TCL: Tightly Coupled Learning Strategy for Weakly Supervised Hierarchical Place Recognition

Yanqing Shen, Ruotong Wang, Weiliang Zuo, *Member, IEEE*, and Nanning Zheng

***Abstract*—Visual place recognition (VPR) is a key issue for robotics and autonomous systems. For the trade-off between time and performance, most of methods use the coarse-to-fine hierarchical architecture, which consists of retrieving top-N candidates using global features, and re-ranking top-N with local features. However, since the two types of features are usually processed independently, re-ranking may harm global retrieval, termed *re-ranking confusion*. Moreover, re-ranking is limited by global retrieval. In this paper, we propose a tightly coupled learning (TCL) strategy to train triplet models. Different from original triplet learning (OTL) strategy, it combines global and local descriptors for joint optimization. In addition, a bidirectional search dynamic time warping (BS-DTW) algorithm is also proposed to mine locally spatial information tailored to VPR in re-ranking. The experimental results on public benchmarks show that the models using TCL outperform the models using OTL, and TCL can be used as a general strategy to improve performance for weakly supervised ranking tasks. Further, our lightweight unified model is better than several state-of-the-art methods and has over an order of magnitude of computational efficiency to meet the real-time requirements of robots.**

***Index Terms*—Recognition, localization, SLAM, representation learning, place recognition.**

I. Introduction

To QUICKLY and accurately recognize the location of a given image (i.e. for a robot), a task called visual place recognition (VPR), is essential for full simultaneous localization and mapping (SLAM) systems. However, VPR still faces key issues: 1) Appearance, illumination, and viewpoint of a place may vary greatly over time. 2) Different locations may look similar, called perceptual aliasing [1].

VPR is typically regarded as an image retrieval task [2], which searches for the most similar image to the query in a geo-tagged database. The common way to represent a single image is to use local or global descriptors. Local descriptors [3]–[6] are generated separately for key-points (or regions), and then are used for cross matching [7], [8]. Global descriptor-based methods [9], [10] transform an image into a compact vector. Local descriptors are viewpoint invariant, but encoding and matching require a lot of computational memory and time. Conversely, global descriptors are compact and appearance invariant but suffer from viewpoint changes.

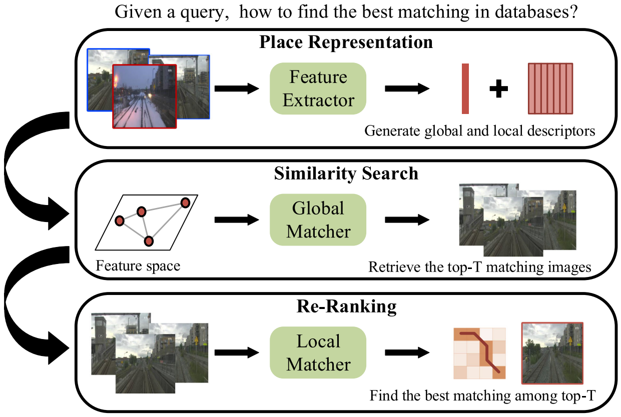
Given a query, how to find the best matching in databases?
Place Representation
Feature Extractor
+
Generate global and local descriptors
Similarity Search
Feature space
Global Matcher
Retrieve the top-T matching images
Re-Ranking
Local Matcher
Find the best matching among top-T

Fig. 1. **Hierarchical retrieval architecture of VPR.** (i) After finetuning a single backbone on VPR task, the unified feature extractor provides the global and local descriptors. (ii) Given a query, top-N candidates in database images are retrieved by similarity searching. (iii) The best matching result (R@1) is obtained by local feature-based re-ranking algorithms.

Considering the complementary advantages of local and global descriptors, a widely used architecture is to rank the database by global features, and then re-rank the top candidates using local features, as shown in Fig. 1. Many studies proved that re-ranking could improve the performance [2], [8], [11], [12]. For example, Patch-NetVLAD [12] derived patch-features from global features for local matching. However, the fact is that the improvement of re-ranking is limited by global retrieval results, and bad re-ranking may harm ranking results, called *re-ranking confusion*. This is mainly because the original triplet learning (OTL) strategy separates global and local descriptors, even in a holistic system.

The goal of this paper is to tightly couple global and local descriptors so that they can reinforce each other to improve the global perception capability and enrich the local details, and improve their consistency. To achieve this goal, we make the following contributions: 1) From the perspective of coupling them in the learning process, we propose a tightly coupled learning (TCL) strategy of triplet networks, and present the re-ranking loss and adapted ranking loss for weakly supervised scenario to achieve joint optimization. 2) We deploy TCL on different backbones to train lightweight unified models that combine superior performance with improved computational efficiency, which are also suitable for real-time scenarios. 3) To improve the accuracy

Manuscript received October 7, 2021; accepted January 4, 2022. Date of publication January 10, 2022; date of current version January 31, 2022. This letter was recommended for publication by Associate Editor M. Cornia and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants 62088102 and 61773312. *(Corresponding author: Nanning Zheng.)*

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qing1159364090@stu.xjtu.edu.cn; wrt072@stu.xjtu.edu.cn; weiliang.zuo@xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3141663, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3141663

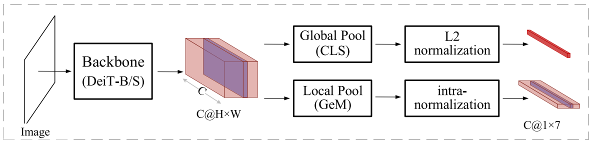

Fig. 2. Unified feature extractor that we trained in Fig. 3 consists of a deep network backbone, a global pooling layer and a local pooling layer.

of re-ranking, we propose bidirectional search dynamic time warping (BS-DTW) algorithm, which can mine more spatial information designed for VPR under reasonable path assumptions.

