m), attitudes, and scenes over a broad contiguous area, bridging the gap with real-world scenarios by introducing partial matching and meter-based evaluation. However, being grounded in a Western virtual environment, it struggles to generalize to the distinct urban morphology of East Asian cities such as South Korea. Moreover, all aforementioned datasets are predominantly confined to Chinese urban environments or Western virtual environments, failing to reflect the distinct architectural styles and spatial characteristics of South Korean urban areas. Furthermore, while some existing datasets provide a single oblique or nadir view, few existing benchmarks simultaneously provide both nadir and multiple oblique views from a single flight position alongside multi-scale satellite imagery.

KoSim-GL, proposed in this paper, is the first cross-view geo-localization dataset targeting Korean urban environments, addressing all the limitations outlined above. Cen-

tered on the urban area of Daejeon, South Korea, it provides drone images across multiple altitudes (100–600 m), multiple scenes, and multiple satellite image scales over a contiguous area. In addition, five simultaneous viewing angles are captured at each flight position, comprising one nadir view and four oblique views, enabling robust localization evaluation even in feature-sparse environments where nadir-only matching is prone to fail.

3. The KoSim-GL Dataset

3.1. Problem Definition

Cross-view geo-localization aims to estimate a drone's location by retrieving, from a geo-referenced satellite image database $\mathcal{G} = \{g_1, g_2, \ldots, g_N\}$, the satellite image that corresponds to the same geographic location as the drone-view query. We formulate the task under two settings: a single-view setting, which uses a single drone-view image as the query, and a multi-view setting, which uses the five synchronized drone-view images captured at the same flight position.

Single-view setting. Given a single drone-view query image q, the goal is to retrieve the satellite image g^* that best matches q:

$$g^* = \arg\max_{g_i \in \mathcal{G}} \text{sim}(f(q), f(g_i)), \tag{1}$$

where $f(\cdot)$ denotes the feature extraction function and $\text{sim}(\cdot, \cdot)$ denotes the similarity between two feature vectors.

Multi-view setting. At each flight position, five synchronized drone-view images are captured, denoted $\mathcal{Q} = \{q^{(1)}, q^{(2)}, \ldots, q^{(5)}\}$, comprising one nadir view and four oblique views. For each view $q^{(v)}$, let $\mathcal{R}_K(q^{(v)})$ be the set of its top-K retrieved satellite images:

$$\mathcal{R}_K(q^{(v)}) = \text{Top-}K\Big(\text{sim}\Big(f(q^{(v)}), f(g_i)\Big) \;\Big|\; g_i \in \mathcal{G}\Big). \tag{2}$$

- **Input:** In the single-view setting, a single drone-view image; in the multi-view setting, five drone-view images captured simultaneously by the onboard camera system (one nadir view and four oblique views), together with their corresponding GPS coordinates (latitude, longitude) and flight-altitude metadata.
- **Output:** Top-K candidate satellite images retrieved from $\mathcal{G}$, along with their corresponding GPS coordinates as the estimated drone location.

3.2. Multi-View Camera System

Previous drone-based geo-localization datasets have provided only a single drone-view image per flight position. However, this approach has limitations: the observable ground information becomes increasingly restricted at higher altitudes, and it is difficult to extract sufficient feature points from building facades or forested areas whose appearance changes seasonally. Furthermore, few existing datasets simultaneously provide both nadir and oblique views from a single flight position, which can lead to degraded localization accuracy in feature-sparse environments. To address these limitations, we designed a multi-view camera system that simultaneously captures nadir and multi-directional oblique views from a single flight position, as illustrated in Figure 1.

Each drone carries five cameras. The Center CAM is mounted at the drone body center, facing directly downward (nadir direction) to provide a top-down view. The Front, Back, Left, and Right CAMs are each positioned 1 m from the center along their respective axes and tilted 15° outward from the nadir direction, providing oblique views in four cardinal directions. This four-directional oblique camera configuration was designed to acquire complementary spatial information in multiple directions, including building facades, road

layouts, and topographical features that may be missed in nadir-only views. As shown in Figure 1C, the five cameras simultaneously capture the same geographic location from different viewpoints, demonstrating that each view provides distinct and complementary spatial information. The number of cameras was limited to five because increasing the number of cameras caused a rapid increase in ROS-based data processing load, leading to system instability. Therefore, five cameras represented the maximum stable configuration in our simulation environment.

