Distill Structural Knowledge by Weighting Samples for Visual Place Recognition Supplementary Material

Yanqing Shen Sanping Zhou Jingwen Fu Ruotong Wang Shitao Chen Nanning Zheng*†
National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

This supplementary material is structured as follows. In Section 1, we explain the reasons for choosing the datasets in the main paper and present the detailed dataset configurations. In Section 2, we show some additional quantitative results and analysis on datasets, as well as results on the computation time. This section also contains several qualitative results. Section 3 provides the experimental details of methods, including detailed training settings of our method, ablation studies and the implementation of SOTA methods. Finally, Section 4 contains some additional ablation studies.

1. Detailed Dataset Configuration

To facilitate an informed assessment of the results, we further detail datasets and the usage, which were briefly mentioned in Section 4.2 of the main paper. We evaluate our method on 4 key benchmark datasets.

Mapillary Street Level Sequences (MSLS). MSLS [15] is introduced to promote lifelong place-recognition research, and contains over 1.6 million images recorded in urban and suburban areas over 7 years. Compared to other datasets, it covers the most comprehensive variation (dynamic objects, season, light, viewpoint, and weather), and we only evaluate the image-to-image task. GPS coordinates and compass angles are provided for each image, and the ground truth corresponding to a query is the reference images located within 25m and 40° from the query. The dataset is divided into a training set, a public validation set and a withheld test set (MSLS challenge)[1]. In training, we define a distance d_{qp} to represent the FOV overlap between query q and positive p:

$$d_{qp} = \left\| \mathbf{x}_q - \mathbf{x}_p \right\|_2 / 25 + \left(\theta_q - \theta_p \right) / 40 < 1 \tag{1}$$

*Corresponding author.
†Supported by National Science Foundation of China (No. 62088102).
[1] https://codalab.lisn.upsaclay.fr/competitions/865

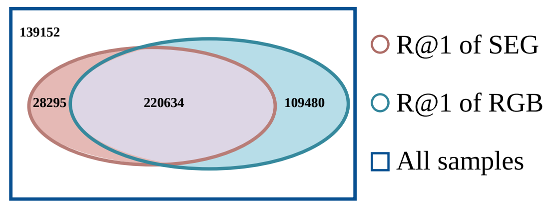


Figure 1. Visualization of sample distribution of MSLS training set. The numbers indicate the number of samples.

where $\mathbf{x}$ is the GPS coordinate and θ is the angle. Eq. (1) ensures an overlapping area between a query and a positive.

Nordland. The Nordland dataset [13] contains four timestamp-aligned image sequences recorded in four seasons, and hence it contains challenging appearance changes and different weather conditions while few viewpoint variations. We use the partitioned dataset [9] containing 3450 images per sequence, with summer as reference and winter as query. Same as [6, 9], we also remove all black tunnels and times when the train is stopped. Ground truth tolerance is set to 2 frames, that means that one query image corresponds to 5 reference images.

Pittsburgh. The Pittsburgh dataset [14] contains 250k images derived from Google Street View panoramas. The data is generated by 24 perspective images (two pitch and twelve yaw directions) at each place, which results in significant viewpoint variations, along with dynamic objects. As only GPS information is available, the ground truths for evaluation are defined as reference images within 25m from the query. In our experiments, we use the subset, Pitts30k, and the weakly supervised sample mining strategy [1] in training. It contains 30k database images and 24k queries, which are geographically divided into train/val./test sets.

2. Additional Results

Complementarity of RGB and SEG. In the main paper, we mentioned that there is a sample-level complemen-

