Dark Reciprocal-Rank: Teacher-to-student Knowledge Transfer from Self-localization Model to Graph-convolutional Neural Network

Takeda Koji Tanaka Kanji

***Abstract*— In visual robot self-localization, graph-based scene representation and matching have recently attracted research interest as robust and discriminative methods for self-localization. Although effective, their computational and storage costs do not scale well to large-size environments. To alleviate this problem, we formulate self-localization as a graph classification problem and attempt to use the graph convolutional neural network (GCN) as a graph classification engine. A straightforward approach is to use visual feature descriptors that are employed by state-of-the-art self-localization systems, directly as graph node features. However, their superior performance in the original self-localization system may not necessarily be replicated in GCN-based self-localization. To address this issue, we introduce a novel teacher-to-student knowledge-transfer scheme based on rank matching, in which the reciprocal-rank vector output by an off-the-shelf state-of-the-art teacher self-localization model is used as the dark knowledge to transfer. Experiments indicate that the proposed graph-convolutional self-localization network (GCLN) can significantly outperform state-of-the-art self-localization systems, as well as the teacher classifier. The code and dataset are available at https://github.com/KojiTakeda00/Reciprocal_rank_KT_GCN.**

I. Introduction

In visual robot self-localization, graph-based scene representation and matching have attracted recent research interest as robust and discriminative methods for self-localization. For example, in [1], an image-based self-localization application was addressed by representing each view image frame as a graph node and by connecting neighboring image frames via graph edges. In [2], a subimage-based self-localization application was addressed by representing semantically segmented regions as graph nodes and connecting neighboring segments via graph edges. In these applications, a query scene graph is matched against each map graph according to the similarity of graph node descriptors (e.g., image descriptors [3], region descriptors [4]) and the graph structure. Although they are effective, their computational and storage costs increase in proportion to the environment size and do not scale well to large environments.

To alleviate this problem, we formulate self-localization as a graph classification task and use the graph convolutional neural network (GCN) as a graph classification engine. Our approach, wherein the GCN is used as a scene graph classifier, is analogous to the recent paradigm of using a convolutional neural network (CNN) as a scene image classifier [5]. We inherit desirable properties of the classification task formulation, such as the flexibility in defining place classes [6], compressed classifier model [7], and high classification speed [8]. A key difference from the image classifier tasks is that the input visual data must be translated to graph data before being input to the GCN. This problem is the main focus of the present study.

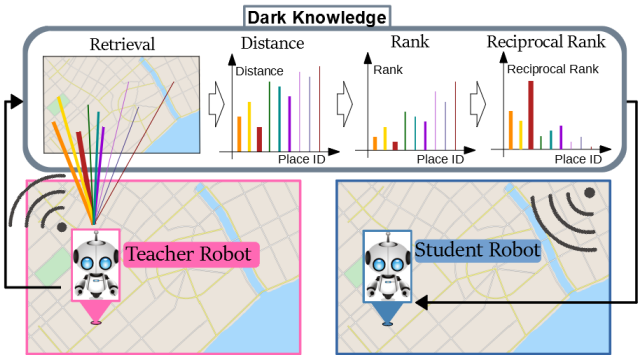

Fig. 1. We propose the use of the reciprocal-rank vector as the dark knowledge to be transferred from a self-localization model (i.e., teacher) to a graph convolutional self-localization network (i.e., student), for improving the self-localization performance.

A straightforward approach is to employ visual feature descriptors used by state-of-the-art self-localization systems, such as CNN-based [3], GAN-based [9], and autoencoder-based features [10], directly as graph node features. However, the main concern is that visual feature descriptors are not optimized for graph convolutions. In theory, their superior performance in the original self-localization system may not necessarily be replicated in GCN-based self-localization. Our experimental results indicated that the self-localization performance deteriorated when visual feature descriptors were directly used as a graph node feature descriptor in the GCN model.

To address this issue, we introduce a novel teacher-to-student knowledge-transfer scheme based on rank matching [11] (Fig. 1), inspired by our previous studies [12]–[14]. The basic idea is to introduce a state-of-the-art self-localization model (e.g., bag-of-words image retrieval [15], object proposal and matching [4], and deep image feature descriptor [3]) as a teacher classifier. This approach is inspired by the rank-matching loss used by recent transfer-learning schemes [11], where rank values are employed as the dark knowledge transferred from the teacher classifier to the student classifier. While rank transfer has been used in transfer learning with CNNs, its use in feature transfer with GCNs is non-trivial and was addressed for the first time in this study.

Our work has been supported in part by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) 17K00361, and (C) 20K12008.

