Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition

<Supplementary Material>

Stephen Hausler Sourav Garg Ming Xu Michael Milford Tobias Fischer
QUT Centre for Robotics, Queensland University of Technology
{s.hausler, s.garg, m22.xu, michael.milford, tobias.fischer}@qut.edu.au

Overview

The Supplementary Material is structured as follows. In Section 1, we show results obtained on the RobotCar Seasons v2 and Extended CMU Seasons datasets split by query condition. Section 2 contains additional quantitative results on the Pittsburgh 30k and Tokyo 24/7 datasets, as well as additional results on the computation time split across the feature extraction and feature matching processes. Section 3 contains a variety of additional ablation studies, some of them further demonstrating the robustness of our method, while others detail experiments that might logically be expected and were conducted, but that were not fruitful. Section 4 contains various qualitative results, showcasing both challenging success cases of Patch-NetVLAD and some failure cases. This section also contains examples of incorrect dataset annotations, where Patch-NetVLAD actually found the correct match but this match was not within the ground-truth matches due to errors in the ground-truth. Finally, in Section 5 we describe in detail the six key benchmark datasets on which we evaluate Patch-NetVLAD.

1. Results Split by Condition on RobotCar Seasons v2 and Extended CMU Seasons

RobotCar Seasons v2: Suppl. Table 1 contains results obtained from the training split of the RobotCar Seasons v2 dataset split by condition. Tables 1 and 2 of the main paper are summary statistics on the query set, where the different conditions are weighted by the number of images contained within each condition. Utilizing the training set allows us to further split results by specific appearance change conditions, providing an additional set of fine-grained comparisons between Patch-NetVLAD and existing state-of-the-art over the main paper.

Patch-NetVLAD outperforms SuperGlue [12] by 1.3% absolute recall on the tightest error thresholds (.25m translational error and 2 degrees orientation error) when considering the summary statistic. There are some conditions where SuperGlue has a slight performance advantage for the looser error thresholds, in particular the night traverses. As stated in the Conclusions section of the main paper, it would be interesting to train a neural network-based feature matcher similar to SuperGlue that uses our proposed Patch-NetVLAD features instead of the original SuperPoint [6] features. This approach would likely yield more robust matching than a standard mutual nearest neighbors matching technique, which combined with outlier rejection will likely yield a significant performance improvement.

Interestingly, while DELG performs well on datasets like Nordland and Pittsburgh, both the global retrieval only as well as the global + local re-ranking DELG perform relatively poorly on RobotCar Seasons v2 where a low ground truth pose error tolerances are required (Patch-NetVLAD outperforms DELG global and local re-ranking by 4.6% and 7.0% absolute recall in the summary statistic respectively). In future works, it would be interesting to investigate why local re-ranking in this case worsens performance.

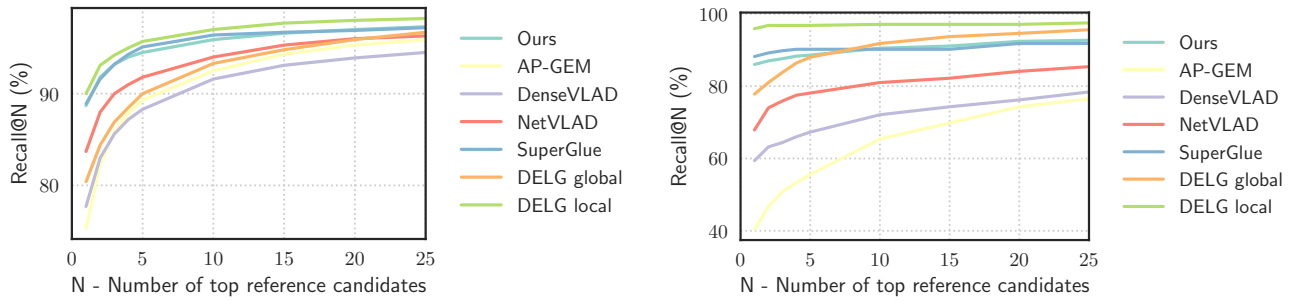
Extended CMU Seasons: In Suppl. Table 2 we similarly show detailed results for the Extended CMU Seasons dataset, split by Urban, Suburban and Park environments. Patch-NetVLAD consistently outperforms all comparison methods, including our competitive SuperGlue baseline and DELG, on all conditions and all error thresholds by relatively large margins, with two exceptions being the park condition where SuperGlue performs slightly better for the largest error threshold, and the suburban condition where SuperGlue performs slightly better for the medium error threshold.

