

Lidar-based Human Detection for a Mobile Robot in Social Environment using Deep Learning

1st Alif Wicaksana Ramadhan
Department of Electrical Engineering
Politeknik Elektronika Negeri Surabaya (PENS)
 Surabaya, Indonesia
 alifwr@outlook.com

2nd Bima Sena Bayu Dewantara
Dept. of Informatics and Computer Engineering
Politeknik Elektronika Negeri Surabaya (PENS)
 Surabaya, Indonesia
 bima@pens.ac.id

3rd Setiawardhana
Dept. of Informatics and Computer Engineering
Politeknik Elektronika Negeri Surabaya (PENS)
 Surabaya, Indonesia
 setia@pens.ac.id

4th Fursan Akzada
Dept. of Informatics and Computer Engineering
Politeknik Elektronika Negeri Surabaya (PENS)
 Surabaya, Indonesia
 fursanakzada@ce.student.pens.ac.id

Abstract—Nowadays, robotics implementation spans industries such as manufacturing, healthcare, and logistics, integrating AI-driven capabilities to enhance efficiency and collaboration in diverse operational environments. The ability to respond and interact while considering human presence is crucial for a mobile robot to operate effectively and be accepted in a human social environment. It is necessary to make a robot that can detect the presence of humans around it to make it capable of navigating while considering the social environment. Meanwhile, considering that the lidar sensor is almost always available in many robot navigation cases, it becomes an accessible tool for detecting the presence of humans in its surroundings. We brought the one-dimensional lidar data into two-dimensional image data, then involved some object selection based on weak characteristics, such as the acceptable range radius of the object and the acceptable distance between two legs. The result of the process is a group of human candidates. We use a deep learning model with a pre-trained mobilenetv3 model as the feature extractor to ensure whether the candidates are humans and a fully connected layer of neural networks as the classifier. We defined several scenarios in an indoor environment to test the performance of the proposed algorithm. The experiment results show the system's performance with a precision score of 0.902, a recall score of 0.782, and an F1 score of 0.837.

Index Terms—Mobile Robot, Lidar Sensor, Object Selection, Deep Learning, MobileNet

I. INTRODUCTION

The development of robot navigation has been a significant area of research in recent years. The increasing usage of robots nowadays is in more than just industrial manufacturing. Various fields, such as healthcare, transportation, and even other daily activities, have started to enforce the existence of robots [1] [2] [3]. These conditions increase the need for robots to navigate autonomously and effectively in their environments, especially mobile robots. This condition has led to the development of new techniques and algorithms for robot navigation, ranging from simple path planning to complex SLAM (Simultaneous Localization and Mapping) [4] [5] [6]. Despite significant progress in robot navigation, many

challenges still need to be addressed to fully understand this complex issue and develop practical solutions. One of the challenges is to make the robot able to work in a human environment without disturbing human activities. Numerous studies have been conducted in order to find solutions to these problems.

Several thorough reviews of human-aware robot navigation [7] [8] make a significant addition to the field of robotics. The review sheds light on the various methods and uses of human-aware navigation. It emphasizes how crucial it is for robots' secure and effective operation in confined places. The paper's conclusion discusses the difficulties and potential paths for future study in human-aware robot navigation. In [9], a socially conscious behavior control has been implemented. By implementing the social force model (SFM), the robot can create behavior that considers obstacles around it and humans as dynamic obstacles. The authors also improved the adaptive ability of the SFM to the environment by using a reinforcement learning algorithm to update the parameters of the SFM. Repiso [10] shows another example of implementing a robot for social activities. The article offers an innovative approach for adaptive side-by-side social robot guidance that is efficient and trustworthy in a practical setting, significantly adding to human-robot interaction. The authors also implemented the SFM algorithm and improved it by defining several scenarios for the robot to adapt its behavior to the current conditions. Outdoor and crowded robot navigation has been tested [11]. The authors proposed a neural network model that utilizes the information on the robot's target direction, the information on pedestrian positions, and the information about the static obstacles. A reinforcement learning method was also implemented to improve the ability to adapt to the conditions. The proposed method is social obstacle avoidance using deep reinforcement learning (SOADRL). The results show that the method has the best success rate among the other compared methods.

