sensors

MDPI

Article

CNN-Based Fault Detection of Scan Matching for Accurate SLAM in Dynamic Environments

Hyein Jeong and Heoncheol Lee *

Department of IT Convergence Engineering, Kumoh National Institute of Technology,
Gumi 39177, Republic of Korea

* Correspondence: hclee@kumoh.ac.kr; Tel.: +82-54-478-7458

Abstract: This paper proposes a method for CNN-based fault detection of the scan-matching algorithm for accurate SLAM in dynamic environments. When there are dynamic objects in an environment, the environment that is detected by a LiDAR sensor changes. Thus, the scan matching of laser scans is likely to fail. Therefore, a more robust scan-matching algorithm to overcome the faults of scan matching is needed for 2D SLAM. The proposed method first receives raw scan data in an unknown environment and executes ICP (Iterative Closest Points) scan matching of laser scans from a 2D LiDAR. Then, the matched scans are converted into images, which are fed into a CNN model for its training to detect the faults of scan matching. Finally, the trained model detects the faults when new scan data are provided. The training and evaluation are performed in various dynamic environments, taking real-world scenarios into account. Experimental results showed that the proposed method accurately detects the faults of scan matching in every experimental environment.

Keywords: deep learning; scan matching; 2D SLAM

check for updates

Citation: Jeong, H.; Lee, H. CNN-Based Fault Detection of Scan Matching for Accurate SLAM in Dynamic Environments. *Sensors* **2023**, *23*, 2940. https://doi.org/10.3390/s23062940

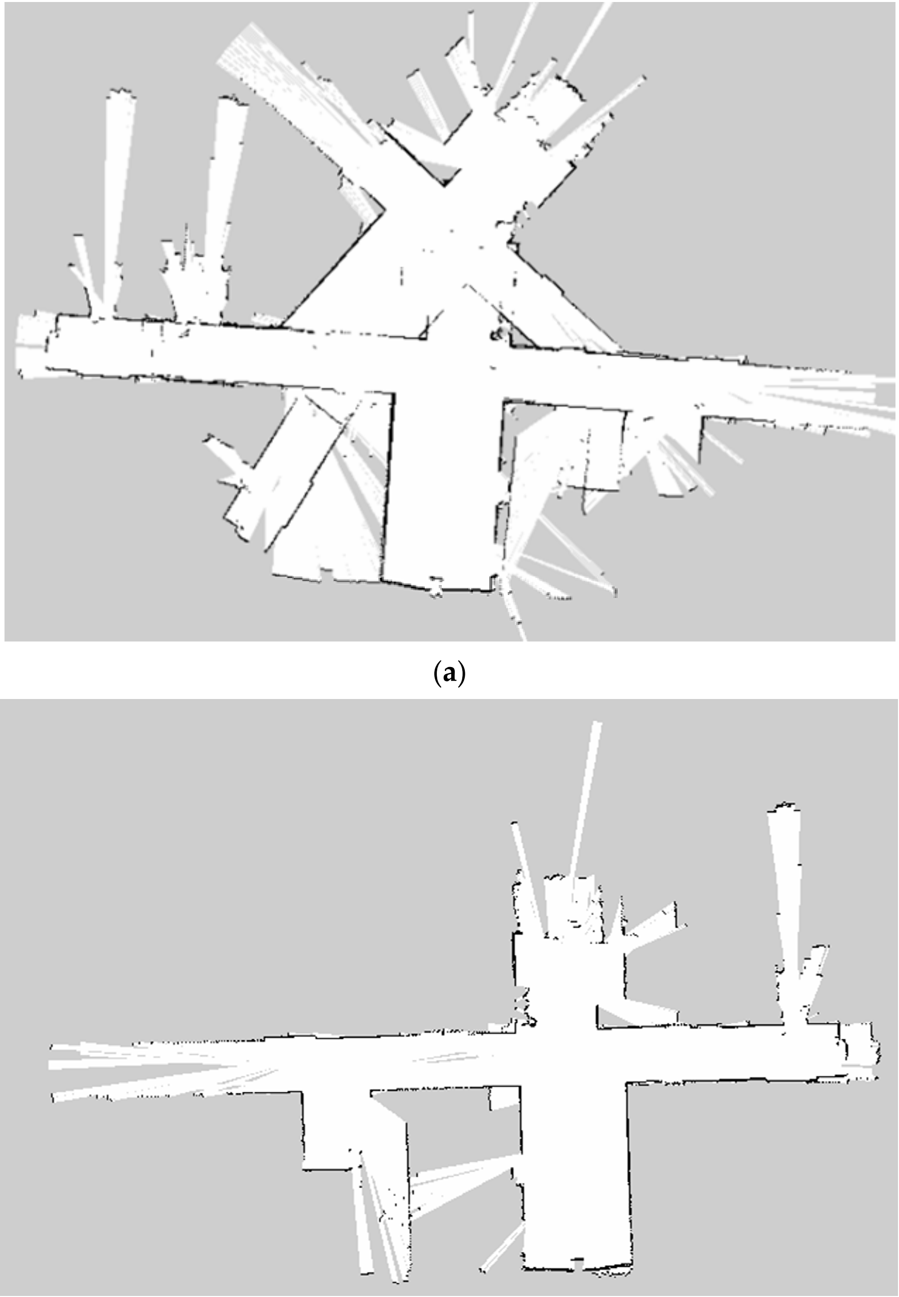
Academic Editor: Sergio A. Velastin

Received: 19 January 2023
Revised: 24 February 2023
Accepted: 2 March 2023
Published: 8 March 2023

1. Introduction

SLAM (Simultaneous Localization and Mapping) is a method which allows a mobile robot to build a map and estimate its location simultaneously in an unknown environment [1–3]. Despite recent studies in novel SLAM algorithms, SLAM in dynamic environments is still being studied for its more robust performance [4,5]. In 2D SLAM, when objects are dynamic or interrupt the light from a LiDAR (Light Detection and Ranging) in an environment, the detected environment by the sensor changes by seconds [6,7]; thus, the accuracy of SLAM degenerates, as shown in Figure 1.

SLAM problems in dynamic environments have been explored with various solutions. The authors of Ref. [8] proposed a filter-based dynamic object detection and removal method that requires both a camera and a laser scanner for visual SLAM and laser-based robot path planning. Ref. [9] suggested a distance filter to eliminate dynamic objects in laser scan data with multi-layer searching [10] for robot localization for 2D SLAM. Meanwhile, various methods utilizing deep learning for SLAM have been proposed. Ref. [11] is a visual SLAM with the LIFT network used to extract features of challenging environments. Ref. [12] suggested a moving object tracking algorithm that utilizes ML-RANSAC and CNN for SLAM in a dynamic environment. Ref. [13] suggested a SLAM system that uses local features obtained with CNN instead of traditional hand-made features. Regarding deep learning-based 2D SLAM, most existing methods use a method to convert range values of laser scans in Polar coordinates to panoramic grayscale images for the training of CNN models [14–16]. In this paper, for the training processes, we have used a new form of images which represent the laser range values as scan points in Cartesian coordinates with multiple colors. Table 1 compared the related works in the aspects of dynamic objects, the use of deep learning, and the type of sensors used.



(**a**)

(**b**)

Figure 1. An example of the accuracy degeneration of SLAM in dynamic environments (**a**) SLAM in dynamic environments; (**b**) SLAM in static environments.

Table 1. The comparison of related works.

Reference	Dynamic Objects	DL-Based	Sensors
Proposed	O	O	Only LiDAR
[8]	O	X	Vision, LiDAR
[9]	O	X	Only LiDAR
[11,13]	X	O	Vision
[12]	O	O	Vision, LiDAR
[14,15]	X	O	Only LiDAR
[16]	X	O	LiDAR, IMU

As described previously, most of the existing SLAM solutions in dynamic environments or those which are deep learning-based require vision sensors to perform object detection for visual SLAM. An undesirable consequence of this is that the image data have a large data size and take much more time to process than laser scan data. In this regard, solutions for SLAM problems in dynamic environments with only a LiDAR, while not using vision sensors, are worth further exploration. Additionally, deep learning applications for 2D SLAM are still open to be thoroughly studied.

The main contributions of this paper are as follows.

- An online process which includes raw scan data acquisition, scan matching, matched scan image generation, and the fault detection of the scan matching has been proposed and performed successfully, as shown in Figure 2.
- A method to form the training images which represent two consecutive laser scans, thereby taking advantage of the effective CNN-model training, has been proposed for the first time.
- The fault detection of scan matching has been conducted with high accuracy in various dynamic real environments.