2. Additional Quantitative Results

Additional Recall Plots: Fig. 3 in the main paper shows the recall@N performance on the Mapillary validation set. Similarly, Suppl. Fig. 1 shows the recall@N performance for the Pittsburgh 30k and Tokyo 24/7 datasets.

Supplementary Table 1. Performance comparison RobotCar Seasons v2

	day conditions							**night conditions**	
	dawn	dusk	OC-summer	OC-winter	rain	snow	sun	night	night-rain
m	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
AP-GEM [11]	1.4 / 14.2 / 65.9	9.6 / 29.4 / 82.9	2.4 / 19.1 / 80.5	3.6 / 20.3 / 78.1	4.4 / 21.5 / 86.0	4.5 / 15.8 / 75.9	1.8 / 7.5 / 58.2	0.0 / 0.2 / 6.8	0.1 / 1.2 / 15.8
DenseVLAD [17]	4.5 / 24.3 / 79.6	12.5 / 38.9 / 89.1	3.8 / 27.4 / 90.8	4.1 / 27.1 / 85.6	5.4 / 29.0 / 91.4	6.7 / 25.5 / 85.1	3.2 / 11.0 / 67.1	1.4 / 2.7 / 23.2	0.6 / 5.2 / 29.8
NetVLAD [1]	2.2 / 16.8 / 73.3	11.4 / 31.0 / 85.9	3.2 / 21.5 / 90.9	4.1 / 22.6 / 84.0	4.2 / 22.2 / 89.4	5.2 / 20.1 / 80.8	2.4 / 10.4 / 70.3	0.2 / 1.2 / 9.1	0.3 / 0.9 / 8.8
DELG global [5]	1.6 / 10.9 / 66.4	8.9 / 23.9 / 81.3	2.1 / 16.5 / 77.6	3.5 / 18.5 / 73.6	3.9 / 20.5 / 87.9	3.6 / 13.5 / 73.5	1.0 / 6.4 / 59.6	0.2 / 0.7 / 7.6	0.1 / 1.6 / 13.8
DELG local [5]	1.7 / 10.4 / 78.3	2.5 / 7.3 / 76.8	1.1 / 8.9 / 84.2	1.2 / 9.1 / 83.2	1.2 / 4.5 / 76.8	3.5 / 10.9 / 80.8	3.3 / 12.6 / 85.2	**1.4 / 7.6 / 38.6**	**2.4 / 11.9 / 53.0**
SuperGlue [12]	4.3 / 24.6 / 84.8	12.7 / 40.3 / 88.6	5.0 / 31.5 / **95.0**	4.5 / 30.2 / 88.6	**5.9 / 30.1** / 91.8	7.0 / 25.4 / 87.2	3.3 / 17.1 / **83.9**	0.5 / 2.2 / 27.9	0.9 / 5.4 / 31.8
Ours	**4.8 / 29.4 / 86.2**	**13.5 / 41.9 / 89.5**	**5.3 / 33.5** / 94.5	**6.3 / 32.7 / 89.8**	**5.9** / 29.3 / **92.1**	**7.8 / 27.3 / 87.9**	**4.8 / 20.1** / 83.4	0.5 / 2.7 / 24.9	1.0 / 5.4 / 30.8



Supplementary Figure 1. **Comparison with state-of-the-art.** We show the Recall@N performance of Ours (Multi-RANSAC-Patch-NetVLAD) compared to AP-GEM [11], DenseVLAD [17], NetVLAD [1] and SuperGlue [12], on the Pittsburgh (left) and Tokyo 24/7 (right) datasets.

Supplementary Table 2. Performance comparison Extended CMU Seasons

	Urban	Suburban	Park
m	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
AP-GEM [11]	9.4 / 24.4 / 83.6	2.7 / 10.3 / 66.7	1.9 / 7.4 / 48.1
DenseVLAD [17]	14.1 / 35.5 / 87.3	5.2 / 18.8 / 80.1	5.1 / 19.4 / 72.2
NetVLAD [1]	12.1 / 31.5 / 91.3	3.7 / 14.0 / 78.4	2.9 / 10.9 / 62.0
DELG global [5]	7.6 / 21.1 / 82.6	2.5 / 9.7 / 69.4	1.2 / 4.8 / 41.4
DELG local [5]	6.3 / 18.2 / 89.4	4.1 / 17.0 / 96.6	7.1 / 29.0 / 93.6
SuperGlue [12]	17.1 / 43.6 / 96.9	7.5 / **30.5** / 96.5	7.5 / 30.5 / **96.5**
Ours	**19.2 / 48.0 / 97.2**	**8.2** / 28.8 / **97.0**	**9.5 / 34.9** / 94.3

