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

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

Abstract

Visual Place Recognition is a challenging task for robotics and autonomous systems, which must deal with the twin problems of appearance and viewpoint change in an always changing world. This paper introduces Patch-NetVLAD, which provides a novel formulation for combining the advantages of both local and global descriptor methods by deriving patch-level features from NetVLAD residuals. Unlike the fixed spatial neighborhood regime of existing local keypoint features, our method enables aggregation and matching of deep-learned local features defined over the feature-space grid. We further introduce a multi-scale fusion of patch features that have complementary scales (i.e. patch sizes) via an integral feature space and show that the fused features are highly invariant to both condition (season, structure, and illumination) and viewpoint (translation and rotation) changes. Patch-NetVLAD achieves state-of-the-art visual place recognition results in computationally limited scenarios, validated on a range of challenging real-world datasets, including winning the Facebook Mapillary Visual Place Recognition Challenge at ECCV2020. It is also adaptable to user requirements, with a speed-optimised version operating over an order of magnitude faster than the state-of-the-art. By combining superior performance with improved computational efficiency in a configurable framework, Patch-NetVLAD is well suited to enhance both stand-alone place recognition capabilities and the overall performance of SLAM systems.

1. Introduction

Visual Place Recognition (VPR) is a key prerequisite for many robotics and autonomous system applications, both as a stand-alone positioning capability when using a prior map and as a key component of full Simultaneous Localization And Mapping (SLAM) systems. The task can prove challenging because of major changes in appearance, illumination and even viewpoint, and is therefore an area of active research in both the computer vision [3, 19, 35, 43, 79, 81, 82] and robotics [10, 11, 12, 27, 39, 45] communities.

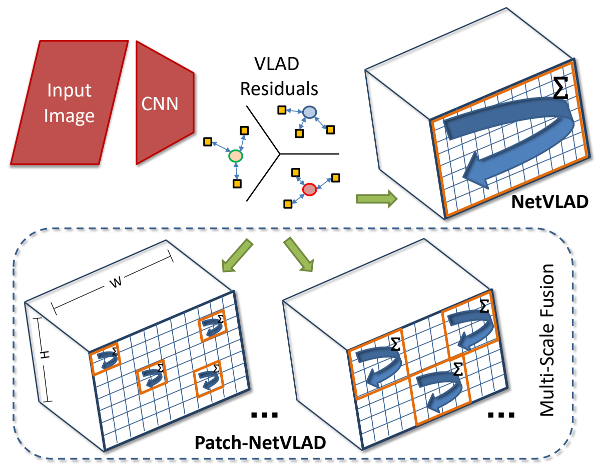

Figure 1. **Patch-NetVLAD** is a novel condition and viewpoint invariant visual place recognition system that produces a similarity score between two images through local matching of *locally-global* descriptors extracted from a set of *patches* in the feature space of each image. Furthermore, by introducing an integral feature space, we are able to derive a multi-scale approach that fuses multiple patch sizes. This is in contrast with the original NetVLAD paper, which performs an appearance only aggregation of the whole feature space into a single global descriptor.

VPR is typically framed as an image retrieval task [45, 59, 76, 24], where, given a query image, the most similar database image (alongside associated metadata such as the camera pose) is retrieved. There are two common ways to represent the query and reference images: using global descriptors which describe the whole image [3, 79, 27, 59, 10, 58, 82], or using local descriptors that describe areas of interest [20, 52, 18, 23, 14]. Global descriptor matching is typically performed using nearest neighbor search between query and reference images. These global descriptors typically excel in terms of their robustness to appearance and illumination changes, as they are directly optimized for place recognition [3, 59]. Conversely, local descriptors are usually cross-matched, followed by geometric verification. Local descriptor techniques prioritize spatial precision, predominantly on a pixel-scale level, using a fixed-size spatial neighborhood to facilitate highly-accurate 6-DoF pose estimation. Given the complementary strengths of local and global approaches, there has been little research [9, 65, 76] attempting to combine them. The novel Patch-NetVLAD system proposed here combines the mutual strengths of local and global approaches while minimizing their weaknesses.

To achieve this goal, we make a number of contributions (see Fig. 1): First, we introduce a novel place recognition system that generates a similarity score between an image pair through a spatial score obtained through exhaustive matching of *locally-global* descriptors. These descriptors are extracted for densely-sampled local patches within the feature space using a VPR-optimized aggregation technique (in this case NetVLAD [3]). Second, we propose a multi-scale fusion technique that generates and combines these hybrid descriptors of different sizes to achieve improved performance over a single scale approach. To minimize the computational growth implications of moving to a multi-scale approach, we develop an integral feature space (analogous to integral images) to derive the local features for varying patch sizes. Together these contributions provide users with flexibility based on their task requirements: our final contribution is the demonstration of a range of easily implemented system configurations that achieve different performance and computational balances, including a performance-focused configuration that achieves state-of-the-art recall performance when tight error thresholds are required, a balanced configuration that performs almost as well as the state-of-the-art while being $3\times$ faster than SuperGlue and $28\times$ faster than DELG, and a speed-focused configuration that is at least an order of magnitude faster than the state-of-the-art.

We extensively evaluate the versatility of our proposed system on a large number of well-known datasets [71, 46, 80, 79, 82] that capture all challenges present in VPR. We compare Patch-NetVLAD with several state-of-the-art global feature descriptor methods [3, 59, 79], and the recent global and local descriptor DELG [9]. We additionally introduce new SuperPoint [18] and SuperGlue [62]-enabled VPR pipelines as competitive local descriptor baselines. Patch-NetVLAD outperforms the global feature descriptor methods by large margins (from 6% to 330% relative increase) across all datasets, and achieves superior performance (up to a relative increase of 54%) when compared to SuperGlue. Patch-NetVLAD outperforms DELG [9] in some datasets while performing worse in others; however, the order-of-magnitude difference in compute speed makes Patch-NetVLAD much more suitable in practical scenarios. Patch-NetVLAD won the Facebook Mapillary Long-term Localization Challenge as part of the ECCV 2020 Workshop on Long-Term Visual Localization. To characterise the system's properties in detail, we conduct numerous ablation studies showcasing the role of the individual components comprising Patch-NetVLAD, particularly the robustness of the system to changes in various key parameters. To foster future research, we make our code available for research purposes: `https://github.com/QVPR/Patch-NetVLAD`.