The remainder of this paper is organized as follows: In Section 2, we describe the problem of deterioration of SLAM and scan matching in dynamic environments. Section 3 proposes the process of training image generation from raw scan data and a CNN model for the fault detection of the scan matching. Section 4 describes the experimental environments and reports the results. Finally, we conclude in Section 5.

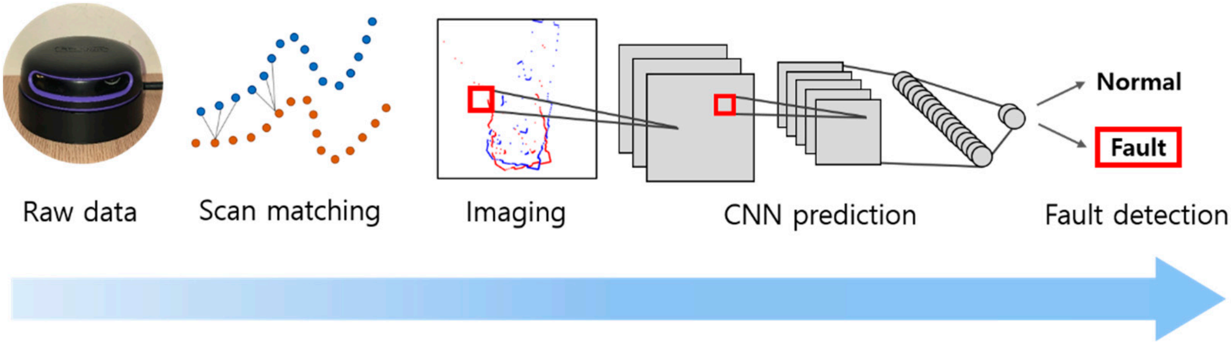


Figure 2. The concept overview of the proposed method. As soon as the raw scan data are received, the process of scan matching, the generation of images of matched scans, and fault detection are performed sequentially in each time step.

2. Problem Description

2.1. SLAM in Dynamic Environments

Various 2D SLAM algorithms with a LiDAR have been developed. This work has utilized the GMapping algorithm, one of the widely used SLAM algorithms based on RBPF (Rao-Blackwellized Particle Filter), to build a 2D grid map and a robot trajectory [17,18]. During the filtering process, the estimated robot trajectory is defined as $p(x_{1:t}|z_{1:t},\ u_{1:t-1})$, where x is the robot position, z is the sensor observation, and u is control information from the robot odometer. Then, while using the estimated robot positions and observations from the sensor, the estimated map m is defined as: $p(m|x_{1:t},\ z_{1:t})$.

However, in dynamic environments, the sensor data for the computations of the two posterior probabilities of the robot trajectory and the map are disrupted, resulting in bad SLAM performance. To improve this situation, our proposed method detects the faults of scan matching for the final decision to reflect the resulting scans from scan matching or otherwise on the computations as described in Figure 3, which represents the whole structure of 2D SLAM using the proposed method.

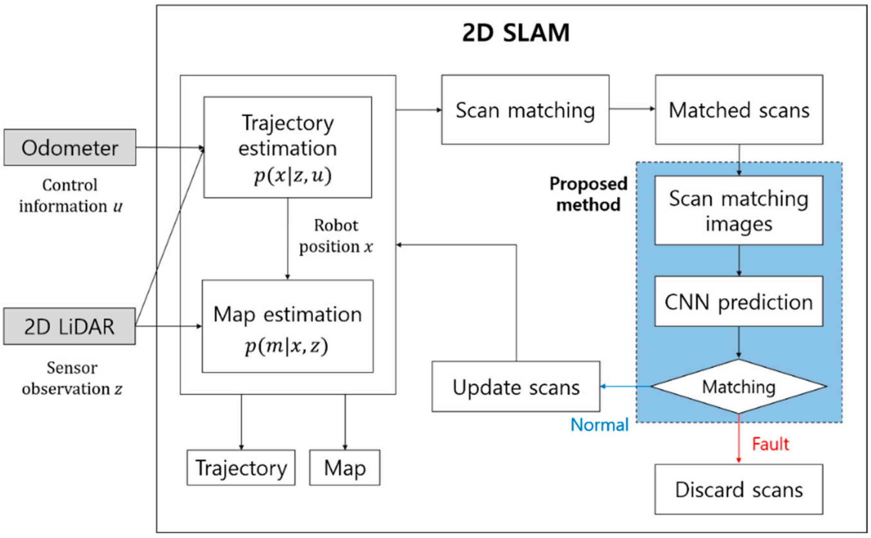


Figure 3. The whole structure of 2D SLAM with the proposed method.

In 2D SLAM, a new laser scan is matched to a previous scan. To take the results of matching, the matched scans are converted into scan images. When the trained CNN model confirms that the scan matching is successful, the new laser scan is used for updates of the robot position and the map. If the scan matching is predicted to be faulty, the new laser scan is discarded, but only the odometer data at the time are used. Not using the unaligned scan points prevents an incorrect estimation of the map and the robot trajectory.

2.2. ICP (Iterative Closest Points)

ICP is a scan-matching method that iteratively matches two consecutive laser scans. A transformation matrix that minimizes the distance between two scans is needed to align the current scan with the previous scan.

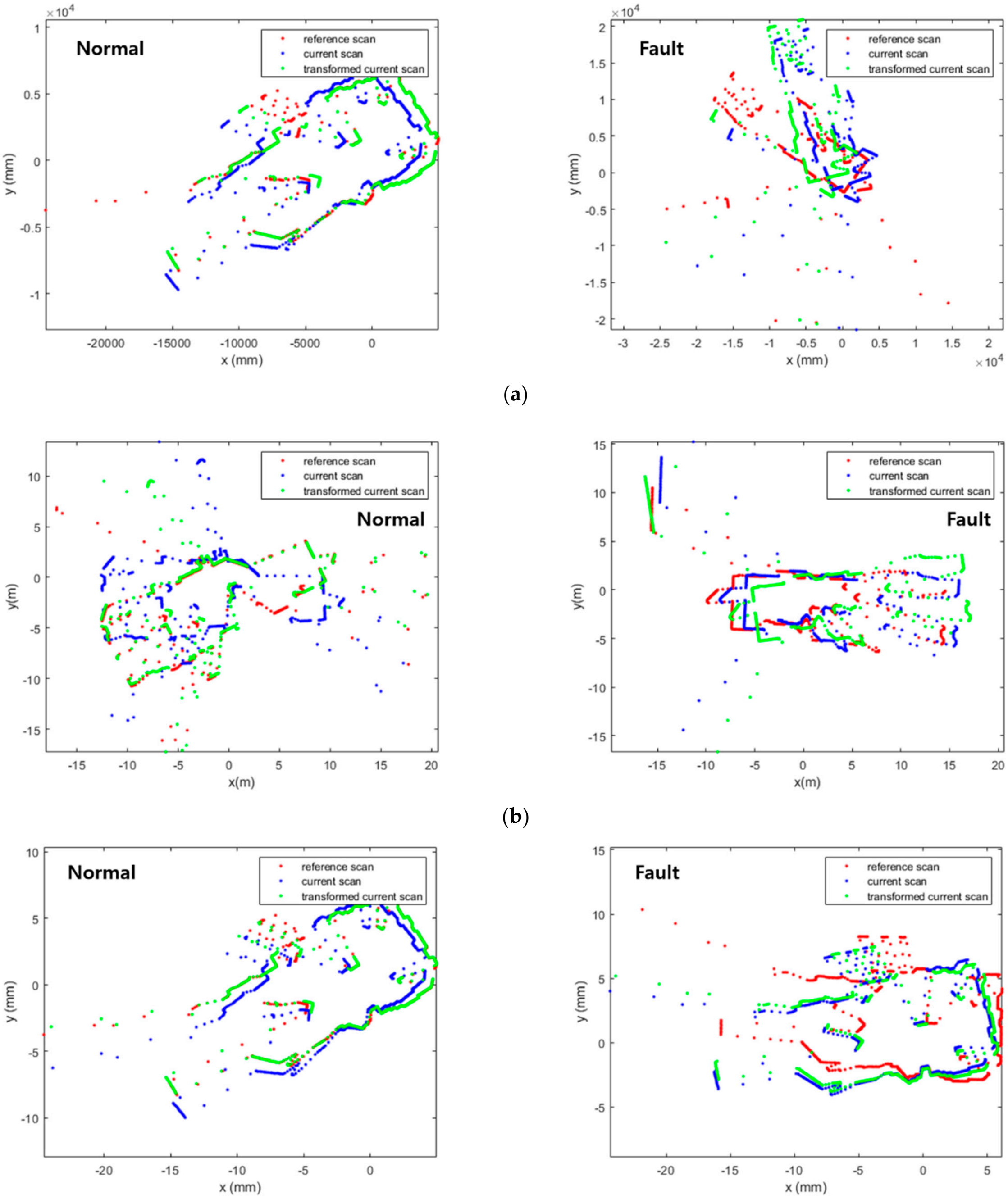
The closest points p_i in the current scan to the points q_i in the previous scan are searched. A cross-covariance matrix K is defined as:

$$K = \sum_{i=1}^{N} p_i \cdot {q_i}^T = \begin{bmatrix} cov(P_x,\ Q_x) & cov(P_x,\ Q_y) \\ cov(P_y,\ Q_x) & cov(P_y,\ Q_y) \end{bmatrix} \tag{1}$$