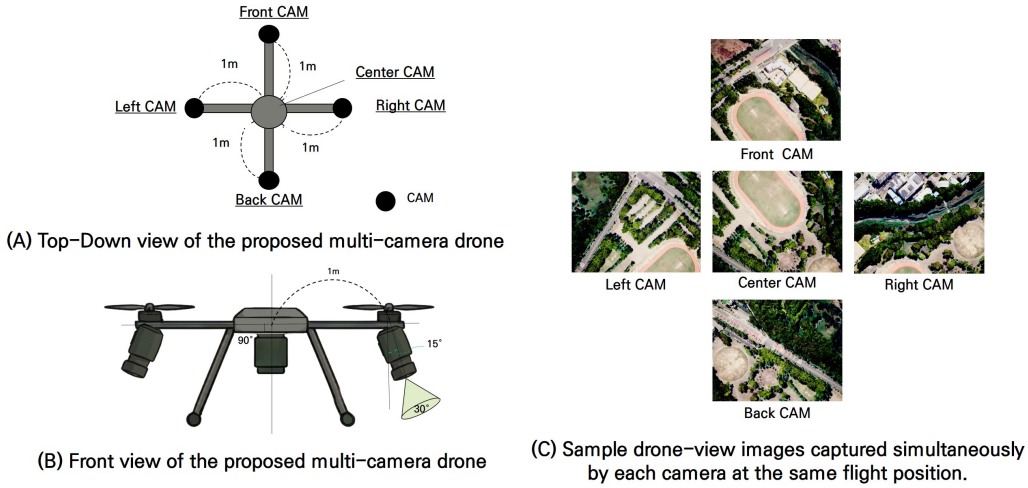


Figure 1. Multi-camera drone configuration and sample views. (**A**) Top-down view of the proposed multi-camera drone platform, showing five cameras arranged in a cross pattern with 1 m spacing from the center. (**B**) Front view illustrating the tilt angle of the oblique cameras. The Center CAM faces directly downward (90° from horizontal, i.e., nadir), while each oblique CAM is tilted 15° from the nadir direction with a 30° Field of View (FOV). (**C**) Sample drone-view images captured simultaneously by each camera at the same flight position. The Center CAM provides a nadir view of the ground, while the Front, Back, Left, and Right CAMs capture oblique views in their respective directions, providing complementary spatial information including building facades and surrounding terrain.

All cameras were configured with a unified resolution of 640 × 480 and a Field of View (FOV) of 30°. The narrow FOV of 30° was selected to minimize lens distortion at image edges. The 15° tilt angle was chosen so that the near edge of each oblique camera's footprint coincides with the nadir point directly below the drone, ensuring spatial continuity between the nadir and oblique views. The resulting ground-level overlap between the center and oblique footprints can be derived geometrically as follows. Let h denote the flight altitude. With a half-FOV of $\alpha = 15^\circ$ and a tilt angle of $\theta = 15^\circ$ from nadir, the near and far edges of the oblique footprint are located at ground distances of $h\tan(\theta - \alpha) = h\tan(0^\circ) = 0$ and $h\tan(\theta + \alpha) = h\tan(30^\circ)$ from the nadir point, respectively. The oblique footprint width is therefore $W_{\text{obl}} = h\tan(30^\circ)$. The center (nadir) camera covers a ground half-width of $h\tan(\alpha) = h\tan(15^\circ)$, so the overlap width equals $h\tan(15^\circ)$. The overlap ratio relative to the oblique footprint is thus:

$$r = \frac{h\tan 15^\circ}{h\tan 30^\circ} = \frac{\tan 15^\circ}{\tan 30^\circ} \approx \frac{0.268}{0.577} \approx 46\% \tag{3}$$

which is independent of flight altitude. The 1 m physical offset of each oblique camera from the center introduces a fixed near-edge shift of 1 m on the ground, which amounts to less than 2% of the footprint width even at the minimum altitude of 100 m and is therefore negligible. The drone flew at a speed of 3 m/s across six altitude levels from 100 m to 600 m in 100 m increments. At each altitude, the five camera images were grouped by index.

Groups in which the average distance between the GPS points of the five cameras exceeded 1 m were considered to have positional inconsistencies and were excluded from the dataset.

3.3. Data Collection

The KoSim-GL dataset was constructed using the AirSim (v1.8.1) [43] simulator and ROS 2 Galactic [44]. AirSim is an open-source, cross-platform simulator for drones and vehicles built on Unreal Engine, and ROS is a standard framework for robot software development that enables efficient communication and data exchange between heterogeneous hardware and software modules. The overall data generation pipeline is illustrated in Figure 2. Simulation-based data collection effectively addresses the time, cost, and safety constraints associated with real-world drone operations, while offering the advantage of systematically controlling diverse altitudes, viewpoints, and environmental conditions. However, a sim-to-real gap may exist due to visual differences between simulated and real environments, such as lighting and seasonal conditions, which remains a challenge to be addressed in future work. The Daejeon simulation environment was constructed as a three-dimensional space from drone footage captured directly over the actual city, faithfully reproducing real building geometries, road layouts, textures, and vegetation. The remaining sim-to-real gap is therefore primarily temporal and photometric, arising from seasonal differences between the model's capture date and the satellite imagery and from the fixed clear-morning illumination used in AirSim. Nevertheless, KoSim-GL serves as a large-scale dataset reflecting the diverse geographical and architectural characteristics of Korean urban environments, providing a foundation for future real-world drone localization research.

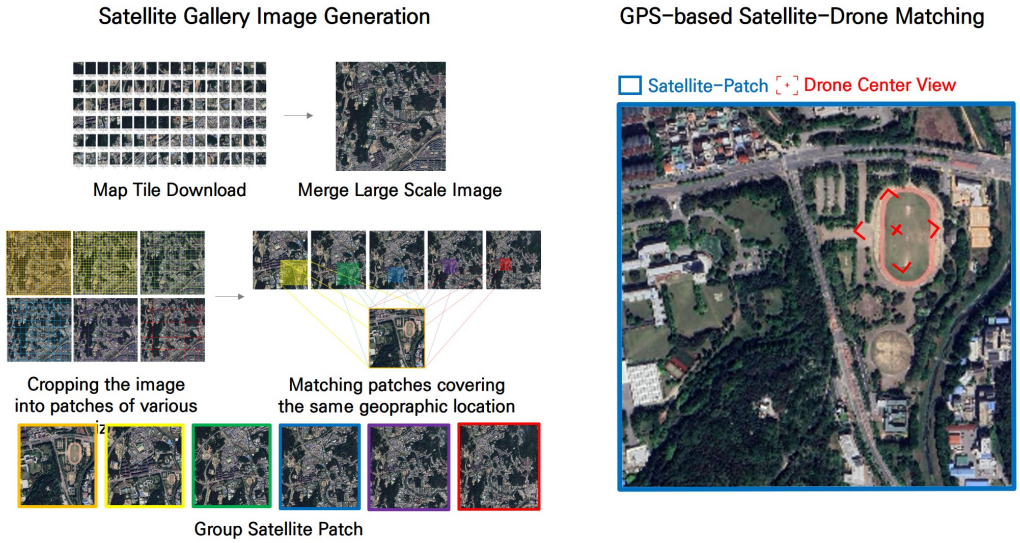


Figure 2. Satellite and drone data generation pipeline. This pipeline shows the overall process of constructing the KoSim-GL dataset, including virtual drone flight simulation using AirSim and ROS, multi-view image capture at multiple altitudes, and satellite patch extraction based on drone GPS coordinates. The colored boxes indicate satellite patches at different altitudes: orange (100 m), yellow (200 m), green (300 m), blue (400 m), purple (500 m), and red (600 m).

