Beyond Paired Data: Self-Supervised UAV Geo-Localization from Reference Imagery Alone

Supplementary Material

8. ViLD dataset details

As described in Section 3.1 of the main paper, our dataset ViLD is composed of real-world UAV images captured during two actual flights conducted by an ultralight aircraft, equipped with a drone's camera and Inertial Measurement Unit (IMU), following typical drone flight trajectories.

8.1. First flight

The first flight, composed of 30,909 images, is the one we used to test our CAEVL method. As shown on Figure 7, the drone had a fairly stable altitude throughout the flight, with the exception of a couple of zones. Those induced errors for CAEVL, as shown in Figure 6 of the main paper.

Figures 8 and 9 show interesting examples of UAV and reference images, typical of this flight. For instance, the far left column on Figure 8 shows the difference in viewpoints between UAV and reference images when the drone's altitude starts to drop significantly. So does the far right column of Figure 9. The third column on Figure 9 shows the small shift in viewpoint that we can observe when the pitch or roll of the drone is not null.

8.2. Second flight

The second flight recorded 59,438 images. The landscapes recorded during this flight are similar to those recorded during the first flight. This flight has a less complex trajectory than Flight 1, *i.e.* the trajectory of Flight 2 is mainly composed of straight lines rather than sharp turns like in Flight 1. However, as shown on Figure 10, the images are recorded at higher altitudes during Flight 2, and with more altitude changes. Indeed, Flight 1 has an average altitude of 571m with a standard deviation of 116m, while Flight 2 has an average altitude of 1007m with a standard deviation of 402m. Figure 11 shows the evolution of the drone's altitude throughout the flight. Like for the first flight, the altitude is fairly stable, but has some sudden up or down changes.

Figures 12, 13 and 14 show random examples of UAV images from Flight 2, as well as the closest reference images. Those examples show that the hypothesis of null pitch and roll holds better for this flight, as the trajectory has more straight lines and fewer turns. However, the larger altitude range creates bigger differences in viewpoints be-

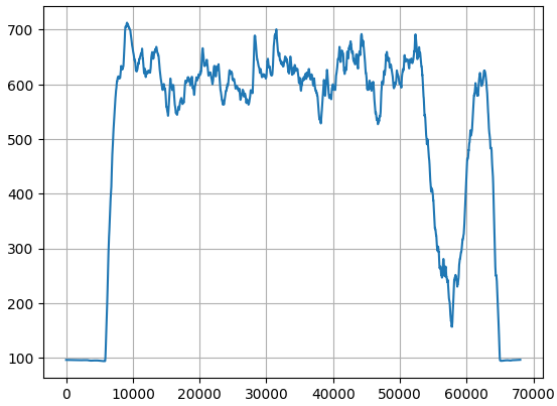

Figure 7. Altitude of drone throughout Flight 1. Altitude (m) is shown on the y-axis, and the x-axis represents each timestamp throughout the flight.

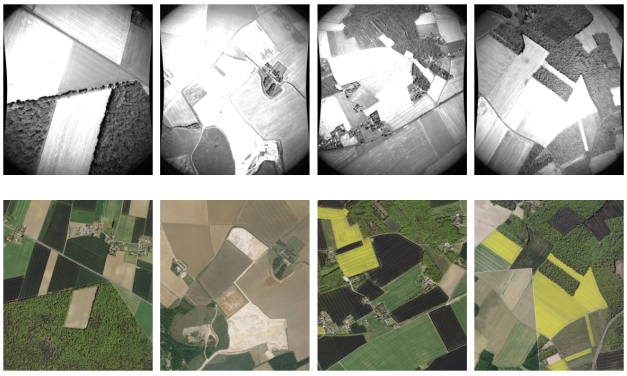

Figure 8. First set of examples of randomly picked UAV images (upper row) from Flight 1 and their geographically closest reference images (lower row).

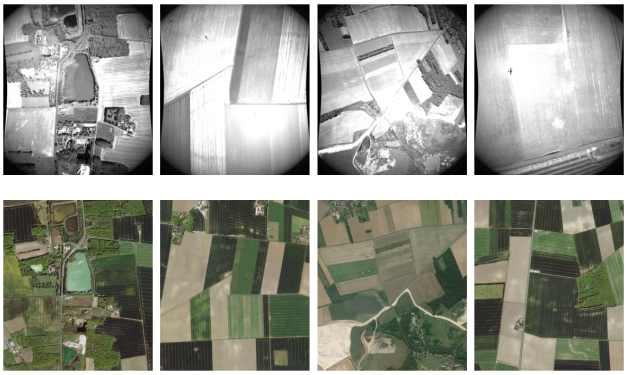

Figure 9. Second set of examples of randomly picked UAV images (upper row) from Flight 1 and their geographically closest reference images (lower row).

Table 4. Coverage percentage by land cover type for Flight 1. Columns 3 and 4 also show the highest and lowest coverage percentage per image per class.

Land Cover Type	Coverage (%)	Highest (%)	Lowest (%)
Noise	0.311	38.373	0
Low vegetation	0.236	5.016	0
High vegetation	2.953	47.391	0
Forest	17.023	83.074	0
Field	76.483	99.204	6.171
Stone	0.0001	0.669	0
River / lake	0.487	17.376	0
Road	0.587	6.171	0
Building	0.155	4.539	0
Industrial zone	0.00003	0.0754	0
Railroad	0.0008	0.336	0
Field path	1.760	7.746	0.0170

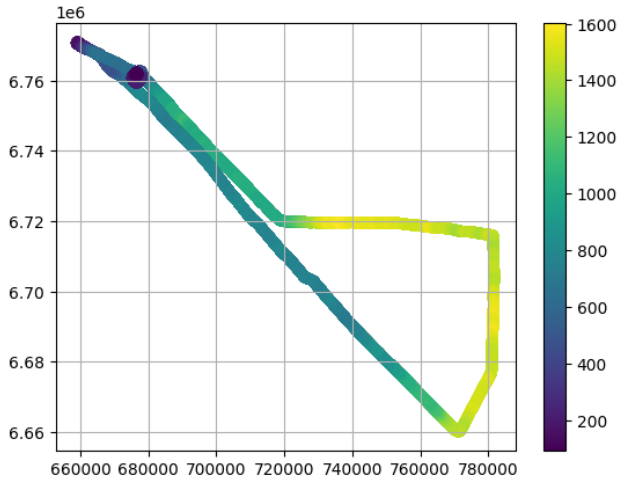

Figure 10. Illustration of the altitude of the drone throughout the second whole trajectory of flight 2, in meters

tween UAV and reference images.