Then, to find the rotation elements in the transformation matrix, K is decomposed as: $K = U\Sigma V^T$ by a singular value decomposition, where U is the left singular vector and V is the right singular vector. The rotation matrix R is defined as: $R = UV^T$. The transformation matrix is the combination of rotation and translation elements. Thus, the translation matrix t is defined as: $t = \mu_Q - R \cdot \mu_P$, where μ_Q and μ_P are the centers of mass of the previous scan and current scan. The rotation and translation matrix are iteratively computed until the scan matching error defined as: $err = \| \mu_Q - R \cdot \mu_p \|$, is smaller than a preset threshold. In this paper, the maximum number of iteration times and the error threshold were set as 20 and 0.01 (m), respectively. Disallowing too many iterations, the matching was terminated when the distance between two scans was less than 1cm. However, in this paper, most of the matchings were not finished before 20 iterations because the scan data were collected in environments which were dynamically set.

2.3. The Problem of Scan Matching in Dynamic Environments

Scan matching is an important technique which allows SLAM to match the current scan to the previous scan of two consecutive laser scans by searching the overlapping parts [19]. ICP (Iterative Closest Points) [20–24], PSM (Polar Scan Matching) [25–27], and NDT (Normal Distributions Transform) [28–30] are representative scan matching methods. However, in dynamic environments, objects which move over time lessen the resemblance of two consecutive scans, making it challenging to perform accurate scan matching, as shown in Figure 4. In GMapping, scan matching is used to compute the maximum likelihood of the current robot position. The corrected robot position achieved via scan matching is defined as: $x_t = argmax\{p(z_t|x_t,\ m_{t-1})\ p(x_t|u_t,\ x_{t-1})\}$. However, scan matching may fail in dynamic environments, since the sensor data for the computations of the two posterior probabilities of the robot trajectory and the map are disrupted.



(**a**)

(**b**)

(**c**)

Figure 4. Normal and faulty scan matching with dynamic objects. (**a**) ICP (Iterative Closest Points); (**b**) PSM (Polar Scan Matching); (**c**) NDT (Normal Distributions Transform).

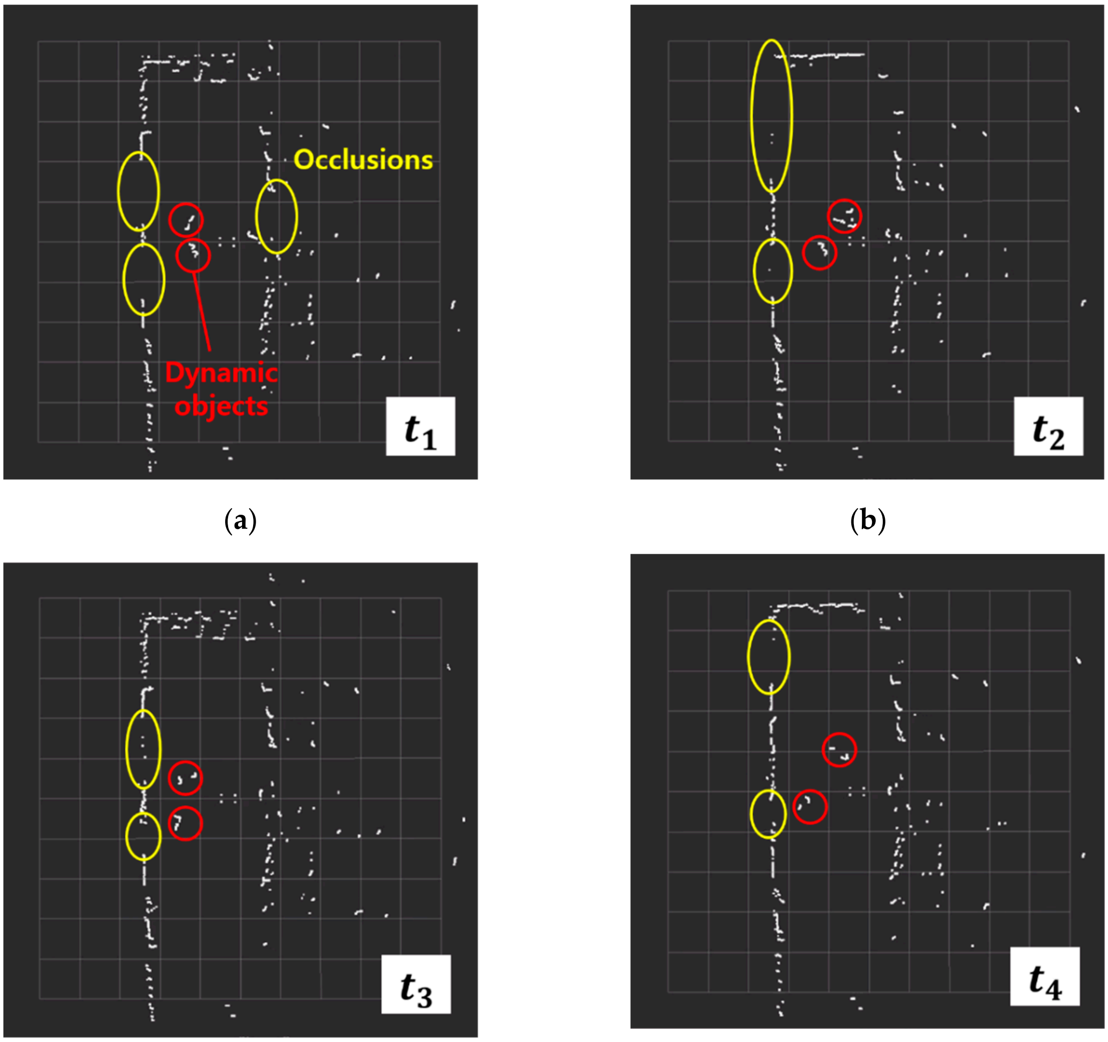
Consequently, the scan matching in GMapping, as well as the traditional scan matching methods, can be faulty due to dynamic objects and may cause errors in robot position estimation and map building. Even though various methods have been developed to resolve the problem of scan matching in dynamic environments, this remains an ongoing problem in SLAM communities. Therefore, this paper proposes a practical and reliable method to resolve the problem by detecting faulty scan matching based on deep learning.

3. Proposed Method

In environments with various dynamic scenarios, the raw scan data are acquired by a 2D LiDAR. The scans are then matched with ICP (Iterative Closest Points) scan matching. The matched scans are converted into scan images to be fed as training data for the CNN model for fault detection. When new scan data are acquired, the trained model detects faults in scan matching. The sequential tasks of acquisition of raw scan data, scan matching, conversion of matched scans into images, and fault detection are performed in a single process.

3.1. Data Acquisition

To obtain a large amount of laser scan data for the training of the CNN model, raw scan data from a high-resolution LiDAR are received in ROS (Robot Operating System) sensor messages. The raw scan data include ranges values of dynamic objects and their scan occlusions, as shown in Figure 5.