Data was collected over the Sinseong-dong, Doryong-dong, and Gajeong-dong areas of Daejeon, South Korea. This region was selected for its diverse geographical features, including forests, dense residential areas, high-rise apartment complexes, research institutes, and landmarks, making it well suited for dataset construction.

To ensure dataset diversity and generalization performance, the drone flew along five distinct paths as shown in Figure 3. Each path was designed to include a variety of

geographical features, such as high-rise apartment complexes, low-rise residential areas, forests, stadiums, and research institutes.

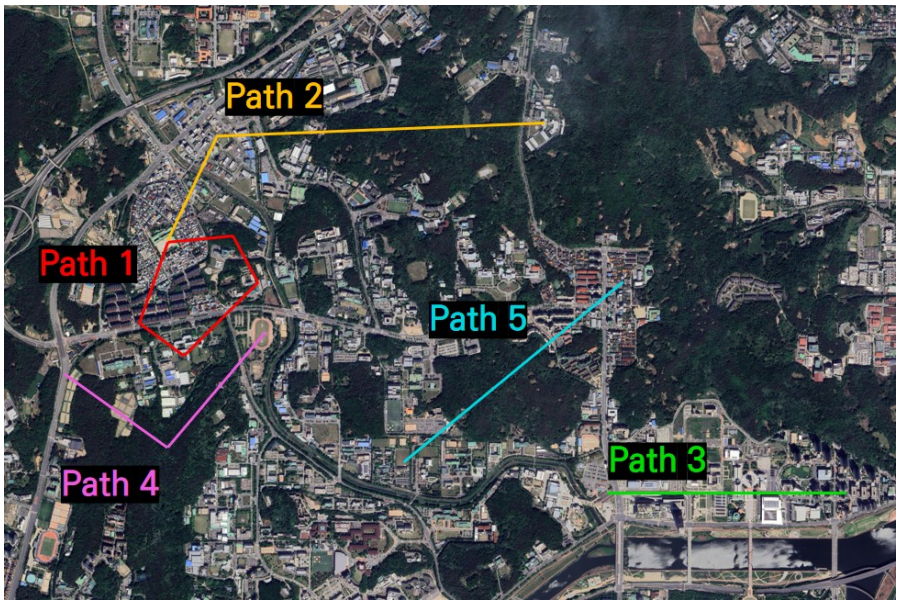


Figure 3. Flight path overview in Daejeon region. This figure shows five distinct flight paths collected over the Sinseong-dong area of Daejeon, South Korea. Each path was flown at multiple altitudes ranging from 100 m to 600 m in 100 m increments, covering diverse geographical and architectural environments including high-rise apartment complexes, low-rise residential areas and mountainous terrain, and research institutes and educational facilities.

The GPS coordinates (latitude, longitude) and flight altitude of each drone image are stored as metadata. These GPS coordinates serve two purposes. First, they are used as the reference for matching drone images to their corresponding satellite patches. Second, they serve as ground truth for evaluating the accuracy of localization algorithms.

The flight paths are as follows:

- **Path 1 (2.280 km):** Geumseong Elementary School → Lucky Hana Apartment → Hanwha Solutions Central Research Institute → Korea Research Institute of Chemical Technology → Geumseong Elementary School
- **Path 2 (2.593 km):** Geumseong Elementary School → Hanwoo Kimsatgat Restaurant → Samyang Group R&D Center
- **Path 3 (1.380 km):** National Science Museum → TJB Daejeon Broadcasting
- **Path 4 (1.679 km):** Sinseong Crossroads → Sinseong Neighborhood Park (Mountain) → Daedeok Research Complex Sports Complex
- **Path 5 (1.780 km):** International Intellectual Property Training Institute → Daedeok High School

3.4. Satellite Image Construction

A satellite reference map was constructed using the Google Maps Tile API [45]. First, GPS boundary coordinates covering the entire Sinseong-dong area were defined, and all satellite tiles within this region were downloaded at zoom level 17 of the Google Maps tile coordinate system. This corresponds to a ground sampling distance of 1.194 m/pixel at the equator and approximately 0.96 m/pixel at the dataset's latitude of 36.35° N in Daejeon, South Korea. The downloaded tiles were sorted according to their tile coordinates (x, y) and merged into a single large-scale satellite map image. The GPS coordinates for each tile were calculated using an inverse transformation function.

From this large-scale satellite map, satellite patches were generated by cropping the map into six sizes corresponding to drone flight altitudes from 100 m to 600 m. Using the 100 m altitude patch as the reference, patches at 200–600 m altitudes that fully encompass the reference patch area were matched to construct a set of six multi-scale satellite patches for each location. This design ensures that a corresponding satellite patch exists for any drone flight altitude, enabling robust localization across varying altitudes. Gallery folders were created based on the number of 100 m patches, and a drone image was matched to a satellite patch as a gallery–query pair when the ground center point (latitude, longitude) of its nadir center-camera view fell within the geographic boundaries of the satellite patch. All satellite maps and patches were reprojected into the UTM coordinate system for precise geo-localization.