2. Related Work

Global Image Descriptors: Notable early global image descriptor approaches include aggregation of local keypoint descriptors either through a Bag of Words (BoW) scheme [67, 16], Fisher Vectors (FV) [37, 56] or Vector of Locally Aggregated Descriptors (VLAD) [38, 2]. Aggregation can be based on either sparse keypoint locations [67, 38] or dense sampling of an image grid [79]. Re-formulating these methods through deep learning-based architectures led to NetVLAD [3], NetBoW [48, 54] and NetFV [48]. More recent approaches include ranking-loss based learning [59], novel pooling [58], contextual feature reweighting [40], large scale re-training [82], semantics-guided feature aggregation [28, 64, 75], use of 3D [53, 81, 43], additional sensors [32, 55, 22], sequences [25, 82], and image appearance translation [1, 57]. Place matches obtained through global descriptor matching are often re-ranked using sequential information [26, 85, 49], query expansion [31, 13], geometric verification [41, 27, 52] and feature fusion [83, 86]. Distinct from existing approaches, this paper introduces Patch-NetVLAD, which reverses the local-to-global process of image description by deriving multi-scale patch features from a global descriptor, NetVLAD.

Local Keypoint Descriptors: Local keypoint methods are often used to re-rank initial place match candidate lists produced by a global approach [23, 74, 61]. Traditional hand-crafted local feature methods such as SIFT [44], SURF [6] and ORB [60], and more recent deep-learned local features like LIFT [84], DeLF [52], SuperPoint [18] and D2Net [20], have been extensively employed for VPR [52, 9, 17], visual SLAM [50] and 6-DoF localization [63, 77, 20, 61]. The two most common approaches of using local features for place recognition are: 1) local aggregation to obtain global image descriptors [52] and 2) cross-matching of local descriptors between image pairs [74].

Several learning-based techniques have been proposed for spatially-precise keypoint matching. These include a unified framework for detection, description and orientation estimation [84]; a 'describe-then-detect' strategy [20]; multi-layer explicit supervision [21]; scale-aware negative mining [68]; and contextual similarity-based unsupervised training [69]. However, the majority of these learning-based methods aim for 3D pose estimation, optimizing *keypoint*-level descriptors and nearest neighbor matching. Local descriptors can be *further* improved by utilizing the larger spatial context, especially beyond the CNN's inherent hierarchical feature pyramid, a key motivation for our approach.

Local Region/Patch Descriptors: [72] proposed *ConvNet Landmarks* for representing and cross-matching large image regions, explicitly derived from Edge Boxes [91]. [12] discovered landmarks implicitly from CNN activations using mean activation energy of regions defined as '8-connected' feature locations. [8] composed region features from CNN activation tensors by concatenating individual spatial elements along the channel dimension. However, these off-the-shelf CNNs or handcrafted region description [87] approaches are not optimized for place recognition, unlike the use of the VPR-trained network in this work.

Learning region descriptors has been studied for specific tasks [65, 47] as well as independently on image patches [70]. In the context of VPR, [29] designed a regions-based self-supervised learning mechanism to improve global descriptors using *image-to-region* matching during training. [89] modeled relations between regions by concatenating local descriptors to learn an improved global image descriptor based on K-Max pooling rather than sum [4] or max pooling (R-MAC) [78]. [11] proposed a 'context-flexible' attention mechanism for variable-size regions. However, the learned attention masks were only employed for *viewpoint-assumed* place recognition and could potentially be used for region selection in our proposed VPR pipeline. [76] proposed *R-VLAD* for describing regions extracted through a trained landmark detector, and combined it with selective match kernels to improve global descriptor matching, thus doing away with cross-region comparisons. [39] proposed *RegionVLAD* where region features were defined using average activations of connected components within different layers of CNN feature maps. These region features were then *separately* aggregated as a VLAD representation. Unlike [39, 76], we remove this separate step by generating region-level VLAD descriptors through NetVLAD, thus reusing the VPR-relevant learned cluster membership of spatial elements.

Existing techniques for multi-scale approaches typically fuse information at the descriptor level, which can lead to loss of complementary or discriminative cues [83, 86, 10, 51, 30, 90] due to pooling, or increased descriptor sizes due to concatenation [42, 89, 7, 8]. Distinct from these methods, we consider multi-scale fusion at the final scoring stage, which enables parallel processing with associated speed benefits.

3. Methodology

Patch-NetVLAD ultimately produces a similarity score between a pair of images, measuring the spatial and appearance consistency between these images. Our hierarchical approach first uses the original NetVLAD descriptors to retrieve the top-k (we use $k = 100$ in our experiments) most likely matches given a query image. We then compute a new type of patch descriptor using an alternative to the VLAD layer used in NetVLAD [3], and perform local matching of *patch-level descriptors* to *reorder* the initial match list and refine the final image retrievals. This combined approach minimizes the additional overall computation cost incurred by cross matching patch features without sacrificing recall performance at the final image retrieval stage. An overview of the complete pipeline can be found in Fig. 2.

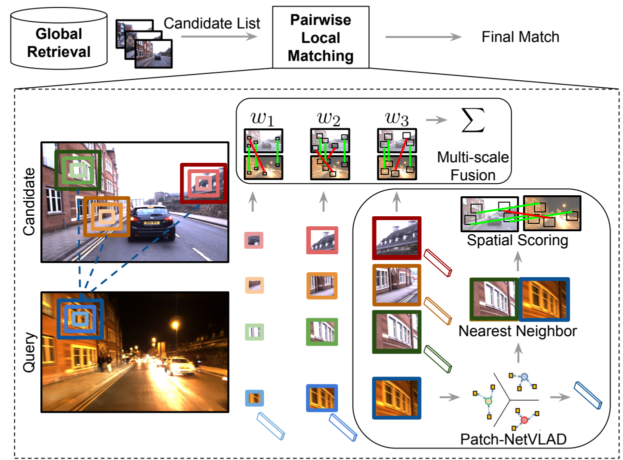

Figure 2. **Proposed algorithm schematic.** Patch-NetVLAD takes as input an initial list of most likely reference matches to a query image, ranked using NetVLAD descriptor comparisons. For top-ranked candidate images, we compute new *locally-global* patch-level descriptors at multiple scales, perform local cross-matching of these descriptors across query and candidate images with geometric verification, and use these match scores to re-order the initial list, producing the final image retrievals.