Computational Time Requirements: Fig. 4 of the main paper shows the number of seconds required to process each query by a variety of our system configurations, as well as SuperGlue and DELG. The processing times presented in Fig. 4 of the main paper show the *accumulated* times of feature extraction and feature matching. In Suppl. Fig. 2 we show the compute times *split* into feature extraction time only and feature matching time only; as well as the accumulated time.

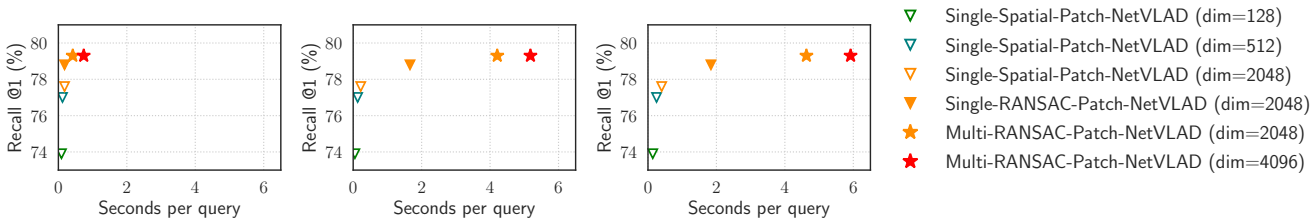
3. Further Ablation Studies

Ablation of Multi-Scale Fusion Weights and Patch Sizes: In Fig. 6 of the main paper, we demonstrated that Patch-NetVLAD is robust to the choice of particular patch sizes that are fused in our multi-scale approach. In Suppl. Table 3, we further validate that our proposed multi-scale fusion of spatial scores across several patch sizes is robust to changes in patch size *and* weightings by presenting results for the Mapillary dataset. Note that, as stated in the main paper, the set of weights used *across all experiments and all datasets* was determined using a grid-search on the training set of the RobotCar Seasons v2 dataset.

While we fuse three patch sizes in the main paper, our method is not constrained to fusing any particular number of patch sizes. An investigation regarding this is shown in Suppl. Table 4 – there all patch sizes are fused with equal weights for simplicity. An interesting observation is that increasing the number of different patch sizes used (from three up to five) does not improve the recall performance beyond the best combination of three patch sizes. We can infer that the span of patch sizes (the difference between the smallest size and the largest size) is more important than the number of patch sizes used.

Early Match Fusion: In Section 3.5 of the main paper, we describe our multi-scale fusion approach that merges the spatial scores obtained from different patch sizes. An alternative to this post-processing fusion is an early fusion where mutual nearest neighbors (Section 3.3 of the main paper) are found across patches of different scales, and a joint spatial score is calculated from all these mutual nearest neighbors.

However, we found that this early fusion approach does not work as well as the proposed post-processing fusion. Specifically, on the Mapillary validation set, we find that



(a) Feature extraction (b) Feature matching (c) Combined

Supplementary Figure 2. **Computational time requirements.** The number of seconds required to process each query are shown on the x-axis, with the resulting R@1 shown on the y-axis, for the Mapillary dataset. (a) indicates the times taken for feature extraction only, while (b) shows the feature matching time. In (c) we show the combined time (as in Fig 4 in the main paper). Triangles indicate single-scale Patch-NetVLAD, while stars indicate multi-scale Patch-NetVLAD. Filled symbols are used for RANSAC matching, while hollow symbols are used for the rapid spatial verification. The color indicates varying PCA dimensions.