K. Takeda is with Graduate School of Engineering, University of Fukui, Japan. K. Tanaka is with Faculty of Engineering, University of Fukui, Japan. {takedakoji00, tanakakanji}@gmail.com

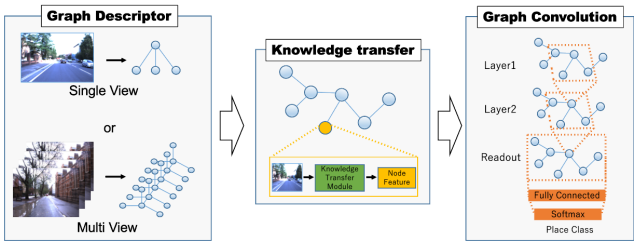

Fig. 2. System architecture.

Fig. 3. Single-view subimage-level scene graph (SVSL).

The main contributions of this work are summarized as follows: (1) We propose a novel graph node descriptor, which transfers the prediction of an off-the-shelf state-of-the-art teacher self-localization model to the student GCN classifier. (2) We show that a class-specific reciprocal-rank vector is a proper and effective representation of the dark knowledge to transfer. (3) We experimentally show that the proposed graph-convolutional self-localization network (GCLN) can significantly outperform state-of-the-art self-localization systems, as well as the teacher classifier. (4) We make the code and dataset publicly available[1].

II. Related Work

Visual feature descriptors for visual robot self-localization have been intensively studied. Recently, the use of intermediate features of deep neural networks (e.g., CNN [3], GAN [9], and autoencoders [10]) as discriminative and invariant visual image descriptors has become common. Additionally, the bag-of-words model has been used as a highly efficient image descriptor in state-of-the-art self-localization systems [15]. Moreover, it is straightforward to extend such an image descriptor to a subimage descriptor by segmenting images into subimages [4]. The effectiveness of image features has also been shown in several graph-based place recognition frameworks. In [16], the feature similarity between query and database images is represented in graphs, and performance is improved over a simple feature matching method by using diffusion operation. On the other hand, in [17], performance of SeqSLAM is improved by removing the assumption that the robot is moving at a constant speed, which is based on a directed acyclic graph -based method. In [18], a covisibility graph is used to encode the geometric information between visual words, which is based on the graph kernel for graph matching. However, the use of visual feature descriptors in the GCN framework is not straightforward and has yet to be investigated.

The GCN was recently developed and is one of the most popular types of deep graph neural networks. The GCN provides a flexible and descriptive model and has been successfully used in applications where the traditional CNN proved to be either inefficient or unsuitable (e.g., chemical reactivity [19], web-scale image retrieval [20]). Recently, the GCN has also been used in robotics and vision applications, such as the representation and parsing of spatially sparse three-dimensional (3D) point clouds [21]. Additionally, the use of scene graph representation has recently attracted research interest. In [22], a new view-based GCN was proposed, in which 3D shapes are recognized according to the graphical representation of multiple views. In [23], the worst-case graph matching method was proposed for addressing the challenges caused by appearing and disappearing landmarks, in which the spatial similarity of the landmarks with the worst appearance similarity is maximized. However, in the present study, we revisit a classical visual robot self-localization application with the aim of improving existing solutions.

The problem of visual robot self-localization has been studied with regard to various aspects. Several studies have focused on challenging self-localization scenarios, e.g., homogeneous orchard scenery [24], limited onboard resources [25], and highly dynamic environments [26]. Furthermore, advanced self-localization methodologies have been proposed, such as visual feature selection [27], hierarchical localization [28], quantifying the self-localization safety [29], key-frame selection [30], end-to-end self-localization [31], augmenting the scan context [32], sequence-based matching [33], geometric hashing [34], and persistence reasoning [35]. Additionally, the use of prior domain knowledge, e.g., OpenStreetMap [36] and Google Earth [37], has become common. In our previous studies, the ranking-based scene descriptors were explored in the context of image change detection [12], knowledge distillation [13], as well as rank fusion [14]. However, our approach focuses on the fundamental problem of scene modeling, which is orthogonal to and would facilitate these existing frameworks.

III. Scene Graph Model

In the proposed GCN self-localization framework, two types of scene graph models are used: the single-view subimage-level scene graph (SVSL) and the multi-view image-level scene graph (MVIL), which are described in Sections III-A and III-B, respectively (Fig. 2).