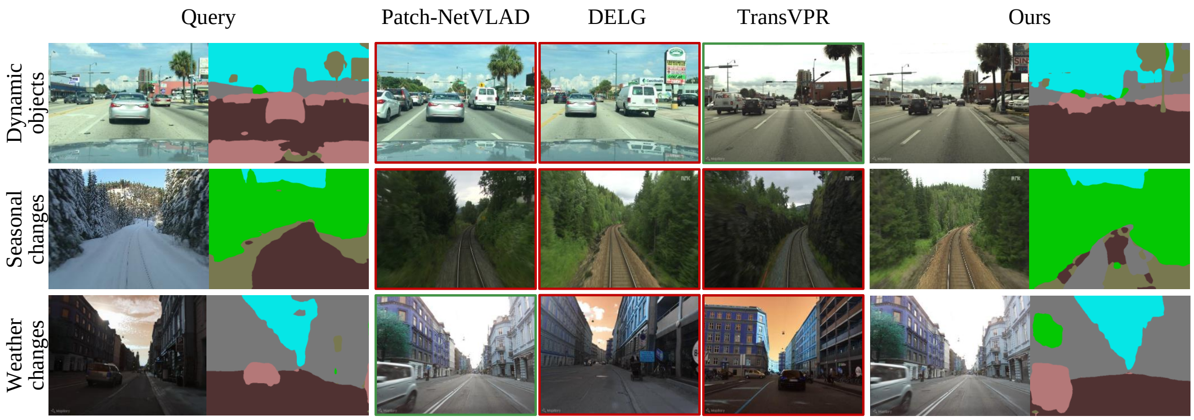


Figure 2. **Qualitative Results.** In these examples, our method successfully retrieves the matching reference images. For other methods, red borders indicate false matches and green borders indicate correct matches.

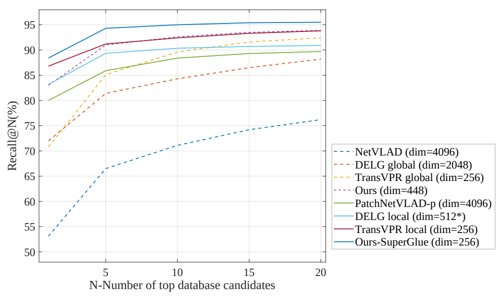
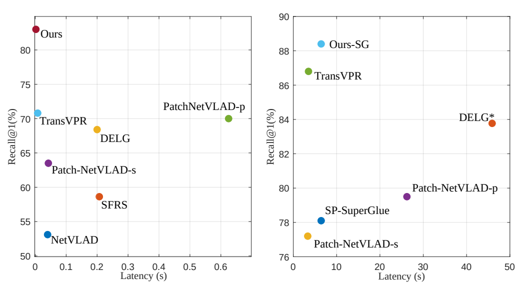


Figure 3. **Comparison with state-of-the-art on MSLS val. set.** We show the comparison of Recall@N performance with other methods. Results w/o re-ranking are depicted in dotted line, while results with re-ranking are depicted in solid line. * indicates unofficially reproduced results, and details are in Section 3.2.



(a) Global methods (b) Two-stage methods

Figure 4. The Recall@1 scores on MSLS validation set are shown on the y-axis, and the x-axis represents the accumulated time of feature extraction and feature matching.

tarity between RGB images and segmentation images for VPR. We calculate the Recall@1 of seg-branch in the query sets with correct/wrong predictions at top-1 ranking list of rgb-branch, respectively, and we swap branches and do the calculation again. The specific results on the MSLS training set are as shown in Figure 1.

Qualitative results. Figure 2 illustrates example

Table 1. Performance of selective distillation with different samples on Nordland dataset. None refers to the *rgb-branch* without distillation and All refers to non-selective distillation.

Group for distillation	Sample ratio	Nordland				
		R@1	R@5	R@10	R@15	R@20
None	0%	48.3	69.2	76.1	80.6	83.0
All	100%	<u>55.4</u>	71.9	76.9	79.4	80.7
$\mathcal{D}_1$	4.11%	51.2	69.1	75.9	79.8	82.5
$\mathcal{D}_2$	50.67%	52.5	72.1	79.8	<u>83.5</u>	<u>85.8</u>
$\mathcal{D}_3$	14.77%	42.4	62.5	70.8	75.1	78.4
$\mathcal{S}_1$	69.55%	54.8	<u>74.0</u>	<u>81.1</u>	83.2	84.8
$\mathcal{S}_2(\mathcal{D}_4)$	30.45%	33.0	54.4	64.2	69.5	72.8
$\mathcal{R}_1$	73.24%	55.2	73.7	79.6	82.6	84.0
$\mathcal{R}_2$	26.76%	51.0	67.4	73.0	75.4	77.8
Ours	69.55%	**56.1**	**75.5**	**82.9**	86.2	**88.3**

Table 2. Performance of *seg-branch* with different number of clustered classes. Note that † indicates another weighted encoding.