Supplementary Table 3. Ablation of Multi-Scale Fusion Weights (R@1)

Weights / Patch sizes	2 / 5 / 8	2 / 4 / 6	2 / 6 / 10
0.33 / 0.33 / 0.33	**79.7**	78.6	78.8
0.2 / 0.6 / 0.2	79.1	78.5	78.4
0.45 / 0.15 / 0.4	79.5	78.9	78.8
0.45 / 0.35 / 0.2	78.8	78.1	79.1

Supplementary Table 4. Ablation of the Number of Fused Patch Sizes

Patch sizes	Recall@1
1 / 3 / 5	78.0
2 / 5 / 8	**79.7**
3 / 5 / 7	79.3
4 / 5 / 6	78.2
1 / 2 / 3 / 4	77.3
1 / 3 / 5 / 7	78.5
2 / 4 / 6 / 8	78.9
1 / 3 / 5 / 7 / 9	79.5
2 / 4 / 6 / 8 / 10	78.8

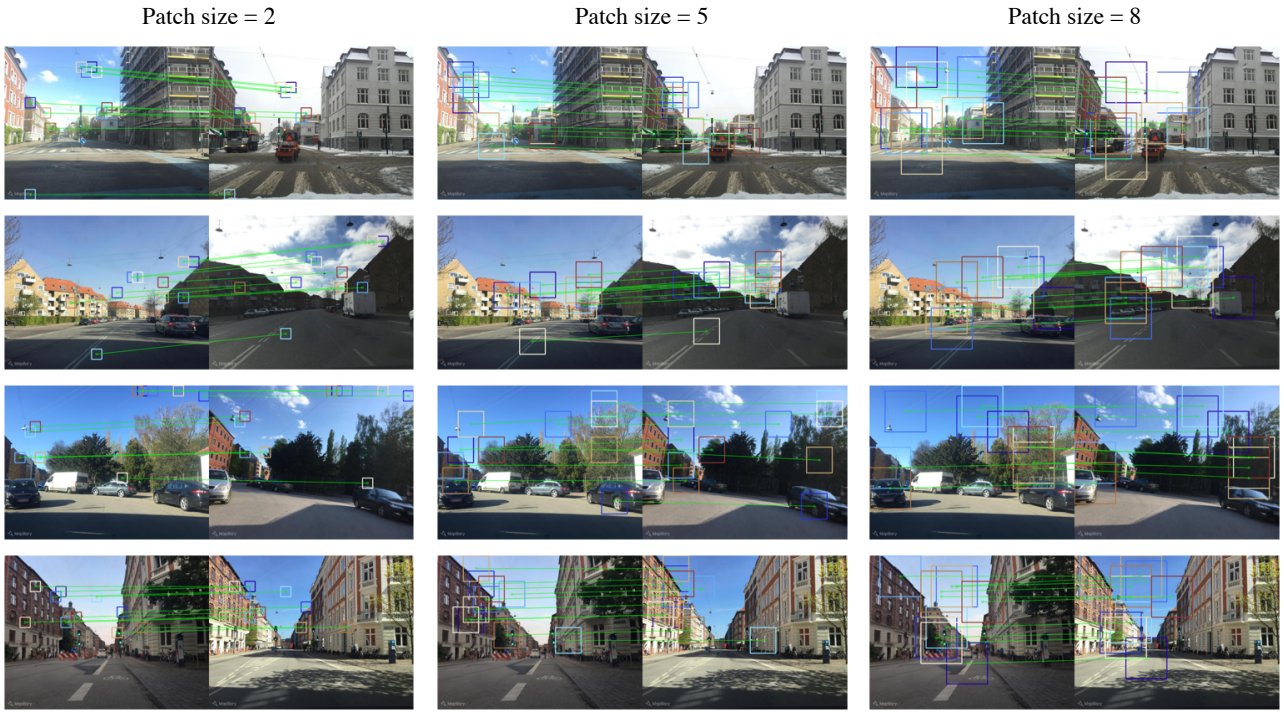
the early fusion approach results in R@1: 77.2%, R@5: 85.3%, and R@10: 87.3%. This compares to R@1: 79.5%, R@5: 86.2% and R@10: 87.7% using our proposed post-processing fusion.

Other Pooling Strategies: We use NetVLAD pooling to aggregate patch features into a single patch descriptor in our proposed approach. Instead of NetVLAD pooling, other pooling strategies such as max-pooling [16] and sum-pooling [2] have been proposed in the literature. Our spatial scoring system based on patch-based matching is in principle applicable with alternative pooling strategies. However, we found that patch-based aggregation does not perform well when applied to these pooling strategies: patch-level average pooling of VGG's Conv-5 layer (all else being equal) improves performance from 60.8% R@1 (vanilla NetVLAD on Mapillary dataset) to 73.6%, which compares to 79.5% using patch-level VLAD pooling (Patch-NetVLAD). Patch-level max-pooling similarly leads to decreased performance when compared to Patch-NetVLAD (R@1: 74.5%). In summary, our Patch-NetVLAD description significantly outperforms those alternative pooling strategies. Further investigation will be required to gain a deeper understanding of the complementary nature of the underlying pooling strategies and our proposed patch-based aggregation.