3.1. Original NetVLAD Architecture

The original NetVLAD [3] network architecture uses the Vector-of-Locally-Aggregated-Descriptors (VLAD) approach to generate a condition and viewpoint invariant embedding of an image by aggregating the intermediate feature maps extracted from a pre-trained Convolutional Neural Network (CNN) used for image classification [66]. Specifically, let $f_\theta : I \rightarrow \mathbb{R}^{H\times W\times D}$ be the base architecture which given an image I, outputs a $H \times W \times D$ dimensional feature map F (*e.g.* the conv5 layer for VGG). The original NetVLAD architecture aggregates these D-dimensional features into a $K \times D$-dimensional matrix by summing the residuals between each feature $\mathbf{x}_i \in \mathbb{R}^D$ and K learned cluster centers weighted by soft-assignment. Formally, for $N \times D$-dimensional features, let the VLAD aggregation layer $f_{\text{VLAD}} : \mathbb{R}^{N\times D} \rightarrow \mathbb{R}^{K\times D}$ be given by

$$f_{\text{VLAD}}(F)(j,k) = \sum_{i=1}^{N} \bar{a}_k(\mathbf{x}_i)(x_i(j) - c_k(j)), \quad (1)$$

where $x_i(j)$ is the j^{th} element of the i^{th} descriptor, $\bar{a}_k$ is the soft-assignment function and $\mathbf{c}_k$ denotes the k^{th} cluster center. After VLAD aggregation, the resultant matrix is then projected down into a dimensionality reduced vector using a *projection layer* $f_{\text{proj}}\colon \mathbb{R}^{K\times D} \to \mathbb{R}^{D_{\text{proj}}}$ by first applying intra(column)-wise normalization, unrolling into a single vector, L2-normalizing in its entirety and finally applying PCA (learned on a training set) with whitening and L2-normalization. Refer to [3] for more details.

We use this feature-map aggregation method to extract descriptors for *local patches* within the whole feature map $(N \ll H \times W)$ and perform cross-matching of these patches at multiple scales between a query/reference image pair to generate the final similarity score used for image retrieval. This is in contrast to the original NetVLAD paper, which sets $N = H \times W$ and aggregates *all* of the descriptors within the feature map to generate a global image descriptor.

3.2. Patch-level Global Features

A core component of our system revolves around extracting global descriptors for densely sampled sub-regions (in the form of *patches*) within the full feature map. We extract a set of $d_x \times d_y$ patches $\{P_i, x_i, y_i\}_{i=1}^{n_p}$ with stride s_p from the feature map $F \in \mathbb{R}^{H\times W\times D}$, where the total number of patches is given by

$$n_p = \left\lfloor \frac{H - d_y}{s_p} + 1 \right\rfloor * \left\lfloor \frac{W - d_x}{s_p} + 1 \right\rfloor, d_y, d_x \leq H, W \tag{2}$$

and $P_i \in \mathbb{R}^{(d_x\times d_y)\times D}$ and x_i, y_i are the set of patch features and the coordinate of the center of the patch within the feature map, respectively. While our experiments suggest that square patches yielded the best generalized performance across a wide range of environments, future work could consider different patch shapes, especially in specific circumstances (*e.g*. environments with different texture frequencies in the vertical and horizontal directions).

For each patch, we subsequently extract a descriptor yielding the patch descriptor set $\{\mathbf{f}_i\}_{i=1}^{n_p}$ where $\mathbf{f}_i = f_{\text{proj}}\left(f_{\text{VLAD}}\left(P_i\right)\right) \in \mathbb{R}^{D_{\text{proj}}}$ uses the NetVLAD aggregation and projection layer on the relevant set of patch features. In all experiments we show how varying the degree of dimensionality reduction on the patch features using PCA can be used to achieve a user-preferred balance of computation time and image retrieval performance (see Section 4.5). We can further improve place recognition performance by extracting patches at multiple scales and observe that using a combination of patch sizes which represent larger sub-regions within the original image improves retrieval (see Section 3.5). This multi-scale fusion is made computationally efficient using our IntegralVLAD formulation introduced in Section 3.6.

Compared to local feature-based matching where features are extracted for comparatively small regions within the image, our patch features implicitly contain *semantic* information about the scene (*e.g*., building, window, tree) by covering a larger area. We now introduce the remaining parts of our pipeline, which is comprised of mutual nearest neighbor matching of patch descriptors followed by spatial scoring.

3.3. Mutual Nearest Neighbors

Given a set of reference and query features $\{\mathbf{f}_i^r\}_{i=1}^{n_p}$ and $\{\mathbf{f}_i^q\}_{i=1}^{n_p}$, (we assume both images have the same resolution for simplicity), we obtain descriptor pairs from mutual nearest neighbor matches through exhaustive comparison between the two descriptor sets. Formally, let the set of mutual nearest neighbor matches be given by $\mathcal{P}$, where

$$\mathcal{P} = \left\{(i, j)\colon i = \text{NN}_r(\mathbf{f}_j^q),\ j = \text{NN}_q(\mathbf{f}_i^r)\right\} \tag{3}$$

and $\text{NN}_q(\mathbf{f}) = \operatorname{argmin}_j \|\mathbf{f} - \mathbf{f}_j^q\|_2$ and $\text{NN}_r(\mathbf{f}) = \operatorname{argmin}_j \|\mathbf{f} - \mathbf{f}_j^r\|_2$ retrieve the nearest neighbor descriptor match with respect to Euclidean distance within the query and reference image set, respectively. Given a set of matching patches, we can now compute the spatial matching score used for image retrieval.

3.4. Spatial Scoring

We now introduce our spatial scoring methods which yield an image similarity score between a query/reference image pair used for image retrieval. We present two alternatives, a RANSAC-based scoring method which requires more computation time for higher retrieval performance and a spatial scoring method which is substantially faster to compute with only a minor decrease in image retrieval performance.

RANSAC Scoring: Our spatial consistency score is given by the number of inliers returned when fitting a homography between the two images, using corresponding patches computed using our mutual nearest neighbor matching step for patch features. We assume each patch corresponds to a 2D image point with coordinates in the center of the patch when fitting the homography. We set the error tolerance for the definition of an inlier to be the stride s_p. We also normalize our consistency score by the number of patches, which is relevant when combining the spatial score at multiple scales as discussed in Section 3.5.

Rapid Spatial Scoring: We also propose an alternative to the RANSAC scoring approach which we call *rapid spatial scoring*. This rapid spatial scoring significantly reduces computation time as we can compute this score directly on the matched feature pairs without requiring sampling.