8.3. Differences with existing UAV datasets

We talk, in Section 3.4 of the main paper, about the differences between ViLD and other UAV datasets in the literature. We show here, with Figures 15 and 16, some examples of reference-query image pairs from different datasets to underline the said differences.

9. Implementation details

As explained in Section 5, the autoencoder architecture is based on [28] and [50]. The encoder consists of four strided convolutional layers with batch normalization and *LeakyReLU* activation, producing a 1024-dimensional latent space. Only the encoder is deployed on the UAV, and it contains 33 million parameters. The decoder mirrors the encoder's structure, with a *Sigmoid* activation on its final

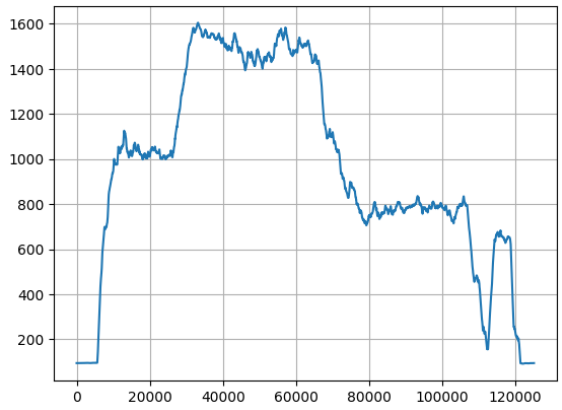

Figure 11. Altitude of drone throughout flight 2. Altitude (m) is shown on the y-axis, and the x-axis represents each timestamp throughout the flight.

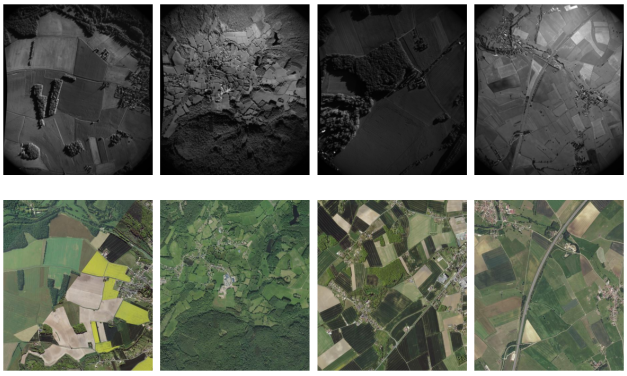

Figure 12. First set of examples of randomly picked UAV images (upper row) from Flight 2 and their geographically closest reference images (lower row).

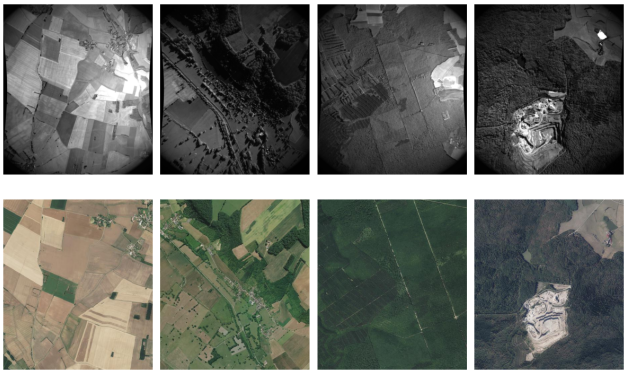

Figure 13. Second set of examples of randomly picked UAV images (upper row) from Flight 2 and their geographically closest reference images (lower row).

layer to accommodate the binary contour input images. In Eq. (1), β is set to 1. The global projection head consists

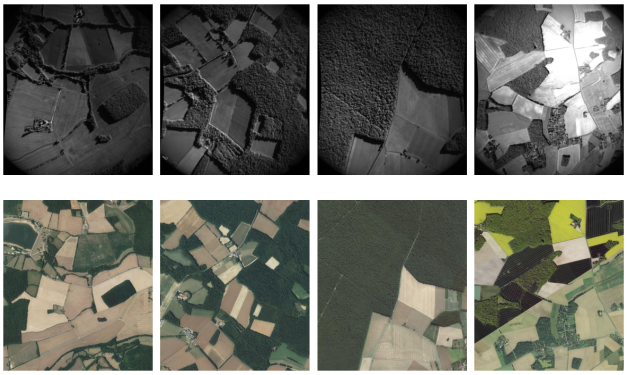

Figure 14. Third set of examples of randomly picked UAV images (upper row) from Flight 2 and their geographically closest references images (lower row).

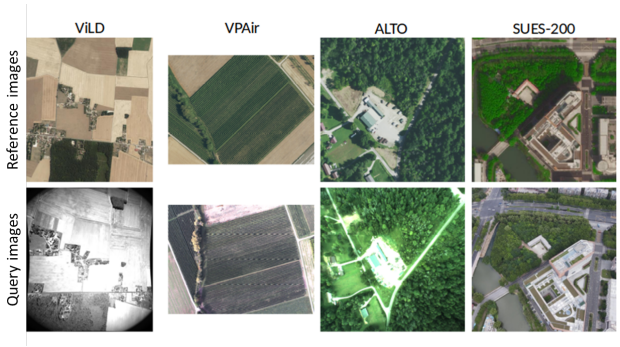

Figure 15. Comparison of a randomly selected query-reference pair from different datasets.

Figure 16. Comparison of a randomly selected query-reference pair from different datasets.

of 2 linear layers with batch normalization and *ReLU*, followed by a third linear layer, all of size 4096. The local projection head has the same structure, but each layer is sized at 560. Following [4], α in Eq. (4) is 0.75, and γ is set to 20 for the top-γ pairs retained in Eq. (2) and (3). Models were trained on reference images for 100 epochs using the AdamW optimizer and then tested on the UAV images.

The vignetting augmentation was applied by multiplying the input image with a centered Gaussian filter matrix, matching the image size, with a standard deviation of 70. The translation offset was selected uniformly within [-x, x], where x is the pixel distance equivalent to 10 meters in real life. Rotations were randomly chosen between -30° and 30°, and cropping was centered on a random point, with scale set between 70% and 100%. Noise and blur were added using a Gaussian distribution and a kernel size of 5, while brightness and contrast factors varied uniformly between 0 and 2. All augmentations were applied to grayscale images prior to Canny edge extraction.

