International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

End-to-end navigation for robots using learning from demonstration with 2D-laser range data

Md. Siddik, Abdullah Al Niloy *, Md Nur Hossain, Fahim Afsar Raihan, Md. Sagor Ahmed and Md Rabiul Islam

Department of Electrical and Electronic Engineering, Independent University, Dhaka 1212, Bangladesh.

International Journal of Science and Research Archive, 2026, 19(02), 202-209

Publication history: Received on 23 March 2026; revised on 04 May 2026; accepted on 06 May 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.2.0960

Abstract

One of the continuing research topics is the application of learning from demonstration to motion planning. The purpose of this research is to offer a model that is capable of learning the complex mapping from raw 2D-laser range discoveries and a target location to the necessary steering signals for the robot. This work introduces the first approach that, to the best of our knowledge, trains a target-oriented end to-end navigation model for a robotic platform. The expert demonstrations that are generated in simulation with an existing motion planner serve as the foundation for the supervised model training sessions. We provide evidence that the navigation model that has been taught may be directly transferred to environments that have not been encountered before, both in the virtual world and, in the real world. It can safely lead the robot through areas that are packed with obstacles to reach the targets that have been set. A comprehensive quantitative and qualitative evaluation of the motion planner is presented, and a comparison is made between it and a grid-based global strategy, which was tested in simulation as well as in trials conducted in the real world.

Keywords: Autonomous Ground Robots; Data-DRIVEN Approach; Complex Mapping

1. Introduction

Making robots function in a manner that is desired by human operators is one of the most significant issues in the field of robotics. Even though objectives such as a safe distance to obstacles are obvious to the actual human operator, they generally require time-demanding manual tweaking to ensure that the robot moves in the manner that is required [1]. Furthermore, motion planning systems in a traditional manner necessitate several successive steps of the preprocessing of data, which are often separated from one another. The input from the sensors must be preprocessed, and prospective items must be identified for the planner to be able to respond appropriately at a later stage [2]. A strategy that goes in the opposite direction is shown in this work. The goal of this approach is to reduce the amount of hand-tuning that is required for many procedures to obtain the target navigation performance [3]. The end-to-end motion planner that is driven by data. Within a specific virtual training environment, the robot is given demonstrations from trained professionals on how to navigate using the environment. An operator of a robot can demonstrate to the robot the behavior and navigation technique that they want it to exhibit in this manner [4]. In contrast, the multiple-layer mapping-based systems described in our method don't necessitate the use of a universal map to traverse. The architecture of the technique is founded on the notion that our machine learning-based approach may perform similarly, even in situations that have not been experienced before, if it is taught an awareness of the surroundings and the navigational features of an operator [5]. The technique is not restricted to any one kind of environment based on its design. On the other hand, the focus of our analysis in this work is precisely on navigation in surroundings that are static.

* Corresponding author: Abdullah Al Niloy

It is necessary to have a model that is capable of capturing the required properties during the process of local motion planning. Among the many different techniques of machine learning, those that are based on deep neural networks (DNN) have the greatest potential to model complicated dependencies [6].

Figure 1 Menitoned robotic platform is capable of securely maneuvering through a maze-like environment by relying solely on laser (local) and target data (top left). The ultimate traversed-like trajectory is depicted on the mentioned map (right). The robot's visual representation is solely intended for imaging reasons and does not serve as an input

In a variety of applications, to mention a few, they have demonstrated their capability of comprehending complicated physical scenes and extracting features from those scenes. This approach is used to train the motion planner. We use a front 270-degree laser finder as the sole robotic sensor [7]. Additionally, the achievement of the taught end-to-end motion model is evaluated on an actual platform in addition to being assessed in simulation. When conducting experiments in environments that have not been observed before, it is of utmost importance to conduct experiments. Therefore, we put the robot through its paces by putting it through its paces in situations that were unfamiliar to us. This scenario is organized in the following manner for the remaining sections: Presented in Section II is an outline of the study that is linked to this topic. The problem is formulated in Section III, and our approach is presented in this section. Following the presentation of the experiments and the outcomes of those experiments in Section IV, we will then move on to Section V to discuss the outcomes and Section VI to conclude.