We evaluate the effectiveness of our approach and compare with several state-of-the-art methods [10]–[13] on well-known datasets (i.e. Mapillary Street Level Sequences (MSLS), Pittsburgh, Nordland and Gardens Point Walking (GPW) [14]–[17]). Our approach outperforms speed-focused Patch-NetVLAD and the global feature-based methods across all test sets. The performance of our approach is is better than best-performing Patch-NetVLAD and DELG-local in some datasets while worse in others, but our computational cost is lower (at least 200 times faster). We also carry out experiments to prove our model is suitable for real-time scenarios, including New College [18] and City Centre [19]. Furthermore, ablation studies are carried out to showcase the effectiveness of TCL and BS-DTW, particularly the potential of TCL to provide more powerful representations that are tailored to the ranking tasks.

II. Related Work

Global Descriptors: Global descriptors can provide a compact representation by aggregating local descriptors or directly processing the whole image. BoW [20], Fisher Vectors [21], and VLAD [22] have been used to aggregate hand-crafted local features through a visual codebook, and then have been incorporated into CNN-based architectures [10]. Other CNN-based works mainly focused on designing specific pooling layers on top of convolutional trunks [23], [24]. There have also been some methods using graph-based strategies [25], [26] to combine semantic and geometric information.

Local Descriptors: The hand-crafted local features such as SIFT [3], SURF [4] and ORB [5], and more recent CNN-based features including SuperPoint [27] and D2Net [28], have been extensively used in VPR and SLAM systems [2], [12], [19], [29], [30]. In addition to local aggregation to obtain global image descriptors, local descriptors are often used to re-rank initial candidates produced by a global approach through cross-matching between image pairs [7], [8], [31]. Distinct from existing methods, we propose TCL to combine global and local descriptors in learning process to achieve their additional joint optimization. Most related to our BS-DTW, a non-learning re-ranking algorithm, STA-VPR [8] proposed an adaptive DTW algorithm to align local features from the spatial domain. However, the assumptions about start and end points are not reasonable enough for VPR.

III. Proposed Method

As illustrated in Fig. 2, global and local features are extracted from the unified backbone. TCL strategy makes improvements in tuples mining strategy and loss function definition (Section III-B). Triplet tuples are selected for the next step of training through the global-local matcher, and an adapted ranking loss and a new re-ranking loss are introduced for joint optimization. In Section III-C, a bidirectional search DTW algorithm is designed for image spatial alignment in re-ranking to provide the local distance.

A. Place Representation

This section introduces the feature extractor and some symbols used in the subsequent description. As illustrated in Fig. 2, global and local features are extracted from a **unified** model by adding specific pooling layers on a lightweight backbone. The unified model further reduces calculation time and requires no additional or learned local feature extractor.

1) Backbone Network: Considering that cascaded self-attention modules in vision transformer (ViT) can capture long-distance feature dependencies, lightweight pretrained ViT models [32] are used as backbones. They are used to compare with SOTA methods and illustrate the effectiveness and portability of the TCL strategy in ablation studies.

2) Pooling Layer: Given an image I, the output consists of $M+1$ vectors corresponding to M input patches and a `[class]` token. Following the spirit of ViT [33], we view normalized `[class]` as a global descriptor, $G(I)$.

Reshaping the feature map as a $C \times H \times W$-dimensional (channel by height by width) tensor, we split it vertically into N separate tensors and apply Generalized Mean (GeM) pooling on each tensor. In this way, an image can be represented as a sequence of C-dimensional normalized local features, $\{L_1(I), \ldots, L_k(I), \ldots, L_N(I)\}$.

3) Distances: Global distance is denoted as d_G and used for ranking. *Local distance* is denoted as d_L and used for re-ranking, with details in Section III-C.

B. Tightly Coupled Learning Strategy

This section introduces the improvements of TCL strategy from two aspects: mining training tuples that are more effective for training speed and model performance, and defining additional loss function.

1) Training Tuples Mining Strategy: For Google Street View data, each query image q have a set of potential positives $\{p_i^q\}$ and a set of definite negatives $\{n_j^q\}$. The potential positives mean that at least one positive image is co-visible with q, but it's not clear which one it is. To address this problem in the training process, previous mining strategy [10] chose the best matching positive image for each $(q, \{p_i^q\}, \{n_j^q\})$ as

$$p_{i*}^q = \underset{p_i^q}{\operatorname{argmin}}\, d_G\left(q, p_i^q\right), \tag{1}$$

where d_G is the **global distance** in the current learning phase, $d_G(q, p) = \|G(q) - G(p)\|$.

In order to address the separation of global and local descriptors in learning, we present the *global-local matcher* in our strategy. As shown in Fig. 3, it chooses the best matching positive image $p^q(*)$ in two steps, different from p_{i*}^q above. First, global features are used to select top-T nearest images $\{p_{i*}^q\}_T$ from potential positives as (1). Second, local feature-based algorithm