Patch Crops in the Image Space Instead of Feature Space: In Patch-NetVLAD, pooling is performed from a set of patches in the *feature space* of an image. One could instead perform forward passes on patch-crops in the *image space*. The main problem with this approach is that processing overlapping patches is prohibitive in terms of compute and storage (as each patch needs to be separately passed through VGG). However, overlapping patches are crucial for achieving high task performance – we found that overlapping patches are key to achieving viewpoint invariance. Therefore, performing forward passes on patch-crops in the image space is not a viable alternative to our proposed pooling of patches in the feature space.

Matching Across Different Patch Sizes: In the proposed method we match patches with other patches of the same size, but there is the possibility to match between patches of different sizes. For instance, a patch of size 2x2 could find a nearest neighbor match to a patch of size 5x5. Experiments revealed that such a cross-patch-size matching leads to sub-optimal performance: R@1 reduces from 79.5% to 78.1%. In future works, we would like to explore other matching strategies such as a coarse-to-fine matching scheme. We would also note that, conceptually, images of different zoom levels should not be matched, as they could have been taken from different places.

Complementarity of Patch Sizes: Suppl. Fig. 3 shows examples of correspondences split by patch size. We randomly sampled 10 correspondences per patch size and indicate the area covered by each patch. We include examples where small/medium/large patch sizes (*i.e.* $d_p = \{2, 5, 8\}$)



Supplementary Figure 3. **Complementarity of Patch Sizes.** The three columns indicate different patch sizes, from small (*i.e.* $d_p = 2$) over medium (*i.e.* $d_p = 5$) to large (*i.e.* $d_p = 8$). It can be observed that a small patch size is able to find matches where smaller spatial context is more intuitive, for example, near boundaries between sky and buildings (first row, left column) or between sky and power lines (third row, left column). On the other hand, a larger patch size provides complementary cues by spanning over large building surfaces, enabling matching despite significant illumination variations (second row, right column). Note that the size of the squares does not reflect the receptive field sizes of the underlying features; different sizes are used for visualization purposes only.

result in particularly good matches, as well as one example (the bottom row) where all patch sizes work well for the same image pair but in distinct areas of the image.

4. Additional Qualitative Results

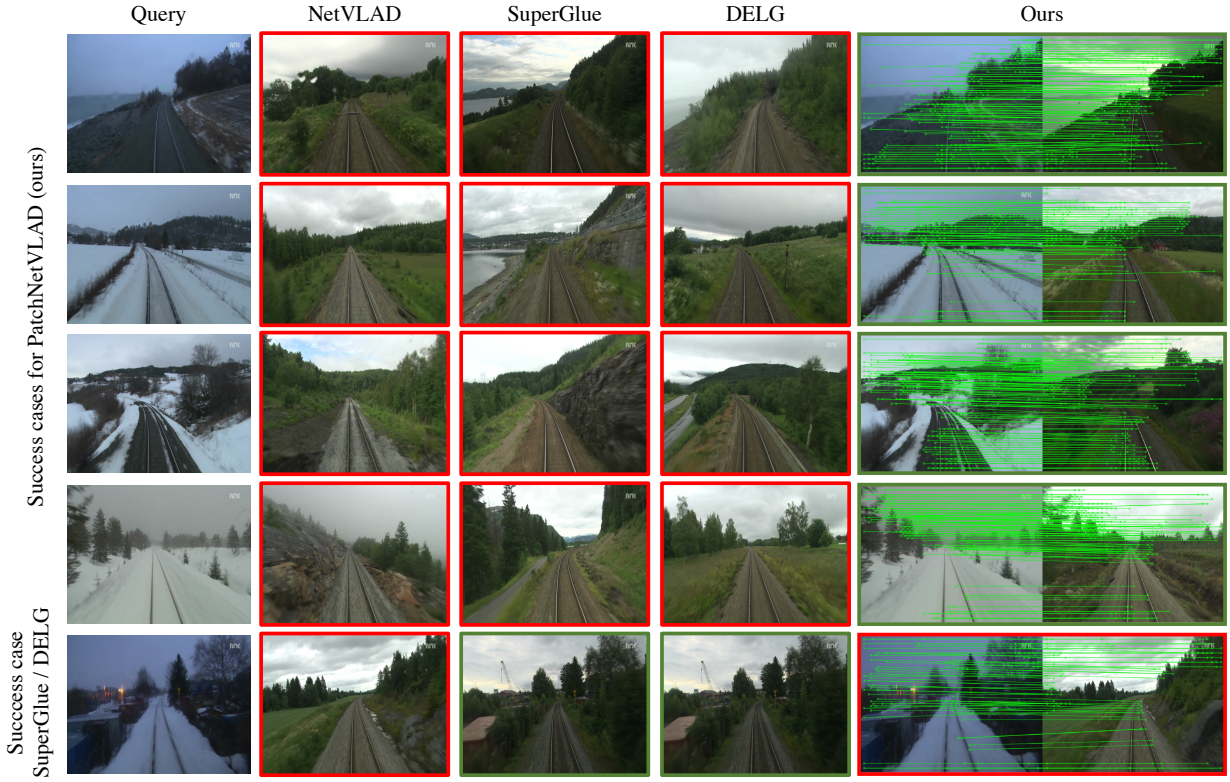
Suppl. Figs. 4, 5, 6 and 7 contain additional qualitative results on the Mapillary, Nordland, Pittsburgh and Tokyo 24/7 datasets respectively. For all these results, correct matches are represented with green borders, and incorrect matches with red borders. We show success cases of Patch-NetVLAD where all other methods failed to retrieve a correct match (with the exception of Tokyo 24/7, where DELG is also able to identify the correct image whenever Patch-NetVLAD is able to). Besides success cases, we also include failure cases where DELG and our proposed competitive SuperGlue baseline find the correct match, but Patch-NetVLAD does not localize correctly. Many of these matches are challenging to recognize as the same place, even for a human observer.

