Sensors and Materials, Vol. 34, No. 1 (2022) 251–260
MYU Tokyo

S & M 2808

Object Detection of Road Facilities Using YOLOv3 for High-definition Map Updates

Tae-Young Lee, Myeong-Hun Jeong,* and Almirah Peter

Department of Civil Engineering, Chosun University,
309 Pilmun-daero, Dong-gu, Gwangju 61452, Republic of Korea

(Received November 15, 2021; accepted January 4, 2022)

Keywords: high-definition (HD) map, object detection, autonomous driving, deep learning, YOLOv3

Autonomous driving technology is significantly based on the fusion of high-definition (HD) maps and sensors. Therefore, the construction and update of HD maps must be emphasized to achieve full driving automation. Herein, a method is proposed to detect road facilities using object detection with images, particularly for HD map updates utilizing the You Only Look Once version 3 (YOLOv3) algorithm. The proposed approach, a deep-learning-based object detection method, utilizes transfer learning, which can detect objects in road facilities and record road sections that require maintenance. To test the effectiveness of the detection method, we analyze video footage captured in the Korean road environment. The experimental results show that this method achieves a mean average precision (mAP) of 58 and can update HD maps using a crowdsourcing framework.

1. Introduction

Autonomous driving technology is currently being developed owing to industry demand for a more robust detection system to ensure safety during driving. An example of such a detection system is the high-definition (HD) map.(1) HD maps contain a 3D layout of environmental information regarding roads in advance, afforded by the use of a driving vehicle equipped with a mobile mapping system (MMS) that includes sensors such as an inertial navigation system, radio detection and ranging sensors, light detection and ranging (LiDAR) sensors, cameras, and global navigation satellite systems. In addition, HD maps further enhance the features of the detection system. In complex road environments, HD maps enable one to recognize road facilities (e.g., road signs or traffic lights) and be aware of a vehicle's surroundings. However, most HD map construction processes are currently performed manually, which is both costly and time-consuming. Hence, methods to achieve automatic HD map construction are being investigated actively. In particular, a system that automatically detects any changes on roads and corrects them synchronously on a map is highly recommended. Research pertaining to HD map updating focuses on real-time systems that use a crowdsourcing framework.(2–5) Although a significant amount of time and money is required to complete HD maps, updates to reflect

*Corresponding author: e-mail: mhjeong@chosun.ac.kr
https://doi.org/10.18494/SAM3732

ISSN 0914-4935

volatile road conditions in real time are necessary to ensure safety during autonomous driving. This framework can be categorized into two stages: the stage of recognizing objects to identify changes and the stage of updating HD map features via an Internet server. This study aims to improve the initial detection stage of identifying changes through object recognition and realize an HD map updating system for autonomous driving in the Republic of Korea. In particular, the proposed method explicitly addresses Korean road facilities based on the road and traffic signs of the Republic of Korea. Methodologies for developing an efficient change detection system in a road environment are tested in this study. Notably, the You Only Look Once version 3 (YOLOv3) algorithm was utilized via transfer learning to detect traffic signs, road signs, and traffic lights in a video featuring a road environment in real time.

In Sect. 2, the features of our approach are reviewed on the basis of previous relevant studies. Section 3 presents the experimental data and the proposed method. Section 4 presents and discusses the results of the study. Finally, Sect. 5 provides conclusions, including suggestions for future research.