Figure 4 provides an overview of the diverse environments captured in KoSim-GL, including urban areas, forests, farmland, schools, and research institutes. The dataset was split into training and test sets by randomly partitioning 284 locations into 255 training and 29 test locations. Since each location is uniquely defined by its GPS coordinates and matched to satellite patches based on geographic boundaries, spatial overlap between training and test queries does not occur at the image level. Furthermore, the diversity introduced by six altitude levels and five simultaneous viewpoints at each location reduces the likelihood of localization through simple memorization of nearby scenes. We note, however, that since the dataset is collected over a contiguous area using a limited number of flight paths and a multi-directional (five-view) camera configuration, neighboring satellite patches and adjacent trajectories may introduce a degree of spatial correlation between nearby training and test locations; a stricter geographically disjoint split is left for future work. This random partitioning strategy is consistent with existing benchmarks such as GTA-UAV, which also employs random splitting over contiguous areas. The training set consists of 255 locations with 2,188,425 drone images and 1530 satellite images, while the test set consists of 29 locations with 261,890 query drone images and 174 gallery satellite images. In total, KoSim-GL comprises 2,450,315 drone images and 1704 satellite images across 284 locations.

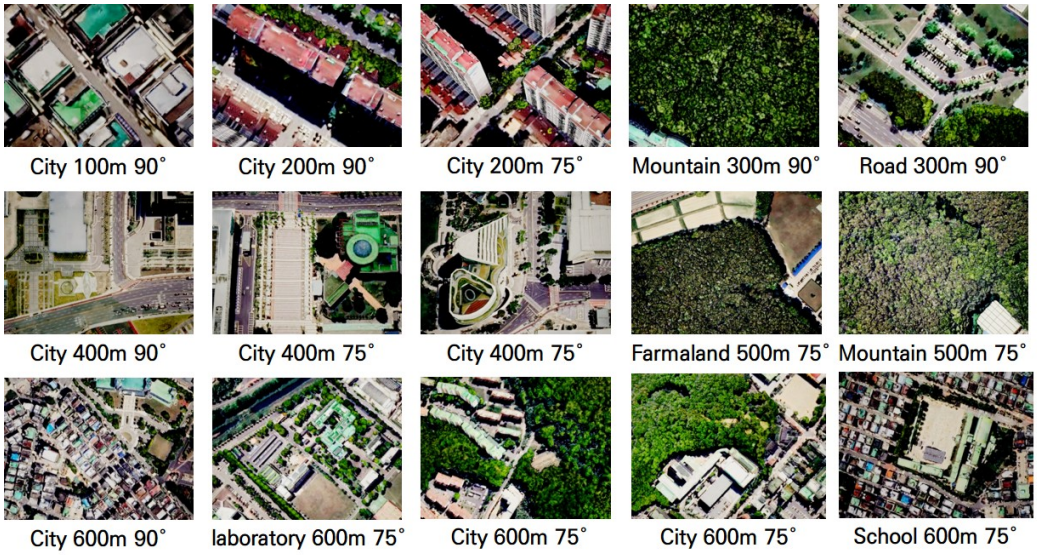


Figure 4. Dataset overview. The KoSim-GL dataset contains multi-scene, multi-altitude, and multi-view UAV images taken across diverse urban areas in Daejeon, South Korea, as well as satellite reference maps. Each UAV image shows its flight altitude and pitch angle below.

4. Experiments

4.1. Experimental Setup

To validate the benchmark reliability of the proposed KoSim-GL dataset, we conduct single-view and multi-view cross-view geo-localization experiments. Representative methods covering diverse backbone architectures, including CNN, ViT [41], and ConvNeXt [19], are selected to verify the generalizability of the dataset without bias toward any specific model architecture. All models are retrained from scratch on KoSim-GL rather than directly evaluated with weights pretrained on other datasets. For fair comparison, all methods strictly follow the official implementations and default configurations provided by the original authors—including the optimizer, learning-rate schedule, and other hyperparameters—and no additional hyperparameter tuning is performed; each backbone is initialized with ImageNet-pretrained weights.

The only exceptions are CVCities and MFRGN: under their official batch sizes, the GPU memory was insufficient, so we reduced the batch size and correspondingly re-adjusted the learning rate. These changes were made solely to accommodate hardware constraints, not to improve performance. For CVCities, the adjustment applies only to the multi-view setting (batch size 128 and learning rate 0.04), whereas the single-view setting follows the official configuration (batch size 16 and learning rate 0.005). For MFRGN, the adjustment applies to both the single-view and multi-view settings. All experiments were conducted on NVIDIA GeForce RTX 3090 GPUs (NVIDIA Corporation, Santa Clara, CA, USA). The per-method training configurations are summarized in Table 2.

Table 2. Training configurations for each method. † Batch size and learning rate are adjusted due to memory constraints (CVCities: multi-view setting only; MFRGN: both settings).

Method	View	Input Size	LR	Batch Size	Epochs
MuseNet (PR'24) [16]	-	256 × 256	0.005	8	210
DWDR (TGRS'24) [21]	-	256 × 256	0.003	8	120
LPN (TCSVT'22) [13]	-	256 × 256	0.001	8	120
FSRA (TCSVT'21) [14]	-	256 × 256	0.01	8	120
CVCities † (JSTARS'24) [23]	single multi	448 × 448	0.005 0.04	16 128	40
MCCG (TCSVT'24) [18]	-	256 × 256	0.01	8	200
Sample4Geo (ICCV'23) [22]	-	384 × 384	0.01	8	1
DAC (TCSVT'24) [20]	-	384 × 384	0.001	24	1
CAMP (TGRS'24) [15]	-	384 × 384	0.001	24	1
MFRGN † (ACM MM'24) [17]	-	384 × 384	0.00017	32	3

4.2. Metric

Model performance is evaluated using Recall@K (K = 1, 5, 10) and Average Precision (AP), which are widely adopted metrics in cross-view geo-localization. Recall@K measures the proportion of queries for which the correct match is found within the top-K retrieved results, while AP represents the mean precision across all ranking positions.

