Localizer-to-Mapper Knowledge Transfer: Real-time Diagnosis of Deep SLAM in Everyday Navigation

Yoshida Mitsuki Tanaka Kanji

***Abstract*— In this study, we address a novel "domain-shift localization (DSL)" problem, by which "user robots" of a deep SLAM system localize a domain-shifted region in the robot workspace during their daily navigation. Furthermore, we present a case study pertaining to a simple deep SLAM system comprising a visual place classifier and visual odometry as exteroceptive and proprioceptive modules, respectively. Such a DSL method enables "mapper robots" to focus on available resources (e.g., time, energy, and computation) in the domain-shifted region, rather than the entire workspace, thereby significantly reducing the cost of per-domain DNN maintenance. Unlike conventional scenarios of SLAM diagnosis, the deep SLAM system comprises deep neural networks (DNNs) with black-box characteristics, which render it difficult to directly diagnose the internal signals of SLAM modules. Hence, we present a novel diagnosis algorithm that does not rely on internal signals but uses only the input/output signals of DNNs as input to DSL. Experiments demonstrate that, compared with a vanilla deep SLAM system that does not reflect fault diagnosis, the proposed deep SLAM framework can achieve a path that is more similar to the actual measured GPS path. This study is a first step towards a real-time approach to diagnose aging of deep SLAM systems.**

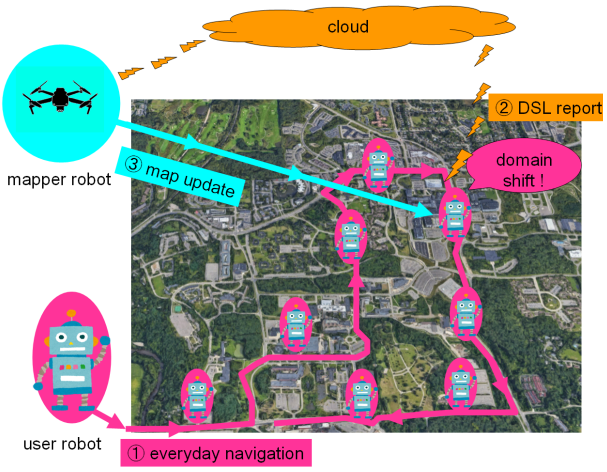


Fig. 1. DSL by a user robot during everyday navigation.

I. Introduction

Deep learning alternatives to SLAM modules have garnered research interest recently [1]–[3]. These alternatives aim to replace traditional exteroceptive/proprioceptive measurement modules of SLAM (e.g., landmark detector, and wheel odometry) using deep neural networks (DNNs). Such DNNs provide significantly improved performances using "user robots" in an in-domain scenario, where they are assumed to be trained and tested in the same domain (e.g., weather, time of the day, and season). However, their performances can become deteriorated in general cross-domain scenarios because of domain shifts [4], e.g., appearance/removal of landmark objects, falling leaves, snow cover, and building construction. This issue can be addressed using a "mapper robot" to explore the entire workspace such that a new training set can be obtained and the DNNs can be retrained. However, such an exploration requires significant per-domain cost in terms of travel distance, time, energy, and collision risks. Therefore, an alternative efficient method for maintaining DNNs is desired.

In this study, we address a novel "domain-shift localization (DSL)" framework, in which user robots localize a domain-shifted region in the robot workspace during their daily navigation (Fig. 1). Furthermore, we present a case study pertaining to a simple deep SLAM system comprising a visual place classifier (VPC) and visual odometry (VO) as exteroceptive and proprioceptive modules, respectively. Such a DSL method enables the mapper robot to focus on available resources (e.g., time, energy, computation) in the domain-shifted region, rather than the entire workspace, thereby significantly reducing the cost of per-domain DNN maintenance. DSL is complicated by the black-box nature of DNN modules, which renders it difficult to directly diagnose the internal signals of SLAM modules. Hence, we present a novel diagnosis algorithm that does not rely on the internal signals but uses only the input/output signals of DNNs as input to DSL. Although the diagnosis of SLAM has been addressed previously, to the best of our knowledge, this is the first study that is associated with the deep learning alternatives of SLAM modules. For our case study, VPC/VO modules were built on two types of DNNs, i.e., deep convolutional neural networks (CNNs) and deep recurrent neural networks (RNNs), which were then combined with a lightweight SLAM backend using an extended Kalman filter (EKF). Experiments using the publicly available NCLT dataset validate the effectiveness of the proposed algorithm.