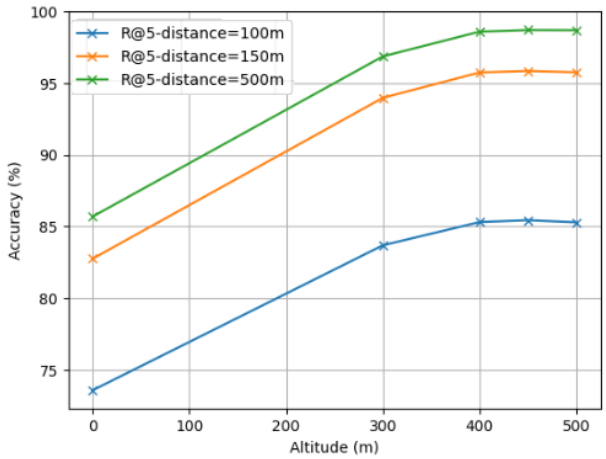

Figure 17. Localization performance when removing images below a given altitude.

10. Experiments and analyses

10.1. Altitude drops

As shown on Figure 6 and discussed in Section 5.1 of the main paper, CAEVL can handle moderate altitude variations when localizing query images, but large altitude changes make the method struggle to correctly localize UAV images. To assess this phenomenon, we calculated localization scores for images captured above 300m, 400m, 450m, and 500m. As shown in Figure 17, while CAEVL remains robust to drops of 100-200m, performance declines when altitude drops exceed 200m. Excluding images below 300m or 500m minimally impacts model performance.

10.2. Localization performance on a subset of queries

To better compare the localization performance of SuperPoint + LightGlue to the performance of the methods reported in Table 1, we reproduce the same experiment that we did for SuperPoint + LightGlue. Indeed, as we explained in Table 1, localizing one query image against the whole reference database would take us a very long time with Super-

Point + LightGlue. As we still wanted to be able to compare the results of this method against other vector-based methods, we decided to sample 500 query images from the test set. To localize each of the 500 query images, we gathered all the reference images that were less than 1km away from the query image, and we localized the query image only against those close reference images. Results are shown on Table 5.

CAEVL consistently outperforms SuperPoint + LightGlue up to 250m. At the 500m mark, the performance levels converge, which is expected given that the search is limited to a 1km radius around each query image. Interestingly, SuperPoint + LightGlue achieves slightly better results for R@5 and R@10 at 500m, though at the cost of a much longer computational time. This advantage may come from its strong robustness to extreme viewpoint variations, such as large heading and altitude differences, among others. While this may help in retrieving database images that resemble the query image —*i.e*. bumping up the localization performance at 500m—, it could also make it harder for the method to precisely identify the closest match, as it remains invariant to these transformations.

10.3. Robustness to image perturbations

To assess whether the learned model overly relies on the synthetic spatial warping augmentations used during training, we evaluate its robustness to different perturbations applied to the reference images at inference time. Specifically, we consider three types of perturbations:

- Rotation, where the image is randomly rotated within a range of 0° to 60° to simulate the variations in UAV heading throughout a flight.
- Center crop and resize, where the image is cropped and resized to simulate variations in UAV altitude, with crop factors ranging from 100% (no crop) to 50% of the original image.
- Color jittering, where random changes in brightness, contrast, saturation, and hue are applied with parameters ranging from (0,0,0,0) to (0.4,0.4,0.4,0.1).

For each perturbation type and severity level, we perform five runs, each time applying the perturbations at random to the reference images. This stochastic evaluation ensures that the reported results are not dependent on a single perturbation configuration and provides a robust estimate of the model's performance under challenging conditions.

Figure 18 shows that CAEVL's ability to adapt to perturbed images is on par with the other methods we compare it to.

Interestingly, for the perturbation simulating altitude drops, the performance of CAEVL peaks at a 75% center crop before declining at smaller crops. We attribute this to two complementary effects. First, moderate cropping removes peripheral regions that often contain noisy or irrelevant edges, yielding cleaner Canny maps and forcing the model to focus on stable, central structures such as roads and buildings. However, when the crop becomes too aggressive (e.g., 50%), important structural context is lost, as roads and intersections are truncated, reducing the distinctiveness of the representation. This explains the bell-shaped performance curve observed in our experiments.

10.4. Edge detector impact

While our method primarily uses the Canny operator for edge detection, we conducted an ablation study to evaluate other detectors, including the classical Sobel and LSD [58], as well as two deep-learning-based methods: HED [63] and DexiNed [56].

Table 6 reports the results. Deep edge detectors like DexiNed slightly improve retrieval accuracy but are significantly slower. Canny achieves a good balance between accuracy and computational efficiency, which justifies its use in our main experiments. Furthermore, several prior works in UAV localization have also relied on Canny for edge detection [1, 50, 55], reinforcing its relevance as a baseline for this task, which was the first reason that led us to choose it.

Another advantage of Canny is the balance it achieves in the amount of visual information preserved. Detectors such as LSD tend to remove too much structural detail, which can cause a loss of discriminative features. Conversely, deep-learning-based detectors like HED or DexiNed often produce dense edge maps, potentially introducing noise and spurious details that may confuse the model during training and retrieval. Canny provides a middle ground, retaining the most salient structures while discarding less informative textures, which aligns well with the goal of learning robust and generalizable image embeddings for UAV localization. Figures 19, 20, 21, 22 and 23 show a few examples of images, from ViLD and other datasets such as DenseUAV and ALTO, that were processed by each edge detector.

10.5. Predictions examples

Figures 24, 25 and 26 provide a qualitative comparison of the top-5 predictions for our method, CAEVL, against key baselines: MixVPR, EigenPlaces, FSRA, and DAC on the ViLD dataset. Furthermore, Figures 27, 28, 29, 30 and 31 provide qualitative comparison examples on other datasets, namely the ALTO, VPAIR and DenseUAV datasets.