A. Single-view Subimage-level Scene Graph (SVSL)

The SVSL takes as input a single-view scene and converts it into a subimage-level scene graph. In the implementation, we use a total of four subimage nodes that correspond to the entire image region [0, 0]×[1080, 800] and three bounding boxes: CENTER [270, 200]×[810, 600], RIGHT [780, 0]×[1080, 800], and LEFT [0, 0]×[300, 800]. As shown in Fig. 3, in an SVSL graph, the edges extend in a star shape from the entire image node to the other three subimage nodes. While this scene graph model requires only a single-view image as an input, the invariance of the graph depends significantly on the invariance of the image segmentation. This limitation does not affect the MVIL model (III-B).

[1]https://github.com/KojiTakeda00/Reciprocal_rank_KT_GCN

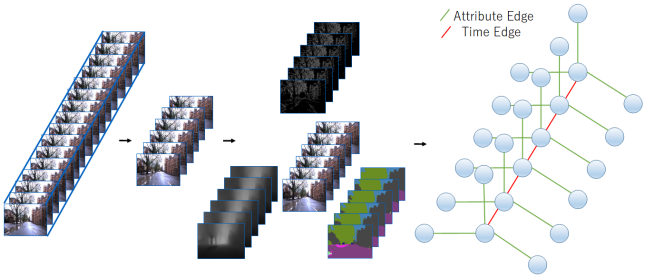

Fig. 4. Multi-view image-level scene graph (MVIL).

It can be effective to use a semantic segmentation (SS) technique to decompose an image into subimages instead of fixed bounding boxes. For example, in [2], the method of decomposing a scene into subimages via SS and connecting the segmented subimages via object-level edges experimentally worked well under an ideal condition of ground-truth segmentations. However, the good performance was not replicated in our current implementation of GCN self-localization. In a preliminary experiment, we attempted to use a state-of-the-art SS technique [38] instead of the fixed segmentation strategy, which was significantly affected by segmentation noise.

B. Multi-view Image-level Scene Graph (MVIL)

The MVIL takes as input a view-sequence scene and represents it as an image-level scene graph with multi-attribute frame image nodes and two types of graph edges: time and attribute edges (Fig. 4). As an example, in experiments, we consider at most $K = 4$ different image nodes, which are obtained by converting an original input RGB image with $(K-1)$ different image filters: Canny, depth regression, and SS, as shown in Fig. 4. A time edge connects the nodes of successive image frames with the same attribute. An attribute edge connects different attribute nodes of the same image frame. $(K-1)$ attribute images are connected to the RGB image node via attribute edges, yielding a star shape from the RGB image node to the $(K-1)$ attribute image nodes, as shown in Fig. 4. To facilitate the invariance of the time edge, the sampling of image frames is controlled so that the travel distance between successive image frames (connected by a time edge) becomes constant with regard to odometry measurements. While a multi-view scene graph requires as input a view-sequence, it is largely unaffected by segmentation noise. This is an appealing property of the MVIL model. The MVIL model is also related to sequence-based approaches such as [39] and [40]. A key advantage of the MVIL model against these existing approaches is that it is able to process multi-modal input data, as we will demonstrate in the experimental section.

The implementation details are as follows: We implemented $(K-1)$ types of attribute images: Canny, depth, and SS images, which were converted from an original RGB image by using the Canny edge detector [41], deep depth regressor [42], and Deeplab v3+ [38], respectively. The weight parameters of the deep depth regressor and Deeplab v3+ were pretrained on the KITTI dataset [43] and the Cityscapes dataset [44], respectively. The pixel values of an SS image were defined by the color map in [44]. The length of the view-sequence for the MVIL model was 10 frames, with 2-m intervals.

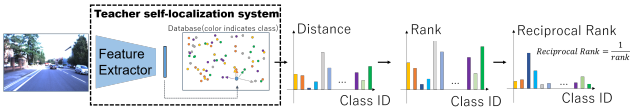

Fig. 5. Knowledge transfer on node feature descriptor.

IV. GCN Self-localization

In this section, we describe the proposed GCLN framework for lightweight and accurate self-localization based on knowledge transfer (Fig. 2). In the proposed framework, a lightweight representation of the scene graph is obtained using knowledge transfer from an external teacher self-localization model. Additionally, the accuracy of self-localization can be higher than that of the teacher self-localization model.

A. Knowledge Transfer

In knowledge transfer [45], the prediction results of a teacher model are often used as the dark knowledge to transfer. In particular, we propose the use of the (reciprocal-) rank vector as the representation of such dark knowledge. Many off-the-shelf self-localization systems (e.g., bag-of-words systems [15], classification systems [46], and map-matching systems [47]) can be modeled as ranking systems. Therefore, our (reciprocal-) rank-based scheme has a broader application area than existing knowledge-transfer schemes, e.g., those where intermediate signals of the teacher systems are used as the dark knowledge to transfer.