One key to implementing socially aware navigation is the robot's ability to recognize humans. Socially aware navigation is only meaningful if there is information about human positions. Many approaches have been proposed to add the ability to detect the position of humans, such as using a camera, a depth sensor, and also a radio signal [12] [13] [14] [15]. Most currently used monitoring techniques emphasize expensive, high-performing instruments like multilayer lidar sensors or high-resolution, short-range radars. Due to the varying lighting, image sensor-based techniques could be more robust. This research aims to build a robust and lightweight human detection system. The proposed method is to use 2D lidar to detect human legs and the position of the human. This approach has also been used in [16] to detect and track a person in a stage for monitor the position of the actors. In this research, the authors used more than one 2D lidar sensor to catch the leg's data from different directions. Those 2D lidar data were then combined into one common coordinate system. The detection and tracking were done in the common coordinate system. Instead of using multiple 2D lidar sensors, we approach this research using only one 2D lidar sensor. The target of our approach is that the method can be applied to social robot navigation.

The remainder of this paper is organized as follows. The architecture of the system and the technique of approach are described in Section II. Section III includes the experiment's design, results, and discussion. We draw inferences about this research in Section IV.

II. SYSTEM DESIGN

In this section, we describe how we acquired 2-dimensional image data from one-dimensional lidar data, preprocessed the lidar data, and how to detect a person by the shape of the legs. The system diagram can be seen in Figure 1.

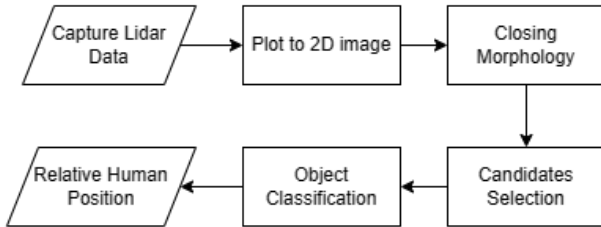


Fig. 1. System Design.

A. 2D Data Acquisition

We use the Robot Operating System (ROS) to integrate the lidar sensor into an embedded computer. The acquired raw data from the lidar sensor is formed as a ROS message containing several parameters. In this work, we only utilize the ranges, a list that stores distance data along the scanning angle, and the angle increment, which stores the value of the scanning increment in degrees. The sensor coverage in this work is entirely 360 degrees, so the data can be formed as a mathematical equation as follows:

$$R = \{r_0, r_1, \dots, r_k\} \quad (1)$$

$$\gamma = \{\gamma_0, \gamma_1, \dots, \gamma_k\} \quad (2)$$

$$\gamma_k = k\alpha \quad (3)$$

with R as a list of values representing the distances from each direction from the lidar to measured obstacles, r_k is the measured distance of the lidar sensor at the index of k , γ is a list of values representing the direction of each distance in degree, the direction of the measured distance in index k , and α is the angle increment from the lidar parameters.

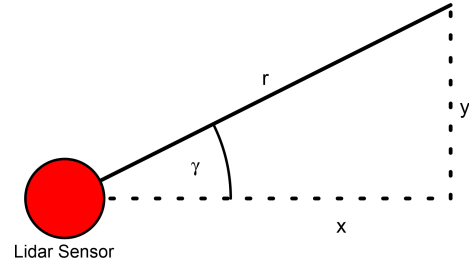


Fig. 2. Conversion from 1D data to 2D coordinate.