2. Background

HD maps provide road environment information and point of interest information, such as road alignment, lane classification, and road signs required for autonomous driving. To update these maps promptly and periodically, studies pertaining to the automatic detection of object changes are currently being conducted using LiDAR data and camera image data from sensor information to perform partial corrections. Road facility object detection using LiDAR data has been investigated extensively.[6–10] Hata and Wolf detected lanes by categorizing LiDAR point data for lanes and asphalt,[6] and Jo *et al*. attempted to identify whether a traffic sign has disappeared or has been added.[7] Ma *et al.* increased the accuracy of detecting and classifying road markings by applying a deep learning framework.[8] Pannen *et al*. constructed a framework that recognizes changes by detecting lanes and immediately providing a crowdsourced updated HD map.[9] Kim *et al*. used a point unit to determine whether shape change has occurred as well as to apply the change immediately; however, they did not specify the changed object.[10] Although using LiDAR data in such a manner provides outstanding accuracy in identifying the shape of an object, to maintain the up-to-dateness of the map, many vehicles equipped with MMS equipment in addition to LiDAR sensors are required to observe all roads in a wide area. However, the use of LiDAR equipment is costly and, therefore, not optimal for updates. By contrast, because modern cameras offer a relatively high resolution, the footage captured through mobile devices can be easily used in image detection systems. Higher resolution footage enables the easy identification of objects using a single device while minimizing costs. Therefore, object detection research using camera footage data is being actively pursued.[11–15] Cai *et al.* conducted an accurate vehicle localization study by detecting lanes through a camera and matching the results with existing HD maps and global navigation satellite system results.[11] Choi *et al*. identified and utilized lane, lane endpoint, and road signs for localization, and Elfring *et al*. detected traffic signs.[12,13] Alcantarilla *et al*. detected changes via masking and performing a pixel-by-pixel comparison of road facility objects in an image; however, the method was

limited by insufficient object identification.[14] Heo *et al.* detected changes by comparing the vector-type object of the HD map and the road facilities of camera image data through adversarial learning for HD map updates.[15] However, the method is not applicable to road signs and traffic lights and can only detect changes in road markings. The aim of this study is to simultaneously detect specific road facilities such as road surface markings, traffic lights, and signs, among other components of the HD map, using a single camera for HD map updates.

3. Materials and Methods

3.1 Area of study and data

To conduct the research, we used the road environment panoramic image artificial intelligence (AI) data provided by the AI Hub site of the Korean Intelligence Information Society Agency.[16] These data are composed of 2711280 images of 189 types of static road environment objects obtained while driving a total of 3400 km on the major roads in Seoul, the Republic of Korea, and are used to obtain training data for automatic recognition models. Among them, 14184 images with objects were used for transfer learning to complete the algorithm for detecting road facilities. Excluding objects that are insufficient for performing deep learning among the objects in the image, Table 1 shows a total of 12 objects managed in the HD map.

The Ministry of Land, Infrastructure and Transport in Korea defined 14 layers, including 189 road facilities for HD maps. This research focused on just 12 road facilities. Our experimental data did not include all training images of the 189 road facilities. We selected 12 road facilities among the experimental data because they provided enough training data. The 12 road facilities covered three layers: road surface markings, traffic lights, and safety signs.

High-performance computation is required to complete an algorithm that automatically detects road facilities by deep learning using camera footage. Amazon Web Services (AWS) was first used to upload and store the data in an S3 bucket, which is a storage space for AWS. Subsequently, a Python environment was established in AWS for data preprocessing and deep learning implementation via elastic cloud computation. This process is shown in Fig. 1. The

Table 1
Overview of variables used in this study.

Variables	Class code
Signs	No-Parking-or-Stopping
	No-Parking
	Maximum-Speed-Limit
	Towing-Zone
Road marks	Speed-Limit
	Right-Turn
	Left-Turn
	Straight
	Straight-or-Right
	Crosswalk
	Pedestrian-Crossing-Ahead
Traffic light	Traffic-Light

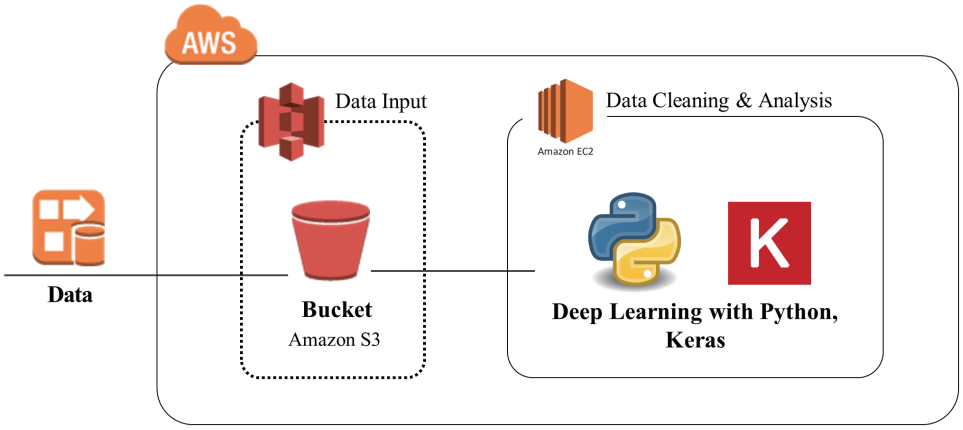


Fig. 1. (Color online) System architecture.