As an example, in our experimental system, a typical nearest neighbor (NN) image classifier with a NetVLAD image descriptor [3] was employed as the teacher model (Fig. 5). The teacher model used a visual image as an input, computed the L2 nearest-neighbor distance from the query image descriptor to the class-specific database image descriptors, and then converted the distance values into a class-specific rank value vector; a smaller rank value corresponded to a better degree of matching. Such pairings of the input image and output rank value vector are used as the dark knowledge to transfer in our scheme. We observed that a reciprocal-rank value vector is a good representation of an attribute-image-node descriptor, as discussed in Section V.

B. GCN Classifier

This subsection describes the procedure for graph convolution, focusing on the equation for forward propagation. A scene graph is represented as $G = (V, E)$, where V represents the set of nodes and E represents the set of edges. Let $v_i \in V$ denote a node and $e_{ij} = (v_i, v_j) \in E$ denote an edge pointing from v_j to v_i. The graph is defined as an undirected graph; i.e., whenever e_{ij} exists, e_{ji} exists. The neighborhood of a node v is defined as $N(v) = \{u \in V | (u, v) \in E\}$. Each

node v has a feature vector $h \in R^D$, where D is the number of dimensions of the feature vector. We performed an experimental ablation study (Section V), in which not only the class-specific reciprocal-rank vector but also the other intermediate representations, such as the original NetVLAD vector, class-specific NN-distance vector, and class-specific rank vector, were considered as the node feature descriptor.

The graph convolution operation takes node v_i in the graph and processes it in the following manner. First, it receives messages from nodes connected by the edge. Then, the collected messages are summed via the SUM function. The result is passed through a single-layer fully connected neural network followed by a nonlinear transformation for conversion into a new feature vector. In this study, we used the rectified linear unit (ReLU) operation as the nonlinear transformation, which is expressed as follows:

$$\mathbf{h}_i^{new} = \mathrm{ReLU}\left(\mathbf{W}\left(\sum_{u \in N(v_i) \cup v_i} \mathbf{h}_u\right)\right). \quad (1)$$

Here, W represents an $R^{D \times F}$ weight matrix, and D and F represent the numbers of dimensions of the node feature vector before and after the linear transformation, respectively. The foregoing process can be generalized to the processing of node features in the l-th GCN layer:

$$\mathbf{h}_i^{(l)} = \mathrm{ReLU}\left(\mathbf{W}^{(l-1)}\left(\sum_{u \in N(v_i) \cup v_i} \mathbf{h}_u^{(l-1)}\right)\right). \quad (2)$$

The process was applied to all the nodes in the graph in each iteration, yielding a new graph that had the same shape as the original graph but updated node features. The iterative process was repeated L times, where L represents the ID of the last GCN layer. After the graph node information obtained in this manner were averaged, the probability value vector of the prediction for the graph was obtained by applying the fully connected layer and the softmax function. This averaging operation is called "Readout." For the probability value vector of the output p, the operation is expressed as follows:

$$\mathbf{p} = \mathrm{Softmax}\left(FC\left(\frac{1}{|V|}\sum_{u \in V} \mathbf{h}_u^L\right)\right). \quad (3)$$

where h_u is a feature of node u after it passes through the last GCN layer. For implementation, we used the Deep Graph Library [48] on the Pytorch backend.

V. Experiments

We conducted self-localization experiments to confirm the effectiveness of the proposed method by using the publicly available Oxford Robotcar Dataset [49].

A. Settings

The Oxford Robotcar Dataset was obtained by a robotic vehicle-mounted camera when a robot car traveled along the same route in different seasons and with different weather and lighting conditions. Table I presents details regarding the dataset used in this study. The onboard camera used was a PointGreyBumblebeeXB3 (BBX3-13S2C-38) trinocular stereo camera (the center camera, 1280×960×3, 16 Hz). To avoid self-reflections and self-occlusions due to the vehicle, we used an image region of 1080×800 pixels (with 100 pixels from the left and right and 160 pixels from the bottom removed).

TABLE I
Statistics of the dataset.

date	weather	#images	detour	roadworks
2015-08-28-09-50-22	sun	31,855	×	×
2015-10-30-13-52-14	overcast	48,196	×	×
2015-11-10-10-32-52	overcast	29,350	×	○
2015-11-12-13-27-51	clouds	41,472	○	○
2015-11-13-10-28-08	overcast, sun	42,968	×	×

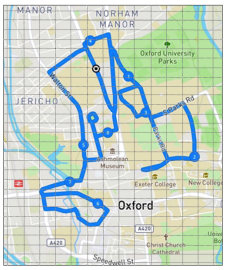

Fig. 6. Example of place partitioning.