To compute the rapid spatial score, let $x_d = \{x_i^r - x_j^q\}_{(i,j)\in\mathcal{P}}$ be the set of displacements in the horizontal direction between patch locations for the matched patches, and y_d be the displacements in the vertical direction. In addition, let $\bar{x}_d = \frac{1}{|x_d|}\sum_{x_{d,i}\in x_d} x_{d,i}$ and similarly $\bar{y}_d$ be the mean displacements between matched patch locations. We can then define our spatial score (higher is better) to be

$$s_{\text{spatial}} = \frac{1}{n_p} \sum_{i \in \mathcal{P}} \Big(|\max_{j \in \mathcal{P}} x_{d,j}| - |x_{d,i} - \bar{x}_d| \Big)^2 + \Big(|\max_{j \in \mathcal{P}} y_{d,j}| - |y_{d,i} - \bar{y}_d| \Big)^2, \quad (4)$$

where the score comprises the sum of residual displacements from the mean, with respect to the maximum possible spatial offset. The spatial score penalizes large spatial offsets in matched patch locations from the mean offset, in effect measuring the coherency in the overall movement of elements in a scene under viewpoint change.

3.5. Multiple Patch Sizes

We can easily extend our scoring formulation to ensemble patches at multiple scales and further improve performance. For n_s different patch sizes, we can take a convex combination of the spatial matching scores for each patch size as our final matching score. Specifically,

$$s_{\text{spatial}} = \sum_{i=1}^{n_s} w_i s_{i,\text{spatial}}, \quad (5)$$

where $s_{i,\text{spatial}}$ is the spatial score for the i^{th} patch size and $\sum_i w_i = 1$, $w_i \geq 0$ for all i.

3.6. IntegralVLAD

To assist the computation of extracting patch descriptors at multiple scales, we propose a novel *IntegralVLAD* formulation analogous to integral images [15]. To see this, note that the aggregated VLAD descriptor (before the projection layer) for a patch can be computed as the sum of all 1×1 patch descriptors each corresponding to a single feature within the patch. This allows us to pre-compute an integral *patch feature map* which can then be used to compute patch descriptors for multi-scale fusion. Let the integral feature map $\mathcal{I}$ be given by

$$\mathcal{I}(i,j) = \sum_{i'<i, j'<j} \mathbf{f}^1_{i',j'}, \quad (6)$$

where $\mathbf{f}^1_{i',j'}$ represents the VLAD aggregated patch descriptor (before projection) for a patch size of 1 at spatial index i', j' in the feature space. We can now recover the patch features for arbitrary scales using the usual approach involving arithmetic over four references within the integral feature map. This is implemented in practice through 2D depth-wise dilated convolutions with kernel K, where

$$K = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (7)$$

and the dilation is equal to the required patch size.

4. Experimental Results

4.1. Implementation

We implemented Patch-NetVLAD in PyTorch and resize all images to 640 by 480 pixels before extracting our patch features. We train the underlying vanilla NetVLAD feature extractor [3] on two datasets: Pittsburgh 30k [80] for urban imagery (Pittsburgh and Tokyo datasets), and Mapillary Street Level Sequences [82] for all other conditions. All hyperparameters for training are the same as in [3], except for the Mapillary trained model that uses 16 instead of 64 clusters for faster training due to the large dataset size.

To find the patch sizes and associated weights, we perform a grid search to find the model configuration that performs best on the RobotCar Seasons v2 training set. This resulted in patch size $d_x = d_y = 5$ (which equates to an 228 by 228 pixel area in the original image) with stride $sp = 1$ when a single patch size is used, and square patch sizes 2, 5 and 8 with associated weights $w_i = 0.45, 0.15, 0.4$ for the multi-scale fusion. We emphasize that this single configuration is used for *all experiments across all datasets*. It is likely that if a highly specialized system was required, further performance increases could be achieved by fine-tuning patch sizes and associated weights for the specific environment.

4.2. Datasets

To evaluate Patch-NetVLAD, we used six of the key benchmark datasets: Nordland [71], Pittsburgh [80], Tokyo24/7 [79], Mapillary Streets [82], RobotCar Seasons v2 [46, 77] and Extended CMU Seasons [5, 77]. Full technical details of their usage are provided in the Supplementary Material; here we provide an overview to facilitate an informed appraisal of the results. Datasets were used in their recommended configuration for benchmarking, including standardized curation (*e.g*. removal of pitch black tunnels and times when the train is stopped for the Nordland dataset [8, 73, 34, 33]) and use of public validation and withheld test sets where provided (*e.g*. Mapillary).

Collectively the datasets encompass a challenging range of viewpoint and appearance change conditions, partly as a result of significant variations in the acquisition method, including train, car, smartphone and general crowdsourcing. Specific appearance changes are caused by different times of day: dawn, dusk, night; by varying weather: sun, overcast, rain, and snow; and by seasonal change: from summer to winter. Nordland, RobotCar, Extended CMU Seasons and Tokyo 24/7 contain varying degrees of appearance change up to very severe day-night and seasonal changes. The Pittsburgh dataset contains both appearance and viewpoint change, while the MSLS dataset [82] in particular includes simultaneous variations in *all* of the following: geographical diversity (30 major cities across the globe), season, time of day, date (over 7 years), viewpoint, and weather. In

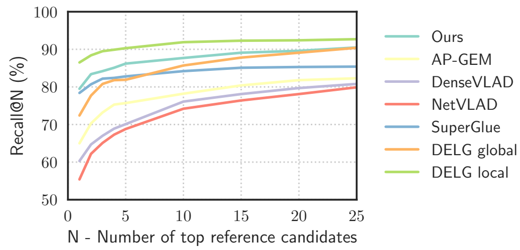

Figure 3. **Comparison with state-of-the-art.** We show the Recall@N performance of Ours (Multi-RANSAC-Patch-NetVLAD) compared to AP-GEM [59], DenseVLAD [79], NetVLAD [3], SuperGlue [62], and DELG [9] on Mapillary val. set.

total, we evaluate our systems on ≈300,000 images.

4.3. Evaluation

All datasets except for RobotCar Seasons v2 and Extended CMU Seasons are evaluated using the Recall@N metric, whereby a query image is correctly localized if at least one of the top N images is within the ground truth tolerance [3, 79]. The recall is then the percentage of correctly localized query images, and plots are created by varying N.

We deem a query to be correctly localized within the standard ground-truth tolerances for all datasets, *i.e.* 10 frames for Nordland [33, 34], 25m translational error for Pittsburgh and Tokyo 24/7 [3], and 25m translational and 40° orientation error for Mapillary [82].