AWS instance option selected for this study was p2.xlarge [i.e., random access memory 61 GiB and graphics processing unit (GPU) 11441 MiB]. In addition, CUDA version 10.0 was used as the GPU, and deep learning was implemented using Keras.

3.2 YOLOv3

Unlike other object detection techniques, YOLO is a one-stage detector that detects objects in an image by simultaneously performing localization, computation of the location of an object in the image, and classification to identify the object. It is a detection technique that enables real-time detection via processing.[17] In addition, owing to its high accuracy, it is well known as an exceptional deep-learning-based object detection algorithm. The localization process first involves the indication of an object's position based on its location and boundaries by a bounding box. YOLO is based on partitioning an image into several grid cells and detecting one object for each cell as illustrated in Fig. 2.

The process of YOLO determining the x,y coordinates (b_x, b_y) of the center of the bounding box, and the width (b_w) and height (b_h) of the bounding box are as follows:

$$\sigma(t_o) = Pr(object) \cdot IoU(b, object), \tag{1}$$

$$b_x = \sigma(t_x) + c_x, \tag{2}$$

$$b_y = \sigma(t_y) + c_y, \tag{3}$$

$$b_w = p_w \cdot e^{t_w}, \tag{4}$$

$$b_h = p_h \cdot e^{t_h}, \tag{5}$$

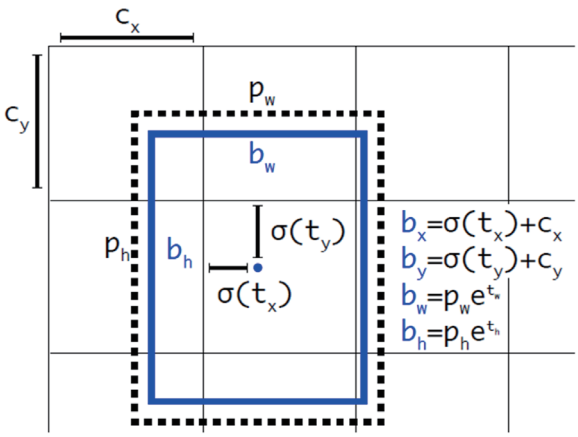


Fig. 2. (Color online) Bounding boxes with location prediction.[17]

where $\sigma(t_o)$ is first calculated to obtain the final x,y coordinates of the bounding box, and it is determined as a value between 0 and 1 by considering whether it is an object using logistic regression and multiplying the corresponding *IoU* value. Subsequently, the result is added to c_x and c_y. c_x and c_y are the x- and y-coordinates of the upper left of each grid cell, respectively. t_x, t_y, t_w, and t_h refer to the predicted model offset values; t_x is the box center x-coordinate; t_y is the box center y-coordinate; t_w is the box width; t_h is the box height shift to obtain the final x- and y-coordinates of the bounding box. p_w and p_h are the width and height of the anchor box, respectively. During learning, t_w and t_h attempt to be approximately 0; if they are 0, then e^{t_w} and e^{t_h} are obtained. Subsequently, e^{t_w} and e^{t_h} return 1 and become the same as the prior value.

Figure 3 shows the classification stage, which is the process of classifying and differentiating the background and object and then determining the object. YOLOv3 detects an object under the assumption that the object exists in each grid cell and predicts the final class of the cell based on the binary cross-entropy loss. In this process, the classification of hierarchical classes such as a person and their gender (man/woman) is enabled using an independent logistic classifier. The independent logistic classifier is used instead of a softmax classifier, and it differentiates classes using values for each class as class probabilities. In addition, it is expressed as a value between 0 and 1 in terms of the objectness score, which conveys the confidence in the final prediction class. Subsequently, it is determined whether it should be recognized as an object. This model was trained on the Microsoft Common Objects in Context (MS COCO) dataset.[18] These data contained a set of various daily life photographs created for computer vision learning, in which each object is segmented and labeled.