To define the place class, the workspace of the Oxford Robotcar Dataset was partitioned into a two-dimensional regular grid of place classes according to the ground-truth global positioning system coordinates (Fig. 6). More formally, an area surrounded by 0.001 degree of latitude and longitude was defined as a one-place class. The number of classes for these test seasons ranged from 82 to 86. The classes that existed only for training in each season pair and the classes with less than five images in the area were not used in the experiments. "Unseen" classes, which existed only in the test season, were used as-is. The parameters of the NetVLAD descriptor were trained on the Pittsburgh (Pitts250k) dataset [3].

A subsequence of length 20 [m] obtained by shifting the first image frame by frame was sampled from the entire image sequence and used as training/test samples. For each dataset, all possible overlapping subsequences with travel distance 20 [m] were sampled from the entire image sequence and used as training/test samples. Those subsequences that straddle different place classes were removed. For single-view methods, the first image frame of each subsequence is used as a query input. For multi-view methods, each subsequence is represented by a length 10 view-sequence with 2-m intervals and used as a query input.

The number of GCN layers L was set as 2. The number of dimensions of the intermediate representation was 256. Thus, when the number of classes was C, the number of dimensions of the feature vectors (from the bottom layer to the top layer) was $C \rightarrow 256 \rightarrow 256 \rightarrow C$. The node aggregation method used the SUM operation and the ReLU activation function. The number of epochs was set as 5. The batch size was 32, and the learning rate was 0.001. The cross-entropy

TABLE II

AVERAGE TOP-1 ACCURACY.

Method	Average Top-1 Accuracy
Ours	92.4
NetVLAD	87.9
SeqSLAM	2.9

loss function and the Adam optimizer were used.

B. Comparison Methods

We used NetVLAD [3] and SeqSLAM [50] as comparison methods (Table II). The implementation of NetVLAD was based on [51]. NetVLAD was used in a single-view image-level self-localization scheme, in which the nearest-neighbor matching of place classes in terms of the Euclidean distance was performed. SeqSLAM was used as a multi-view image-level self-localization scheme. The implementation of SeqSLAM was based on the C++ version of OpenSeqSLAM. The parameters of SeqSLAM were optimized for the Nordland dataset, and no parameter manipulation was performed. The image IDs output by SeqSLAM were converted into the place class IDs to which the images belonged, and then the class IDs were simply used as outputs of the system.

C. Results

Fig. 7 presents example results of the proposed method, where L2 norm nearest neighbor matching is used with the output of the middle layer of GCN. As shown, the proposed method was robust against illumination changes and dynamic objects, owing to the Canny and SS image filters. However, the proposed method was not suitable for homogeneous scenery with no distinctive landmarks. This is because none of the image filters employed by the method (i.e., Canny, depth, SS) were robust against homogeneous scenes. To compensate, prior domain knowledge, e.g., road markings or the road topology, can be used as additional graph node descriptors. We plan to investigate this in a future study.

Computation time for GCN classification was 23.8 msec per graph (Intel (R) Xeon (R) GOLC 6130 CPU @ 2.10 GHz). For the GCN training, the speed was satisfactory (170 sec for a size 31,835 training set) even with CPU. This indicates that our approach can be implemented even on low-cost hardware with moderate performance, such as that used by small, inexpensive robots [52].

The results for the SVSL scene graph are presented in Fig. 8. For an ablation study, in addition to the proposed SVSL scene graph, a naive scene graph without edge connections was tested. As shown, the performance was better when edge connections were used.

The results for the proposed and comparison methods are presented in Figs. 9, 10 and 11. Comparing Figs. 8 and 9 reveals that the MVIL method using the reciprocal-rank vector had the best performance. In Figs. 9, 10 and 11, the proposed method employs an MVIL scene graph with raw RGB, Canny, and SS attribute image nodes (i.e., $K = 3$). First, the result for $K = 2$ is shown. In this study, we

TABLE III

PERFORMANCE RESULTS VERSUS THE GRAPH STRUCTURE.

Method	Average Top-1 accuracy
Attribute edges and Time edges	92.3
w/o all edges	89.0
w/o attribute edges	91.5
w/o time edges	88.7
w/o attribute node/edge	91.7

TABLE IV

PERFORMANCE RESULTS FOR DIFFERENT COMBINATIONS OF K AND IMAGE FILTERS.

number of nodes	combination	Average Top-1 accuracy
k=2	canny	92.1
	depth	91.8
	semantic	92.3
k=3	canny-depth	92.0
	canny-semantic	92.4
	depth-semantic	92.1
k=4	canny-depth-semantic	92.3

investigated which node feature is appropriate, which image conversion method is compatible with it, and which graph structure is optimal. Fig. 10 presents the dependencies of the choice of the attribute feature descriptor on the performance. Next, Fig. 11 presents the dependencies of the choice of the attribute image on the performance, where the feature is fixed to the reciprocal-rank vector, for the case of $K = 2$. As shown, the method using a semantic image had the highest accuracy. Clearly, the proposed method outperforms the comparison methods, i.e., NetVLAD and SeqSLAM.