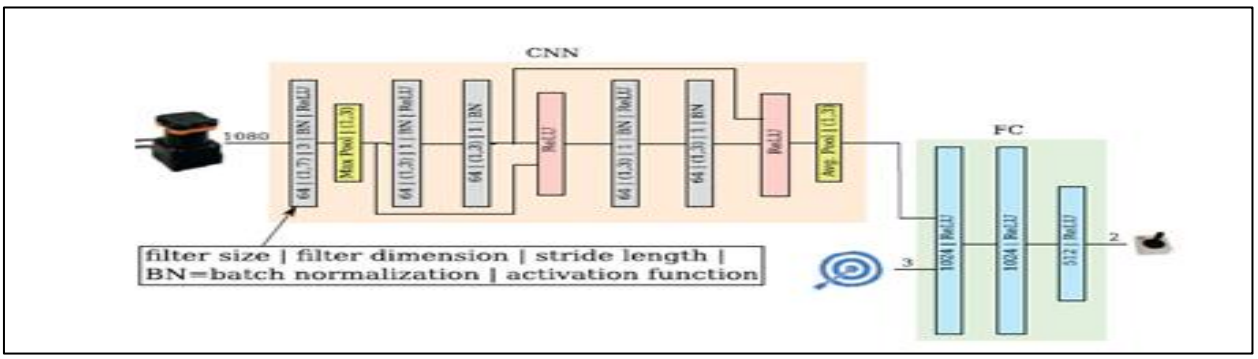

Figure 2 The architecture of the Convolutional Neural Network (CNN). The laser dataset is analyzed by the component of convolution, which is composed of 2 residual blocks as described in reference [18]. The FC component of the network combines the new features with the target. The dimensions of the output & input of the entire model are displayed on the connectors. All parameters are subject to L1 regularisation

2. Approach

2.1. Problem Formulation

The human species has remarkable capabilities in terms of sensing their surroundings, gleaning significant information from that perception, and making reasonable judgments based on the knowledge obtained [8]. On the other hand, they can rely on a vast information base that they have acquired during their entire life to make decisions. Robots must overcome several obstacles for them to be able to make the appropriate judgments, which in our case means moving a robot to the required goal position (including heading) [9]. To begin, the pertinent information must be extracted from

the data collected by the sensors. A model that should be taken needs to be found utilizing this knowledge, which brings us to the second step. During the deployment phase, this model must be utilized to arrive at the appropriate decisions. We offer a methodology that computes optimal steering commands, in contrast to the common practice of solving both tasks independently by many different methods.

$$u = f_\theta(y, g) \qquad (1)$$

With the help of demonstrations, we are attempting to locate a systematic function that transfers goal information and the vector of the sensor's dataset to the particular commands that are wanted. A parameter vector is used to implement the parameters of this function. Function parameters provide the most accurate explanation of a set of training data while we are engaged in supervised training [10].

2.2. End-to-end Model

There is the potential for an arbitrarily complicated model to be produced as a consequence of the interaction between steering commands and given data. DNNs and CNNs are well known among the various techniques of machine learning because of their extraordinary ability. For the sake of this work, we shall be employing a CNN as the representation. To ensure that the whole processing pipeline, including the extraction of information and the identification of the appropriate steering commands, is covered by a single model. The laser data is processed by a CNN to recover spatial scene understanding at tributes. This is done before the FC layers continue to process the model [11]. As can be seen in Fig. 2, the demonstrations of the neural model are not complicated.

Throughout the entirety of this article, we will be looking into two different variants of the model. For the first version, which is known as CNN smallFC, the size of the 3 FC layers are (All are the having values of 256). However, for the nest version, which is known as CNN bigFC, the dimensions of the FC layers are increased to (1024, 1024, 512). There is no change to the convolutional component of either of the networks, as demonstrated in Figure 2. TensorFlow, a framework developed by Google, serves as the foundation for our neural network model implementation [12].