3.3 Transfer learning

The existing YOLOv3 can recognize objects with up to 80 features, including men, women, and dogs. However, owing to the lack of usable training images for recognizing facilities in a road environment, the model must be further trained using these additional images. Therefore,

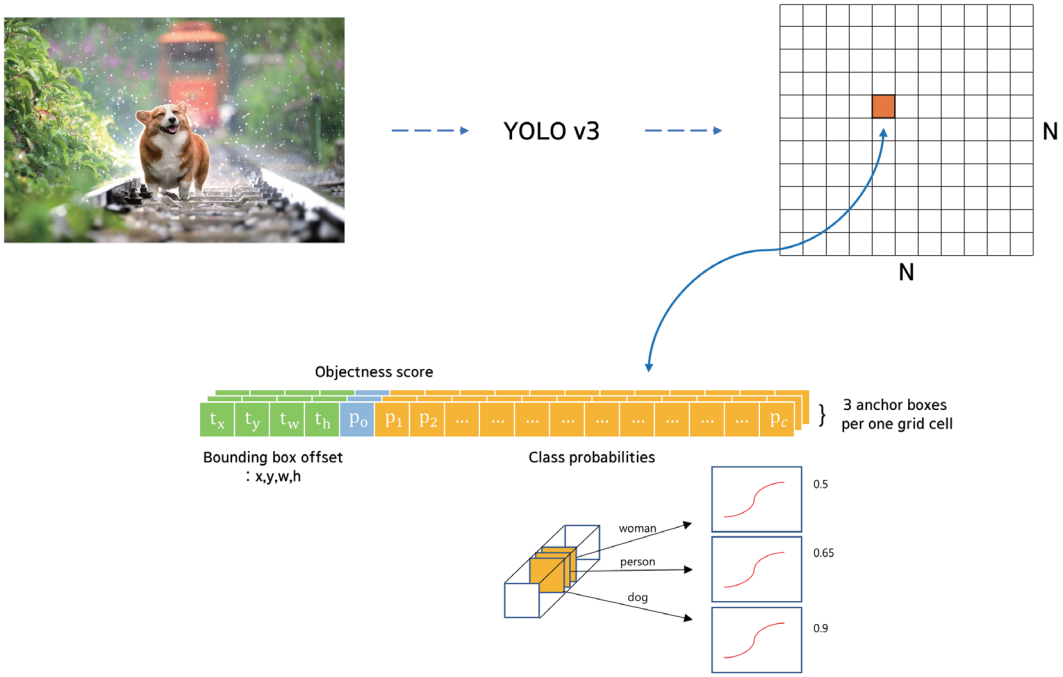


Fig. 3. (Color online) Multilabel classification.

transfer learning was utilized in addition to facilitating the training of the YOLOv3 algorithm to recognize objects in the road environment, such as road surface markings, traffic lights, and road signs. Using this method, we successfully maintained the framework of YOLOv3 and enabled the algorithm to detect new objects while maintaining its advantages. Transfer learning does not solely involve newly learning an entire convolutional neural network (CNN) that extracts features, but uses the weights obtained when completing YOLO in advance to learn new target data. Subsequently, the new target data are learned only in the fully connected layer, thereby completing the algorithm in a shorter time. Hence, transfer learning is an ideal algorithm for establishing an object detection model for individual datasets. Even in an environment where sufficient data for training are difficult to obtain, transfer learning can be implemented using the generated results, providing a relatively high precision.[19–23] Furthermore, according to Yosinski *et al.*, if an entire CNN is trained only with individual datasets, then the model may be biased.[24] Hence, transfer learning was considered appropriate for the analysis. Fine-tuning was performed to find the optimal hyperparameters of transfer learning by adjusting the size of training images, the batch size, and the number of epochs. Among the various experiments, two sets of the initial and final values are presented for comparison in Fig. 4. Table 2 presents the optimal learning rates found through the Keras callback function in the learning process.