Table III presents the results for different graph structures obtained using the reciprocal-rank method and $K = 2$ scene graph structure with the SS attribute image. For nodes without edge connection, there is no inter-node information transmission in the convolution process and thus, the feature translation is done independently for each node. As shown, the As shown, the prediction performance was improved with both the attribute and time edges, relative to the other graph structures.

Table IV presents the performance results for different combinations of image filters and the following numbers of filters: K =2, 3, 4. As shown, the performance was maximized when $K = 3$ filter set was used that consisted of the RGB image, Canny, and SS conditions.

D. Discussion

We now examine why the performance was improved by the proposed GCLN framework. For the SVSL scene graph, it is considered that the performance was largely affected by the degree of invariance of the image segmentation. The MVIL scene graph was successful for many difficult scenes where the NetVLAD method often failed. One reason for this is that knowledge from multiple image frames helped to understand the scene structure, which was captured by the GCN. Additionally, the attribute node inherited the desirable properties of the multiple attribute images, e.g., robustness against illumination changes (Canny), spatial invariance

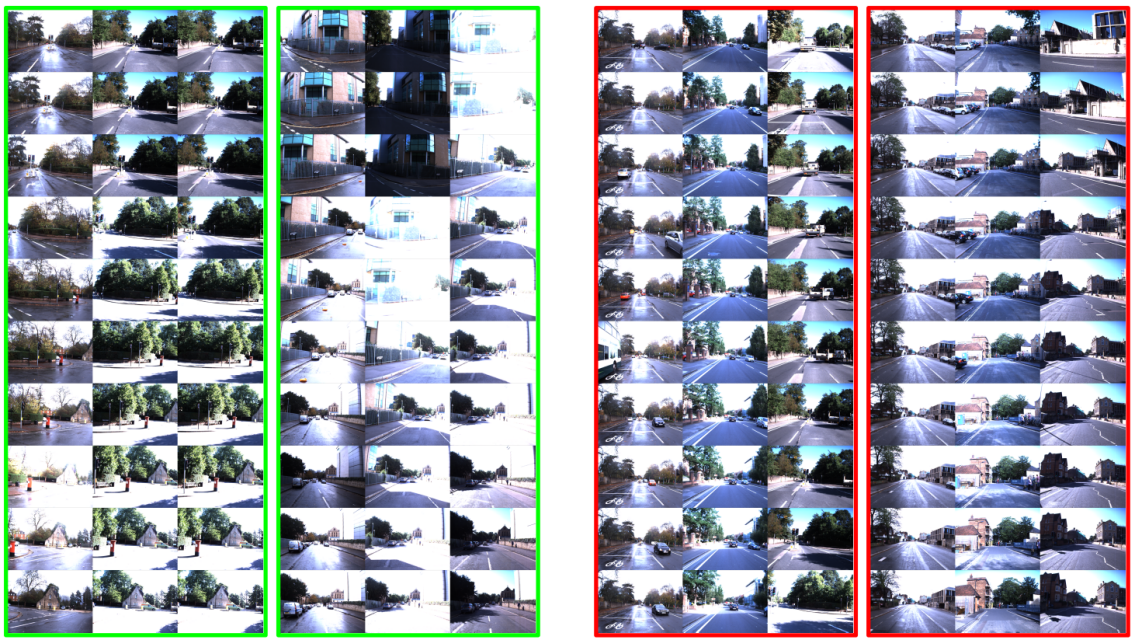

Fig. 7. Example results. From left to right, the panels show the query scene, the top-ranked DB scene, and the ground-truth DB scene. Green and red bounding boxes indicate "success" and "failure" examples, respectively.

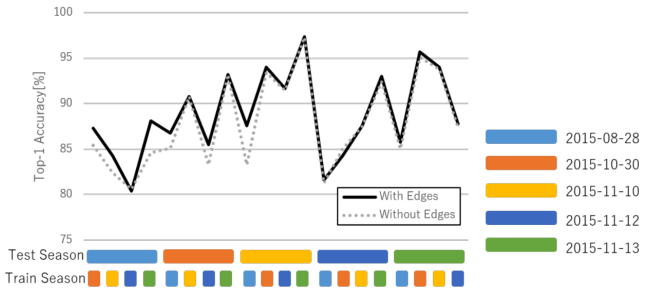

Fig. 8. Performance results for single-view scene graph with and without edge connections.