For RobotCar Seasons v2 and Extended CMU Seasons, we use the default error tolerances [77], namely translational errors of .25, .5 and 5.0 meters and corresponding rotational errors of 2, 5 and 10 degrees. Note that our method is a place recognition system and does not perform explicit 6-DOF pose estimation; the pose estimate for a query image is given by inheriting the pose of the best matched reference image.

For all methods, images are resized to 640 by 480 pixels.

4.4. Comparison to State-of-the-art Methods

We compare against several benchmark localization solutions based on retrieval using global descriptors only: AP-GEM [59], DenseVLAD [79], and NetVLAD [3].

We also compare against two recent methods which additionally perform re-ranking of global retrievals using spatial verification of local features, similar to Patch-NetVLAD. The first method, DELG [9] proposes a network architecture which jointly extracts local and global features for image retrieval. The DELG global descriptor is used for global retrieval. The second method is based on SuperGlue [62] and extracts SuperPoint [18] local features before applying SuperGlue to identify matches. We use the number of proposed matches to re-rank global retrievals from NetVLAD [3].

Table 1 and Fig. 3 contain quantitative comparisons of Patch-NetVLAD and the baseline methods. Patch-NetVLAD outperforms the best performing global descriptor methods, NetVLAD, DenseVLAD, AP-GEM and DELG (global descriptor only), on average by 18.6%, 19.3%, 22.6% and 15.8% (all percentages stated are absolute differences for R@1) respectively. The differences are particularly pronounced in datasets with large appearance variations, *i.e.* Nordland and Extended CMU Seasons (seasonal changes), Tokyo 24/7 (including night time imagery), and RobotCar Seasons as well as Mapillary (both seasonal changes and night time imagery). On Nordland, the difference between Patch-NetVLAD and the original NetVLAD is 34.5%.

Patch-NetVLAD also yields competitive performance to alternative systems which utilize local feature re-ranking. Patch-NetVLAD yields 3.1% better performance on average compared to SuperGlue (a relative increase of 6.0%) despite a lack of a learned local feature matcher. We hypothesize that Patch-NetVLAD's performance could further benefit from SuperGlue's learned matcher and discuss this opportunity further in Section 5. Patch-NetVLAD's performance edge is particularly significant when large appearance variations are encountered in *unseen* environments – not typically used for training local feature methods like SuperGlue. Thus, Patch-NetVLAD achieves superior performance on Nordland with an *absolute* percentage difference of 15.8%.

For DELG, Patch-NetVLAD yields considerably higher performance on the RobotCar Seasons v2 and Extended CMU Seasons dataset where pose error tolerances for a correct retrieval are very low compared to the other datasets. Comparing DELG global retrieval to DELG global + local re-ranking indicates that utilizing local features harms retrieval performance. DELG achieves better performance on the remaining datasets; this comes at a high computational cost (≈ 10 times slower than our best-performing and ≈ 240 times slower than our speed-focused method; see Fig. 4).

Patch-NetVLAD won the Mapillary Challenge at the ECCV2020 workshops, with Table 1 showing that Patch-NetVLAD outperformed the baseline method, NetVLAD, by 13.0% (absolute R@1 increase) on the withheld test dataset. The test set was more challenging than the validation set (48.1% R@1 and 79.% R@1 respectively; note that no fine-tuning was performed on any of the datasets), indicating that the Mapillary test set is a good benchmarking target for further research compared to "near-solved" datasets like Pittsburgh and Tokyo 24/7, where Patch-NetVLAD, SuperGlue and DELG achieve near perfect performance.

Fig. 5 illustrates example matches retrieved with our method compared to NetVLAD, SuperGlue, and DELG.

4.5. Ablation Studies

Single-scale and Spatial Scoring: To analyze the effectiveness of Patch-NetVLAD, we compare with the following variations: 1) Single-RANSAC-Patch-NetVLAD uses a single patch size (*i.e.* 5) instead of multi-scale fusion. 2) Single-Spatial-Patch-NetVLAD employs a simple but rapid spatial

Table 1. Quantitative results

Method	Nordland			Mapillary (Challenge)			Mapillary (Val. set)			Pittsburgh 30k			Tokyo 24/7			RobotCar Seasons v2			Extended CMU Seasons		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	.25m/2°	.5m/5°	5.0m/10°	.25m/2°	.5m/5°	5.0m/10°
AP-GEM [59]	11.1	13.2	16.1	—	—	—	65.0	75.7	78.2	80.7	91.4	94.0	58.4	69.5	74.3	5.7	20.7	69.5	4.4	13.4	65.7
DenseVLAD [79]	11.9	20.8	26.2	—	—	—	60.3	70.1	76.1	78.2	88.8	92.3	59.4	67.0	71.8	7.5	28.3	80.7	7.7	23.8	79.7
NetVLAD [3]	13.0	20.6	25.0	35.1	47.4	51.7	58.6	71.2	76.1	85.1	92.2	94.4	64.4	78.4	81.6	7.0	24.9	76.6	5.9	18.0	76.9
SuperGlue [62]	29.1	33.4	35.0	—	—	—	78.4	82.8	84.2	88.7	95.1	96.4	88.2	90.2	90.2	8.3	32.4	89.9	9.5	30.7	**96.7**
DELG global [9]	23.4	35.4	41.7	—	—	—	72.4	81.9	85.7	80.4	90.0	93.4	77.8	87.9	91.8	5.0	19.4	73.3	3.6	11.4	64.3
DELG local [9]	**60.1**	**63.5**	**64.6**	—	—	—	**86.5**	**90.3**	**91.9**	**90.0**	**95.7**	**97.0**	**95.9**	**96.8**	**97.1**	2.6	9.9	83.4	5.7	21.1	93.6
Ours	44.9	50.2	52.2	**48.1**	**57.6**	**60.5**	79.5	86.2	87.7	88.7	94.5	95.9	86.0	88.6	90.5	**9.6**	**35.3**	**90.9**	**11.8**	**36.2**	96.2