2.3. Model Training

Ultimately, the purpose of the proposed approach that has been provided is to achieve the capability of learning a driving feature that is exhibited to a robot by an experienced operator who is supervised [13]. We resort to the actual simulation, to cancel the load of human-driving datasets on a large scale. Within the realm of robotic applications, where the acquisition of data can be somewhat costly, this is an extremely valuable characteristic [14]. The velocity command, denoted by, u = (v, w) allows for both rotational and translational velocity to be included in this context. A randomization process is performed on the tuples before they are used for training to minimize the temporal correlations that are present in the training data. The Adam optimizer is utilized in conjunction with mini-batch training to carry out the optimization. The main loss function to each supervised step k can be expressed as follows:

$$J_k(\Gamma_B) = \frac{1}{N_B} \sum_{j=i}^{i+N_B} \left| F_{\theta_k}(y_j, g_j) - u_{exp,j} \right| \qquad (2)$$

The model at the current step is denoted by the F_{θ_k} , which is represented by the letter F. Backpropagation is a technique that can be utilized to compute the gradient of this cost-function to the actual parameters of the model.

2.4. Motion Planner Deployment

When compared to alternative methods, neural network models have several advantages, one of which is that their query time during testing is predictable. However, the complexity of a deep neural network (DNN) is not impacted by the environment in which the robot operates, and the query does not change [15]. The displayed neural network model is responsible for computing the steering directives on a frame-by-frame basis. To take into consideration the prior inputs and outputs, neither the internal nor the external memory is utilized.

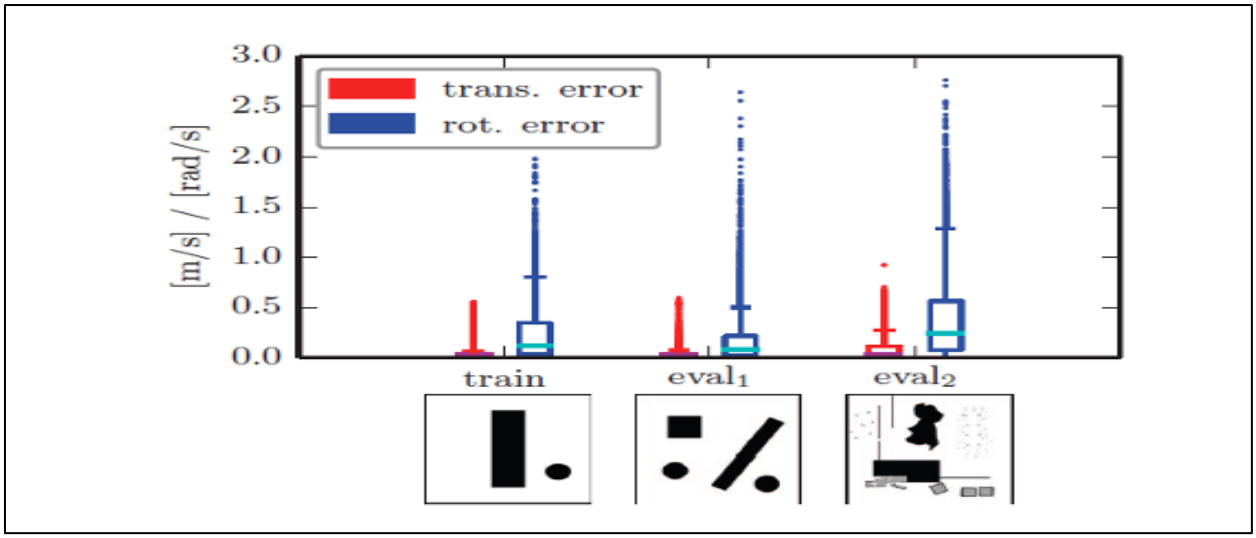

Figure 3 The frame-by-frame error between the deep planner and ROS (expert) is provided. The data was previously unused for training. The 3 miniature figures depict the maps upon which the examination was carried out. Maps are best observed by utilizing the zoom function on a computer screen

3. Experiments

This section talks about the experiments that were carried out and how they were evaluated. To begin, the robotic platform is presented to the audience. In every experiment, it serves as a model for the simulations as well as for the tests that are conducted in the real world. In the second step, the generation of training data is discussed, and then four experiments for evaluation are described, two of which are conducted in simulation and two of which are conducted on the actual platform [16].