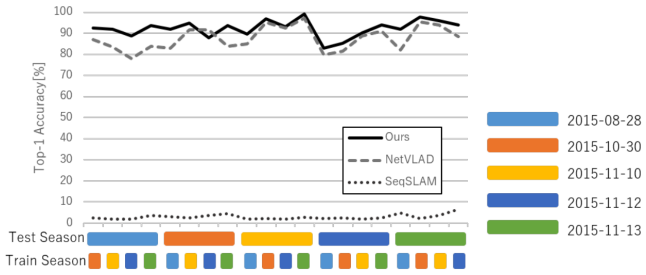

Fig. 9. Performance results for different training and test season pairs.

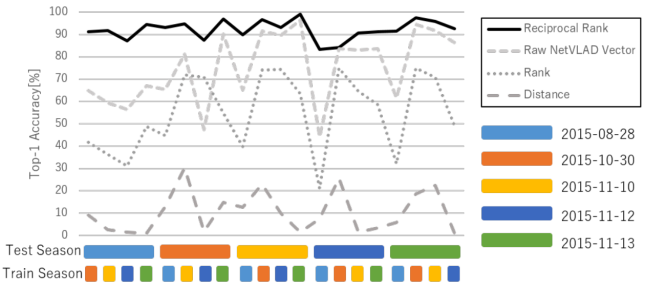

Fig. 10. Performance results versus the choice of attribute image descriptors (K=2).

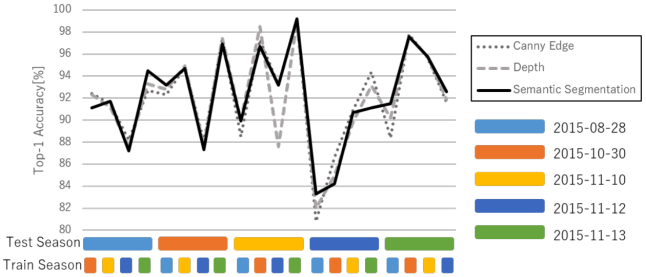

Fig. 11. Performance results for individual training/test season pairs (K=2).

in non-dynamic scenes (depth), and appearance invariance against seasonal shifts. The proposed method can combine the strengths of different attribute images in a computationally efficient manner. Moreover, the average performance is not sensitive to the graph structure, the number of attribute edges, or the combination of attribute images used, indicating the flexibility of the proposed GCLN framework.

VI. Conclusions

We presented a framework for enhancing a visual robot self-localization system using GCN. The proposed GCLN framework combines the accuracy of state-of-the-art self-localization systems and the flexibility and efficiency of the GCN. To this end, a novel teacher-to-student knowledge-transfer scheme based on rank matching was introduced. Experimental results indicated that the proposed framework outperformed the state-of-the-art methods and the teacher self-localization system. The reciprocal-rank vector was found to be effective dark knowledge to transfer, and in a future study, we plan to develop additional knowledge-transfer strategies for improving the GCN self-localization performance.

REFERENCES

[1] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows." in *AAAI*, 2014, pp. 2564–2570.

[2] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.

[3] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2015, pp. 4297–4304.

[5] N. Yang, K. Tanaka, Y. Fang, X. Fei, K. Inagami, and Y. Ishikawa, "Long-term vehicle localization using compressed visual experiences," in *21st Int. Conf. Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2203–2208.

[6] K. Tanaka, "Self-supervised map-segmentation by mining minimal-map-segments," in *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[7] T. Li, J. Li, Z. Liu, and C. Zhang, "Few sample knowledge distillation for efficient network compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 639–14 647.

[8] S. Hofstätter, S. Althammer, M. Schröder, M. Sertkan, and A. Hanbury, "Improving efficient neural ranking models with cross-architecture knowledge distillation," *arXiv preprint arXiv:2010.02666*, 2020.

[9] H. Hu, H. Wang, Z. Liu, C. Yang, W. Chen, and L. Xie, "Retrieval-based localization based on domain-invariant feature learning under changing environments," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2019, pp. 3684–3689.

[10] N. Merrill and G. Huang, "CALC2.0: Combining appearance, semantic and geometric information for robust and efficient visual loop closure," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019.

[11] Y. Chen, N. Wang, and Z. Zhang, "Darkrank: Accelerating deep metric learning via cross sample similarities transfer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[12] K. Tanaka, "Detection-by-localization: Maintenance-free change object detector," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4348–4355.