4. Results

The image data for 12 types of static objects were transferred to YOLOv3 to detect road facilities. Consequently, the newly created YOLOv3 accurately detected the object when applied to the general road footage, which was not used for learning, as shown in Fig. 5. The

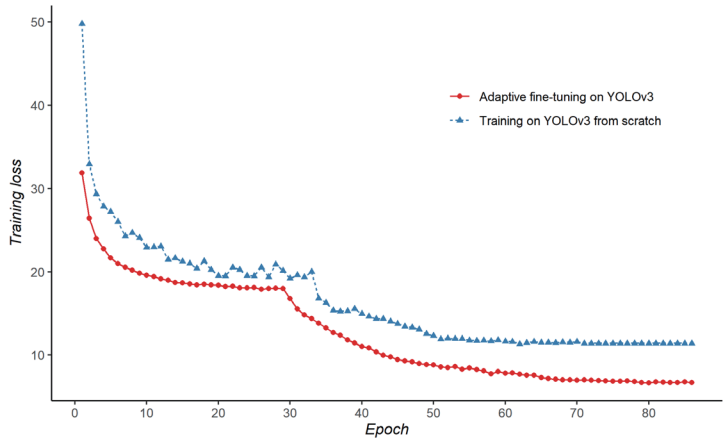


Fig. 4. (Color online) Curves of training loss for different hyperparameter settings.

Table 2
Loss of transfer learning to YOLOv3.

	Check point 1	Check point 2	Check point 3	Check point 4	Check point 5
Loss	17.9966	7.5566	6.9730	6.7041	6.6980
Validation loss	18.7987	7.9946	7.6314	7.2628	7.1726

(a) (b)

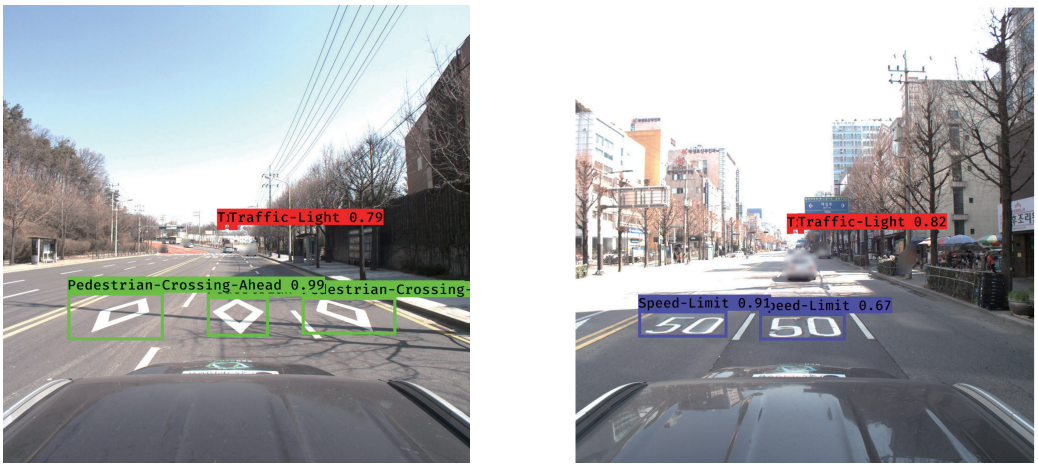
Fig. 5. (Color online) Road facilities detection: (a) straight and straight or right road marks and (b) traffic light and maximum speed limit sign.

detection performance was evaluated by a confidence score, presented next to the class name. This value was obtained by partitioning the intersection area of the bounding box predicted via object detection and the ground-truth bounding box based on the union area of the two bounding boxes, which resulted in a value between 0 and 1. The object detection result of the algorithm was returned as a text result in the form of [class name, confidence score, bounding box top left coordinate, bounding box bottom right coordinate], and the visualization result obtained by matching it with the image is shown in Fig. 5.

As shown in Fig. 5(a), the straight arrow marking on the road surface and the straight and right turn arrow markings were detected accurately. Moreover, Fig. 5(b) shows that the maximum speed limit sign and two traffic lights were detected. Figure 6(a) shows the result of detecting the crosswalk-ahead warning sign and traffic lights, whereas Fig. 6(b) shows that the speed limit road surface marks and traffic lights were detected accurately.