These match example visualizations lead to interesting observations. For example, in Suppl. Fig. 6 (Pittsburgh dataset), we note that a large proportion of cases where Patch-NetVLAD succeeds and DELG/Superglue fail are for images containing a large proportion of sky. We notice that SuperGlue is attempting to find correspondences between points corresponding to clouds in these images. Patch-NetVLAD, on the other hand, uses larger patch-level features which typically include clouds *and* a ground level feature. Suppl. Fig 3 illustrates this effect by showing the corresponding patch sizes at multiple scales superimposed onto the original image.

In Suppl. Fig. 8, we showcase some examples where *all* methods fail to localize correctly – those examples may guide future research to address these open challenges. We also note the ground-truth errors in the Pittsburgh dataset. In the bottom two rows of the Pittsburgh failure cases, both DELG and Patch-NetVLAD are actually finding the correct match but is being incorrectly classified as a failure due to errors in the Pittsburgh ground-truth.

Finally, Suppl. Fig. 9 provides some examples of the Pittsburgh and Mapillary datasets where a manual inspection of Patch-NetVLAD's *failure cases* has shown that Patch-NetVLAD *actually found a correct place match*, which indicates that either the error tolerances are too tight, or that some ground-truth locations are incorrectly annotated.

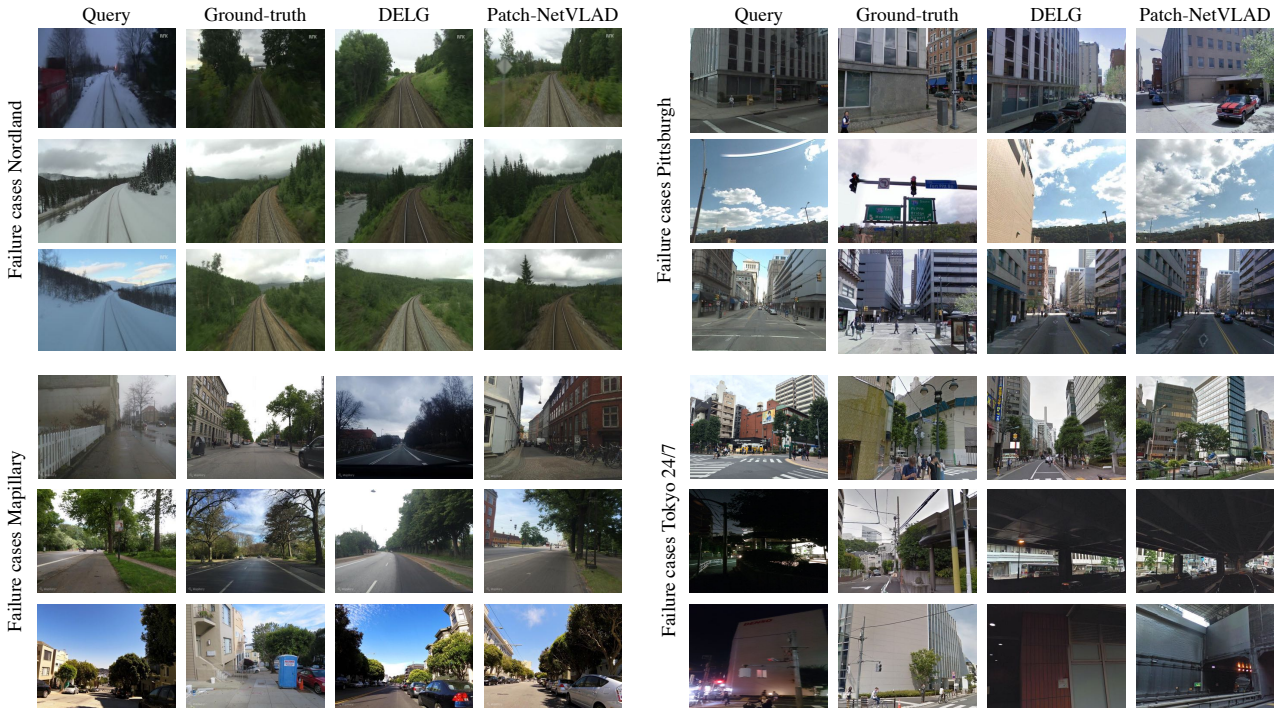
Supplementary Figure 4. **Feature correspondences for the Mapillary dataset.**



Supplementary Figure 5. **Feature correspondences for the Nordland dataset.**

Supplementary Figure 6. **Feature correspondences for the Pittsburgh dataset.**

Supplementary Figure 7. **Feature correspondences for the Tokyo 24/7 dataset.**



Supplementary Figure 8. **Cases where *all* methods fail.** We hope that these cases inform future research in VPR. Failure cases on Nordland (top left block) are mainly due to an unseen environment (for learning-based methods) and significant perceptual aliasing, *i.e.