[13] T. Hiroki and K. Tanaka, "Long-term knowledge distillation of visual place classifiers," in *2019 22st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2019.

[14] K. Tanaka, "Unsupervised part-based scene modeling for visual robot localization," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6359–6365.

[15] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[16] X. Zhang, L. Wang, Y. Zhao, and Y. Su, "Graph-based place recognition in image sequences with cnn features," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 2, pp. 389–403, 2019.

[17] J. Maltar, I. Marković, and I. Petrović, "Visual place recognition using directed acyclic graph association measures and mutual information-based feature selection," *Robotics and Autonomous Systems*, vol. 132, p. 103598, 2020.

[18] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard, "Robust visual slam across seasons," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2529–2535.

[19] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay, and K. F. Jensen, "A graph-convolutional neural network model for the prediction of chemical reactivity," *Chemical science*, vol. 10, no. 2, pp. 370–377, 2019.

[20] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, 2018, pp. 974–983.

[21] L. Zhang and Z. Zhu, "Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks," in *IEEE Int. Conf. 3D Vision*, 2019, pp. 395–404.

[22] X. Wei, R. Yu, and J. Sun, "View-gcn: View-based graph convolutional network for 3d shape analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1850–1859.

[23] P. Gao and H. Zhang, "Long-term place recognition through worst-case graph matching to integrate landmark appearances and spatial relationships," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1070–1076.

[24] O. Shalev and A. Degani, "Canopy-based monte carlo localization in orchards using top-view imagery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2403–2410, 2020.

[25] Q. Yan, L. Jiang, and S. S. Kia, "Measurement scheduling for cooperative localization in resource-constrained conditions," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1991–1998, 2020.

[26] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic slam: The need for speed," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2123–2129.

[27] H. K. Mousavi and N. Motee, "Estimation with fast feature selection in robot visual navigation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3572–3579, 2020.

[28] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging deep visual descriptors for hierarchical efficient localization," in *Conference on Robot Learning*. PMLR, 2018, pp. 456–465.

[29] O. A. Hafez, G. D. Arana, M. Joerger, and M. Spenko, "Quantifying robot localization safety: A new integrity monitoring method for fixed-lag smoothing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3182–3189, 2020.

[30] K. M. Han and Y. J. Kim, "Robust rgb-d camera tracking using optimal key-frame selection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6275–6281.

[31] N. Akai, T. Hirayama, and H. Murase, "Hybrid localization using model-and learning-based methods: Fusion of monte carlo and e2e localizations via importance sampling," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.

[32] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2095–2101.

[33] S. Garg and M. Milford, "Fast, compact and highly scalable visual place recognition through sequence-based matching of overloaded representations," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3341–3348.

[34] I. Hofstetter, M. Sprunk, F. Ries, and M. Haueis, "Reliable data association for feature-based vehicle localization using geometric hashing methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1322–1328.

[35] Z. Hashemifar and K. Dantu, "Practical persistence reasoning in visual slam," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7307–7313.

[36] T. Ort, K. Murthy, R. Banerjee, S. K. Gottipati, D. Bhatt, I. Gilitschenski, L. Paull, and D. Rus, "Maplite: Autonomous intersection navigation without a detailed prior map," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 556–563, 2019.

[37] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," *IEEE Robotics and Automation Letters*, 2020.

[38] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[39] S. Garg, B. Harwood, G. Anand, and M. Milford, "Delta descriptors: Change-based place representation for robust visual localization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5120–5127, 2020.

[40] P. Neubert, S. Schubert, and P. Protzel, "A neurologically inspired sequence processing model for mobile robot place recognition," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3200–3207, 2019.

[41] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[42] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv preprint arXiv:1812.11941*, 2018.

[43] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[44] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[45] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[46] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term lidar localization using scan context image," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.

[47] J. Neira, J. D. Tardós, and J. A. Castellanos, "Linear time vehicle relocation in slam," in *ICRA*. Citeseer, 2003, pp. 427–433.

[48] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, Z. Huang, Q. Guo, H. Zhang, H. Lin, J. Zhao, J. Li, A. J. Smola, and Z. Zhang, "Deep graph library: Towards efficient and scalable deep learning on graphs," *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[49] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[50] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE Int. Conf. Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[51] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2466–2473.

[52] H. Aagela, M. Al-Nesf, and V. Holmes, "An asus_xtion_probased indoor mapping using a raspberry pi with turtlebot robot turtlebot robot," in *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, 2017, pp. 1–5.