For multi-view evaluation, we introduce an Any-Hit Recall@K metric. A query location is considered successfully localized if at least one of the five simultaneously captured views, comprising one nadir view and four oblique views, retrieves the correct match within the top-K results. This evaluation scheme reflects the design intent of KoSim-GL's multi-view configuration, which enables localization through oblique views in feature-sparse environments such as mountains and forests, where nadir-only matching is prone to fail.

4.3. Multi-View Evaluation

Table 3 presents the multi-view evaluation results on KoSim-GL. Localization performance was measured by grouping five simultaneously captured camera views at the same altitude into a single flight position.

Table 3. Multi-view experimental results on the KoSim-GL dataset. The best results are highlighted in **bold**.

Method	R@1	R@5	R@10	AP
MuseNet (PR'24) [16]	64.93	95.75	99.54	36.89
DWDR (TGRS'24) [21]	60.38	91.95	98.86	**38.08**
LPN (TCSVT'22) [13]	62.90	95.64	**99.63**	37.34
FSRA (TCSVT'21) [14]	**65.37**	95.94	**99.63**	37.49
CVCities (JSTARS'24) [23]	45.64	90.76	97.28	25.44
MCCG (TCSVT'24) [18]	56.18	**96.32**	99.21	32.54
Sample4Geo (ICCV'23) [22]	44.67	89.64	95.57	24.41
DAC (TCSVT'24) [20]	45.48	91.59	99.28	25.90
CAMP (TGRS'24) [15]	48.19	92.47	96.76	23.61
MFRGN (ACM MM'24) [17]	44.87	89.77	93.82	25.04

Part-based methods achieved the highest performance in the multi-view setting. LPN [13] and FSRA [14] achieved R@1 of 62.90% and 65.37%, respectively, while MuseNet [16] and DWDR [21] also recorded strong performance at 64.93% and 60.38%. This suggests that methods learning spatially partitioned local features benefit from higher inter-view complementarity, as the discriminative parts activated vary with camera orientation, enabling effective localization in multi-view settings.

In contrast, contrastive learning-based methods such as Sample4Geo [22] (44.67%), DAC [20] (45.48%), CVCities [23] (45.64%), MFRGN [17] (44.87%), and CAMP [15] (48.19%) showed relatively lower performance. This is attributed to their structural emphasis on view-invariant global representations or scene consistency learning, which tends to produce similar feature responses across views regardless of camera orientation, thereby limiting inter-view complementarity in multi-view settings. These results suggest that a fundamental trade-off exists between single-view robustness and multi-view complementarity.

4.4. Single-View Evaluation

Table 4 presents the baseline evaluation results under the single-view setting, conducted for comparison with the multi-view evaluation. In the single-view setting, nadir-view and oblique-view images are each used independently as queries to measure localization performance.

FSRA [14] achieved the highest R@1 of 44.08%, followed by LPN [13] (42.74%) and DWDR [21] (41.11%). Notably, MCCG [18] achieved R@1 of 42.52% and R@5 of 91.15%, showing competitive results among all evaluated methods.

In contrast, CVCities [23] (R@1 37.97%), Sample4Geo [22] (35.53%), DAC [20], CAMP [15], and MFRGN [17] showed relatively lower performance, with AP scores all below 21%, revealing limitations in generalizing to the proposed dataset.

Overall, methods that performed poorly under the single-view setting showed significant performance improvements under the multi-view setting. This suggests that the multi-view configuration effectively compensates for degraded localization accuracy in

feature-sparse environments, demonstrating the value of simultaneously capturing nadir and oblique views in geo-localization benchmarks.

To highlight the effectiveness of the multi-view configuration, we directly compare the single-view and multi-view results. For instance, FSRA improves from R@1 44.08% under the single-view setting to 65.37% under the multi-view setting, representing a gain of 21.29 percentage points. These consistent improvements across diverse methods demonstrate that the proposed multi-view configuration provides substantial localization gains, particularly in environments where a single viewpoint fails to capture sufficient discriminative features.

Table 4. Single-view experimental results on the KoSim-GL dataset. The best results are highlighted in **bold**.

Method	R@1	R@5	R@10	AP
MuseNet (PR'24) [16]	40.68	86.24	95.27	26.08
DWDR (TGRS'24) [21]	41.11	88.18	97.14	27.71
LPN (TCSVT'22) [13]	42.74	85.49	97.60	29.00
FSRA (TCSVT'21) [14]	**44.08**	88.17	97.94	29.13
CVCities (JSTARS'24) [23]	37.97	82.62	90.55	18.65
MCCG (TCSVT'24) [18]	42.52	**91.15**	**98.22**	27.91
Sample4Geo (ICCV'23) [22]	35.53	82.05	91.14	20.43
DAC (TCSVT'24) [20]	37.11	84.82	89.38	19.42
CAMP (TGRS'24) [15]	36.26	84.16	93.13	19.64
MFRGN (ACM MM'24) [17]	38.02	74.37	86.35	**20.82**

4.5. View-Wise Retrieval Analysis

To analyze retrieval performance according to viewing angle, we divided the evaluation queries into three conditions: vertical capture (Nadir), oblique capture (Oblique), and All, which includes both conditions, and measured Recall@*K* and AP under the Any-Hit criterion. The results for the FSRA model are presented in Table 5.

Table 5. View-wise retrieval performance of FSRA under the Any-Hit Recall@*K* metric (%). Nadir uses only the nadir view as the query; Oblique applies Any-Hit over the four oblique views; All applies Any-Hit over all five views. The best results are highlighted in **bold**.