Clustered classes	Nordland				
	R@1	R@5	R@10	R@15	R@20
3	40.4	63.6	<u>73.6</u>	78.0	81.1
6	<u>41.1</u>	<u>63.8</u>	73.2	<u>78.6</u>	<u>81.6</u>
150	25.1	41.4	50.4	55.3	58.9
6 †	34.5	53.0	61.3	65.9	69.1
6 (Ours)	**56.1**	**75.5**	**82.9**	**86.2**	**88.3**

matches with challenging conditions, such as viewpoint changes, occlusions caused by dynamic objects, and seasons. In these examples, other methods show a tendency to retrieve images with similar appearance as the query. Especially in the case of MSLS, the color tone and the vehicle ahead of retrievals of Patch-NetVLAD and DELG are consistent with query. Ours can successfully retrieve images based on structural information, paying more attention to the spatial information of static objects in the background.

Additional recall plots. Table 1 in the main paper shows the Recall@N performance on the benchmark datasets. More intuitively, Figure 3 shows the detailed Recall@N performance for the MSLS validation dataset.

Efficiency-accuracy. Table 2 in the main paper shows the feature extraction, feature matching time and storage required to process each query. In Figure 4, we show the the accumulated time of feature extraction and matching as well as their performances on MSLS validation set. Ours achieves the best trade-off between accuracy and efficiency.

Ablations on other datasets. In Table 1 and Table 2, we provide more results on Nordland dataset for ablation experiments discussed in the main paper.

3. More Implementation Details

3.1. Training Details

Ours. To obtain segmentation images offline, we use ADE20K [19][2] and PSPNet [17] and the open-source codebase [3] with configuration file of "ade20k-resnet50dilated-ppm_deepsup". MobileNetV2 is initialed with the pre-trained weights on ImageNet [4]. MobileNet-L in seg-branch refers to the implementation of depth-stream in MobileSal[5].

In the first learning stage, rgb-branch and seg-branch are fine-tuned with the whole backbone using initial lr=0.001, where rgb-branch starts with a pre-trained model on ImageNet and seg-branch starts with random parameters. In the second learning stage, backbone also starts with a pre-trained model on ImageNet and is fine-tuned as a whole with initial lr=0.0001. We also attempted to continue the second stage on the basis of pre-trained network in the first training stage, but the difference is not significant.

Models are all optimized by AdamW optimizer [18] with 0.0001 weight decay and cosine learning rate decay schedule, and m in VPR loss is 0.1. The network which yields the best recall@5 on the val. set is used for testing.

For running SuperGlue network with SuperPoint based on our global retrieval, the details are shown in Section 3.2.

Concat-input. We concatenate RGB image and encoded segmentation label map in channel C as input, where the C of the first layer changes from 3 to 9 compared with *rgb-branch*. The model is fine-tuned with the whole backbone and initial lr=5e-5. The dim of global features is 448.

Concat-feat. Two separate networks, same as the two branches in the first stage, are used to extract features separately. Then the two features are concatenated, followed by a L_2 normalization step, as final global features to build loss function. The two models are fine-tuned with the whole backbone and initial lr=0.00005. The final loss is the direct sum of the two losses. The dim of global features is 928.

Multi-task. The implementation of decoder refers to U-Net [11] [6]. The model is fine-tuned with the whole backbone and initial lr=0.0001. The final loss is the sum of the vanilla VPR loss and the weighted cross-entropy loss, where the weight of cross-entropy is 0.1. The dim of global features is 448.

It is worth noting that the results of the Multi-task are not as good as expected, and the training process has high requirements for parameter tuning. Furthermore, compared with implicit supervision in the form of encoder-decoder, knowledge distillation is more direct and interpretable for enhancing structural information in features.

3.2. Implementation Details of Baselines

NetVLAD [1]. We use the pytorch implementation[7] and its released model trained on Pitts30k training set with VGG-16 backbone. Note that this method does not resize the image.

SFRS [5]. This work proposes a self-supervised method with image-to-region similarities to fully explore the potential of difficult positive images alongside their sub-regions. We use the official implementation[8] and the released model trained on Pitts30k training set.