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

M. Yoshida and K. Tanaka are with Robotics Course, University of Fukui, Japan. {ha171531, tnkknj}@g.u-fukui.ac.jp

We would like to express our sincere gratitude to Inagami Kazunori for discussion on deep learning architecture, and initial investigation on deep reinforcement learning on the dataset, which helped us to focus on our DSL project.

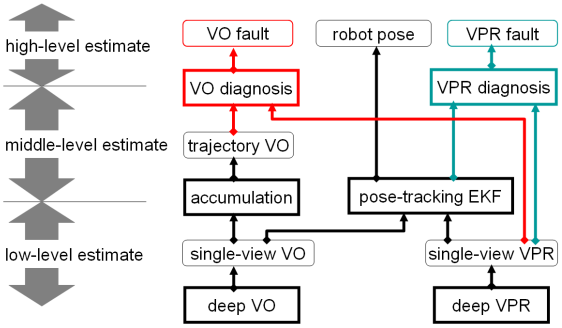


Fig. 2. System architecture.

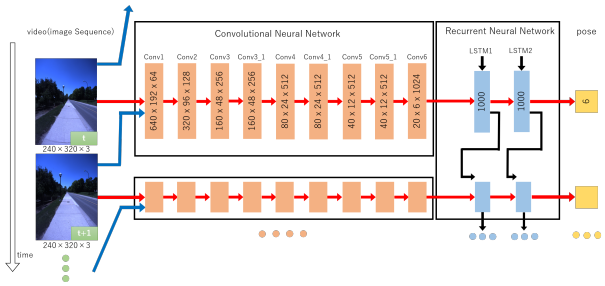


Fig. 3. VO architecture.

II. Approach

Our approach is briefly described as follows: (1) We considered a long-term SLAM scenario. The deep SLAM system comprised component DNNs that have been trained in a certain domain previously based on a set of images with viewpoint annotation. Therefore, when tested in the current domain, the DNNs can become deteriorated by domain shifts between previous and current domains. (2) We assumed that the SLAM system was in the pose-tracking mode during the diagnosis. In other words, the robot poses and environment model (e.g., "map DNNs") have been obtained, and the role of SLAM at that mode was to compensate for drift errors by sequential estimation. Therefore, a Gaussian filter such as an EKF is appropriate for incorporating incremental VO and visual place recognition (VPR) measurements into the SLAM state, as we will describe later. (3) We discovered that the degree of domain shift was highly dependent on the location of the robot's viewpoint. For example, domain shifts such as furniture movements are spatially distributed indoors, whereas domain shifts such as snow cover are distributed outdoors. Therefore, the available resources for retraining (e.g., travel distance, time, energy, and collision risks) should be concentrated on the domain-shifted regions rather than the entire region of the robot workspace.

Hence, we address a novel framework of DSL to identify the visual experience part of a user robot that is significantly affected by domain shifts during its daily navigation. Therefore, we developed a simple deep SLAM system prototype and its diagnosis framework (Fig. 2). Specifically, the deep SLAM system comprised two different types of DNNs and one EKF as the main building blocks. The first DNN was a deep CNN for managing VPR, where a specified view image was classified as a predefined place class. The next DNN was an RNN for managing VO, where ego-motions were estimated from a specified view image sequence. To achieve the goal of DSL, the KF addresses problems pertaining to state estimation and diagnosis, where VO and VPR measurements are incorporated into the SLAM state, and the inconsistency between measurements and estimates are diagnosed simultaneously.