Method	View	R@1	R@5	R@10	AP
FSRA (TCSVT'21) [14]	Nadir	41.79	88.48	97.73	31.12
	Oblique	63.75	95.58	99.40	37.07
	All	**65.37**	**95.94**	**99.63**	**37.49**

The most prominent trend is that the oblique view (Oblique) substantially outperforms the nadir view (Nadir) across all metrics. A gap of 21.96%p in R@1 (63.75% vs. 41.79%) and 5.95%p in AP (37.07% vs. 31.12%) is observed, with the largest difference appearing in the ability to retrieve the correct match as the top-ranked result. This is attributed to the fact that oblique views can capture a wider range of spatial information surrounding the drone's position, including building facades, roads, and adjacent terrain, providing higher discriminability between locations, whereas nadir views focus on the ground directly below, making it difficult to distinguish between visually similar scenes.

The gap between the two views narrows rapidly as K increases, converging to 7.10%p at R@5 and 1.67%p at R@10, with both conditions approaching saturation above 97% at R@10. This suggests that the difficulty of the nadir view lies not in failing to retrieve the correct answer altogether but rather in ranking precision, specifically in distinguishing the correct match from visually similar candidates.

The All condition achieves the highest performance across all metrics with R@1 65.37% and AP 37.49%. This quantitatively demonstrates that the multi-view configuration compensates for the low discriminability of the nadir view with the richer spatial cues provided by oblique views, thereby improving overall retrieval robustness.

4.6. Qualitative Analysis

Figure 5 presents a qualitative comparison of retrieval results under single-view and multi-view settings. The top row shows the single-view result using only the Center CAM, while the bottom row shows the multi-view result using all five cameras simultaneously. Under the single-view setting, the highest-ranked GT patch is located at Rank-4, whereas the multi-view setting successfully retrieves the GT patch at Rank-1. This demonstrates that the proposed multi-view configuration effectively improves localization performance by leveraging complementary spatial information from multiple viewing directions.

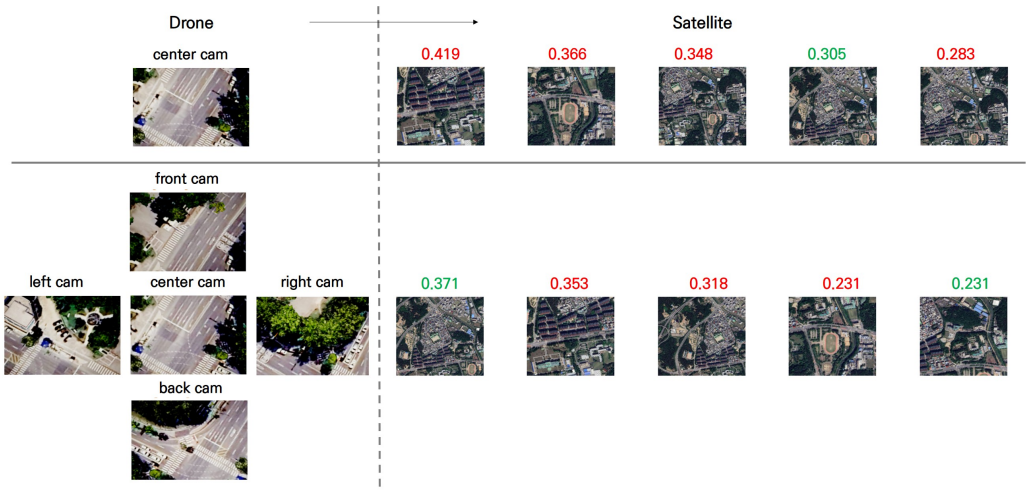


Figure 5. Qualitative comparison of single-view and multi-view retrieval results. The left side shows the drone-view query images and the right side shows the top-5 retrieved satellite images. Green scores indicate ground truth (GT) patches and red scores indicate incorrect matches.

Figure 6 presents a qualitative example in a feature-sparse environment where the Center CAM image is dominated by an open area with few distinguishable features, making nadir-only matching difficult. Under the single-view setting, the top-3 retrieved results are all incorrect, placing the GT patch at Rank-4. In contrast, the multi-view setting leverages complementary spatial information from oblique views, including building facades and surrounding structures, successfully retrieving the GT patch at Rank-1. This demonstrates that the effectiveness of the proposed multi-view configuration becomes more pronounced in feature-sparse environments.

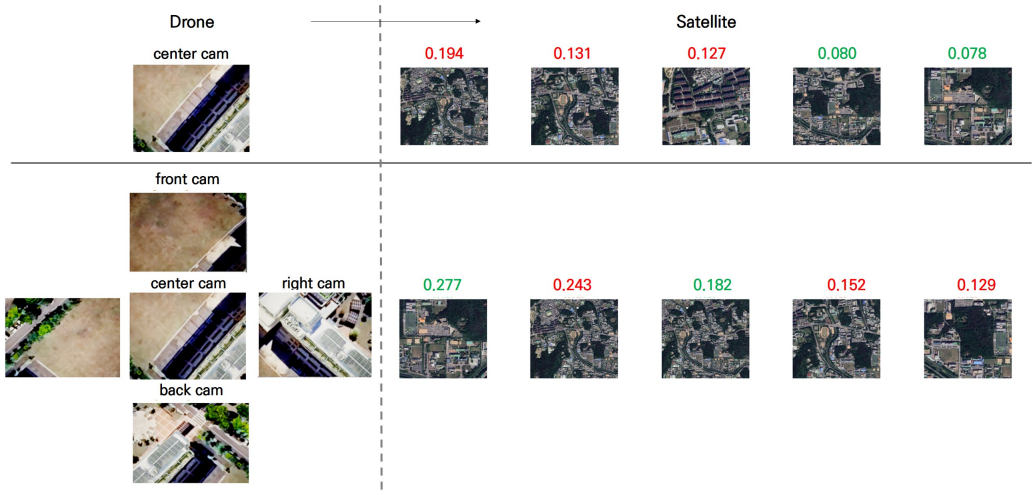


Figure 6. Qualitative comparison in feature-sparse environments. The left side shows the drone-view query images and the right side shows the top-5 retrieved satellite images. Green scores indicate ground truth (GT) patches and red scores indicate incorrect matches.