Method	Satellite-Only Training	100m			150m			250m			500m			Descriptors Dimension	GFLOPs per query
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10		
Random	-	0.03	0.16	0.33	0.07	0.36	0.73	0.21	1.02	2.03	0.86	4.16	8.06	-	-
SuperPoint [20] + LightGlue [41]*	×	11.60	33.80	48.40	24.80	54.00	67.40	49.40	78.20	86.20	86.20	95.60	97.60	256	17,705.44
MixVPR [2] (Zero-shot)	×	15.40	36.60	52.40	29.40	55.80	69.60	59.20	78.00	86.00	87.60	91.40	93.60	4096	10.31
MixVPR (Finetuned)		45.00	68.80	78.20	69.40	83.80	87.40	82.60	89.20	91.40	90.40	94.40	95.00		
EigenPlaces [7] (Zero-shot)	×	15.80	35.60	49.00	27.80	52.00	62.80	49.00	71.20	79.20	81.40	91.80	94.80	2048	19.71
EigenPlaces (Finetuned)		42.40	72.40	82.20	63.80	86.00	89.80	84.00	92.00	93.80	90.80	92.60	95.60		
Megaloc [6] (Zero-shot)	×	5.60	17.60	24.80	10.60	30.60	40.00	29.80	54.80	65.60	66.20	83.80	89.40	8448	54.93
FSRA [18] (Zero-shot)	×	2.00	12.00	18.60	5.20	20.80	32.40	17.60	10.80	53.20	53.80	80.00	88.00	**512**	13.34
FSRA (Finetuned)		38.00	74.00	80.60	67.40	83.80	**86.60**	89.20	91.60	92.60	93.80	95.60	**97.80**		
DAC [62] (Zero-shot)	×	2.80	11.40	15.60	5.40	17.00	24.80	15.20	33.20	45.20	45.60	68.20	78.60	1024	20.55
DAC (Finetuned)		**52.60**	**74.80**	**82.40**	**79.20**	**85.20**	**86.60**	**91.80**	**93.00**	**93.40**	**94.2**	**96.20**	**97.80**		
Di Piazza *et al.* [50] (Finetuned)	✓	34.00	48.20	53.00	42.80	56.20	60.40	46.40	61.20	66.40	55.80	75.00	81.80	1024	**1.42**
CAEVL (Ours)	✓	51.20	**74.80**	81.40	74.60	84.80	86.00	85.40	88.40	88.60	91.80	93.20	97.20	1024	**1.42**

Table 5. Comparison of Recall@K performance for various methods at different localization thresholds (100m, 150m, 250m, and 500m). The performance reported in this table is based on a subset of 500 query images, rather than the entire set of query images. Each query is only matched against reference images within a 1km radius. This setup ensures a fair comparison between all methods and SuperPoint + LightGlue. As noted in Table 1, localizing the full query set against all reference images would be too time-consuming for SuperPoint and LightGlue, which is why we limited the search space.

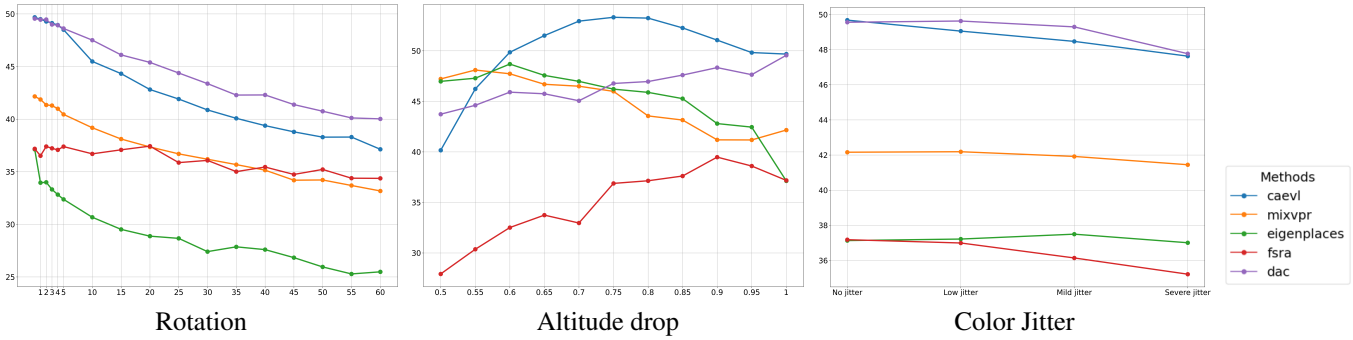

Figure 18. Evolution of Recall@1 at 100 m for CAEVL and four SOTA methods under perturbations applied to the reference images.

Edge Detector	Recall@1 @100m	Recall@1 @150m	Time (ms)
Sobel	14.26	25.07	**1.25**
LSD	44.58	69.83	12.5
HED	48.47	71.45	26.3
DexiNed	**52.27**	**74.44**	35.9
Canny (ours)	49.68	73.93	1.49

Table 6. Comparison of different edge detectors in our pipeline. Recall@1 is reported at 100m and 150m thresholds. Canny provides the best balance between performance and efficiency.