SP-SuperGlue [4, 12]. SuperGlue trains a neural network that matches two sets of local features by jointly finding correspondences and rejecting non-matchable points. The implementation of SP-SuperGlue in our main paper includes: using NetVLAD for global retrieval, then extracting SuperPoint local features, and applying SuperGlue to identify matches and to re-rank candidates. We use the the official implementation [9] and choose the pre-trained outdoor weights on MegaDepth dataset [8].

Patch-NetVLAD [6]. This work derives patch-level features from NetVLAD residuals. We use the official implementation [10] for speed-focused and performance-focused configurations in our main paper. Following the original paper, the model trained on Pitts30k is used for urban imagery (Pittsburgh) , and the model trained on MSLS is used for all other conditions.

DELG [2]. This work unifies global and local features into a single deep model. We refer to the pytorch implementation of two models [11][12] and change the extraction of global features (dim=2048) to the way in the original paper: For global features, we use 3 scales; L_2 normalization is applied for each scale independently, then the three global features are average-pooled, followed by another L_2 normalization step. For local features, all the reproduced re-

[2]The largest open-source dataset for semantic segmentation and scene parsing, similar to the distribution of VPR datasets.

[3]https://github.com/CSAILVision/semantic-segmentation-pytorch

[4]https://download.pytorch.org/models/mobilenet_v2-b0353104.pth

[5]https://github.com/yuhuan-wu/MobileSal

[6]https://github.com/milesial/Pytorch-UNet

[7]https://github.com/Nanne/pytorch-NetVlad

[8]https://github.com/yxgeee/OpenIBL

[9]https://github.com/magicleap/SuperGluepre-trainedNetwork

[10]https://github.com/QVPR/Patch-NetVLAD

[11]https://github.com/feymanpriv/DELG

[12]The results in Table 2 in the main paper are the best of the two models.

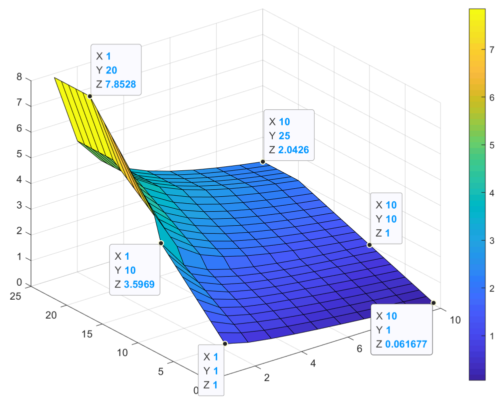


Figure 5. Visualization of the weight function. Some weights of demarcation points are marked. X and Y mean the recall rankings of samples of seg-branch and rgb-branch, same as the main paper.

sults are from the provided model, in which dim is 512 and is different from the original paper (dim=128).

TransVPR. This work proposes a holistic model based on vision Transformers, which can aggregate task-relevant features. We use the official implementation[13]. Same to Patch-NetVLAD, it finetuned the model on MSLS training set and Pitts30k training set.

4. Additional Ablation Studies and Analysis

Prior weights for encoding. Limited to the length of the article and the research content, in Section 4.5 of the main paper, we only manually set three different weighting cases and choose the best of the three, without finding the optimal one. Figure 6 indicates that the value of static buildings should be larger than that of dynamic objects, which is accorded with our intuition. This weighted attempt provides the possibility for follow-up research, and further advancements can be done, such as using grid search or learning methods to obtain the weights through iterative optimization.

Weighting function. For the weight function Eq. (4) given in the main paper, we make the following supplementary explanations. In main paper, we attempted to apply different constant weights to different groups according to the performance of separate group in Table 4.

In fact, we also tried other constant weights, as shown in Table 3. The results show that GP-D(8-4-1-0) and GP-D(4-2-1-0) has a more reasonable weight distribution than others, which proves our prediction of the importance of different groups: the greater the performance improvement when participating in distillation alone, the higher the weight. Moreover, the difference between GP-D(8-4-1-0) and GP-D(4-2-1-0) is small, and this experiment is mainly for providing a numerical reference for the design of our

[13] https://github.com/RuotongWANG/TransVPR-model-implementation

Table 3. Performance of weighted distillation with different weights. The weights correspond to $\mathcal{D}_1$-$\mathcal{D}_2$-$\mathcal{D}_3$-$\mathcal{D}_4$.