Table 2. Ablation study

Method	Nordland			Mapillary (Val. set)			Pittsburgh 30k			Tokyo 24/7			RobotCar Seasons v2			Extended CMU Seasons		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	.25m/2°	.5m/5°	5.0m/10°	.25m/2°	.5m/5°	5.0m/10°
Ours (Single-Spatial-Patch-NetVLAD)	42.9	49.2	51.6	77.2	85.4	87.3	88.0	94.0	95.6	78.1	83.8	87.0	8.7	32.4	88.4	10.0	31.5	95.2
Ours (Single-RANSAC-Patch-NetVLAD)	42.4	48.8	51.2	77.8	85.7	**87.8**	87.3	94.2	95.7	82.2	87.3	89.2	8.7	31.6	88.3	10.0	31.3	94.5
Ours (Multi-Spatial-Patch-NetVLAD)	44.5	50.1	52.0	78.2	85.3	86.9	88.6	**94.5**	95.8	81.9	85.7	87.9	9.4	33.9	89.3	11.1	34.5	**96.3**
Ours (Multi-RANSAC-Patch-NetVLAD)	**44.9**	**50.2**	**52.2**	**79.5**	**86.2**	87.7	**88.7**	**94.5**	**95.9**	**86.0**	**88.6**	**90.5**	**9.6**	**35.3**	**90.9**	**11.8**	**36.2**	96.2

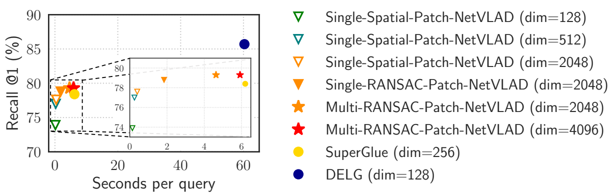

Figure 4. **Computational time requirements.** The time taken to process one query image is shown on the x-axis, with the resulting R@1 shown on the y-axis, for the Mapillary dataset. Our pipeline enables a range of system configurations that achieve different performance and computational balances that either outperform or are far faster than current state-of-the-art.

verification method applied to a single patch size (see Section 3.4). 3) Multi-Spatial-Patch-NetVLAD uses the same rapid spatial verification method, however applied to three patch sizes rather than a single patch size as in the previous variant. Table 2 shows comparison results with these variations and the following numeric results are based on R@1 (recall@1) – the conclusions generally apply to R@5 and R@10 as well. Our proposed multi-fusion approach (Multi-RANSAC-Patch-NetVLAD) performs on average 2.0% better than Single-RANSAC-Patch-NetVLAD, demonstrating that a fusion of multiple patch sizes significantly improves task performance. Our approach also provides some compelling options for compute-constrained applications; our rapid spatial verification approach is 2.9 times faster on a single patch size (Single-Spatial-Patch-NetVLAD), with only a 0.6% performance reduction. Rapid spatial verification applied to multiple patch sizes (Multi-Spatial-Patch-NetVLAD) is 3.1 times faster, with only a 1.1% performance degradation.

Patch Descriptor Dimension: In addition to disabling multi-scale fusion and using our rapid spatial scoring method, the descriptor dimension can be arbitrarily reduced using PCA (as with the original NetVLAD). Here, we choose $D_{\text{PCA}} = \{128, 512, 2048, 4096\}$. Fig. 4 shows the number of queries that can be processed per second by various configurations[1], and the resulting R@1 on the Mapillary validation set. Our proposed Multi-RANSAC-Patch-NetVLAD in a performance-focused configuration (red star in Fig. 4) achieves 1.1% higher recall than SuperGlue (yellow dot) while being slightly (3%) faster. All of our variants are much (10-240 times) faster than DELG. A balanced configuration (orange triangle) is more than 3 times faster than SuperGlue with comparable performance, while a speed-oriented configuration (blue triangle) is 15 times faster at the expense of just 0.6% and 1.7% recall when compared to SuperGlue and our performance-focused configuration respectively. A storage-focused configuration ($D_{\text{PCA}} = 128$) still largely outperforms NetVLAD while having similar memory requirements as a SIFT-like descriptor. Our speed-oriented and storage-focused configurations provide practical options for applications like time-critical robotics. Our approach can also run on consumer GPUs, with our performance configuration requiring 7GB GPU memory (batch-size of 5).

4.6. Further Analysis

We further study the robustness of our approach to the choice of hyperparameters. In Fig. 6 (left) we highlight that Single-Patch-NetVLAD is robust to the choice of the patch size d_p: The performance gradually decays from a peak at $d_p = 4$. Fig. 6 (right) similarly shows that Patch-NetVLAD is robust to the convex combination of the multi-patch fusion in terms of the patch sizes that are fused. The Supplementary Material provides additional ablation studies, including matching across different patch sizes, complementarity of patch sizes and comparison to other pooling strategies.

5. Discussion and Conclusion

In this work we have proposed a novel *locally-global* feature descriptor, which uses global descriptor techniques to further improve the appearance robustness of local descriptors. Unlike prior keypoint-based local feature descrip-

[1] These results include both feature extraction and matching times; the Supplementary Material contains further figures that separate feature extraction and feature matching times.

Figure 5. **Qualitative Results.** In these examples, the proposed Patch-NetVLAD successfully retrieves the matching reference image, while NetVLAD, SuperGlue and DELG produce incorrect place matches (except Tokyo where DELG also finds the correct match). The Tokyo 24/7 match example is particularly challenging, with a combination of day vs night-time, severe viewpoint shift and occlusions.

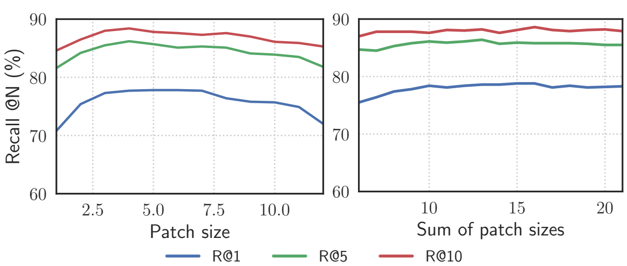

Figure 6. **Robustness studies for single patch sizes and combined patch sizes.** Left: Recall performance of Single-Patch-NetVLAD with varying patch size, using the Mapillary validation dataset. Performance gradually degrades around a peak at $d_p = 4$. The smallest and largest patch sizes perform most poorly, indicating that both local features and global areas are inferior to intermediate size features. An additional issue with large patch sizes is that there are too few patches for effective spatial verification. Right: Recall performance of Multi-Patch-NetVLAD against an indicative measure of cumulative patch dimensions. Our proposed combination of patch dimensions 2, 5 and 8 corresponds to an x-axis value of 15; data points to the left show a reduction in the cumulative patch dimension (*e.g.* $\sum_i d_{p,i} = 14$ for patch sizes 1, 5 and 8; sizes 2, 4 and 8; and sizes 2, 5 and 7) and so forth; and similarly for increasing patch size combinations to the right. As for variations on the single patch size, performance gracefully degrades around the peak and remains high over a large range.

tors [44, 18], our approach considers all the visual content within a larger *patch* of the image, using techniques that facilitate further performance improvements through an efficient multi-scale fusion of patches. Our proposed Patch-NetVLAD's average performance across key benchmarks is superior by 17.5% over the original NetVLAD, and by 3.1% (*absolute* recall increase) over the SuperPoint and SuperGlue-enabled VPR pipeline. Patch-NetVLAD does not outperform DELG, however we note that DELG's feature extraction and matching are significantly slower and thus have less utility for mobile autonomous systems. Our experiments reveal an inherent benefit to fusing multiple patch sizes simultaneously, where the fused recall is greater than any single patch size recall, and provide a means by which to do so with minimal computational penalty compared to single scale techniques.