3.1. Training Data

At the universal level, we make use of a Dijkstra planner that is grid-based, while at the local level, we make use of a DWA planner. A dynamic simulator is utilized in the utilization of Stage 2D [17]. During the process of generating training datasets, the robot will drive to target places on the training map that are 10 meters by 10 meters in size (train). It is guaranteed that the selected target placements will not cause any collisions, meaning that they will be located outside of any potential barriers. Both Fig. 3 and Fig. 4 display the original train map in the column that is located to the left of the column that is now being displayed. When we were doing our tests in the actual world, we utilized fused data from both the map and the eval2 map. The latter consists of debris and other obstacles that the robot would encounter during tests that are conducted in the actual environment. With 2.1 million and 2.2 million input/output tuples, respectively, we were able to create trajectories of 4000 trajectories for the eval2 map and 6000 for the train map. The training of the model on a GPU3 equipped with an Nvidia GeForce GTX 980 Ti takes around eight hours when two million training steps are carried out [18]. Four factors led to the choice to only employ simulation data for training purposes, which was generated using a planner.

3.2. Frame-by-fame Evaluation

The evaluation error of the CNN model about the computed steering directives is the primary focus of the experiment. According to what was stated in Section IV-B, the proper model was trained using just the sample data that were taken from the training map [19]. In the study, the CNN small FC scheme is utilized to observe the results. During the testing phase, the query must be executed solely by the central processing unit (CPU) to be independent of a graphics processing unit (GPU). On an Intel i7-4810 MQ processor operating at 2.8 GHz, the typical query time for a model is 4.3 milliseconds [20]. To conduct the evaluation, we used the expert motion planner to drive to thirty different random target positions on each of the three maps, which were train, eval1, and eval2. This allowed us to generate input/output tuples for the evaluation. This information, which was not visible during the training process, is used to compute the error that exists between the rotational and translational steering directives of the motion planner and the expert planner for every i/o tuple. Taking into consideration the error presented in Fig. 3, it is evident that the train map exhibits the least amount of evaluation error. The big outliers of the velocity command seem to occur more frequently while spinning on the spot, according to our findings. For instance, about the rotational velocity inaccuracy, moving 180

degrees to either the left or the right has significant importance, although the actual behavior of the robot is only marginally affected by this action. The structure of the eval1 map is distinct from that of the train one; yet, the main obstacles take on an outline that is comparable to the train map. The error of the evaluation, particularly about the velocity command, is increased as a result of this condition. Transferring our attention to the eval2 map, we find that the rotational and translational components of the error both grow even higher [21]. Our expectations have been confirmed by this result. The model can transfer the knowledge of navigation and scene interpretation that it has obtained in one environment to another environment. On the other hand, if there is a significant difference in the environment structure, it is possible that the correct scene understanding will not be provided. This will result in a greater disparity between the steering orders issued by the expert planner and those issued by the deep planner.

3.3. Trajectory Comparison

As part of this study, we intend to evaluate when it is utilized on our robotic platform in the capacity of a motion planner. The model of CNN and the training datasets are identical to those used in the experiment that came before it [22]. Creating missions with fixed goal placements is something that is done for both the training environment and the evaluation environment. This is in contrast to the planner, which only gets the particular relative target (represented by the red colored dots in Figure 4. The ROS planner, on the other hand, has universal knowledge about mapping as it learned when there was trained matter. As a result of the deterministic nature of the simulation, a solo mission for every planner is demonstrated. In the subsequent experiment that will take place in the real world, the reproducibility of the approach will be evaluated.