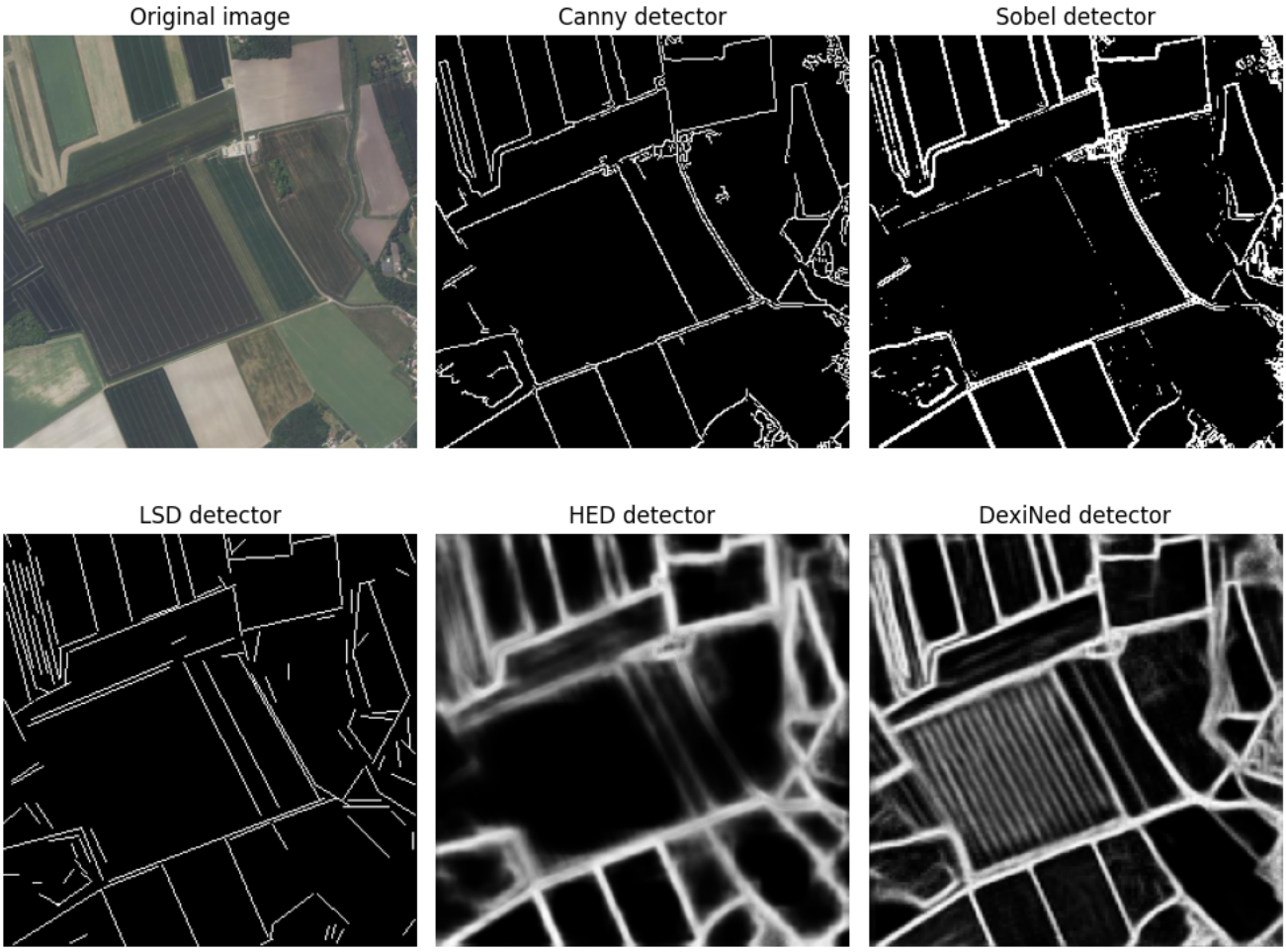

Figure 19. Randomly selected image from ViLD, processed by each edge detector considered in the ablation study.

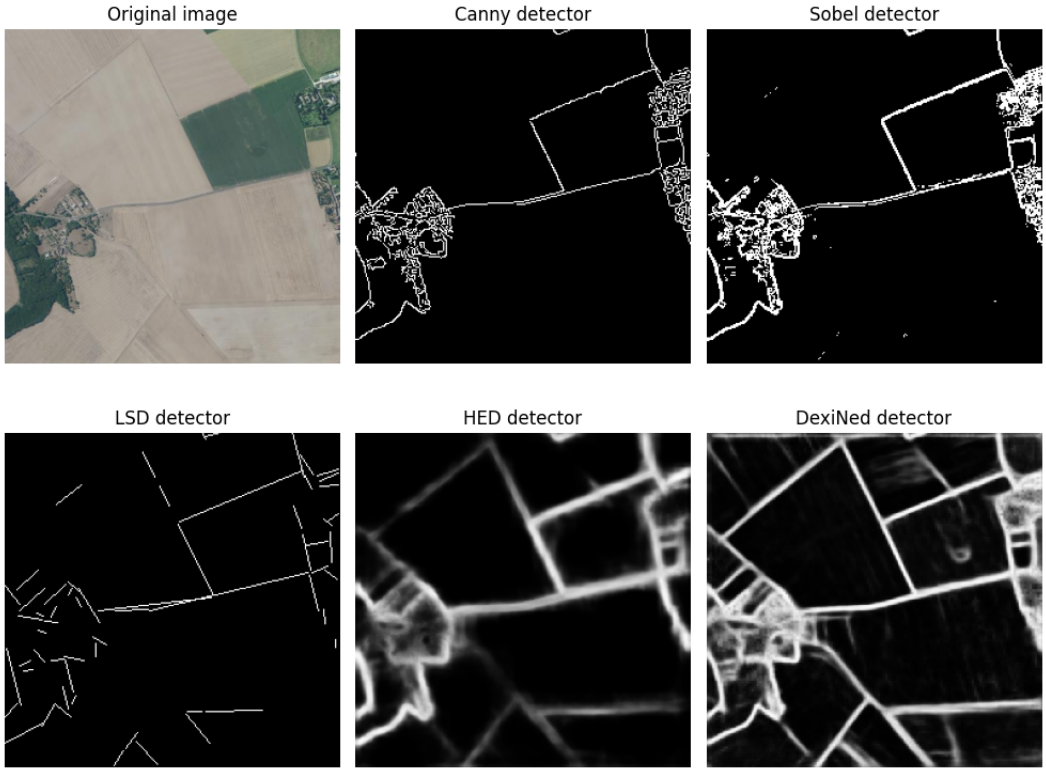

Figure 20. Randomly selected image from ViLD, processed by each edge detector considered in the ablation study.