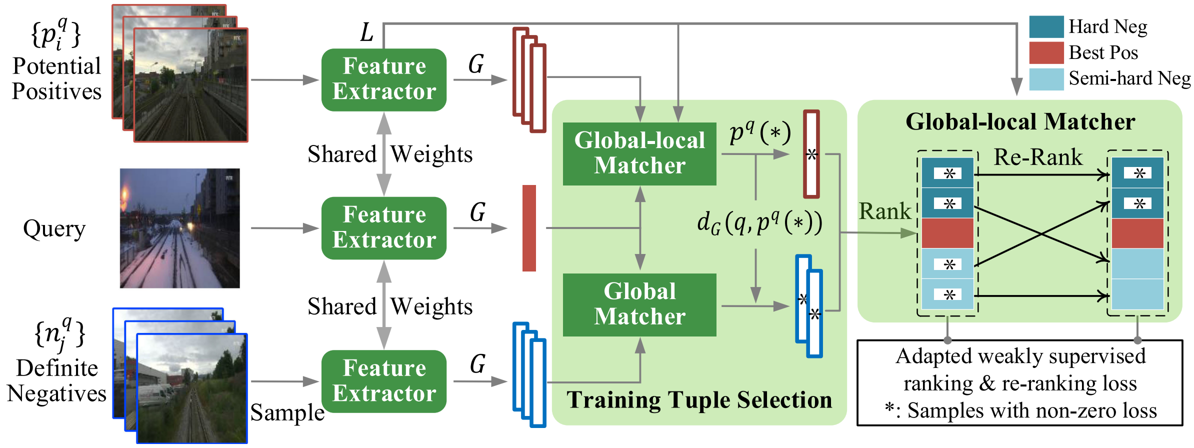

Fig. 3. Tightly Coupled Learning Strategy of Triplet Networks for Weakly Supervised VPR. In order to improve performance in the retrieval architecture as Fig. 1, the TCL strategy couples ranking and re-ranking: selection of training tuples and definition of losses. As shown in the dotted box, *global-local matcher* is a two-stage matching method including ranking and re-ranking instead of only ranking. In addition, we adapt the triplet ranking loss into our scenario and propose the triplet re-ranking loss. $*$ represents the samples that have non-zero losses.

is used to re-rank them and obtain top-1 as

$$p^q(*) = \underset{p_i^q}{\operatorname{argmin}}\, d_L\left(q, \{p_{i*}^q\}_T\right). \tag{2}$$

Subsequently, negatives of hard and semi-hard, $\mathbf{J}_q$, are chosen by *global matcher*, which satisfy

$$d_G\left(q, p^q(*)\right) + m > d_G\left(q, n_j^q\right), \quad j \in \mathbf{J}_q, \tag{3}$$

where m is a constant parameter giving the margin. Note that we first randomly select a subset of negative samples and then rank them only using global features. This is because after measuring the spent time and the gained performance, we find it's not worth re-ranking negative samples. In other words, the main difference from other strategies lies in the selection of positives.

2) Adapted Weakly Supervised Ranking Loss: In [10], triplet ranking loss for $(q, \{p_i^q\}, \{n_j^q\})$ is defined as

$$\sum_{j \in \mathbf{J}_q} h\left(\min_i d_G\left(q, p_i^q\right) + m - d_G\left(q, n_j^q\right)\right), \tag{4}$$

and we adapt (4) to our strategy as

$$L_g = \sum_{j \in J_q} h\left(d_G\left(q, p^q(*)\right) + m - d_G\left(q, n_j^q\right)\right), \tag{5}$$

where h is the hinge loss: $h(x) = \max(x, 0)$ and $p^q(*)$ is chosen by global-local matcher in Fig. 3.

3) Weakly Supervised Re-Ranking Loss: To strengthen the coupling between the ranking and re-ranking in learning, an additional loss is defined to constrain optimization space, namely *weakly supervised re-ranking loss*, as shown below:

$$L_l = \sum_{j \in \mathbf{J}_q} h\left(d_L\left(q, p^q(*)\right) - d_L\left(q, n_j^q\right)\right). \tag{6}$$

This loss is proposed from the point of view of recall in testing to improve the accuracy of re-ranking, so only hard negatives for local distance obtain non-zero losses. It is worth noting that the negative samples in (6) are still from (3), which ensures the consistency with the hierarchical retrieval structure in Fig. 1.

Finally, the total loss is

$$L = w_g L_g + w_l L_l, \tag{7}$$

where w_g and w_l are weights. L_l accelerates the learning process, and makes the parameter space constrained to simultaneously pull out the global and local distances between positives and negatives, thus improving their consistency. We believe that joint optimization leads to better features that are tailored to the ranking tasks.

C. Re-Ranking via BS-DTW

In order to mine spatial alignment information in cross-matching, DTW algorithm has been applied [8]. In this section, we present *bidirectional search DTW (BS-DTW)* algorithm, which is designed for local features alignment in image retrieval. It can yield alignment paths and **local distances** d_L between image pairs.

As illustrated in Fig. 2, each image is represented as a sequence of N local features ($N = 7$ in this paper) in the horizontal dimension, as in STA-VPR [8]. As mentioned in Section III-A, the distance matrix $\mathbf{D}_{N \times N}$ can be established by the distances between pairwise features in two sequences and an optimal path can be obtained:

$$P = \{p_1, p_2, \ldots, p_k, \ldots, p_K\}, \tag{8}$$

where $N < K < 2N - 1$ and the path P represents the alignment between two images, Q and R.

The path needs to satisfy three characteristics: boundary, continuity and monotonicity. The presentation of the real world in different images satisfies the homogeneity, so it is easy to understand that monotonicity and continuity are satisfied in image spatial alignment.

We focus on analyzing the boundary constraint in spatial alignment. Previous DTW algorithms require exact start and end points, e.g., $p_1 = (1, 1)$ and $p_K = (N, N)$ in [8]. However, through analyzing the matched images from the perspective of co-vision, we find that some segments in an image contain something that does not exist in another image.