A. VO

We applied the recently developed deep VO technique presented in [1], which is an end-to-end VO framework using FlowNet [5] and an RNN [6]. FlowNet uses a fully convolutional neural network to determine the number of pixels from the first and second images that has moved between adjacent frames in a video; subsequently, it estimates the optical flow. FlowNet was used because it allowed the use of the DNN architecture for learning geometric feature representations (i.e., optical flows) to address geometric problems (i.e., VO). Furthermore, it was used because of the necessity of inheriting the information between consecutive image frames, which can be achieved straightforwardly using an RNN.

Figure 3 shows the network architecture of DeepVO. DeepVO addresses two non-trivial issues: model building and stochastic reasoning. The goal is to obtain the conditional probabilities of the robot pose $Y_t = (y_1, ..., y_t)$ specified in the sequence of the monocular RGB image $X_t = (x_1, ..., x_t)$. The DNN obtains the VO's optimal parameter θ maximizing the following equation:

$$p(T_t|X_t) = p(y_1, ..., y_t|x_1, ..., x_t). \quad (1)$$

To train the hyperparameters of DNN, we minimize the Euclidean distance between the ground-truth pose (P_k, φ_k) and the estimated pose $(\hat{P}_k, \hat{\varphi}_k)$ at time k. The learning of hyperparameters can be obtained from the following equation:

$$\theta^* = \arg\max_{\theta} p(Y_t|X_t; \theta). \quad (2)$$

The DeepVO loss function comprises the mean squared error (MSE) of all the locations $\{P_k\}_{k=1}^t$ and orientations $\{\varphi_k\}_{k=1}^t$. The MSE is calculated as follows:

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{t} \left| \hat{P}_k - P_k \right|_2^2 + \alpha \left| \hat{\varphi}_k - \varphi_k \right|_2^2. \quad (3)$$

Here, $|\cdot|_2$ is the L2 norm; $\alpha = 100$ s a scale factor that balances the weights in the location and orientation direction; N is the number of samples.

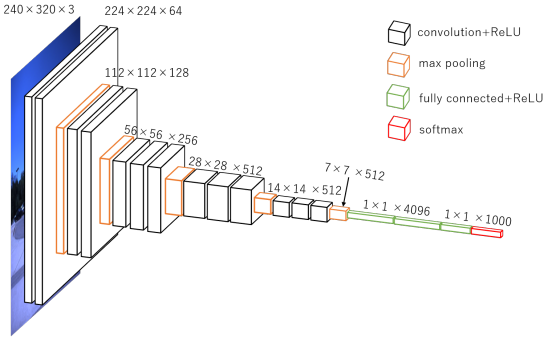


Fig. 4. VPR architecture.

B. VPR

We adopted a deep CNN, i.e., the Vgg16 presented in [7], for VPR (Fig. 4). The VPR task was formulated as a classification task over a set of predefined place classes, as in [2]. Following [8], the VPR task was based on the following two assumptions pertaining to training data and place classes. (1) First, we assumed that a view image sequence with viewpoint annotation, termed visual experience, was available in the training domain. Such a visual experience can often be reconstructed from the view sequence via [9] or SfM [10]. (2) Second, we assumed a predefined set of place classes, each of which corresponded to a local area of the robot workspace. Specifically, in our study, the place classes were defined by grid-based place partitioning, in which the robot workspace was partitioned into a regular grid with cells measuring 30 m $\times$ 30 m.

Based on the two assumptions above, a set of place-specific training data with a ground-truth place label was obtained. Subsequently, the labeled training set was fed to a training process for fine-tuning the publicly available pretrained Vgg16 to the VPR task.

The robot's viewpoint trajectory in the training domain was used as a prior to translate a place class to a fine-grained viewpoint coordinate. First, the top-one ranked place class with the highest probability value was predicted by the VPR classifier—it indicates that the robot is likely to be located at one of the viewpoints of the training images that belong to the top-one region. Subsequently, the location and orientation vectors of these viewpoints were averaged and used to predict the viewpoint coordinate.

C. Deep SLAM System

We employed an EKF for pose tracking and diagnosis. Using the EKF, the SLAM state $x(k+1)$ at the next time step $(k+1)$ was predicted based on the prediction $x(k)$ at current time step k, current measurement $y(k)$, and control input $u(k)$, using measurement and state-space models. Linearization was performed to calculate the Taylor series expansion of a nonlinear function using partial differentials and linear approximations (Jacobian) by truncating the series with the first order.