* different places look very similar. Similarly, on the Mapillary dataset (bottom left block) both SuperGlue and Patch-NetVLAD retrieve places that have a very similar structure to the query – consider for example the last row where both the query and retrieved image from Patch-NetVLAD have buildings on the left, a road in the middle, and trees on the right. On the Pittsburgh dataset (top right block), an additional challenge are the extreme viewpoint variations. Failures on Tokyo 24/7 (bottom right block) are mainly due to extreme viewpoint and appearance changes, as the query images are captured at night-time while reference images are captured at day-time. As mentioned in the Conclusions, we think that adding semantic information might aid Patch-NetVLAD in these extremely challenging cases.

5. Detailed Dataset Description

In this Section, we further detail the datasets that were introduced in Section 4.2 of the main paper. To recap, we evaluate Patch-NetVLAD on six of the key benchmark datasets: Nordland [13], Pittsburgh [18], Tokyo24/7 [17], Mapillary Streets [19], Oxford Seasons v2 [9, 15] and Extended CMU Seasons [3, 15]. Collectively the datasets encompass a wide and challenging range of viewpoint change, appearance change and acqusition methods.

Nordland: The Nordland dataset [13] is recorded from a train traveling 728km through Norway in four different seasons. The four recordings are aligned frame-by-frame using available GPS information. We use the summer and winter traverses as the reference and query sets respectively, as these traverses have the highest appearance dissimilarity and are typically considered in the literature [4, 10, 8, 7]. As in [13], we use the entire 728km traverse and subsample the video at 1 fps. Like previous works using this dataset [4, 14, 8, 7], we remove all tunnels and times when the train is stopped, which resulted in 27,592 images for both reference and query sets.

RobotCar Seasons v2: The RobotCar dataset [9] is a collection of traverses through Oxford, recorded with an autonomous car across multiple times of day and seasons. RobotCar Seasons v2 [15] is a standardized benchmark subset of the RobotCar dataset, where the reference images are recorded in overcast conditions, and the query images are captured at a variety of times and conditions: dawn, dusk, sun, rain, overcast summer, overcast winter, snow, night and night-rain. Similarly to Nordland, RobotCar Seasons v2 mainly captures appearance changes, while viewpoint changes are relatively minor.