To alleviate such a limitation, we design a loose boundary constraint for VPR: at least one element in $\{Q1, R1\}$ must have corresponding contents in another image (the same is

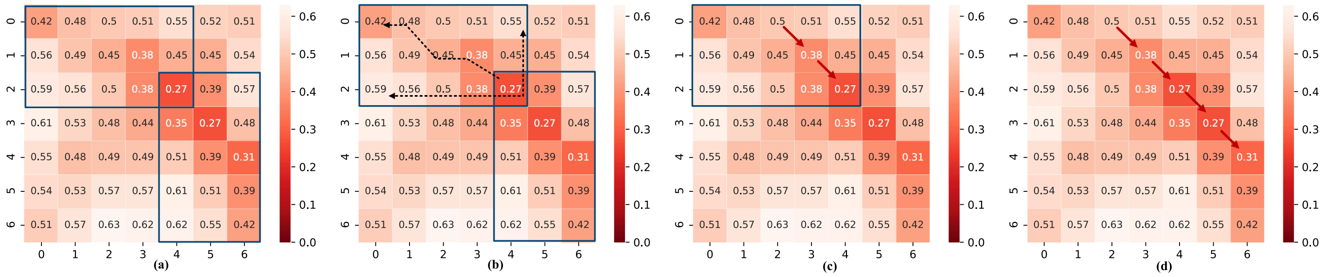

Fig. 4. Flow diagram of BS-DTW. The four figures correspond to the steps to find an optimal warping path. In (c), P_{ul} is marked with red arrows.

true for $\{Q7, R7\}$), called *"edge-touching principle"*. The three constraints can be expressed mathematically as follows:

- Boundary: The start point p_1 is on the leftmost column or uppermost row, that is, $p_1 = (1, J_1)$ or $(I_1, 1)$, where $I_1, J_1 \in [1, N]$. Similarly, the end point $p_K = (N, J_N)$ or (I_N, N), where $I_N \in [I_1, N], J_N \in [J_1, N]$.
- Continuity: If $p_k = (i, j)$ and $p_{k+1} = (i,' j')$, then $i' \leq i + 1$ and $j' \leq j + 1$.
- Monotonicity: If $p_k = (i, j)$ and $p_{k+1} = (i,' j')$, then $i \leq i'$ and $j \leq j'$.

Without the need for exact start and end points, BS-DTW more reasonably assumes that the path needs to contain the position with the highest local similarity, theoretically corresponding to the minimum value in $\mathbf{D}_{N\times N}$. The computation is summarized as follows:

1) Given $\mathbf{D}_{N\times N}$ between two images, find the minimum value and the corresponding index (i^*, j^*). To reduce the risk of extreme incorrect selection, we add additional judgement that more than two of 8-neighborhood are in the top-13 minimum of $\mathbf{D}_{N\times N}$. Then two exploration spaces are generated as shown in Fig. 4(a) and P can be divided into two sub-paths, P_{ul} for upper-left part and P_{lr} for lower-right part:

$$P_{ul} = \{p_1, \ldots, (i^*, j^*)\},$$
$$P_{lr} = \{(i^*, j^*), \ldots, p_K\}.$$

2) Taking P_{ul} as an example, potential start point satisfies $p_1' \in \mathbf{S}_1$, where $\mathbf{S}_1 = \{(1, j), (i, 1)\}$ and $i \in [1, i^*], j \in [1, j^*]$. For every p_1', the optimal path to (i^*, j^*) and the normalized distance are obtained by (9) and (10).
3) Select the point with the shortest normalized distance in $\mathbf{S}_1$ as the start point p_1 in P.
4) Repeat step 2)-3) for P_{lr} and obtain the end point p_K, and path P is shown in Fig. 4(d).
5) Compute the distance of P, s, and the normalized distance of P as the *local distance*, d_L.

Note that the above distance is calculated by the standard DTW methods [34], and the distance $s_{i,j}$ of the optimal path from (1,1) to (i, j) is shown in this formula, referring to the expression of (3) in [8]:

$$s_{i,j} = \begin{cases} d_{i,j} & i = 1, j = 1 \\ d_{i,j} + s_{i,j-1} & i = 1, j \in [2, N] \\ d_{i,j} + s_{i-1,j} & i \in [2, N], j = 1 \\ d_{i,j} + \min\left\{s_{i-1,j-1}, s_{i-1,j}, s_{i,j-1}\right\} & i, j \in [2, N] \end{cases}, \tag{9}$$

where $d_{i,j}$ is the Euclidean distance between local features, $L_i(Q)$ and $L_j(R)$. The normalized local distance between two

TABLE I
SUMMARY OF THE DATASETS. NO. REFERS TO THE NUMBER OF IMAGES. ++ INDICATES SEVERE CHANGES AND + INDICATES MEDIUM CHANGES

Dataset	No. (R/Q)	Description	Variation	
			App.	View.
MSLS-train [14]	900k/500k	urban,suburban,	++	++
MSLS-val	19k/11k	long-time	++	++
Pitts30k-train [15]	10k/7,416	urban,	++	++
Pitts30k-test	10k/6,816	pitch variation	++	++
Nordland [16]	3,450/3,450	train journey,	++	none
synth-Nord	3,450/3,450	seasonal	++	++
GPW [17]	200/200	campus	++	+

images is

$$d_L = \frac{s}{l_P}, \tag{10}$$

where l_P is the length of the optimal path. In our algorithm, the loose boundary constraints make the top-left and bottom-right vertices not mandatory to be included in the path, which is more in line with the practical situations.

IV. EXPERIMENTS

In this section we first describe the datasets and evaluation metrics, and details about our implementation. We then evaluate TCL strategy and BS-DTW via detailed ablation studies, and compare our trained lightweight unified model with state-of-the-art place recognition methods. Finally, we show the results of our model in real-time SLAM scenarios.