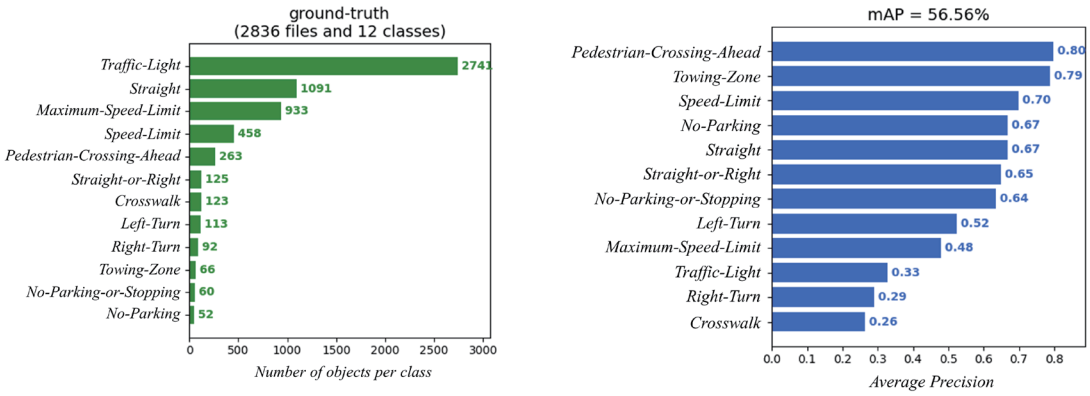
Figure 7 shows the results of quantifying the performance of the object detection model for the 12 classes used to detect the road surface, signs, and traffic lights mentioned in Table 1 by the average precision (AP). AP is an index used to consider both recall and precision, and its value is derived from a precision–recall curve and the calculated area under the curve. In object detection, precision refers to the class matching accuracy of the discovered object, which is important; however, the recall of matching the number of objects in the image is equally important. Although a tradeoff exists between these two scores, they ensure that the model generates relevant results in proportion to the number of predictions. Figure 7(a) shows the 12 classes used as ground truth in 2836 test image files, as well as the number of objects per class in the image. Figure 7(b) shows the AP for each class and the mean average precision (mAP). When evaluating the mAP, evaluation criteria may differ depending on the number of objects present in the image and the difficulty in distinguishing between objects. On the basis of an evaluation, the target road facility object detection model of this study indicated a mAP value of 56.56, which afforded a performance level similar to that of the existing YOLOv3 (mAP value of 57.9), while completing the algorithm for detecting road facilities in real time.

On the basis of the AP values for each class in Fig. 7(b), although most road facility objects were detected accurately, it was confirmed that the object corresponding to a specific road surface and the traffic light demonstrated low performance. In most cases, the inferior detection performance was due to the annotation of smaller objects that were difficult to observe with the naked eye. In particular, it was confirmed that the detection performance



(a) (b)

Fig. 6. (Color online) Road facility detection: (a) pedestrian crossing ahead and traffic lights and (b) speed limit road marks and traffic light.



(a) (b)

Fig. 7. (Color online) Performance of object detection model.

was worse than that of the sign because the road surface markings were affected by perspective; therefore, it was difficult to discern the shape of the object owing to distance.

5. Conclusions

A rapid and accurate update of the HD map is required to promote and implement a safer autonomous driving system. For this HD map update, we conducted a study to identify a method to automatically recognize objects in a road environment. A deep-learning-based object detection model was constructed to detect objects in the HD map with the precision from road driving footage. The evaluation of the model indicated a mAP value of 56.56 as a result of transfer learning using images containing 12 types of road facilities. This study enables the presence or absence of change to be determined by comparing it with the existing HD map by accurately discovering objects in real time using a single camera. Applying this method to the crowdsourcing framework enables simultaneous updates to many vehicles on a road by changing the road environment information. However, owing to the rapid development of state-of-the-art object detection algorithms, we plan to conduct further studies using advanced algorithms to improve the detection of objects on road facilities.

Acknowledgments

This study was supported by a research fund from Chosun University, 2021.