(**a**) (**b**)

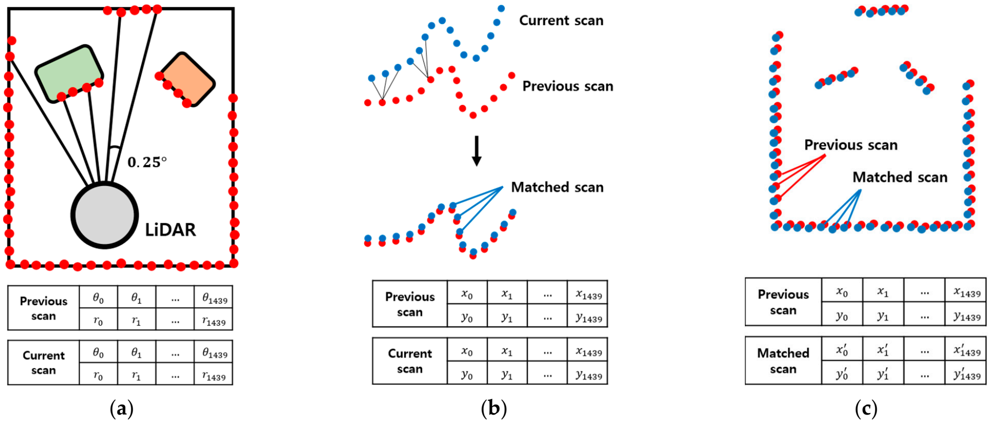
(**c**) (**d**)

Figure 5. Dynamic objects and scan occlusions in raw scan data. The positions of objects and the occlusions of scans are changed over time. (**a**) t = t_1; (**b**) t = t_2; (**c**) t = t_3; (**d**) t = t_4.

3.2. Training Images

To conduct effective training of the CNN model, the scan points are converted into simple scan-matching images with clear features. To generate scan-matching images, two consecutive scans in the raw scan data are scan matched at each timestamp using ICP.

The matched scans with Cartesian values are plotted as point graphs, as shown in Figure 6. To differentiate two scans in a pair, the previous scan is plotted in red, and the current scan in blue. Since the images contain only two scans in different colors on the white background, it is advantageous to extract features for the CNN model. In addition, to maximize the meaningful parts in images, the plotted areas of scan pairs are not fixed, but they change according to the ranges in which scans exist, as shown in Figure 7.



Previous scan	θ_0	θ_1	…	θ_{1439}
	r_0	r_1	…	r_{1439}
Current scan	θ_0	θ_1	…	θ_{1439}
	r_0	r_1	…	r_{1439}

(**a**)

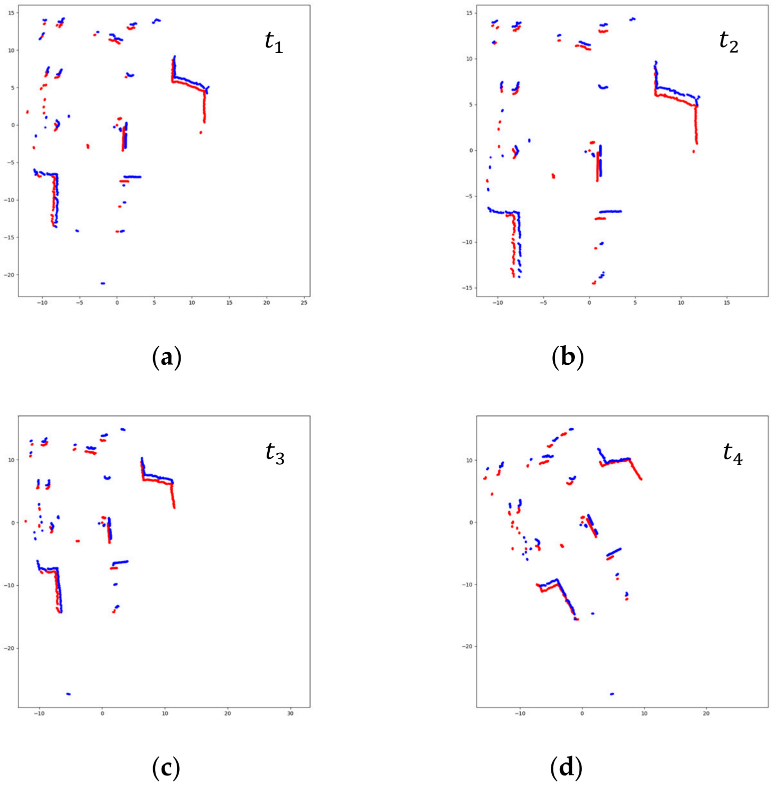
Previous scan	x_0	x_1	…	x_{1439}
	y_0	y_1	…	y_{1439}
Current scan	x_0	x_1	…	x_{1439}
	y_0	y_1	…	y_{1439}

(**b**)

Previous scan	x_0	x_1	…	x_{1439}
	y_0	y_1	…	y_{1439}
Matched scan	x'_0	x'_1	…	x'_{1439}
	y'_0	y'_1	…	y'_{1439}

(**c**)

Figure 6. The generation of training images. (**a**) Raw scan range data in polar form; (**b**) ICP scan matching; (**c**) plotting of matched scans in Cartesian form.

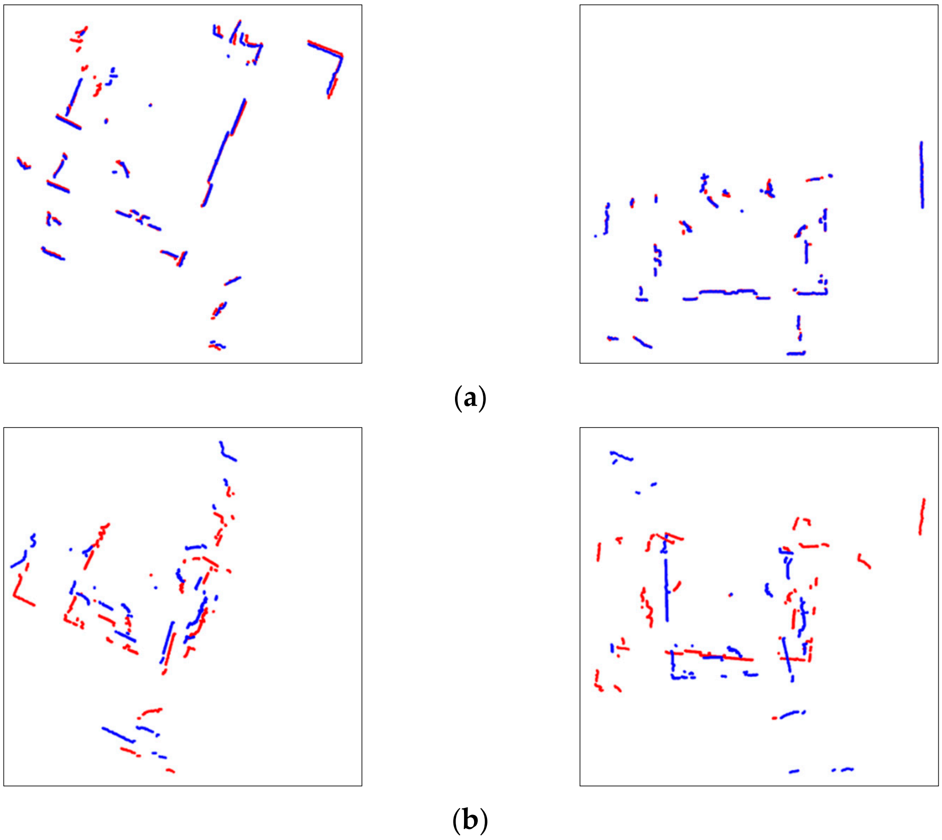


(**a**) (**b**)

(**c**) (**d**)

Figure 7. Conversion of scan pairs into images before preprocessing (Red: previous scan, blue: current scan). The areas where scans are plotted are not fixed. (**a**) t = t_1; (**b**) t = t_2; (**c**) t = t_3; (**d**) t = t_4.

For the preprocessing of plotted images, the unnecessary edge parts of scales are eliminated, and the sizes are reduced by about 1/16, to a final total of 256 × 256 pixels. The images are reduced in size for fast CNN model training while maintaining the features of scan matching. The processed images are automatically saved, then classified into binary class labels of normal and fault, as shown in Figure 8. The image generation and preprocessing were performed using matplotlib, a python visualization tool, which has a much shorter processing time than MATLAB.



(**a**)

(**b**)

Figure 8. Classification of scan matching images after preprocessing (Red: previous scan, blue: current scan). (**a**) Normal; (**b**) fault.

3.3. Fault Detection

The CNN model used to perform the fault detection of scan matching has an architecture shown in Figure 9. For the model to distinguish two scans in different colors, the input images have 256 × 256 pixels with three dimensions of RGB. The raw pixel values of 0 to 255 are normalized in values between 0 and 1. For the feature extraction of images, five 2D convolutional layers are composed of the activation function ReLU (Rectified Linear Unit). Since the training images consist of primarily white backgrounds and limited areas of scan pairs, max-pooling layers are added for each convolutional layer to take meaningful values in feature maps. In addition, a quarter of neurons are dropped out after each max pooling to prevent over-fitting. Fully connected layers to flatten the extracted features follow the convolutional layers, resulting in the likelihood of class labels with the activation function Softmax; thereby, the scan matching can be found to be normal or faulty. For training and validation, binary_crossentropy loss for binary classification and Adam optimizer are utilized. The training epochs are set to 17, since the iteration number was sufficient to train images. Too much repetition of training may cause an over-fitted model. The batch size is set to 32 for the fast training; this value is not too large to affect the training accuracy. Both the training and validation loss desirably decrease during the training repetitions, as shown in Figure 10.