The estimation steps of EKF are as follows. First, the nonlinear state-space model (state equation, observation equation) is expressed by the following equation.

$$x(k+1) = f(x(k), u(k)) + v(k) \tag{4}$$

$$y(k) = h(x(k)) + w(k) \tag{5}$$

where $f(\cdot)$ is the nonlinear system function, $h(\cdot)$ the nonlinear observation function, $v(k)$ the system noise vector, and $w(k)$ the observation noise vector. The linearization of these two nonlinear functions yields the following equation:

$$A(k) = \frac{\partial f(x,u)}{\partial x}|_{x=\hat{x}(k),u=u(k)} \tag{6}$$

$$C(k) = \frac{\partial h(x)}{\partial x}|_{x=\hat{x}(k)} \tag{7}$$

The EKF estimation step begins with the input vector $u(k)$ at time k and the pre-state estimate $x(k)$ from the estimated state $x(k)$ to time $(k+1)$. The pre-estimated value refers to the predicted estimated value at time $k+1$ based on the data available up to time k.

$$\hat{x}^-(k+1) = f(x(k), u(k)) \tag{8}$$

Also, the prior error covariance matrix $P^-(k+1)$ is calculated using $A(k)$ and $Q(t)$, which is the error covariance matrix of the input vector.

$$P^-(k+1) = A(k)P(k)A^T(k) + Q(k) \tag{9}$$

Subsequently, the Kalman gain $G(k+1)$, which is the internal ratio of the pre-estimated value, and the observed value are calculated using $C(k+1)$ and the error covariance matrix of the observed vector $R(k+1)$.

$$G(k+1) = P^-(k+1)C^T(k+1) \left[C(k+1)P^-(k+1)C^T(k+1) + R(k+1)\right]^{-1} \tag{10}$$

Subsequently, the state vector is updated using the observation vector, the residual $y(k+1) - h(\hat{x}^-(k+1))$ between the observation vector and the state estimate above, and the Kalman gain. Next, the state estimated value $x^-(k+1)$ is calculated as follows:

$$\hat{x}(k+1) = \hat{x}^-(k+1) + G(k)\left[y(k) - h(\hat{x}^-(k+1))\right] \tag{11}$$

Here, $\hat{x}^-(k+1)$ is the updated estimate (filtering estimate) of x using $y(k+1)$, which represents the data available up to time $(k+1)$. Furthermore, $\hat{x}^-(k+1)$ is the estimated value to be obtained using the EKF. Additionally, we obtained the posterior error covariance matrix $P(k+1)$. The EKF repeats the prediction step above and updates the procedures to obtain the estimated value until the set time T.

In this study, the EKF uses the VO and VPR outputs as the input u and observation y vectors, respectively. Furthermore, the covariance matrix $\Sigma(k) = P(k+1)$ is used for the Mahalanobis distance-based diagnosis, as will be described later.

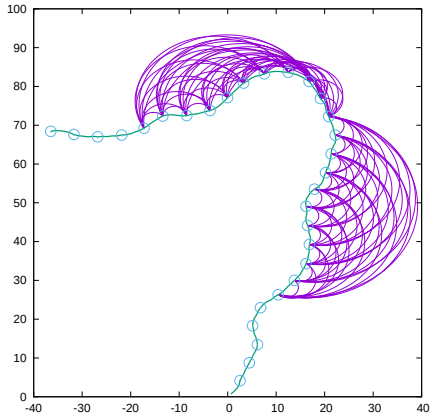


Fig. 5. Trajectory and waypoints for VO diagnosis.