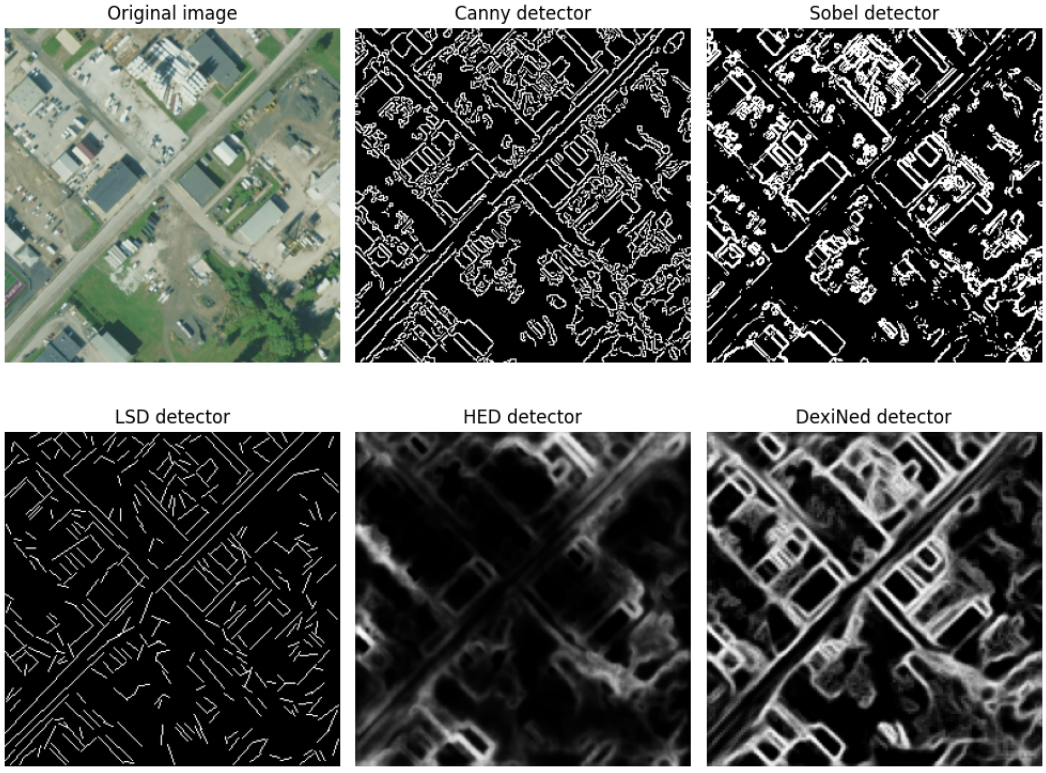

Figure 21. Randomly selected image from the Alto dataset, processed by each edge detector considered in the ablation study.

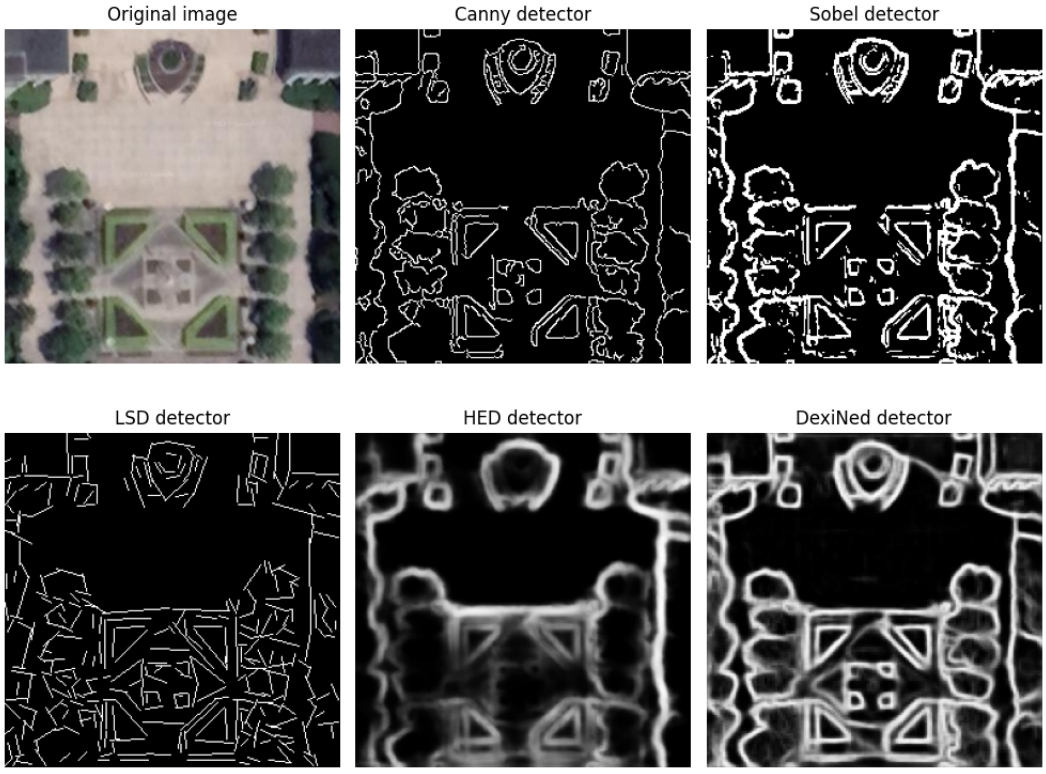

Figure 22. Randomly selected image from the DenseUAV dataset, processed by each edge detector considered in the ablation study.

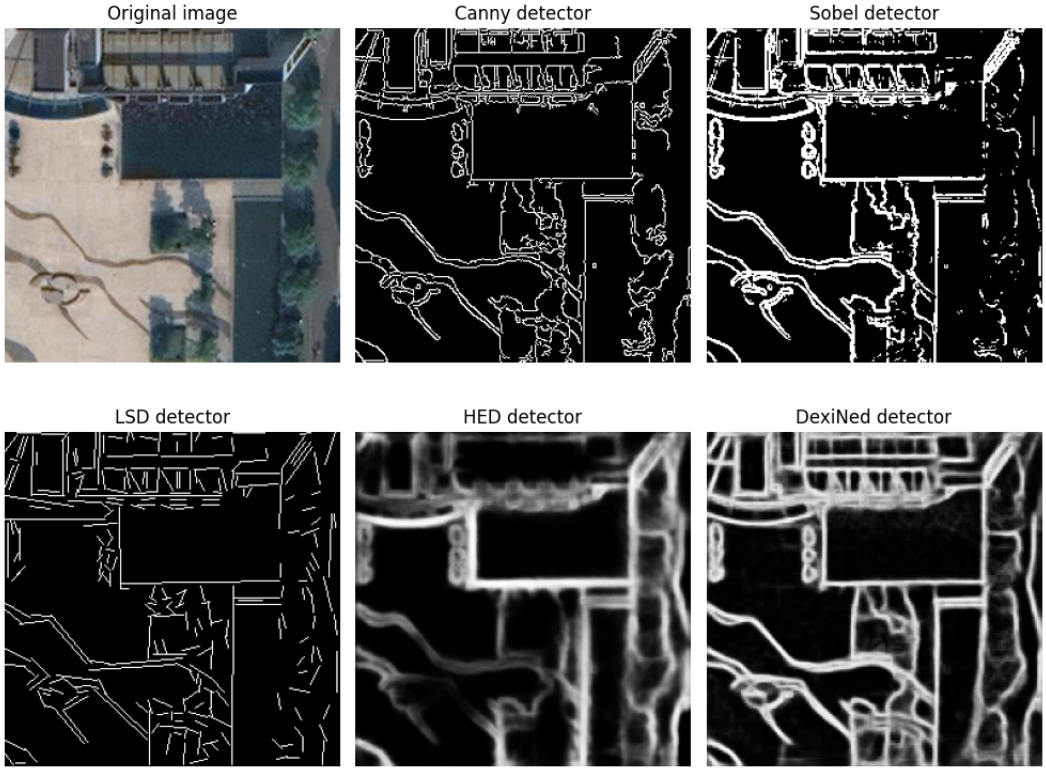

Figure 23. Randomly selected image from the DenseUAV dataset, processed by each edge detector considered in the ablation study.