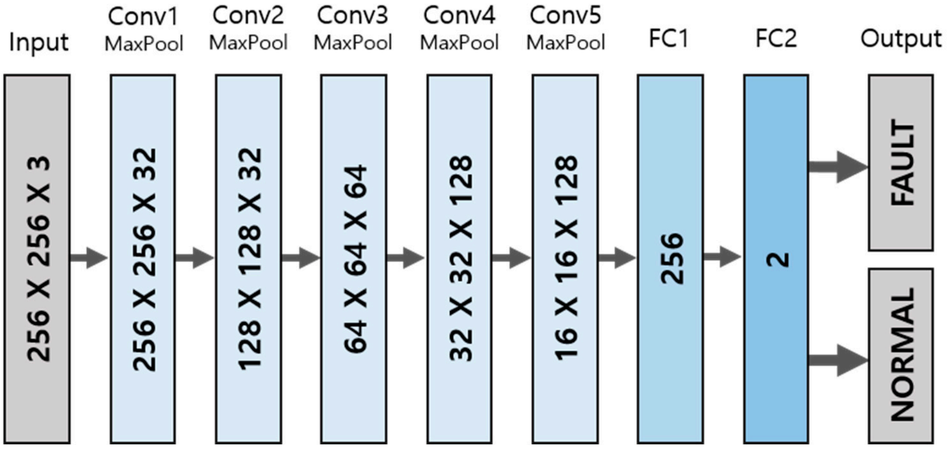


Figure 9. The architecture of CNN model for fault detection.

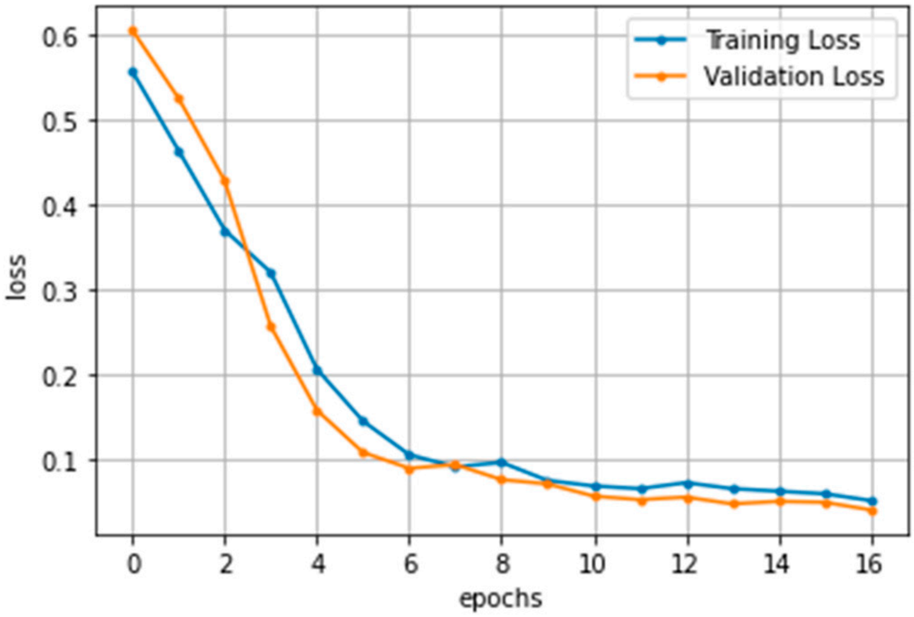


Figure 10. Training and validation loss of CNN model.

4. Experiments and Results

The environments for the experiments are set with multiple dynamic scenarios for an adaptive CNN model, as shown in Figure 11. To show the feature of each environment clearly, the people have been excluded from the figure. Environment 1 is a lobby with dynamic objects and clear landmarks. Landmarks with features are advantageous for scan matching, but the dynamic objects disturb the sensor data. Environment 2 is a narrow corridor with few features. This generally degrades the performance of scan matching. Environment 3 is an indoor environment with mirrors (lifts) and windows; this makes the range values from the sensor unreliable. Environment 4 is a wide hall; its structure partially exceeds the detection distance of LiDAR. In addition to these challenging conditions, Environments 1, 2, 3, and 4 were stuffed with dozens of people. Lastly, the Willow Garage dataset [31] was used as Environment 5. The environment is an indoor office with a few moving objects. For precise validation of the performance, the data collected in environment 5 were not used for training, but only to test the trained model.

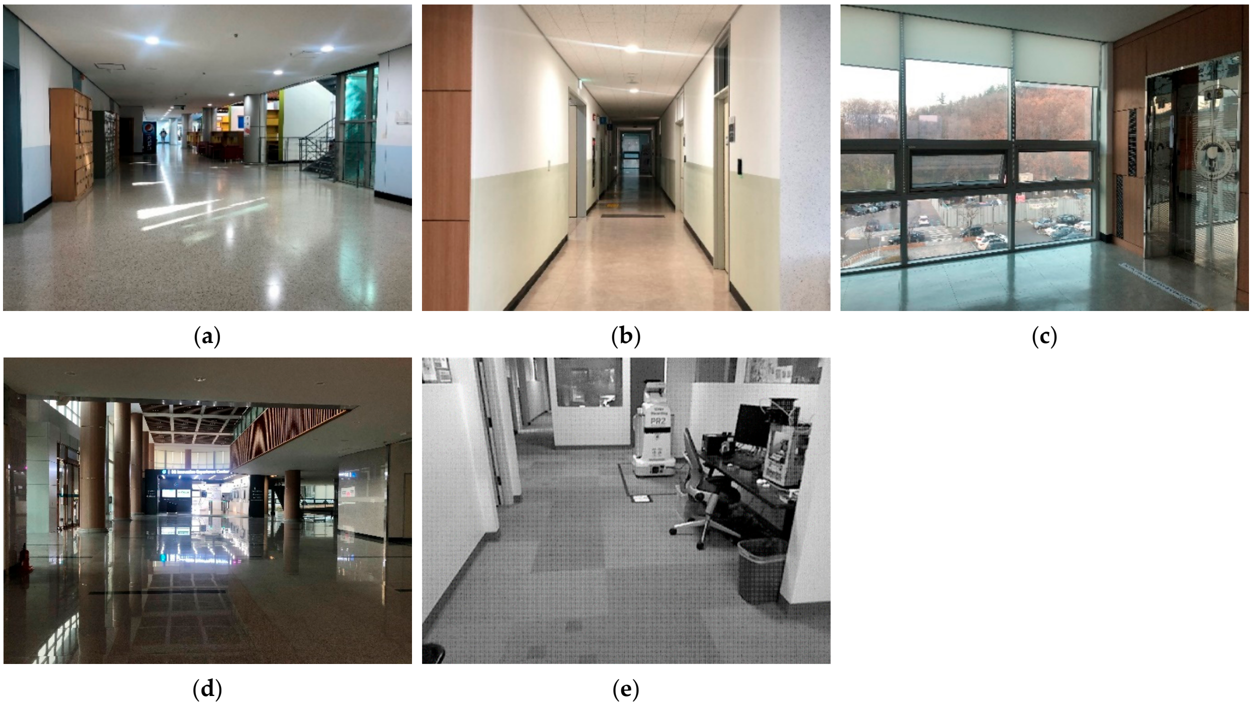

(**a**) (**b**) (**c**)
(**d**) (**e**)

Figure 11. Experimental environments. (**a**) Environment 1; (**b**) Environment 2; (**c**) Environment 3; (**d**) Environment 4; and (**e**) Environment 5.

The mobile robot used for experiments is Omorobot R1, which has a maximum speed of 1.2 m/s and an odometer for the control information, as shown in Figure 12. For the detection of environments, RPLidar A3 is used with a maximum detection distance of 25 m, taking 1440 points for each scan with 360° of FoV (Field of View). With the sensor with high angular resolution (0.25°), acquiring large amounts of data is possible. The Willow Garage dataset has exceptive data, of which laser scans have 1040 points with 270° of FoV.