References

1 R. Liu, J. Wang, and B. Zhang: J. Navig. **73** (2020) 324. https://doi.org/10.1017/S0373463319000638
2 P. Zhang, M. Zhang, and J. Liu: J. Sens. **21** (2021) 2477. https://doi.org/10.3390/s21072477
3 D. Pannen, M. Liebner, W. Hempel, and W. Burgard: Proc. 2020 IEEE Int. Conf. Robotics and Automation (IEEE, 2020) 2288–2294. https://doi.org/10.1109/ICRA40945.2020.9197419
4 L. He, S. Jiang, X. Liang, N. Wang, and S. Song: eprint arXiv:2107.07030 (2021). https://arxiv.org/abs/2107.07030

5 K. Kim, S. Cho, and W. Chung: IEEE Rob. Autom. Lett. **6** (2021) 1895. https://doi.org/10.1109/LRA.2021.3060406
6 A. Hata, and D. Wolf: Proc. In 17th Int. IEEE Conf. Intelligent Transportation Systems (IEEE, 2014) 584–589. https://doi.org/10.1109/ITSC.2014.6957753
7 K. Jo, C. Kim, and M. Sunwoo: J. Sens. **18** (2018) 3145. https://doi.org/10.3390/s18093145
8 L. Ma, Y. Li, J. Li, Y. Yu, J. M. Junior, W. N. Gonçalves, and M. A. Chapman: IEEE Trans. Intell. Transp. Syst. **22** (2020) 1981. https://doi.org/10.1109/TITS.2020.2990120
9 D. Pannen, M. Liebner, W. Hempel, and W. Burgard: Proc. 2020 IEEE Int. Conf. Robotics and Automation (IEEE, 2020) 2288. https://doi.org/10.1109/ICRA40945.2020.9197419
10 C. Kim, S. Cho, M. Sunwoo, P. Resende, B. Bradaï, and K. Jo: IEEE Access **9** (2021) 8028. https://doi.org/10.1109/ACCESS.2021.3049482
11 H. Cai, Z. Hu, G. Huang, D. Zhu, and X. Su: J. Sens. **18** (2018) 3270. https://doi.org/10.3390/s18103270
12 M. J. Choi, J. K. Suhr, K. Choi, and H. G. Jung: IEEE Access **7** (2019) 149846. https://doi.org/10.1109/ACCESS.2019.2947287
13 J. Elfring, S. Dani, S. Shakeri, H. Mesri, and J. W. van den Brand: Proc. 2020 IEEE 23rd Int. Conf. Intelligent Transportation Systems (IEEE, 2020) 1–6. https://doi.org/10.1109/ITSC45102.2020.9294208
14 P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi: Auton. Robots **42** (2018) 1301. https://doi.org/10.1007/s10514-018-9734-5
15 M. Heo, J. Kim, and S. Kim: Proc. 2020 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IEEE, 2020) 10218–10224. https://doi.org/10.1109/IROS45743.2020.9340757
16 The Ministry of Science and ICT in Korea: https://aihub.or.kr (accessed June 2021).
17 J. Redmon and A. Farhadi: eprint arXiv:1804.02767 (2018). https://arxiv.org/abs/1804.02767
18 T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, and C. L. Zitnick: Proc. European Conf. Computer Vision (Springer, 2014) 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
19 M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa: Meas.: J. Int. Meas. Confed. **167** (2021) 108288. https://doi.org/10.1016/j.measurement.2020.108288
20 A. Mehmood, S. Yang, Z. Feng, M. Wang, A. S. Ahmad, R. Khan, and M. Yaqub: Neuroscience **460** (2021) 43. https://doi.org/10.1016/j.neuroscience.2021.01.002
21 J. G. A. Barbedo: Comput. Electron. Agric. **153** (2018) 46. https://doi.org/10.1016/j.compag.2018.08.013
22 Z. Chen, T. Zhang, and C. Ouyang: Remote Sens. **10** (2018) 139. https://doi.org/10.3390/rs10010139
23 M. S. Khan, S. B. Jeon, and M. H. Jeong: Remote Sens. **13** (2021) 4976. https://doi.org/10.3390/rs13244976
24 J. Yosinski, J. Clune, Y. Bengio, and H. Lipson: eprint arXiv:1411.1792 (2014). https://arxiv.org/abs/1411.1792