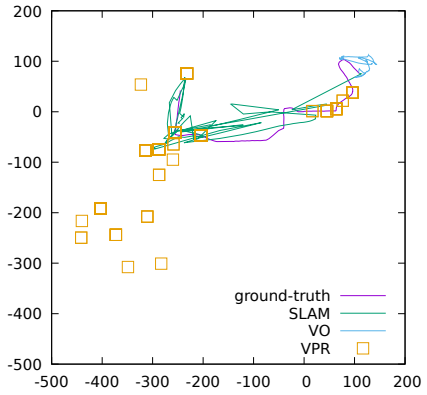
D. Fault Diagnosis

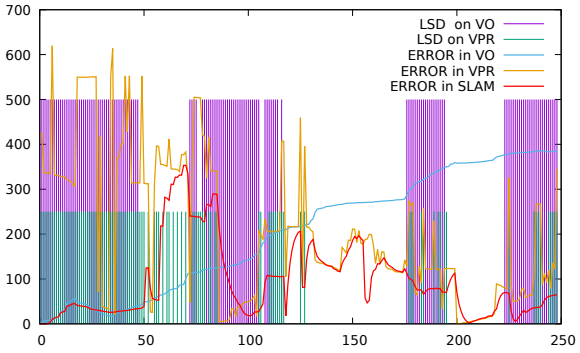
The VO and VPR modules exhibit vastly different noise characteristics. VO noise is cumulative, which is not detrimental to short distance travels but can be catastrophic to long-distance travels. Meanwhile, VPR noise is significant and non-cumulative. These characteristics should be considered when diagnosing individual modules. VPR diagnosis was performed at each time step k by assessing the inconsistency between the latest VPR measurement and the latest state vector of the EKF. Let the output signal of VO at time k be $u(k)$, the state vector of EKF be $\mu(k)$, and the variance-covariance matrix be $\Sigma(k)$. Therefore, the Mahalanobis distance $d(k)$ can be computed as follows:

$$d(k) = \left[(u_k - \mu_k)^T \Sigma_k^{-1} (u_k - \mu_k) \right]^{\frac{1}{2}} \tag{12}$$

If $d(k)$ is larger than the threshold value $K_d = 5$ (m), then the VO at time t is diagnosed as containing the fault.

The VO diagnosis (Fig. 5) is triggered every time the robot travels a certain distance $K_t = 50$ (m). Hence, we conducted the diagnosis at the subtrajectory level with length K_t by considering $(Kw+1)$ waypoints $(K_w = 10)$, where the first and (K_w+1)-th waypoints are defined as the start and goal locations of the trajectory, respectively. Since each subtrajectory corresponds to a waypoint pair, $(K_w+1)Kw/2$ subtrajectories exist. The likelihood of each subtrajectory containing the fault is evaluated by the inconsistency between VO and VPR in the predicted relative ego motion. Specifically, the prediction obtained was the Euclidean distance between the start and goal locations, which were provided by VO or VPR. Subsequently, the inconsistency obtained was the difference between VO and VPR in the prediction. Typically, multiple subtrajectories overlap. In that case, the inconsistency score of each point on the trajectory is evaluated by the median over the difference values of all the subtrajectories to which the point belongs (Fig. 5). The final decision of the subtrajectory-level fault diagnosis is determined by whether the score exceeds the threshold value of $K_d = 5$ (m). The performance of the VO diagnosis above relies on the reliability of the VPR measurements. Hence, we





(2012/01/11, 2012/04/29)

Fig. 6. Prediction results.

distinguished the cases where VPR predicted the same place class at the start and goal points of the subtrajectory and excluded non-informative subtrajectories from the evaluation process. This is because such subtrajectories indicate that the robot remained at the same place class throughout the subtrajectory. This strategy is simple yet effective.

TABLE I
PRECISION-RECALL PERFORMANCE OF DSL [%].

	TP	TN	FP	FN
VO	69.0	31.0	0.0	0.0
VPR	45.5	13.6	7.7	33.2

III. EXPERIMENT

In this experiment, we used the public NCLT dataset. The NCLT dataset is a large-scale long-term dataset obtained every fortnight from January 8, 2012 to April 5, 2013 using a Segway-type robot on the north campus of the University of Michigan. The robot was equipped with an omni-directional camera Lady Bug3 and a GPS. In this study, images from a camera (Cam5) and GPS data were used. A pairing of training and test sets, (2012/01/11, 2012/04/29) was used. For the test set, the first 1/6 part of the sequence is used. These image sets comprised 26,209 and 21,511 images, respectively. Images are resize to 320×240 before being input to the training and test processes.