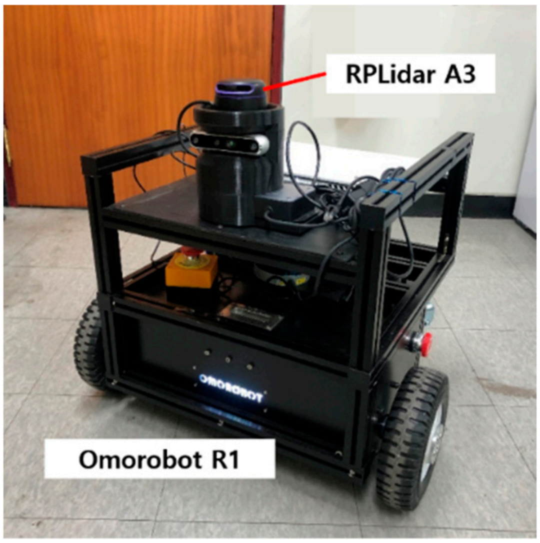
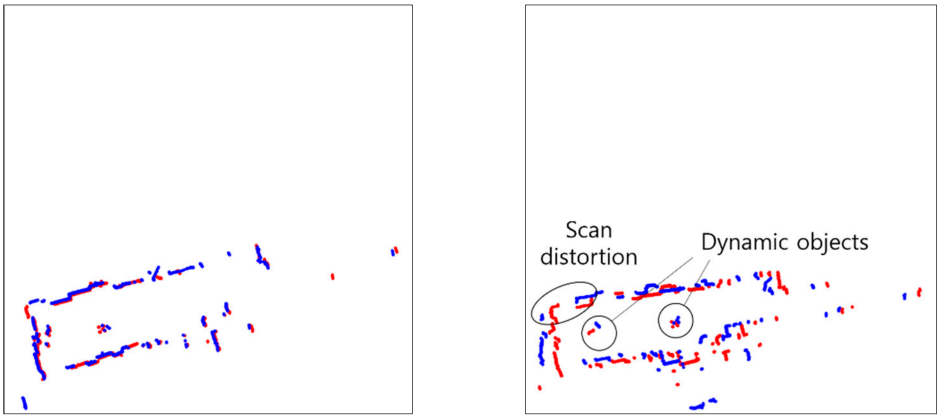


Figure 12. Robot and sensor used for experiments.

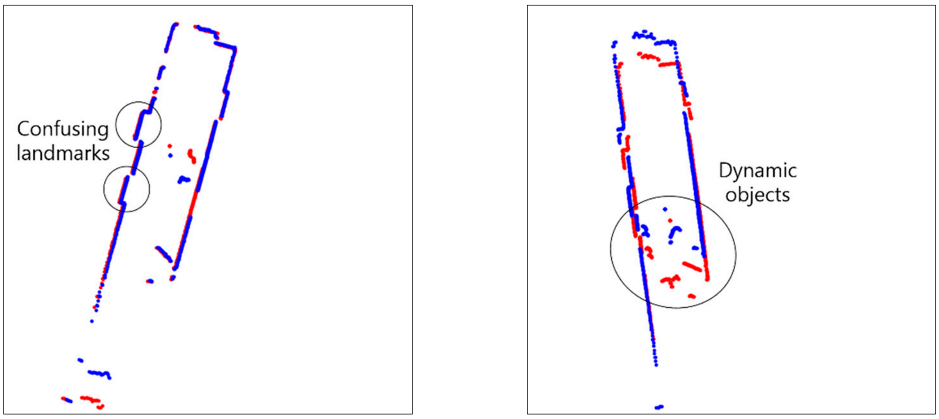
4.1. Scan Matching

The results of scan matching in Environments 1, 2, 3, 4, and 5 are shown in Figures 13–17. In Environment 1, despite landmarks that provide good features for scan matching, dynamic objects disturb the detection of LiDAR. As a result, scan matching occasionally fails. Scan matching in Environment 2 is more challenging than in Environment 1, which has few landmarks except the walls of the corridor. This is especially true when the mobile robot moves in a straight line without rotation, as the consecutive scans from the motion still have large overlapping areas of walls before scan matching. That is, the structural features which are straight and continuous can stop the iterations of scan matching, even though the scans are not yet adequately matched. When objects are added to this situation, the performance of scan matching deteriorates further. Environment 3 shows the faults of scan matching caused by both dynamic objects and the discouraging factors of sensor detection. The mirrors (lifts) reflect the sensor light; thus, the scan points representing the structure are scattered. Likewise, windows allow light to pass through them; therefore, the sensor detection becomes unreliable and the environment map highly deteriorates. In Environment 4, the scan points on the parts which are out of the range of the sensor are received as infinite values, and the values are discarded for scan matching. This results in small matching areas, and the areas are further reduced by dynamic objects, which hide the landmarks in the environment from LiDAR detection. Environment 5, the public Willow Garage dataset, has data with a smaller detection angle than the others, and many objects hide a large part of its original structure.



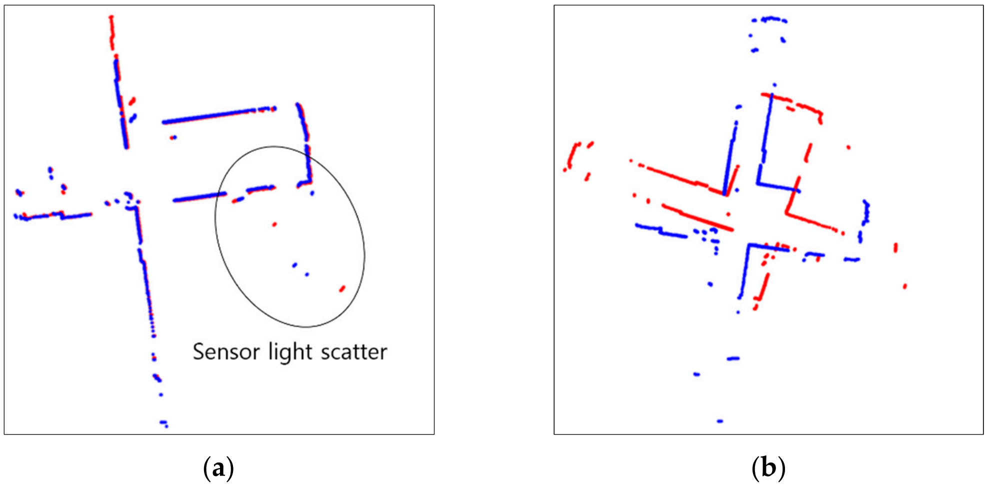
(**a**) (**b**)

Figure 13. Scan matching images in environment 1 (Red: Previous scan, blue: current scan). (**a**) Normal matching images; (**b**) faulty matching images.



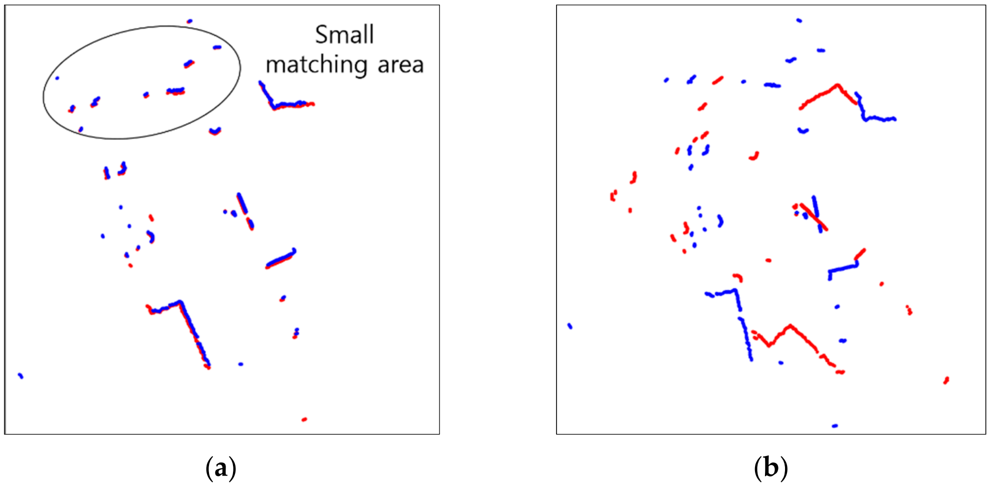
(**a**) (**b**)

Figure 14. Scan matching images in environment 2 (Red: Previous scan, blue: current scan). (**a**) Normal matching images; (**b**) faulty matching images.