Weight	MSLS val			MSLS challenge		
	R@1	R@5	R@10	R@1	R@5	R@10
GP-D(1-4-8-0)	80.3	86.9	90.2	61.2	76.1	81.2
GP-D(1-2-4-0)	80.6	87.6	90.4	61.5	76.3	81.1
GP-D(4-2-1-0)	81.2	**89.3**	91.4	61.7	78.3	**82.5**
GP-D(8-4-1-0)	**82.2**	88.9	**91.5**	**62.3**	**78.9**	82.4

weight function. Therefore, we finally choose GP-D(8-4-1-0) as the reference.

Based on reference constant weights, the specific function design includes the following consideration:

- φ cannot be negative;
- The non-zero part of φ should be proportional to $y-x$ and inversely proportional to x;
- The value of φ cannot be too large;
- The partial derivatives should be different for 3 groups.

The prototype of the function can be denoted as $\frac{f(y-x)}{g(x)}$, where $f(\cdot)$ and $g(\cdot)$ are monotonically increasing functions. Considering x should play an more important role in weights than $y-x$, we choose liner for $f(\cdot)$ and natural logarithm for $g(\cdot)$ (see Figure 5).

Throughout the design process, we did not perform rigorous tuning of the parameters, but simply chose representative design to demonstrate our insight and motivation. In Table 4, we show more ablation experiments by replacing $\frac{y-x}{ln(x+1)}$ with $\frac{y-x}{x}$ in (4). Combined with the performance of distillation with fixed weights in Table 3, it can be seen that the function performs better than the discrete fixed weights and prototype function, showing the advantages of (4).

Table 4. Comparisons of functions.

$\frac{y-x}{x}$	MSLS val (81.2/89.5/92.2)	MSLS challenge (64.2/79.1/83.2)
$\frac{y-x}{\ln(x+1)}$	MSLS val (83.0/91.0/92.6)	MSLS challenge (64.5/80.4/83.9)

Sensitivity to hyper-parameters. In Section 4.4 of the main paper, we have performed ablation experiments on the most important hyper-parameters. We further evaluate the sensitivity of our model to changes in the other two hyper-parameters: N_t and N_m in our group partition strategy.

Here we perform unweighted selective distillation with the experimental setup of GP-S, that is, on samples belong to $\mathcal{S}_1$. The results are shown in Table 5 and Figure 6. It can be seen that within the appropriate range of 5-15, the performance is relatively close and we select 10 in the main paper.

After N_t is set as 10, N_m is mainly used to limit the weight range.

Sensitivity to Segmentation Models. In order to use accurate semantic information, some previous works [7, 10]

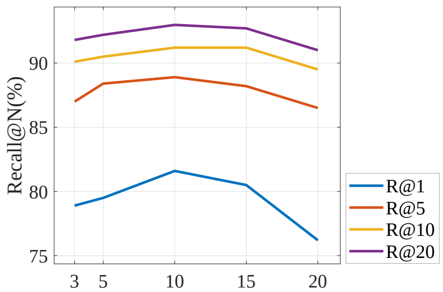


Figure 6. Ablation experiments on the recall performance of StructVPR with different N_t.

Table 5. The sample ratio corresponding to different N_t.

N_t	3	5	10	15	20
$\mathcal{S}_1$	60.75%	64.76%	69.55%	72.11%	73.82%
$\mathcal{S}_2$	39.25%	35.24%	30.45%	27.89%	26.18%

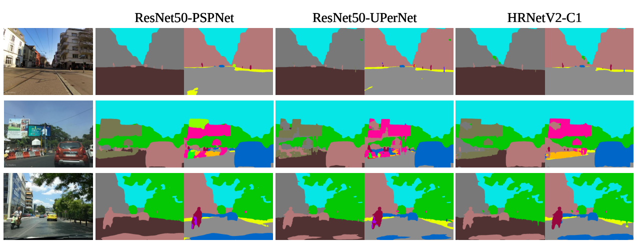


Figure 7. Examples of three semantic segmentation models with 6 classes and 150 classes.

use synthetic virtual datasets for training, and then generalize to real-world datasets through domain adaptation. In the main paper, we use an open-source semantic segmentation model to obtain segmentation images, which greatly reduces the implementation costs and training difficulty.