A. Datasets and Evaluation

We test models on several benchmark datasets and one synthetic dataset, using their recommended configuration and public partition of validation/test sets. A summary of datasets is shown in Table I. Here we introduce the usage of datasets to facilitate an informed assessment of the results. **Mapillary Street Level Sequences (MSLS)** [14] contains over 1.6 million images recorded in urban and suburban areas over 7 years. We evaluate models on the validation set. **Pittsburgh (Pitts30k)** [15] contains 30k database images and 24k queries, which are geographically divided into train/validation/test sets [10]. We evaluate models on the test set. **Nordland** [35] are divided into train and test sets [16], and down-sampled images (224 × 224) of test set in summer (reference) and winter (query) are used.[1] **Synth-Nord** is

[1] Same as previous works [12], [16], we remove all black tunnels and times when the train is stopped.

TABLE II
RESULTS OF DIFFERENT STRATEGIES ON PITTS30K, NORDLAND, AND GPW. ALL MODELS USE OUR BS-DTW TO RE-RANK TOP-100 UNLESS MARKED WITH †, WHICH INDICATES THE USE OF ADAPTIVE DTW OF STA-VPR [8]. ALL MODELS ARE FINETUNED ON PITTS30K FOR PITTS30K AND GPW, MARKED WITH $*$, AND MSLS FOR NORDLAND MARKED WITH $**$. TRAINING TIME REFERS TO THE TIME TAKEN FOR THE BEST R@5 ON PITTS30K VAL. SET

Backbone	#Dim	#Params	Strategies	Training Time	**Pitts30k test***			**Gardens Point Walking***			**Nordland test****		
					R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
DeiT-B	768	86M	NL	/	35.92	59.82	70.32	43.00	75.00	88.00	36.99	41.45	44.00
			OTL	13h53m	77.88	93.48	96.31	54.50	79.00	90.50	40.14	47.59	49.99
			OTL-R†	13h53m	79.85	93.51	96.32	57.00	80.50	91.00	41.75	48.52	50.11
			OTL-R	13h53m	81.55	93.80	96.45	58.50	81.50	94.00	42.61	48.90	50.28
			TCL	2h42m	82.51	94.83	97.13	62.00	86.50	94.00	43.32	52.68	59.20
			TCL-R	2h42m	**86.67**	**95.01**	**97.15**	**70.00**	**88.00**	**95.00**	**45.61**	**54.70**	**60.26**
DeiT-S	384	22M	NL	/	39.22	63.41	73.74	44.50	75.50	87.00	36.17	40.00	42.49
			OTL	8h38m	78.06	93.09	95.83	55.00	79.00	91.50	39.16	44.61	48.64
			OTL-R†	8h38m	79.37	93.21	96.02	57.00	81.50	91.00	41.39	45.94	48.84
			OTL-R	8h38m	81.57	93.82	96.23	59.00	84.50	92.50	42.38	46.26	49.54
			TCL	1h54m	82.40	94.72	97.00	63.00	93.00	95.00	42.75	50.52	53.73
			TCL-R	1h54m	**90.51**	**95.94**	**97.45**	**80.50**	**95.00**	**97.00**	**43.45**	**51.72**	**54.26**

synthesized by cropping the right 30% of databases and the left 10% of queries as [8]. **Gardens Point Walking (GPW)** [17] captures two traverses during the day and night. The day-left (query) and night-right (reference) traverses are used. **City Centre** [19] and **New College** [18] are used for testing real-time SLAM in Section IV-E.

1) Evaluation Metrics: We evaluate the recognition performance mainly based on Recall@N, whereby query is regarded to be correctly localized if at least one of the top N retrieved database images is within the ground truth tolerance. In addition, we evaluate the performance of SLAM by precision-recall scores. Follow the recommended evaluation of datasets, ground truth matches are obtained by given binary matrix from authors for New College [18] and City Centre [19], 25 m translation error for Pitts30k [10], [12], 25 m translation and 40° orientation error for MSLS [14], and 5 closet corresponding frames (i.e. ±2) for two-path-form datasets, i.e. Nordland [16] and GPW [17].

B. Implementation

As [12], we train models on Pitts30k for urban imagery (Pitts30k and GPW), and MSLS for other conditions.

Following the standard data augmentation methods, each image is resized to (256,256) with random horizontal flipping, and then randomly cropped to (224,224). We use two transformer-based models as backbones to build lightweight unified models. The pre-trained weights on ImageNet [36] are from the public implementation1 of DeiT [32] built on the timm library.[2] Our models are finetuned with triplet network using single parameter configuration with the margin $m = 0.1$ and Adam optimizer using a learning rate of 0.000005, a weight decay of 0.0001 and a batch size of 2. All experiments are run on an NVIDIA GeForce RTX 2080 Ti, except for training on MSLS with two cards. The finetuned model with the best recall@5 on the val set is used to test datasets, and the top-10 and top-100 images are respectively re-ranked in our experiments, termed -R10 and -R100.

For Section III-B, potential positives mean the database images within 10 meters, and definite negatives mean images further away than 25 meters. $T = 5$ in global-local matcher and 10 hardest negatives are chosen from 1000 randomly sampled negatives. Considering that global distances and local distances are close, $w_g = w_l = 0.5$.

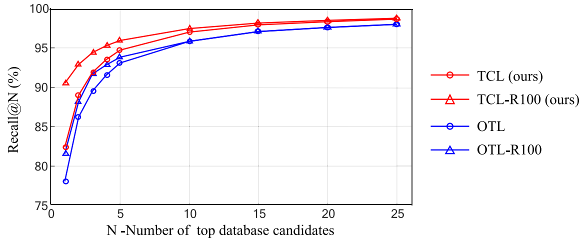

Fig. 5. Comparisons of DeiT-S on the Pitts30k-test dataset. We show the Recall@N performance of TCL in red and OTL in blue.