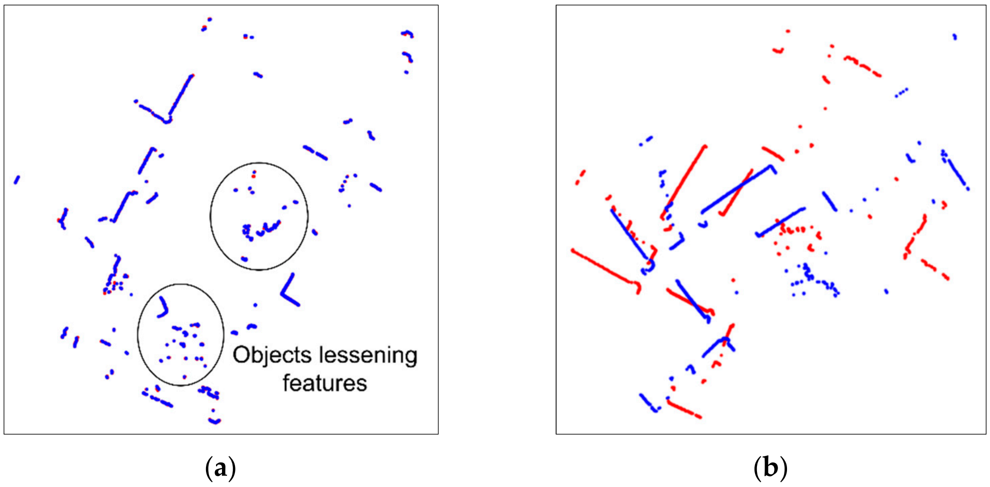
(**a**) (**b**)

Figure 15. Scan matching images in environment 3 (Red: Previous scan, blue: current scan). (**a**) Normal matching images; (**b**) faulty matching images.



(**a**) (**b**)

Figure 16. Scan matching images in environment 4 (Red: Previous scan, blue: current scan). (**a**) Normal matching images; (**b**) faulty matching images.



(**a**) (**b**)

Figure 17. Scan matching images in environment 5 (Red: Previous scan, blue: current scan). (**a**) Normal matching images; (**b**) faulty matching images.

4.2. Fault Detection

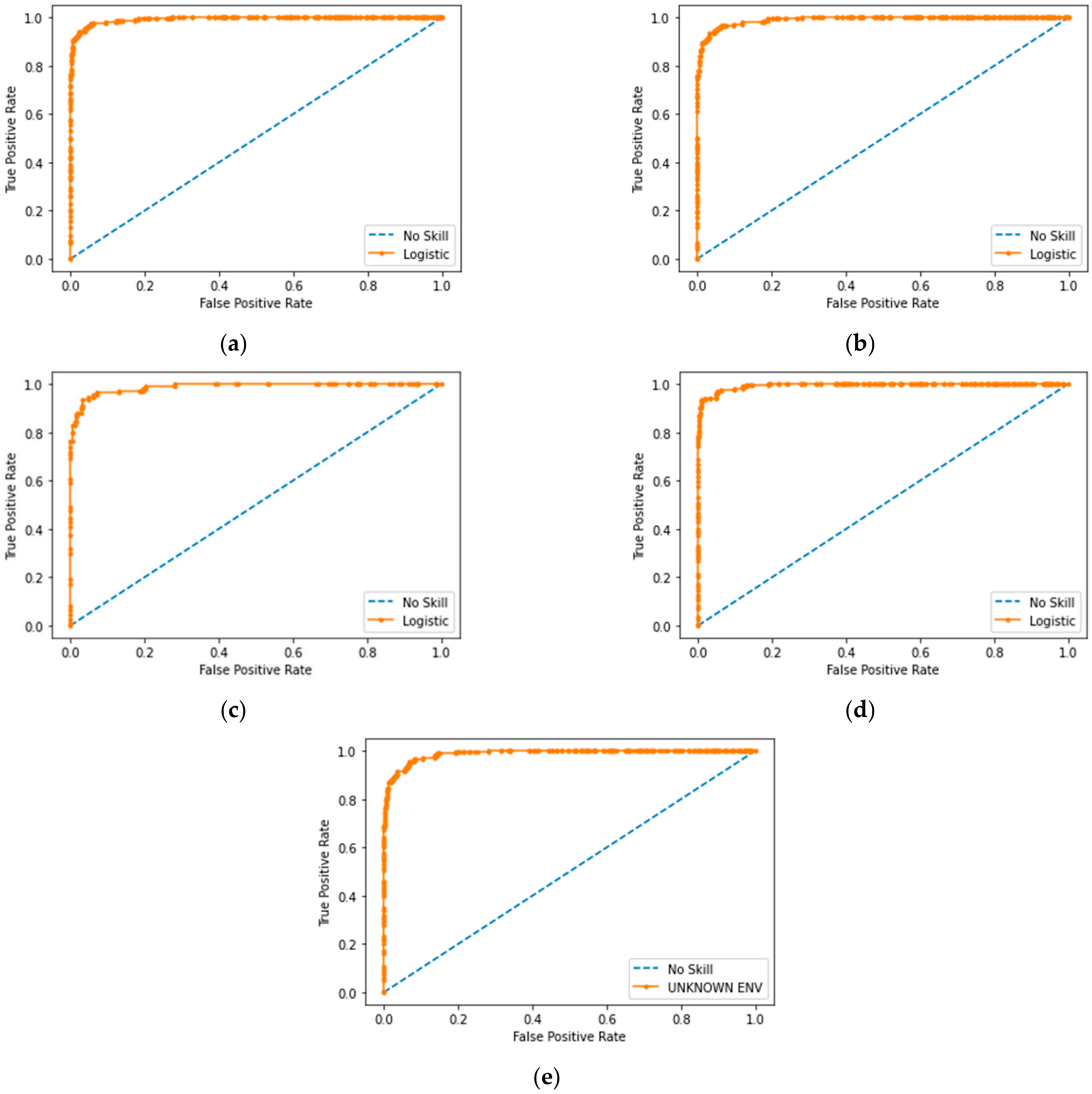
For training and testing, among 20,830 images in total, 13,254 normal images and 7576 faulty images are used at a ratio of three (training) to one (testing). For the validation of the prediction, 2969 normal images and 1283 faulty images of 4252 images are used. Data from all environments described in 4.1 are used for the training and testing, and the validations of label prediction are conducted using the total data of all environments and the respective data of each environment. Table 2 shows the confusion matrix for prediction results of fault detection in various environments.

Environment 1 shows the highest accuracy in detecting faults of 99.6%. This is because the images in this environment more clearly show whether the two consecutive scans are aligned or not, compared to other environments. For the fault detection of the trained model, the mismatch of scans results in more red points of the previous scan in an image. That is, since the mismatched red points in Environment 1 are clearly shown in faulty scan-matching images, the fault detection is successful. Comparatively, in Environment 2, the fault detection has 99.3% accuracy, which is the lowest accuracy. Because the faults of scan matching in the environment are mostly linearly unaligned, the current scan in blue covers the most of the previous scan in red, but the two scans are not yet matched. These faults less clearly show the unconformity of the scans on images than faults in other environments, making the fault detection of the CNN model confusing. However, these errors are reduced by feeding a large number of linearly faulty images; thereby, the trained model is still robust in its predictions. Environment 3 shows good results, except for four incorrectly labeled predictions among 998 images, resulting in 99.5% accuracy. Due to the windows and mirrors, as well as dynamic objects, the red previous scan points partially decrease. The scan matching errors in this situation are wrongly predicted, but they are handled by training data from different trajectories and environments. Likewise, in Environment 4, the scan points of two laser scans are reduced by its spacious structure and objects. The images of this matching error contain fewer features, which demonstrates the quality of scan matching. Still, the trained model predicts it with an accuracy of 99.5%. Only a few slightly unaligned scan images are incorrectly predicted. Environment 5 is used only to test the trained model, and is never used for the fault detection training. Even though the images reduce features as a result of the objects, 98.9% accuracy is achieved due to the large dataset taken from Environments 1–4.

The results are also presented as ROC (receiver operating characteristic) curves in Figure 18. As shown in the graphs, the trained CNN model accurately detects the faults of scan matching, regardless of the environment. The proposed method is adaptive for various environments with their own features. In addition, we compare the proposed method to the original ICP in terms of scan matching error, as shown in Figure 19. The error is defined as the Euclidian distance between the centers of mass of two matched scans, as described earlier in Section 2.2. The smaller the error, the more accurate the scan matching. The inaccuracy in matchings decreases by 75% using our CNN method.

Table 2. Predictions of the proposed method.