We also use another two commonly used semantic segmentation models (UPerNet [16] and HRNetV2 [3]) for training to assess the sensitivity of StructVPR to the segmentation models. We use the code-base[14] with configuration file of "ade20k-resnet50-upernet.yaml" and "ade20k-hrnetv2.yaml".

As shown in Figure 7, HRNetV2 is slightly better than PSPNet and PSPNet is slightly better than UperNet with fewer holes in the case of 150-class segmentation, while in the case of 6-class segmentation, the difference among the three models becomes smaller.

Table 6 shows that StructVPR is compatible with many models and is little affected by segmentation models. It is worth noting that due to time constraints, the results of "StructVPR-HR" and "StructVPR-UPer" are only the best performance among the current checkpoints, and we can provide the latest results later. This also means that

[14] https://github.com/CSAILVision/semantic-segmentation-pytorch

Table 6. Performance of *seg-branch* and StructVPR with different segmentation models. PSP is for ResNet50-PSPNet (main paper), UPer is for ResNet50-UPerNet, and HR is for HRNetV2-C1.

Method	MSLS val			MSLS test			Nordland		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
SEG-UPer	66.2	**80.7**	83.8	41.5	58.7	64.6	39.7	62.4	71.5
SEG-HR	65.2	80.4	**84.1**	**44.0**	**59.5**	64.9	**44.6**	**65.1**	72.5
SEG-PSP	**67.7**	80.0	83.1	43.4	58.9	**65.8**	44.4	64.8	**72.7**
StructVPR-UPer	82.4	90.3	**92.8**	63.4	78.8	82.7	57.2	75.6	82.4
StructVPR-HR	81.62	90.41	91.9	60.8	79.3	83.0	**59.3**	**77.7**	**84.7**
StructVPR-PSP	**83.0**	**91.0**	92.6	**64.5**	**80.4**	**83.9**	56.1	75.5	82.9

StructVPR can achieve excellent performance without relying on semantic annotations ground truth. This is expected since the structural information we extract does not rely on completely accurate pixel-level segmentation, but more on spatial relative positional relationships. Moreover, the clustering operation in SLME also makes StructVPR less sensitive to segmentation results.

Analysis. StructVPR achieves better performance and maintains a low inference cost without re-ranking. Compared to rgb-branch, StructVPR does have more costs in training due to the large amount of training set. Nevertheless, compared to model training, annotation and group partition are not expensive and mostly one-time efforts. At last, considering the robustness of StructVPR to segmentation label map after SLME, we can seek smaller models or reduce the resolution to reduce costs of computing SEG images.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307, 2016. 1, 3

[2] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *ECCV*, pages 726–743, 2020. 3

[3] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, pages 5386–5395, 2020. 5

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, pages 224–236, 2018. 3

[5] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *ECCV*, pages 369–386, 2020. 3

[6] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-NetVLAD: Multi-

scale fusion of locally-global descriptors for place recognition. In *CVPR*, pages 14141–14152, 2021. 1, 3

[7] Hanjiang Hu, Zhijian Qiao, Ming Cheng, Zhe Liu, and Hesheng Wang. Dasgil: Domain adaptation for semantic and geometric-aware image-based localization. *IEEE TIP*, 30:1342–1353, 2020. 4

[8] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 3

[9] Daniel Olid, José M Fácil, and Javier Civera. Single-view place recognition under seasonal changes. *arXiv preprint arXiv:1808.06516*, 2018. 1

[10] Valerio Paolicelli, Antonio Tavera, Carlo Masone, Gabriele Berton, and Barbara Caputo. Learning semantics for visual place recognition through multi-scale attention. In *Int. Conf. Image Anal. Process.*, pages 454–466, 2022. 4

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015. 3

[12] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 3

[13] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *IEEE Int. Conf. Robot. Autom. Worksh.*, page 2013, 2013. 1

[14] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *CVPR*, pages 883–890, 2013. 1

[15] 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 *CVPR*, pages 2626–2635, 2020. 1

[16] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 5

[17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 3

[18] Hui Zhong, Zaiyi Chen, Chuan Qin, Zai Huang, Vincent W Zheng, Tong Xu, and Enhong Chen. Adam revisited: a weighted past gradients perspective. *Frontiers of Computer Science*, 14(5):1–16, 2020. 3

[19] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3