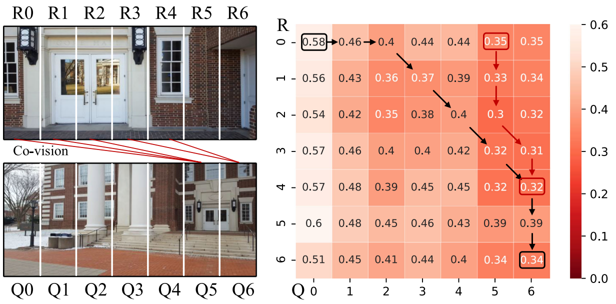

Fig. 6. Qualitative example of warping paths. The red box and arrows indicate ours and the black ones indicate the path by adaptive DTW of STA-VPR [8].

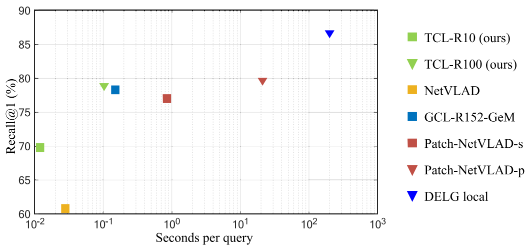

Fig. 7. Accumulated time required to process one query image is shown on the x-axis (log), with the R@1 shown on the y-axis, for MSLS-val.

[2] https://github.com/rwightman/pytorch-image-models

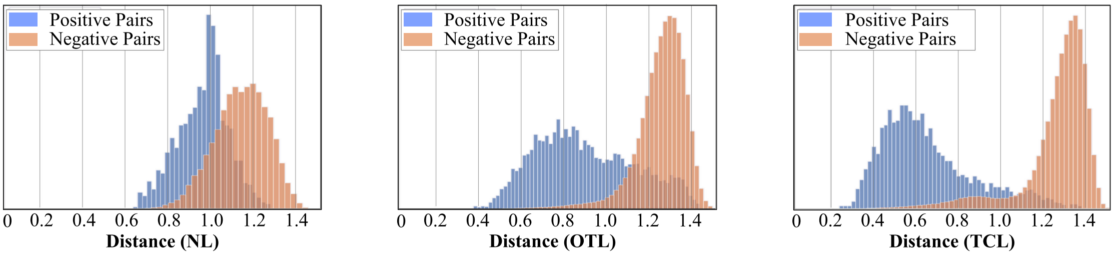

Fig. 8. Histograms of the distances between positive and negative pairs. The overlap in TCL is much smaller than NL and OTL.

C. Ablation Studies

1) Learning strategy: From three perspectives of whether to finetune model, how to finetune, and whether to add re-ranking as backend in testing, we design ablated variants of **weakly supervised** strategies on two base architectures:

- **NL**: No learning strategy.
- **OTL**: Original triplet learning strategy [10].
- **OTL-R**: OTL with Re-Ranking.
- **TCL**: Tightly coupled learning strategy (ours).
- **TCL-R**: TCL with Re-Ranking.

Table II shows the comparison results with above variations in terms of training and testing performance. TCL reduces the training time by a relative 75% over OTL. That is because our learning strategy focuses on selecting more accurate positives during weakly supervised training, and then gradients will be in a more accurate direction to accelerate the convergence with less epochs. Note that the training of TCL-R is same as TCL.

The following numeric analysis on performance are based on R@1 - generally applicable to R@5 and R@10. The percentage of Recall is plotted for different values of N in Fig. 5. DeiT-S trained on Pitts30k performs a little better than DeiT-B, while DeiT-B trained on MSLS is better. That's because the model representation ability should match the size of training dataset to achieve the model generalization ability. The performance of ranking and re-ranking increases significantly on Pitts30k and GPW, while on Nordland is not obvious. This shows that our strategy has more advantages in datasets with viewpoint changes.

Through analyzing the data, we can draw the following conclusions. First, TCL provides more powerful global image representation than OTL for ranking task, thus raising the upper limit of final performance. Second, the gap between TCL to TCL-R and OTL to OTL-R shows that TCL also improves the perception ability of local features and alleviates re-ranking confusion. Last, since the above two points are both valid on different backbones, TCL can be used as a general strategy for weakly supervised ranking tasks.

2) Re-Ranking Method: We conduct experiments to compare BS-DTW with adaptive DTW of STA-VPR [8]. Extended on the same backbone (DeiT-B/S), the two methods are used as a stand-alone spatial alignment backend, differing only in the way of re-ranking. The results of using adaptive DTW of STA-VPR are shown as OTL-R† in Table II. The results show that our algorithm results in better *re-ranking* performance (average relative increase of 80%), especially on Pitts30k and GPW which have viewpoint changes. This indicates that BS-DTW yields a more accurate alignment path when there is a large viewpoint change. It's worth noting again that the limited improvement of re-ranking is due to the confusion problem mentioned above and the limitation of global retrieval accuracy.

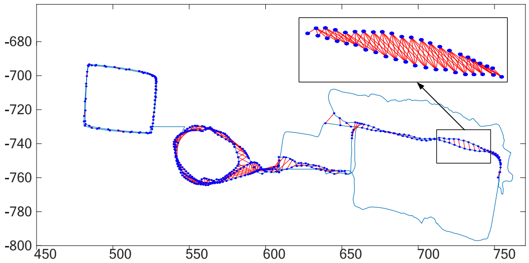

Fig. 9. Loop closure detection results (red lines) of New College.

In Fig. 6, a co-vision image pair with large viewpoint and scale changes is shown on the left. $Q1-Q4$ contains contents that do not exist in R, violating the assumption that p_1 and p_K are fixed in paths. The optimal path of two methods on the right highlights the rationality and flexibility of BS-DTW at the start and end points.