Figure 24. Visual comparison of top-5 retrieval results on a query from ViLD. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 100m of the ground truth, are highlighted with a green frame.

Figure 25. Visual comparison of top-5 retrieval results on a query from ViLD. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 100m of the ground truth, are highlighted with a green frame.

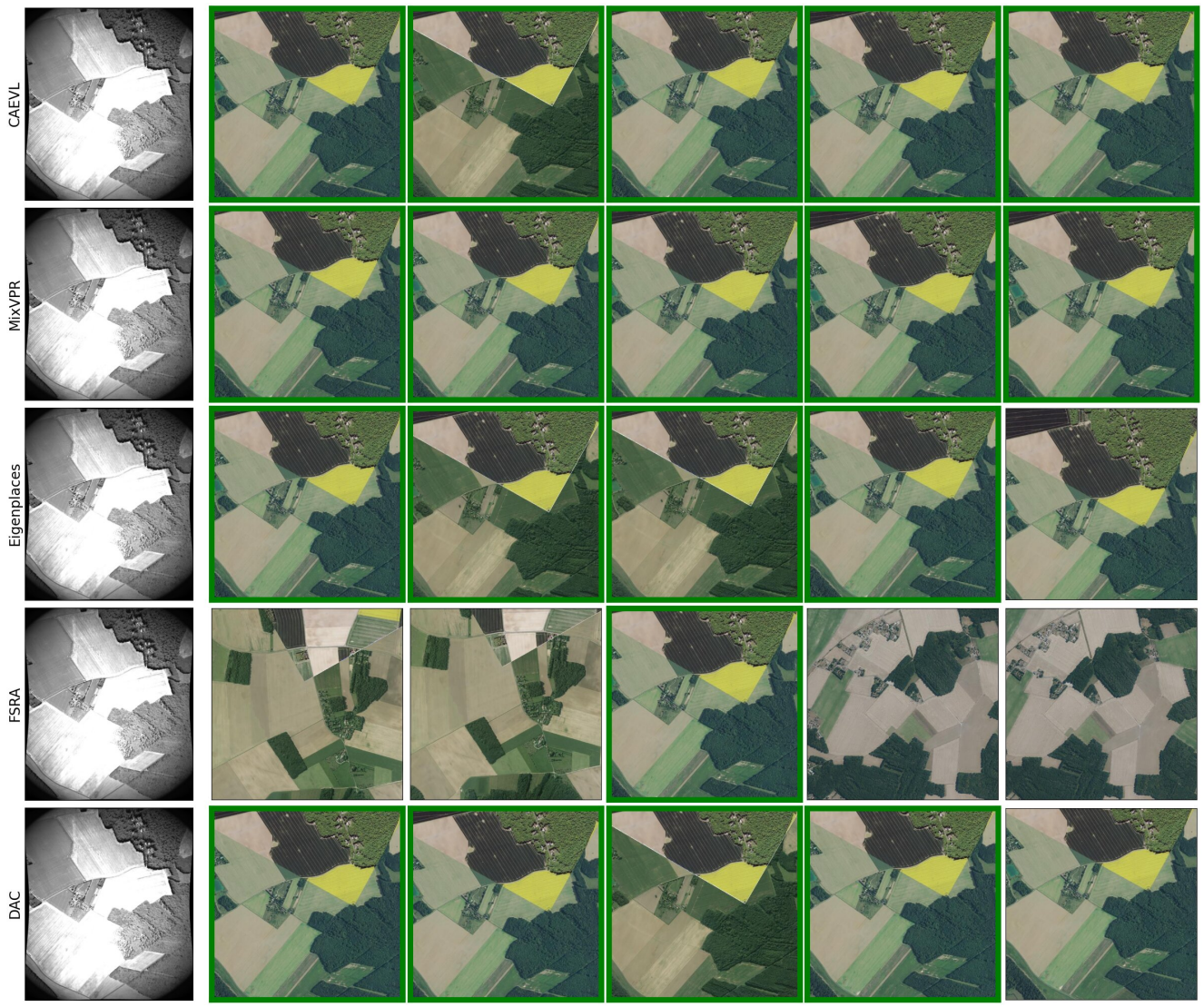

Figure 26. Visual comparison of top-5 retrieval results on a query from ViLD. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 100m of the ground truth, are highlighted with a green frame.