	TP	FN	FP	TN	Recall	Precision	Accuracy
All environments	2946	23	6	1277	99.2%	99.7%	99.3%
Environment 1	978	3	2	274	99.6%	99.7%	99.6%
Environment 2	745	4	3	352	99.4%	99.4%	99.3%
Environment 3	700	3	1	294	99.5%	99.8%	99.5%
Environment 4	534	2	2	355	99.6%	99.6%	99.5%
Environment 5	702	7	4	360	98.0%	99.4%	98.9%



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 18. ROC curve of prediction in (**a**) Environment 1; (**b**) Environment 2; (**c**) Environment 3; (**d**) Environment 4 and (**e**) Environment 5.

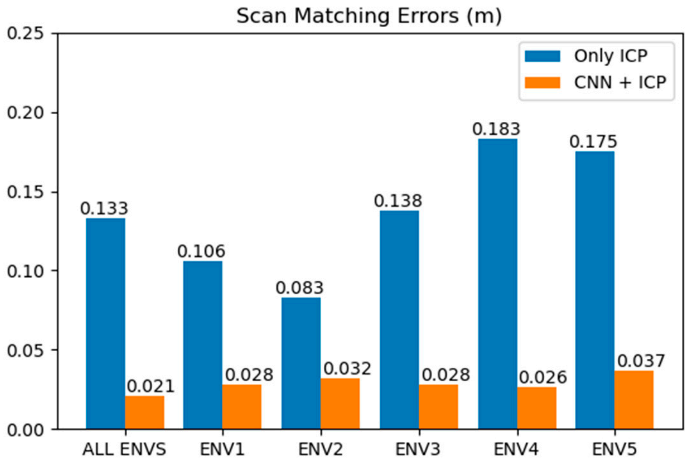


Figure 19. Comparison of methods in terms of scan matching error. The error is Euclidian distance of matched points, which represents the misalignment of scans.

5. Conclusions

We have proposed an applicable CNN model for the fault detection of scan matching to complement SLAM in dynamic environments. For fault detection, as soon as the raw scan data are received, scan matching, the conversion of matched scans into images, and the fault detection by the trained CNN model are performed sequentially at each time step. The scan data for the training of the model are acquired under various dynamic scenarios to compare the model's performances in real-world environments. As the training data for a CNN model, the form of images that represent consecutive scan data in Cartesian coordinates are used for the first time and demonstrate efficiency in the model training. The validation of the trained model shows accuracy of over 99% in the detection of faults of scan matching in every experimental environment.

For future work, the method can be extended to various SLAM frameworks in real time. Depending on the methods of scan matching, mapping, and robot pose estimation of different SLAM algorithms, the steps of scan matching and the application of fault detection in the proposed method can be adapted to enhance existing SLAM and scan registration systems.

Author Contributions: All authors contributed the present paper with the same effort in finding available literatures and writing the paper. H.J. designed and implemented the proposed method. H.L. described the system and problems. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2021R1F1A1064358), and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2021-0-02087, Application Techniques of the 3D Semantic Scene Reconfiguration for Social Interaction-based Multi-Robot Autonomous Navigation), and in part by the Government-wide R&D Fund for Infections Disease Research (GFID), funded by the Ministry of the Interior and Safety, Republic of Korea (grant number: 20014854).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thrun, S.; Fox, D.; Burgard, W.; Dellaert, F. Robust Monte Carlo localization for mobile robots. *Artif. Intell.* **2001**, *128*, 99–141. [CrossRef]
2. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
3. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [CrossRef]
4. Panchpor, A.A.; Shue, S.; Conrad, J.M. A survey of methods for mobile robot localization and mapping in dynamic indoor environments. In Proceedings of the Conference on Signal Processing And Communication Engineering Systems (SPACES), Vijayawada, India, 4–5 January 2018; pp. 138–144. [CrossRef]
5. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and Structure from Motion in dynamic environments: A survey. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef]
6. Mertz, C.; Navarro-Serment, L.E.; MacLachlan, R.; Rybski, P.; Steinfeld, A.; Suppé, A.; Urmson, C.; Vandapel, N.; Hebert, M.; Thorpe, C.; et al. Moving object detection with laser scanners. *J. Field Robot.* **2012**, *30*, 17–43. [CrossRef]
7. Woo, J.; Jeong, H.; Lee, H. Comparison and Analysis of LIDAR-based SLAM Frameworks in Dynamic Environments with Moving Objects. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Gangwon, Republic of Korea, 1–3 November 2021. [CrossRef]
8. Vidal, F.S.; Barcelos, A.D.O.P.; Rosa, P.F.F. SLAM solution based on particle filter with outliers filtering in dynamic environments. In Proceedings of the International Symposium on Industrial Electronics (ISIE), Buzios, Brazil, 3–5 June 2015. [CrossRef]
9. Sun, D.; Geisser, F.; Nebel, B. Towards effective localization in dynamic environments. In Proceedings of the 2016 International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016. [CrossRef]
10. Edwin, O. M3RSM: Many-to-many multi-resolution scan matching. In Proceedings of the International Conference on Robotics and Automation (ICRA), Washington, DC, USA, 26–30 May 2015; pp. 5815–5821.
11. Hudson, M.S.B.; Esther, L.C. LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method. *Neurocomputing* **2021**, *455*, 97–110.

12. Bahraini, M.S.; Rad, A.B.; Bozorg, M. SLAM in Dynamic Environments: A Deep Learning Approach for Moving Object Tracking Using ML-RANSAC Algorithm. *Sensors* **2019**, *19*, 3699. [CrossRef] [PubMed]
13. Rong, K.; Jieqi, S.; Xueming, L.; Yang, L.; Xiao, L. DF-SLAM: A Deep-Learning Enhanced Visual SLAM System based on Deep Local Features. In Proceedings of the 2019 IEEE Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
14. Li, J.; Zhan, H.; Chen, B.M.; Reid, I.; Lee, G.H. Deep learning for 2D scan matching and loop closure. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017. [CrossRef]
15. Spampinato, G.; Bruna, A.; Guarneri, I.; Giacalone, D. Deep Learning Localization with 2D Range Scanner. In Proceedings of the International Conference on Automation, Robotics and Applications (ICARA), Prague, Czech Republic, 4–6 February 2021; pp. 206–210. [CrossRef]
16. Michelle, V.; Cyril, J.; Arnaud, F. An LSTM Network for Real-Time Odometry Estimation. *arXiv* **2019**, arXiv:1902.08536. [CrossRef]
17. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [CrossRef]
18. Murphy, K.; Russell, S. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Sequential Monte Carlo Methods in Practice*; Springer: New York, NY, USA, 2001; pp. 499–515.
19. Yoshitaka, H.; Hirohiko, K.; Akihisa, O.; Shin'Ichi, Y. Mobile Robot Localization and Mapping by Scan Matching using Laser Reflection Intensity of the SOKUIKI Sensor. In Proceedings of the IECON 2006—32nd Annual Conference on IEEE Industrial Electronics, Paris, France, 6–10 November 2006. [CrossRef]
20. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
21. François, P.; Francis, C.; Roland, S. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. Trends Robot.* **2015**, *4*, 1–104.
22. Wang, Y.-T.; Peng, C.-C.; Ravankar, A.A.; Ravankar, A. A Single LiDAR-Based Feature Fusion Indoor Localization Algorithm. *Sensors* **2018**, *18*, 1294. [CrossRef] [PubMed]
23. Zhang, J.; Yao, Y.; Deng, B. Fast and Robust Iterative Closest Point. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3450–3466. [CrossRef] [PubMed]
24. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001. [CrossRef]
25. Diosi, A.; Kleeman, L. Laser scan matching in polar coordinates with application to SLAM. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Edmonton, AB, Canada, 2–6 August 2005. [CrossRef]
26. Diosi, A.; Kleeman, L. Fast Laser Scan Matching using Polar Coordinates. *Int. J. Robot. Res.* **2007**, *26*, 1125–1153. [CrossRef]
27. Friedman, C.; Chopra, I.; Rand, O. Perimeter-Based Polar Scan Matching (PB-PSM) for 2D Laser Odometry. *J. Intell. Robot. Syst.* **2014**, *80*, 231–254. [CrossRef]
28. Biber, P. The Normal Distributions Transform: A New Approach to Laser Scan Matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003.
29. Magnusson, M. The Three-Dimensional Normal-Distributions Transform: An Efficient Representation for Registration, Surface Analysis, and Loop Detection. Ph.D. Thesis, Örebro University, Örebro, Sweden, 2009.
30. Zhou, Z.; Zhao, C.; Adolfsson, D.; Su, S.; Gao, Y.; Duckett, T.; Sun, L. NDT-Transformer: Large-Scale 3D Point Cloud Localisation using the Normal Distribution Transform Representation. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021. [CrossRef]
31. Mason, J.; Marthi, B. An object-based semantic world model for long-term change detection and semantic querying. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.