D. Comparisons With State-of-the-Art Methods

We compare with several retrieval-based localization solutions on image-to-image task: methods with re-ranking (DELG [11] and Patch-NetVLAD [12]) and methods without re-ranking (NetVLAD [10], GCL [13] and DELG-global).

For results not presented in the original paper of baselines, we test their publicly available models (e.g., GCL (ResNet152-GeM-PCA) and Patch-NetVLAD (speed-focused/best-performing, denoted as -s/-p)), and follow their **optimal testing configuration**, such as PatchNetVLAD use models of different training sets to test datasets (mapillary/pittsburgh_WPCA) and ours (DeiT-B-MSLS and DeiT-S-Pitts). Re-Ranking is performed on top-100 candidates.

Table III contains quantitative comparisons with the baseline methods from the perspective of performance and computational efficiency. The accumulated time includes both feature encoding time and feature matching time.[3]

Our method (ours-R100) outperforms all global feature methods, NetVLAD, GCL, DELG-global and Patch-NetVLAD-s on datasets on average by 23.2%, 14.5%, 6.9% and 3.5% (absolute improvement for R@1). Even ours-R10 yielded better results than these global methods, especially on Pitts30k and GPW.

Compared with systems which utilize local feature re-ranking, our method has competitive performance. Our method yields

[3] Matching time means the time taken to find the final R@1 in the database and is proportional to the size of database, which is 10,000 in our testing.

TABLE III
COMPARISON OF THE QUANTITATIVE RESULTS. ‡ INDICATES MODELS TRAINED ON THE PITTS250 K. THE AVERAGE ACCUMULATED TIME REQUIRED FOR EACH QUERY ON PITTS30K-TEST DATASET IS OBTAINED ON AN NVIDIA GEFORCE RTX 2080 TI

Method	Dim	Img Size	Time (s)	**Pitts30k test**			**synth-Nord**			**Gardens Point Walking**			**MSLS val**		
				R@1	R@5	R@10	R@1	R@5	R@10	R @1	R@5	R@10	R@1	R@5	R@10
NetVLAD‡ [10]	32768	original	0.028	81.9	91.2	93.7	12.4	26.4	33.9	38.0	74.5	83.5	60.8	74.3	79.5
GCL [13]	2048	(640,480)	0.15	75.6	91.5	92.6	18.7	28.1	35.8	55.5	85.5	93.0	78.3	85.0	86.9
DELG global [11]	128	(256,256)	-	80.4	90.0	93.4	36.1	49.3	53.2	69.5	92.0	94.0	72.4	81.9	85.7
Patch-NetVLAD-s [12]	512	(640,480)	0.85	87.1	94.0	95.6	38.2	48.2	51.2	71.5	93.5	96.5	77.0	84.2	86.9
Patch-NetVLAD-p [12]	4096	(640,480)	21.04	88.7	94.5	95.9	39.2	52.3	57.1	78.0	94.0	**97.0**	79.5	86.2	87.7
DELG local [11]	128	(256,256)	200	90.0	95.7	97.0	**45.9**	**53.7**	**59.2**	79.0	**95.0**	**97.0**	**86.5**	**90.3**	**91.9**
Ours-R10	384	(224,224)	**0.012**	90.1	95.4	97.0	38.3	48.7	51.8	67.5	84.0	91.0	69.8	76.3	82.2
Ours-R100	384	(224,224)	0.103	**90.5**	**95.9**	**97.5**	42.3	**53.7**	56.5	**80.5**	**95.0**	**97.0**	78.7	82.5	85.3

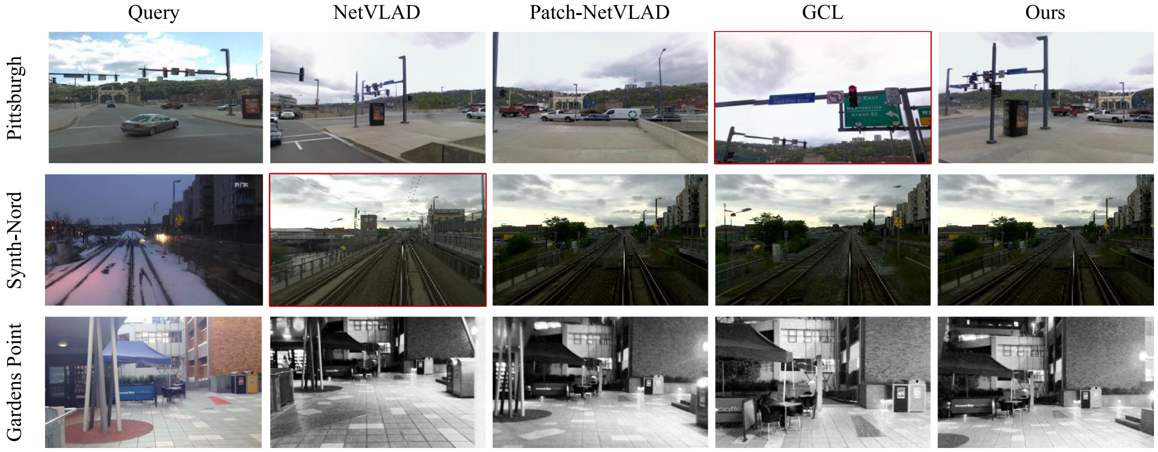

Fig. 10. Qualitative Results. For each query, the R@1 results by our model and other SOTA methods are presented. In these examples, ours and Patch-NetVLAD successfully retrieve the matching reference image, while both NetVLAD and GCL produce some incorrect matches marked with red borders.

considerably higher performance on the urban imagery with distinctive structures, i.e. Pitts30k and GPW. Patch-NetVLAD and DELG achieves better performance on MSLS because of their more complex network structures. But they also comes at a high computational cost on memory and time, and are not suitable for real-time applications. As shown in Table III, our computation is about 8, 200 and 1900 times faster than Patch-NetVLAD-s, Patch-NetVLAD-p and DELG-local. Fig. 7 gives an intuitive result that our algorithm achieves a good balance of accuracy and time.