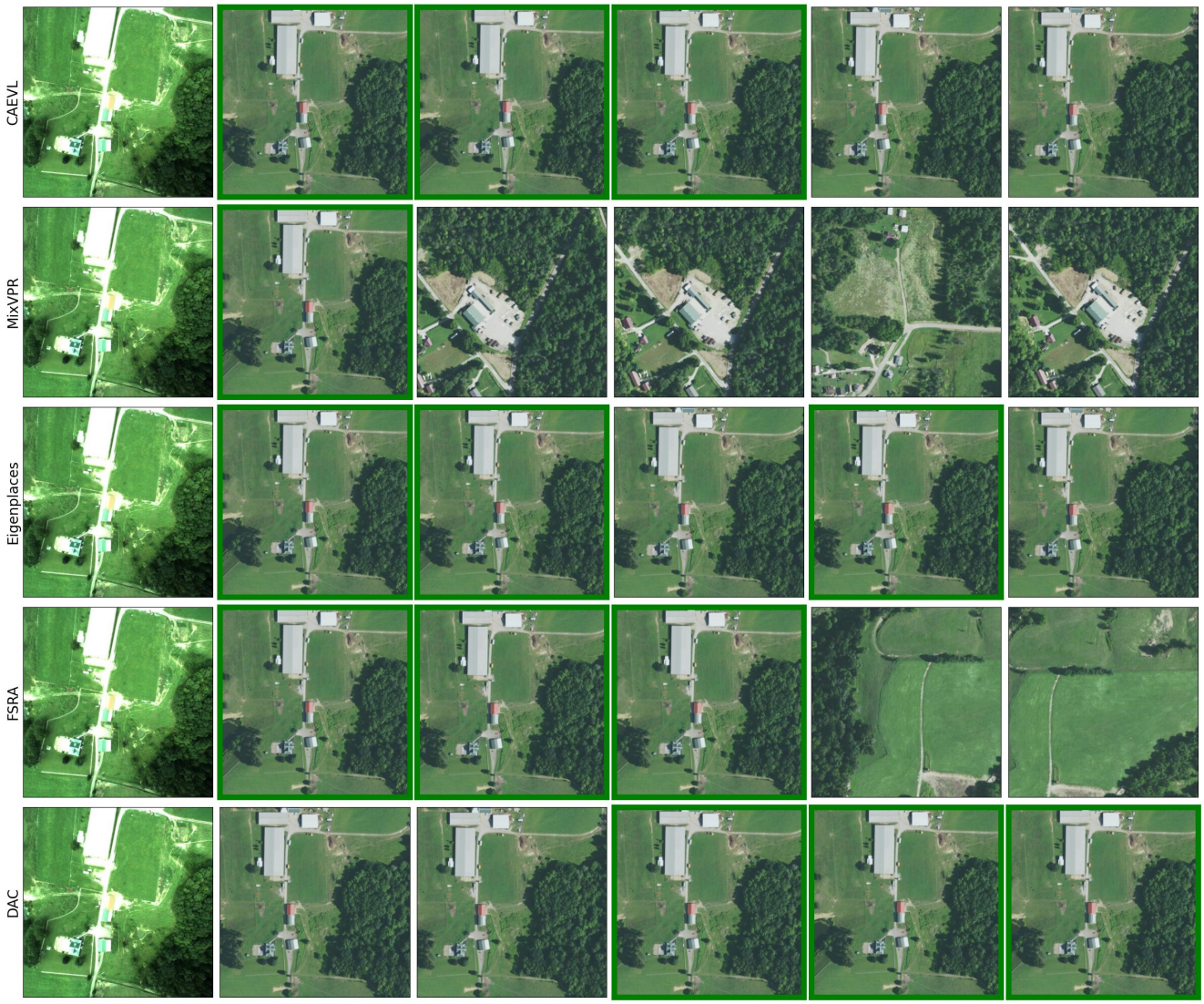

Figure 27. Visual comparison of top-5 retrieval results on a query from the ALTO dataset. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 15m of the ground truth, are highlighted with a green frame.

Figure 28. Visual comparison of top-5 retrieval results on a query from the VPAIR dataset. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 15m of the ground truth, are highlighted with a green frame.

Figure 29. Visual comparison of top-5 retrieval results on a query from the VPAIR dataset. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 15m of the ground truth, are highlighted with a green frame.

Figure 30. Visual comparison of top-5 retrieval results on a query from the DenseUAV dataset. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 15m of the ground truth, are highlighted with a green frame.

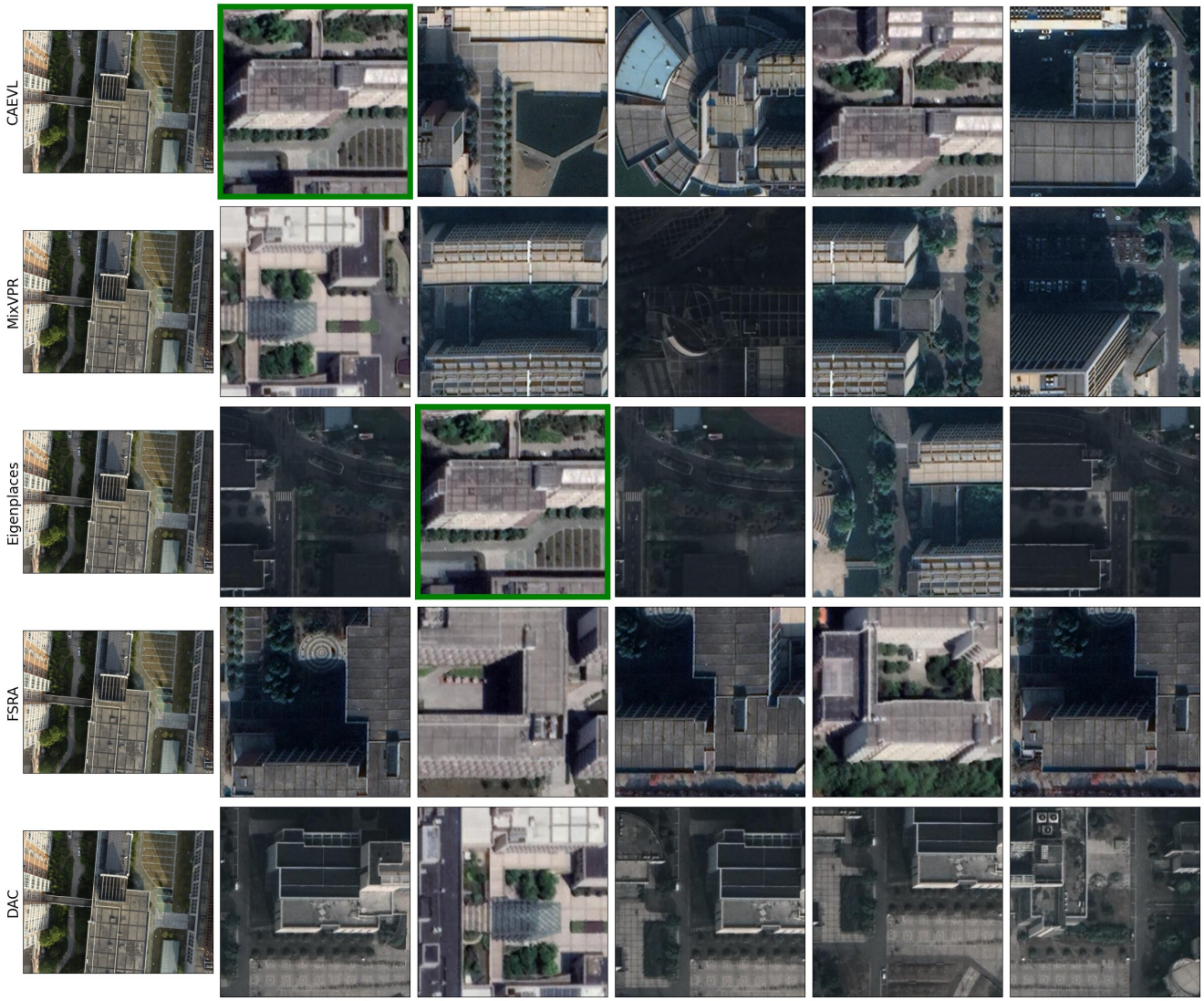

Figure 31. Visual comparison of top-5 retrieval results on a query from the DenseUAV dataset. For each method, the five best matches are shown to the right of a randomly picked query image. Correct predictions, defined as those within 15m of the ground truth, are highlighted with a green frame.