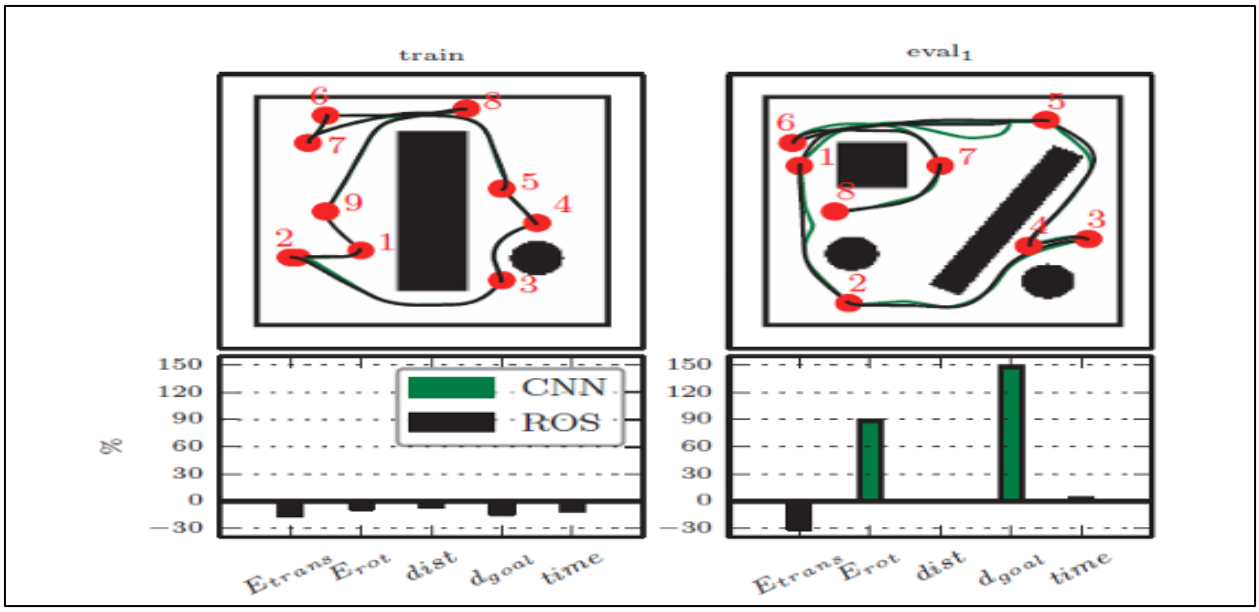

Figure 4 Comparative analysis of the performance between the deep planner and the ROS (expert). Assessing the performance of the trained model on the train (left) and evall map (right). Top: A contrast is created at the trajectory level, with target sites highlighted in red. The ROS/deep planner's relative inaccuracy for assessment metrics comprises the ultimate distance to the target (goal), translational energy (Etrans), traveled distance (dist), rotational energy (Erot), and trip time. The green bars reflect the deep planner's relative error compared to the ROS expert, while the black bars represent the inverse. Consequently, a positive (green) error indicates that the ROS planner outperforms the deep planner, whereas a negative (black) error suggests that the deep planner is superior

As an illustration of the final completed trajectories, Figure 4 (top) is presented here. Fig. 4 (bottom) illustrates the errors that all the planners have made for each of the measures that were chosen. It has been demonstrated through the findings that our comprehensive technique is capable of capturing the navigation of the operator. Even though the deep planner only makes use of local knowledge, the trajectories that are driven by both planners are consistent for the majority of the situations, particularly on the rail map [23]. In terms of the metrics that are provided, the deep planner achieves a little better performance than the expert planner. There is no connection between this and overfitting because neither the deep planner nor the ROS was trained or optimized for the measures.