Let's analyze why our strategy works. First, more accurate tuple mining strategy and fine optimization constraints in weakly supervised training will help the network learn more accurate information. Second, TCL strategy effectively separates positive and negative samples in the feature space. As shown in Fig. 8, compared with NL and OTL, TCL reduces the overlap in positive and negative histogram distribution.

In Fig. 10, we show some matching examples retrieved by our method and SOTA methods.

E. Applications on Robots

In order to prove that our model can be applied to robotics, whether it is loop closure detection or re-localization, we carry out experiments on two challenging datasets, i.e. New College [18] and City Centre [19]. Note that only the left camera image stream is used for evaluation.

We select the **top-10** retrieval results for each query for PR analysis. The maximum recalls at 100% precision and average execution time of our method (DeiT-S-R10) are shown in Table IV. It illustrates that our model can provide accurate and real-time results with appropriate threshold. In Fig. 9, we show the detected matches.

TABLE IV
PERFORMANCE ON THE NEW COLLEGE AND CITY CENTRE DATASETS. WE COMPARE SEVERAL SOTA ALGORITHMS WITH RELATIVELY HIGH REAL-TIME PERFORMANCE

Method	New College		City Centre	
	Recall(%)	Time(ms)	Recall(%)	Time(ms)
NetVLAD	75.1	17.2	66.7	16.9
GCL	82.0	33.6	53.0	33.2
PatchNetVLAD-s	**91.2**	272.2	84.0	289.0
ours	89.2	**3.6**	**85.6**	**3.5**

More results can be viewed in the attached video.

F. Discussion

We found that no single model performed well on all datasets, as was the case with GCL and Patch-NetVLAD, due to the lack of training on one dataset covering the variation of viewpoint, environment and appearance. Moreover, significant perceptual aliasing and extreme viewpoint variations in testing make image-to-image VPR difficult, even for humans.

We consider that generalization of training on different datasets is acceptable with some prior knowledge of the test dataset. For extremely challenging cases, we believe that adding semantic or sequential information might be helpful.

V. Conclusion

In this paper, we propose a tightly coupled learning strategy for weakly supervised hierarchical place recognition. In order to solve the separation between global and local features in hierarchical architecture, we tightly couple the two-style features in learning - (i) selecting the training tuples and (ii) defining the loss function - to obtain the features tailored to the ranking tasks beyond the place recognition task. Further, we also propose a bidirectional search DTW algorithm for spatial alignment in re-ranking. Detailed ablation studies validate the portability and effectiveness of the proposed learning strategy and spatial alignment algorithm. Further experimental results show that our lightweight unified model with the the proposed strategy outperforms several state-of-the-art methods on standard benchmarks and is much more suitable for the actual SLAM scenarios.

References

[1] S. Lowry *et al.*, "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[2] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5109–5118.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157.

[4] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2564–2571.

[6] C. McManus, B. Upcroft, and P. Newman, "Scene signatures: Localised and point-less features for localisation," in *Proc. Robot.: Sci. Syst.*, 2014, pp. 1–9.

[7] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[8] F. Lu, B. Chen, X.-D. Zhou, and D. Song, "STA-VPR: Spatio-temporal alignment for visual place recognition," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4297–4304, Mar. 2021.

[9] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[10] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[11] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 726–743.

[12] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14141–14152.

[13] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Generalized contrastive optimization of siamese networks for place recognition," *CoRR*, 2021, *arXiv:2103.06638*, [Online]. Available: http://arxiv.org/abs/1811.08838.

[14] F. Warburg *et al.*, "Mapillary street-level sequences: A dataset for lifelong place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2626–2635.

[15] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 883–890.

[16] D. Olid, J. M. Fácil, and J. Civera, "Single-view place recognition under seasonal changes," in *IEEE/RSJ Int. Conf. Intell. Robots Syst, Workshop on Planning, Perception and Navigation for Intell. Vehicles*, 2018, *arXiv:1808.06516*.

[17] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2015, pp. 4297–4304.

[18] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *Int. J. Robot. Res.*, vol. 28, no. 5, pp. 595–599, 2009.

[19] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.

[20] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. Int. Conf. Comput. Vis.*, 2003, vol. 3, pp. 1470–1470.

[21] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3384–3391.

[22] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

[23] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4938–4947.

[24] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019.

[25] S. Cascianelli *et al.*, "Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features," *Robot. Auton. Syst.*, vol. 92, pp. 53–65, 2017.

[26] M. Abdollahyan *et al.*, "Visual localization in the presence of appearance changes using the partial order kernel," in *Proc. Eur. Signal Process. Conf.*, 2018, pp. 697–701.

[27] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 224–236.

[28] M. Dusmanu *et al.*, "D2-Net: A trainable CNN for joint description and detection of local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8092–8101.

[29] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[30] S. Garg and M. Milford, "SeqNet: Learning descriptors for sequence-based hierarchical place recognition," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4305–4312, Jul. 2021.

[31] H. Taira *et al.*, "InLoc: Indoor visual localization with dense matching and view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7199–7209.

[32] H. Touvron *et al.*, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[33] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 3–7.

[34] J. Zhao and L. Itti, "shapeDTW: Shape dynamic time warping," *Pattern Recognit.*, vol. 74, pp. 171–184, 2018.

[35] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? Challenging seqslam on a 3000 km journey across all four seasons," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, Art. no. 2013.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.