5. Conclusions

In this paper, we propose KoSim-GL, a large-scale cross-view geo-localization dataset targeting Korean urban environments. Constructed using an AirSim- and ROS-based drone flight simulator, KoSim-GL covers the Sinseong-dong area of Daejeon, South Korea, comprising 2,450,315 drone images and 1704 satellite images, making it the first geo-localization dataset for Korean urban environments.

By providing multiple altitudes (100–600 m), multi-directional views (nadir and oblique), contiguous area coverage, free-path flight trajectories, and multi-scale satellite imagery through a five-camera multi-view platform, KoSim-GL addresses several key limitations of existing datasets.

Baseline evaluations of 10 representative geo-localization models under both single-view and multi-view settings confirm that the multi-view configuration effectively compensates for degraded localization accuracy in feature-sparse environments. In particular, FSRA improves Recall@1 from 44.08% to 65.37% (a 21.29 percentage-point gain) when moving from the single-view to the multi-view setting, demonstrating that KoSim-GL serves as a challenging dataset for existing methods and a solid experimental foundation for future drone localization research in Korean urban environments.

We note, however, that the dataset is limited to outdoor urban and natural scenes, is constructed in simulation, and currently covers only a single urban region in Korea (Daejeon). It also does not include fully GNSS-denied environments such as tunnels or indoor spaces, where satellite signals are physically blocked.

In future work, we plan to extend the dataset to cover more diverse cities, environments, and weather conditions and to validate the simulation-to-real transfer through real drone flights. We also identify breakdown analyses by altitude, scene type, and feature-sparse versus urban regions as explicit targets for future releases of KoSim-GL and further aim to synthesize the five camera views into a panoramic representation that integrates 360° spatial information into a single input.

Author Contributions: Conceptualization, H.A., C.L. and S.L.; methodology, H.A., C.L. and S.L.; software, H.A., C.L. and S.L.; validation, H.A., C.L. and S.L.; formal analysis, H.A., C.L. and S.L.; investigation, H.A., C.L. and S.L.; data curation, H.A., C.L. and S.L.; writing—original draft preparation, H.A., C.L. and S.L.; writing—review and editing, H.A., C.L. and S.L.; visualization, H.A., C.L. and S.L.; resources, H.W., I.J. and D.-G.C.; funding acquisition, H.W., I.J. and D.-G.C.; supervision, D.-G.C.; project administration, D.-G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (26ZR1110, Research on Intelligent Industrial Convergence), and partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2026-25491676) and the Regional Innovation System & Education (RISE) program through the Daejeon RISE Center, funded by the Ministry of Education (MOE) and the Daejeon Metropolitan City, Republic of Korea (2026-RISE-06-002).

Data Availability Statement: The KoSim-GL dataset is publicly available at https://github.com/ahn-heejin/KoSim-GL.git (accessed on 16 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gyagenda, N.; Hatilima, J.V.; Roth, H.; Zhmud, V. A review of GNSS-independent UAV navigation techniques. *Robot. Auton. Syst.* **2022**, *152*, 104069. [CrossRef]
2. Arafat, M.Y.; Alam, M.M.; Moh, S. Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges. *Drones* **2023**, *7*, 89. [CrossRef]
3. Couturier, A.; Akhloufi, M.A. A review on deep learning for UAV absolute visual localization. *Drones* **2024**, *8*, 622. [CrossRef]
4. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
5. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tuts.* **2019**, *21*, 2334–2360. [CrossRef]
6. Bai, Y.; Zhao, H.; Zhang, X.; Chang, Z.; Jäntti, R.; Yang, K. Toward autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches. *IEEE Commun. Surv. Tuts.* **2023**, *25*, 3038–3067. [CrossRef]
7. Dong, R.; Wang, B.; Cao, K.; Tian, J.; Cheng, T. Secure transmission design of RIS enabled UAV communication networks exploiting deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 8404–8419. [CrossRef]
8. Zheng, Z.; Wei, Y.; Yang, Y. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1395–1403.
9. Zhu, R.; Yin, L.; Yang, M.; Wu, F.; Yang, Y.; Hu, W. SUES-200: A multi-height multi-scene cross-view image benchmark across drone and satellite. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 4825–4839. [CrossRef]
10. Dai, M.; Zheng, E.; Feng, Z.; Qi, L.; Zhuang, J.; Yang, W. Vision-based UAV self-positioning in low-altitude urban environments. *IEEE Trans. Image Process.* **2024**, *33*, 493–508. [CrossRef] [PubMed]
11. Xu, W.; Yao, Y.; Cao, J.; Wei, Z.; Liu, C.; Wang, J.; Peng, M. UAV-VisLoc: A large-scale dataset for UAV visual localization. *arXiv* **2024**, arXiv:2405.11936.
12. Ji, Y.; He, B.; Tan, Z.; Wu, L. Game4Loc: A UAV geo-localization benchmark from game data. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25–27 February 2025; Volume 39, pp. 3913–3921.
13. Wang, T.; Zheng, Z.; Yan, C.; Zhang, J.; Sun, Y.; Zheng, B.; Yang, Y. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 867–879. [CrossRef]
14. Dai, M.; Hu, J.; Zhuang, J.; Zheng, E. A transformer-based feature segmentation and region alignment method for UAV-view geo-localization. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 4376–4389.
15. Wu, Q.; Wan, Y.; Zheng, Z.; Zhang, Y.; Wang, G.; Zhao, Z. CAMP: A cross-view geo-localization method using contrastive attributes mining and position-aware partitioning. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5637614. [CrossRef]
16. Wang, T.; Zheng, Z.; Sun, Y.; Yan, C.; Yang, Y.; Chua, T.-S. Multiple-environment self-adaptive network for aerial-view geo-localization. *Pattern Recognit.* **2024**, *152*, 110363. [CrossRef]
17. Wang, Y.; Zhang, J.; Wei, R.; Gao, W.; Wang, Y. MFRGN: Multi-scale feature representation generalization network for ground-to-aerial geo-localization. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, Australia, 28 October–1 November 2024; pp. 2574–2583.