3.4. Reaction to Sudden Change

The results of this final experiment provide a qualitative demonstration of the performance of the deep planner when confronted with impediments that abruptly emerge and disappear [24]. This experiment yielded the findings that are displayed in Fig. 6 as well as the video that is attached. Even though the model of the mentioned navigation solely calculates solo steering, the constant velocity path for this steering command is displayed. The robot must travel from one end of a hallway to the other end of the corridor. At the beginning of the process, the road to the objective is unobstructed (Fig. 6a). However, as the process progresses, the robot is confronted with an object that suddenly appears in its

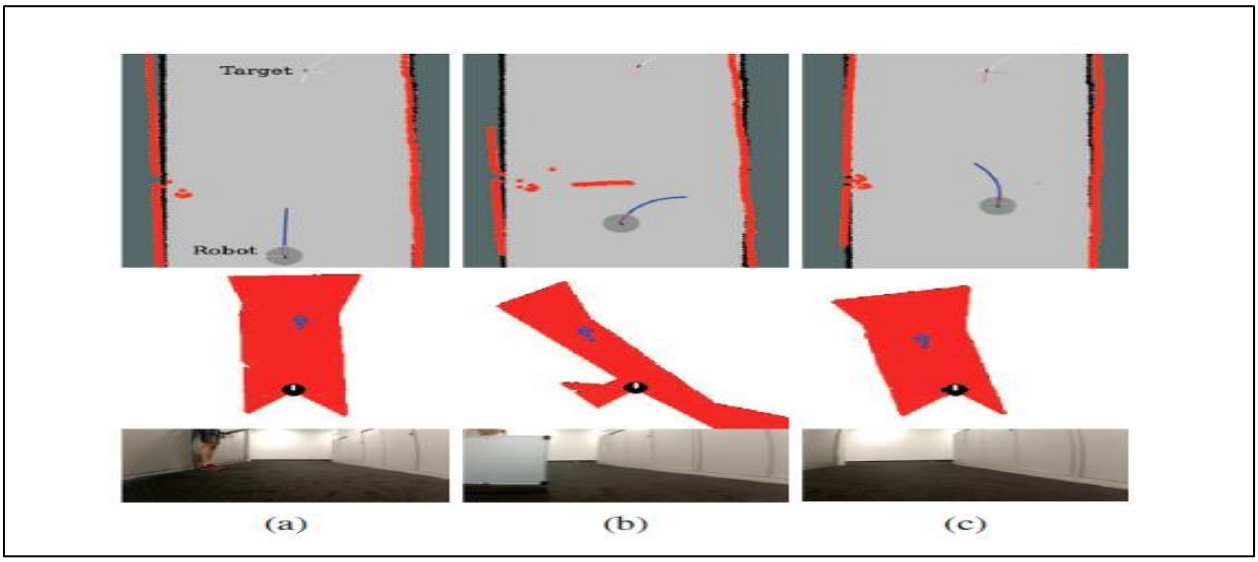

Figure 5 Response to unexpected entities. While navigating toward the desired location, the robot's trajectory is obstructed by an obstacle, but later it becomes unobstructed. The figure displays the constant velocity route in the top row, which is visualized utilizing the calculated steering and the entire setup. The mid row displays the findings of the laser range that is centered around the robot, as well as the position of the target relative to the robot. The photos in the last row, on the other hand, are solely utilized for visualization purposes

path (Fig. 6b). After being confronted by the object, the deep planner makes a sharp turn to the right. Following the removal of the obstruction, the robot adjusts its trajectory to get closer to the objective in the shortest amount of time (Fig. 6c).

4. Conclusion

Within the scope of this work, we proposed a planning strategy for a platform that is data-driven from beginning to end. Our method is efficient in evaluating the necessary steerings for a differential platform, provided that the local range of the laser results and a target location are taken into consideration. The CNN serves as the foundation for the model, which is capable of acquiring navigational methods from an experienced operator and transferring this information to a variety of different situations. We proved that a navigation model can be trained using simulation data and then deployed on a real robotic platform in a previously unknown environment. We also provided a complete evaluation of simulated and real-world tests.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] D. Wang, X. Liang, G. I. Mofack, and O. Martin-Ducup, “Individual tree extraction from terrestrial laser scanning data via graph pathing,” Forest Ecosystems, vol. 8, pp. 1–11, 2021.

[2] J. Para, J. Del Ser, A. J. Nebro, U. Zurutuza, and F. Herrera, "Ana lyze, sense, preprocess, predict, implement, and deploy (asppid): An incremental methodology based on data analytics for cost-efficiently monitoring the industry 4.0," Engineering Applications of Artificial Intelligence, vol. 82, pp. 30–43, 2019.