While this demonstration of Patch-NetVLAD occurred in a place recognition context, further applications and extensions are possible. One avenue for future work is the following: while we match Patch-NetVLAD features using mutual nearest neighbors with subsequent spatial verification using RANSAC, recent deep learned matchers [62, 88] could further improve the global re-localization performance of the algorithm. Although our method is by no means biologically inspired, it is worth noting that the brain processes visual information over multiple receptive fields [36]. As a result, another potentially promising direction for future research is to explore and draw inspiration from how the task of visual place recognition, rather than the more commonly studied object or face recognition tasks, is achieved in the brain. Finally, another line of work could consider the correlation between the learned VLAD clustering and semantic classes (*e.g.* car, pedestrian, building), in order to identify and remove patches that contain dynamic objects.

Acknowledgements: We would like to thank Gustavo Carneiro, Niko Suenderhauf and Mark Zolotas for their valuable comments in preparing this paper. This work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571. The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics.

References

[1] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *IEEE Int. Conf. Robot. Autom.*, pages 5958–5964, 2019.

[2] Relja Arandjelovic and Andrew Zisserman. All about VLAD. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1578–1585, 2013.

[3] 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.

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

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

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Eur. Conf. Comput. Vis.*, pages 404–417, 2006.

[7] Luis G Camara and Libor Přeučil. Spatio-semantic convnet-based visual place recognition. In *Eur. Conf. Mobile Robot.*, 2019.

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

[9] 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.

[10] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *IEEE Int. Conf. Robot. Autom.*, pages 3223–3230, 2017.

[11] Zetao Chen, Lingqiao Liu, Inkyu Sa, Zongyuan Ge, and Margarita Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robot. Autom. Lett.*, 3(4):4015–4022, 2018.

[12] Zetao Chen, Fabiola Maffra, Inkyu Sa, and Margarita Chli. Only look once, mining distinctive landmarks from ConvNet for visual place recognition. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 9–16, 2017.

[13] Ondřej Chum, Andrej Mikulik, Michal Perdoch, and Jiří Matas. Total recall II: Query expansion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 889–896, 2011.

[14] Titus Cieslewski, Konstantinos G. Derpanis, and Davide Scaramuzza. SIPs: succinct interest points from unsupervised inlierness probability learning. In *3DV*, pages 604–613, 2019.

[15] Franklin C. Crow. Summed-area tables for texture mapping. In *Conf. Comput. Graph. Interact. Techn.*, pages 207–212, 1984.

[16] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Eur. Conf. Comput. Vis. Worksh.*, 2004.

[17] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.*, 27(6):647–665, 2008.

[18] 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.

[19] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. Scalable place recognition under appearance change for autonomous driving. In *Int. Conf. Comput. Vis.*, pages 9319–9328, 2019.

[20] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable CNN for joint description and detection of local features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8092–8101, 2019.

[21] Mohammed E Fathy, Quoc-Huy Tran, M Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2D and 3D geometric correspondences. In *Eur. Conf. Comput. Vis.*, pages 803–819, 2018.

[22] Tobias Fischer and Michael Milford. Event-based visual place recognition with ensembles of temporal windows. *IEEE Robot. Autom. Lett.*, 5(4):6924–6931, 2020.

[23] Sourav Garg, V Babu, Thanuja Dharmasiri, Stephen Hausler, Niko Suenderhauf, Swagat Kumar, Tom Drummond, and Michael Milford. Look no deeper: Recognizing places from opposing viewpoints under varying scene appearance using single-view depth estimation. In *IEEE Int. Conf. Robot. Autom.*, pages 4916–4923, 2019.

[24] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? *arXiv preprint arXiv:2103.06443*, 2021.

[25] Sourav Garg, Ben Harwood, Gaurangi Anand, and Michael Milford. Delta descriptors: Change-based place representation for robust visual localization. *IEEE Robot. Autom. Lett.*, 5(4):5120–5127, 2020.

[26] Sourav Garg and Michael Milford. SeqNet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robot. Autom. Lett.*, 2021.

[27] Sourav Garg, Niko Suenderhauf, and Michael Milford. Semantic–geometric visual place recognition: A new perspective for reconciling opposing views. *Int. J. Robot. Res.*, 2019.

[28] Sourav Garg, Niko Sünderhauf, Feras Dayoub, Douglas Morrison, Akansel Cosgun, Gustavo Carneiro, Qi Wu, Tat-Jun Chin, Ian Reid, Stephen Gould, et al. Semantics for robotic mapping, perception and interaction: A survey. *Foundations and Trends® in Robotics*, 8(1–2):1–224, 2020.

[29] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Eur. Conf. Comput. Vis.*, pages 369–386, 2020.

[30] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Eur. Conf. Comput. Vis.*, pages 392–407, 2014.

[31] Albert Gordo, Filip Radenovic, and Tamara Berg. Attention-based query expansion learning. *Eur. Conf. Comput. Vis.*, pages 172–188, 2020.

[32] Jiadong Guo, Paulo VK Borges, Chanoh Park, and Abel Gawel. Local descriptor for robust place recognition using lidar intensity. *IEEE Robot. Autom. Lett.*, 4(2):1470–1477, 2019.

[33] 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.

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

[35] Ziyang Hong, Yvan Petillot, David Lane, Yishu Miao, and Sen Wang. Textplace: Visual place recognition and topological localization through reading scene texts. In *Int. Conf. Comput. Vis.*, pages 2861–2870, 2019.

[36] D H Hubel and T N Wiesel. Functional architecture of macaque monkey visual-cortex. *Proc. Royal Soc. B*, 198:1–59, 1977.

[37] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Adv. Neural Inform. Process. Syst.*, pages 487–493, 1999.