18. Shen, T.; Wei, Y.; Kang, L.; Wan, S.; Yang, Y.-H. MCCG: A ConvNeXt-based multiple-classifier method for cross-view geo-localization. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 1456–1468. [CrossRef]
19. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
20. Xia, P.; Wan, Y.; Zheng, Z.; Zhang, Y.; Deng, J. Enhancing cross-view geo-localization with domain alignment and scene consistency. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 13271–13281.
21. Wang, T.; Zheng, Z.; Zhu, Z.; Gao, Y.; Yang, Y.; Yan, C. Learning cross-view geo-localization embeddings via dynamic weighted decorrelation regularization. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5647112. [CrossRef]
22. Deuser, F.; Habel, K.; Oswald, N. Sample4Geo: Hard negative sampling for cross-view geo-localisation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 16847–16856.
23. Huang, G.; Zhou, Y.; Zhao, L.; Gan, W. CV-Cities: Advancing cross-view geo-localization in global cities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 1592–1606. [CrossRef]
24. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.* **2024**.
25. Ding, L.; Zhou, J.; Meng, L.; Long, Z. A practical cross-view image matching method between UAV and satellite for UAV-based geo-localization. *Remote Sens.* **2021**, *13*, 47.
26. Zhuang, J.; Dai, M.; Chen, X.; Zheng, E. A faster and more effective cross-view matching method of UAV and satellite images for UAV geolocalization. *Remote Sens.* **2021**, *13*, 3979. [CrossRef]
27. Zhu, R.; Yang, M.; Yin, L.; Wu, F.; Yang, Y. UAV's status is worth considering: A fusion representations matching method for geo-localization. *Sensors* **2023**, *23*, 720. [CrossRef] [PubMed]
28. Cui, Z.; Zhou, P.; Wang, X.; Zhang, Z.; Li, Y.; Li, H.; Zhang, Y. A novel geo-localization method for UAV and satellite images using cross-view consistent attention. *Remote Sens.* **2023**, *15*, 4667.
29. Sun, J.; Pei, Y.; He, Y.; Li, X. A contrastive learning based multiview scene matching method for UAV view geo-localization. *Remote Sens.* **2024**, *16*, 3039. [CrossRef]
30. Li, X.; Zhu, R.; Yu, X.; Wang, X. A cross-view geo-localization algorithm using UAV image and satellite image. *Sensors* **2024**, *24*, 3719. [PubMed]
31. Wang, G.; Chen, J.; Dai, M.; Zheng, E. WAMF-FPI: A weight-adaptive multi-feature fusion network for UAV localization. *Remote Sens.* **2023**, *15*, 910.
32. Chen, J.; Zheng, E.; Dai, M.; Chen, Y.; Lu, Y. OS-FPI: A coarse-to-fine one-stream network for UAV geo-localization. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 8508–8521. [CrossRef]
33. Zhang, X.; Li, X.; Yu, X.; Wang, X. UAV geo-localization dataset and method based on cross-view matching. *Sensors* **2024**, *24*, 6905. [PubMed]
34. Boccadoro, M.; Striccoli, D.; Scarano, G.; Trotta, A.; Di Felice, M. A review of indoor positioning systems for UAV localization with machine learning algorithms. *Electronics* **2023**, *12*, 1533. [CrossRef] [PubMed]
35. Röhe, S.; Werner, M.; Weinmann, A.; Brucherseifer, E. Quantifying the simulation–reality gap for deep learning-based drone detection. *Electronics* **2023**, *12*, 2197.
36. Zhao, M.; Guo, Y.; Liu, J.; Xue, W. Autonomous UAV navigation with adaptive control based on deep reinforcement learning. *Electronics* **2024**, *13*, 2432. [CrossRef]
37. Yang, H.; Lu, X.; Zhu, Y. Cross-view geo-localization with layer-to-layer transformer. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2021; Volume 34, pp. 29009–29020.
38. Lin, J.; Zheng, Z.; Zhong, Z.; Luo, Z.; Li, S.; Yang, Y.; Sebe, N. Joint representation learning and keypoint detection for cross-view geo-localization. *IEEE Trans. Image Process.* **2022**, *31*, 3780–3792. [CrossRef] [PubMed]
39. Zhu, S.; Shah, M.; Chen, C. TransGeo: Transformer is all you need for cross-view image geo-localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 1162–1171.
40. Tian, X.; Shao, J.; Ouyang, D.; Shen, H.T. UAV-satellite view synthesis for cross-view geo-localization. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 4804–4815. [CrossRef]
41. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
43. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*; Hutter, M., Siegwart, R., Eds.; Springer: Cham, Switzerland, 2018; pp. 621–635.

44. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A. ROS: An open-source robot operating system. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; p. 5.
45. Google LLC. Google Maps Platform Tile API. Available online: https://developers.google.com/maps/documentation/tile (accessed on 31 August 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.