An important detail of the structure of the RobotCar Seasons v2 dataset is that the reference images of the original RobotCar Seasons v2 dataset is split into 49 disjoint submaps which comprise the full traverse. Query images are captured from 17 of these submaps for all conditions and furthermore, for each query image the identity of the corresponding submap is provided. For our evaluation, we merge *all* 49 submaps for the reference traverse and localize each query image against the whole merged reference traverse *without using the given submap identity* provided

Supplementary Figure 9. **Cases where Patch-NetVLAD retrieved a correct match**, but this match was deemed outside the error tolerance, suggesting either that the error tolerances are too tight (compared to what a human would consider as the same place), or the possibility of slight ground truth errors. The left columns contain examples from the Mapillary dataset, while the right columns contain examples from the Pittsburgh dataset.

with the dataset. This presents a substantially more challenging image retrieval task which showcases the difference between our proposed method and alternative approaches.

Extended CMU Seasons: CMU Seasons [3] is similar to the RobotCar dataset: a car was driven around an 8.8km long route in Pittsburgh covering urban, residential, and park-like settings. Extended CMU Seasons [15] is a subset of the original CMU Seasons dataset that has been standardized for benchmark purposes. Extended CMU Seasons covers a single reference set and multiple query traverses under varying seasonal conditions spanning a one-year time frame. Unlike the RobotCar Seasons, CMU does not contain images that were captured at nighttime.

Pittsburgh: The Pittsburgh dataset [18] contains 250k images collected via Google Street View. The reference and query images are captured at different times of the day and several years apart. For each place, 24 perspective images (two pitch and twelve yaw directions) are generated, which leads to high variations in both viewpoints and appearance. We use the Pitts 30k subset as described in [1], which contains 10k reference images and 6816 query images.

Tokyo 24/7: The Tokyo 24/7 dataset [17] contains images at 125 distinct locations captured with smartphones at three different viewing directions and at three different times of the day. Contrary to the Nordland and Pittsburgh datasets, Tokyo 24/7 includes nighttime images.

Mapillary Street Level Sequences (MSLS): The MSLS dataset [19] has recently been introduced with the aim of facilitating lifelong place recognition research. It contains over 1.6 million images recorded in 30 major cities across the globe in urban and suburban areas over a period of 7 years. Compared to the other datasets, it includes variations in *all* of the following: geographical diversity, season, time of day, viewpoint, and weather. Similarly to RobotCar Seasons v2 and Extended CMU Seasons, it contains a public validation set and a withheld test set. While the dataset is suited for sequence-based methods, we only evaluate the image-to-image task.

References

[1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1437–1451, 2018.

[2] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Int. Conf. Comput. Vis.*, pages 1269–1277, 2015.

[3] Hernan Badino, Daniel Huber, and Takeo Kanade. Visual topometric localization. In *IEEE Intell. Veh. Symp.*, pages 794–799, 2011.

[4] Luis G Camara and Libor Přeučil. Visual place recognition by spatial matching of high-level CNN features. *Robot. Auton. Syst.*, 133:103625, 2020.

[5] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Eur. Conf. Comput. Vis.*, pages 726–743, 2020.

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 224–236, 2018.

[7] Stephen Hausler, Adam Jacobson, and Michael Milford. Multi-process fusion: Visual place recognition using multiple image processing methods. *IEEE Robot. Autom. Lett.*, 4(2):1924–1931, 2019.

[8] Stephen Hausler and Michael Milford. Hierarchical multi-process fusion for visual place recognition. In *IEEE Int. Conf. Robot. Autom.*, pages 3327–3333, 2020.

[9] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *Int. J. Robot. Res.*, 36(1):3—15, 2017.

[10] P. Neubert, N. Sünderhauf, and P. Protzel. Appearance change prediction for long-term navigation across seasons. In *Eur. Conf. Mobile Robot.*, pages 198–203, 2013.

[11] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Int. Conf. Comput. Vis.*, pages 5107–5116, 2019.

[12] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4938–4947, 2020.

[13] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In *IEEE Int. Conf. Robot. Autom. Workshops*, 2013.

[14] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of ConvNet features for place recognition. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 4297–4304, 2015.

[15] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, et al. Long-term visual localization revisited. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[16] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *Int. Conf. Learn. Represent.*, 2016.

[17] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 Place Recognition by View Synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(2):257–271, 2018.

[18] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. Visual place recognition with repetitive structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(11):2346–2359, 2015.

[19] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2626–2635, 2020.