The size of a grid cell in the grid-based place partitioning used in the VPR was 30(m)×30(m).

An ideal method for performance evaluation would be to (1) conduct SLAM with DSL for a sufficiently large number of test visual-experiences ("journeys") in the robot workspace, (2) update the training set by replacing the portion corresponding to the localized region in the training data with the corresponding data in the current domain, (3) fine-tune the DNNs with the updated training data, and (4) evaluate the performance by comparing the SLAM performance with and without the updated training data. However, Step 3 requires two DNNs to be trained per dataset sample, which is cost prohibitive. Hence, we used an alternative lightweight method for evaluation. This method is different from the methods outlined in Steps 3 and 4 above. Instead of performing Steps 3 and 4, it modifies the test data by excluding the "portion" identified in Step 2. This performance evaluation is appropriate because as in the case of retraining, the DNNs can be precluded from being affected by the input data of the domain-shifted region localized by the fault diagnosis. This performance evaluation is extremely efficient because it does not require fine-tuning DNNs per test journey.

The ground-truth data for VO/VPR diagnosis were created using the same diagnosis method but with modified input data. For VO diagnosis, the EKF was used using the ground-truth VPR measurements while performing DSL using the same procedure as outlined in Section II-D. For VPR diagnosis, consistency was assessed between the raw VPR measurements and ground-truth VO measurements.

Diagnosis performance was evaluated based on indictors of precision and recall. Based on the fault diagnosis results and the ground-truth data, the numbers of true positives/negatives (TPs/TNs) and false positives/negatives (FPs/FNs) were calculated. The results are shown in Table I.

Among the results in Table I, the number of correctly diagnosed faults (TP/TN) was on average, 84.5 % for VO and 58.6 % for VPR, and the number of incorrectly diagnosed faults (FP, FN) was 15.5 % for VO and 41.4 % for VPR. It was observed that the numbers of correctly diagnosed faults in VO and VPR were larger than the number of mistaken fault diagnoses.

Figure 6 shows the results predicted by VO. The performance was evaluated based on the abovementioned evaluation procedures. As shown, the prediction result of VO was affected significantly by noise. This occurred because the visual appearance of the ground changed significantly because of domain shifts, which resulted in the significant accumulation of errors. The result shows that the SLAM performance improved when the results of domain-shift localization were incorporated to the SLAM system.

IV. Conclusions

In this study, we developed a new deep SLAM framework that can perform fault diagnosis. Unlike conventional scenarios pertaining to SLAM diagnosis, the deep SLAM comprised DNNs with black-box characteristics. Hence, we used only the input/output signals of the DNNs to diagnosis the SLAM system. The proposed approach can diagnose both single-view (i.e., VPR) and multiview (i.e., VO) signals using the EKF and through consistency verification. Experiments demonstrated that compared with a vanilla deep SLAM system that could not perform DSL, the proposed deep SLAM framework yielded a path that was more similar to the actual measured GPS path.

References

[1] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*. IEEE, 2017, pp. 2043–2050. [Online]. Available: https://doi.org/10.1109/ICRA.2017.7989236

[2] X. Fei, K. Tanaka, Y. Fang, and A. Takayama, "Long-term ensemble learning for cross-season visual place classification," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 22, no. 4, pp. 514–522, 2018. [Online]. Available: https://doi.org/10.20965/jaciii.2018.p0514

[3] R. Li, S. Wang, and D. Gu, "Deepslam: A robust monocular slam system with unsupervised deep learning," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3577–3587, 2020.

[4] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 2758–2766. [Online]. Available: https://doi.org/10.1109/ICCV.2015.316

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[8] H. Tomoe and T. Kanji, "Long-term knowledge distillation of visual place classifiers," in *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*. IEEE, 2019, pp. 541–546. [Online]. Available: https://doi.org/10.1109/ITSC.2019.8917199

[9] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[10] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *arXiv preprint arXiv:1701.08493*, 2017.