[3] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in 2017 ieee international conference on robotics and automation (icra). IEEE, 2017, pp. 1527 1533.

[4] F. Gul, W. Rahiman, and S. S. Nazli Alhady, "A comprehensive study for robot navigation techniques," Cogent Engineering, vol. 6, no. 1, p. 1632046, 2019.

[5] D. C. Guastella and G. Muscato, "Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review," Sensors, vol. 21, no. 1, p. 73, 2020.

[6] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," Archives of Computational Methods in Engineering, vol. 27, pp. 1071 1092, 2020.

[7] M. Bansal, B. Matei, B. Southall, J. Eledath, and H. Sawhney, "A lidar streaming architecture for mobile robotics with application to 3d structure characterization," in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 1803–1810.

[8] A. Linson, A. Clark, S. Ramamoorthy, and K. Friston, "The active inference approach to ecological perception: general information dynamics for natural and artificial embodied cognition," Frontiers in Robotics and AI, vol. 5, p. 21, 2018.

[9] H. Seraji and A. Howard, "Behavior-based robot navigation on challenging terrain: A fuzzy logic approach," IEEE transactions on robotics and automation, vol. 18, no. 3, pp. 308–321, 2002.

[10] H. Bhavsar and A. Ganatra, "A comparative study of training algorithms for supervised machine learning," International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 4, pp. 2231–2307, 2012.

[11] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al Shamma, J. Santamar´ıa, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," Journal of big Data, vol. 8, pp. 1–74, 2021.

[12] A. Gulli, A. Kapoor, and S. Pal, Deep learning with TensorFlow 2 and Keras: regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API. Packt Publishing Ltd, 2019.

[13] D. A. Pomerleau, "Knowledge-based training of artificial neural net works for autonomous robot driving," in Robot learning. Springer, 1993, pp. 19–43.

[14] E. Zereik, M. Bibuli, N. Mi˘ skovi´c, P. Ridao, and A. Pascoal, "Challenges and future trends in marine robotics," Annual Reviews in Control, vol. 46, pp. 350–368, 2018.

[15] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," IEEE access, vol. 6, pp. 24411–24432, 2018.

[16] T. Strasser, M. Stifter, F. Andr´en, and P. Palensky, "Co-simulation training platform for smart grids," IEEE Transactions on Power Systems, vol. 29, no. 4, pp. 1989–1997, 2014.

[17] A. ElNimr, M. Fagiar, and Y. Mohamed, "Two-way integration of 3d visualization and discrete event simulation for modeling mobile crane movement under dynamically changing site layout," Automation in construction, vol. 68, pp. 235–248, 2016.

[18] R. P. Wiewiora, "Rigorous construction of markov state models for con formationally selective drug design," Ph.D. dissertation, Weill Medical College of Cornell University, 2020.

[19] J. H. Uhl, S. Leyk, Y.-Y. Chiang, W. Duan, and C. A. Knoblock, "Automated extraction of human settlement patterns from historical topographic map series using weakly supervised convolutional neural networks," IEEE Access, vol. 8, pp. 6978–6996, 2019.

[20] U. Ali Ahmad, Y. Kasahara, S. Matsuda, M. Ozaki, and Y. Goto, "Automatic detection of lightning whistlers observed by the plasma wave experiment onboard the arase satellite using the opencv library," Remote sensing, vol. 11, no. 15, p. 1785, 2019.

[21] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pf¨andler, U. Angst, R. Siegwart, and J. Nieto, "An omnidirectional aerial manipulation plat form for contact-based inspection," arXiv preprint arXiv:1905.03502, 2019.

[22] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016.

[23] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis et al., "Semantic trajectories modeling and analysis," ACM Computing Surveys (CSUR), vol. 45, no. 4, pp. 1–32, 2013.

[24] M. I. Pavel, S. Y. Tan, and A. Abdullah, "Vision-based autonomous vehicle systems based on deep learning: A systematic literature review," Applied Sciences, vol. 12, no. 14, p. 6831, 2022.