[38] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3304–3311, 2010.

[39] Ahmad Khaliq, Shoaib Ehsan, Zetao Chen, Michael Milford, and Klaus McDonald-Maier. A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes. *IEEE Trans. Robot.*, 36(2):561–569, 2019.

[40] Hyo Jin Kim, Enrique Dunn, and Jan Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3251–3260, 2017.

[41] Zakaria Laskar, Iaroslav Melekhov, Hamed Rezazadegan Tavakoli, Juha Ylioinas, and Juho Kannala. Geometric image correspondence verification by dense pixel matching. In *IEEE Winter Conf. Appl. Comput. Vision*, pages 2521–2530, 2020.

[42] Duc Canh Le and Chan Hyun Youn. City-scale visual place recognition with deep local features based on multi-scale ordered VLAD pooling. *arXiv preprint arXiv:2009.09255*, 2020.

[43] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In *Int. Conf. Comput. Vis.*, pages 2831–2840, 2019.

[44] David G Lowe. Object recognition from local scale-invariant features. In *Int. Conf. Comput. Vis.*, pages 1150–1157, 1999.

[45] Stephanie Lowry, Niko Sunderhauf, Paul Newman, John J. Leonard, David Cox, Peter Corke, and Michael J. Milford. Visual place recognition: A survey. *IEEE Trans. Robot.*, 32(1):1–19, 2016.

[46] 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.

[47] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. In *IEEE Winter Conf. Appl. Comput. Vision*, pages 1034–1042, 2019.

[48] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2017.

[49] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE Int. Conf. Robot. Autom.*, pages 1643–1649, 2012.

[50] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans. Robot.*, 31(5):1147–1163, 2015.

[51] Peer Neubert and Peter Protzel. Beyond holistic descriptors, keypoints, and fixed patches: Multiscale superpixel grids for place recognition in changing environments. *IEEE Robot. Autom. Lett.*, 1(1):484–491, 2016.

[52] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Int. Conf. Comput. Vis.*, pages 3456–3465, 2017.

[53] Amadeus Oertel, Titus Cieslewski, and Davide Scaramuzza. Augmenting visual place recognition with structural cues. *IEEE Robot. Autom. Lett.*, 5(4):5534–5541, 2020.

[54] Eng-Jon Ong, Syed Sameed Husain, Mikel Bober-Irizar, and Miroslaw Bober. Deep architectures and ensembles for semantic video classification. *IEEE Trans. Circuits Syst. Video Technol.*, 29(12):3568–3582, 2018.

[55] Edward Pepperell, Peter I Corke, and Michael J Milford. All-environment visual place recognition with smart. In *IEEE Int. Conf. Robot. Autom.*, pages 1612–1618, 2014.

[56] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8, 2007.

[57] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *IEEE Int. Conf. Robot. Autom.*, pages 1011–1018, 2018.

[58] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(7):1655–1668, 2018.

[59] 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.

[60] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Int. Conf. Comput. Vis.*, pages 2564–2571, 2011.

[61] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12716–12725, 2019.

[62] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature match-

ing with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4938–4947, 2020.

[63] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8601–8610, 2018.

[64] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6896–6906, 2018.

[65] René Schuster, Oliver Wasenmuller, Christian Unger, and Didier Stricker. SDC-Stacked dilated convolution: A unified descriptor network for dense matching tasks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2556–2565, 2019.

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015.

[67] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Int. Conf. Comput. Vis.*, pages 1470–1477, 2003.

[68] Jaime Spencer, Richard Bowden, and Simon Hadfield. Scale-adaptive neural dense features: Learning via hierarchical context aggregation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6200–6209, 2019.

[69] Jaime Spencer, Richard Bowden, and Simon Hadfield. Same features, different day: Weakly supervised feature learning for seasonal invariance. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6459–6468, 2020.

[70] Arulkumar Subramaniam, Prashanth Balasubramanian, and Anurag Mittal. NCC-net: normalized cross correlation based deep matcher with robustness to illumination variations. In *IEEE Winter Conf. Appl. Comput. Vision*, pages 1944–1953, 2018.

[71] 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.

[72] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robot.: Sci. Syst.*, 2015.

[73] 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.

[74] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7199–7209, 2018.

[75] Hajime Taira, Ignacio Rocco, Jiri Sedlar, Masatoshi Okutomi, Josef Sivic, Tomas Pajdla, Torsten Sattler, and Akihiko Torii. Is this the right place? Geometric-semantic pose verification for indoor visual localization. In *Int. Conf. Comput. Vis.*, pages 4373–4383, 2019.

[76] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5109–5118, 2019.

[77] 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.

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

[79] 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.

[80] 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.

[81] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4470–4479, 2018.

[82] 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.

[83] Zhe Xin, Xiaoguang Cui, Jixiang Zhang, Yiping Yang, and Yanqing Wang. Real-time visual place recognition based on analyzing distribution of multi-scale cnn landmarks. *J. Intell. Robot. Syst.*, 94(3-4):777–792, 2019.

[84] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *Eur. Conf. Comput. Vis.*, pages 467–483, 2016.

[85] Peng Yin, Rangaprasad Arun Srivatsan, Yin Chen, Xueqian Li, Hongda Zhang, Lingyun Xu, Lu Li, Zhenzhong Jia, Jianmin Ji, and Yuqing He. MRS-VPR: A multi-resolution sampling based global visual place recognition method. In *IEEE Int. Conf. Robot. Autom.*, pages 7137–7142, 2019.

[86] Jun Yu, Chaoyang Zhu, Jian Zhang, Qingming Huang, and Dacheng Tao. Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE Trans. Neural Netw. Learn. Syst.*, 31(2):661–674, 2019.

[87] Mubariz Zaffar, Shoaib Ehsan, Michael Milford, and Klaus McDonald-Maier. Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. *IEEE Robot. Autom. Lett.*, 5(2):1835–1842, 2020.

[88] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Int. Conf. Comput. Vis.*, pages 5845–5854, 2019.

[89] Yingying Zhu, Biao Li, Jiong Wang, and Zhou Zhao. Regional relation modeling for visual place recognition. In *ACM Conf. Res. Develop. Inf. Retrieval*, pages 821–830, 2020.

[90] Yingying Zhu, Jiong Wang, Lingxi Xie, and Liang Zheng. Attention-based pyramid aggregation network for visual place recognition. In *Int. Conf. Multimedia*, pages 99–107, 2018.

[91] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Eur. Conf. Comput. Vis.*, pages 391–405, 2014.