From the distance and direction data, two-dimensional data is built by using mathematical trigonometry as shown in Equations (4), (5), and also visualized as in Figure 2. The result of the conversion is then converted from meter units to image pixels and plotted as a two-dimensional map formed as an image.

$$x = r \cos(\gamma) \quad (4)$$

$$y = r \sin(\gamma) \quad (5)$$

The image formed from this step has some underexposed areas at distances. This issue happens due to the lidar sensor's lack of sampling per degree. We implement a morphology operation, Closing Morphology, a closing operation, to fix this issue using a provided function from OpenCV [17]. This operation also utilizes the other two methods, dilation and erosion. The operations are explained in Algorithms 1, 2, and 3, respectively. Erosion and dilation are mathematical morphological operations that process binary or grayscale images. Both operations involve scanning the image with a small structuring element, or kernel, which defines the shape and size of the operation.

B. Candidates selection

We clustered several groups of points from an image by its continuousness. Thanks to [18] has provided an algorithm to determine the surrounding relations among the outer boundaries of a binary image is available in the OpenCV library. Using the algorithm, we obtained a list of contours in the

Algorithm 1: Closing Morphology

Input: Input image I , structuring element SE
Output: Output image O after closing operation
 $B \leftarrow \text{convertToBinary}(I);$
 $D \leftarrow \text{dilation}(B, SE);$
 $E \leftarrow \text{erosion}(D, SE);$
 $O \leftarrow E;$
return O

Algorithm 2: Dilation Operation

Input: Input binary image B , structuring element SE
Output: Output binary image D after dilation operation
 $D \leftarrow \text{createEmptyImage}(B.\text{width}, B.\text{height});$
for $x \leftarrow 0$ **to** $B.\text{width} - 1$ **do**
 for $y \leftarrow 0$ **to** $B.\text{height} - 1$ **do**
 $R \leftarrow \text{placeStructuringElement}(B, x, y, SE);$
 $D[x, y] \leftarrow \max(R);$
 end
end
return D

image that are candidates for the human legs. The next step is to eliminate the candidates with several rules, as follows:

- The radius allowed for the candidates is more than a and less than b , which means the average radius of human legs in a pixel is between a and b .
- The allowed candidates must have one nearest neighbor with a maximum distance of l , which means the maximum distance of a pair of human legs when travelling.

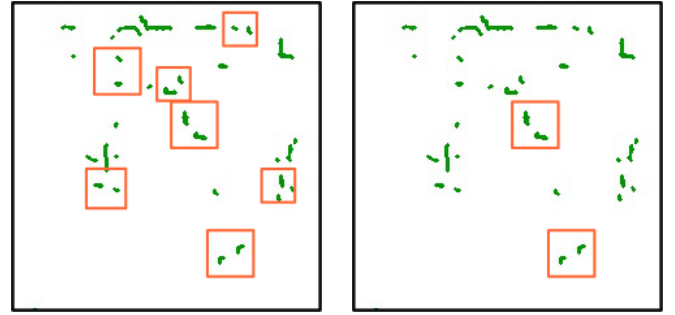
The value of the rules parameter will be defined in the experiment. The result of this step is a list of human leg candidates with more confidence than in the previous step but still needing to be more confident. It is visualized in Figure 3 (a).

C. Candidates Classifier

The final step is ensuring the candidate has a pair of human legs. To achieve this, we transfer learning from a

Algorithm 3: Erosion Operation

Input: Input binary image B , structuring element SE
Output: Output binary image E after erosion operation
 $E \leftarrow \text{createEmptyImage}(B.\text{width}, B.\text{height});$
for $x \leftarrow 0$ **to** $B.\text{width} - 1$ **do**
 for $y \leftarrow 0$ **to** $B.\text{height} - 1$ **do**
 $R \leftarrow \text{placeStructuringElement}(B, x, y, SE);$
 $E[x, y] \leftarrow \min(R);$
 end
end
return E



(a) Before elimination process

(b) After elimination process

Fig. 3. Human leg candidates.

popular deep learning network architecture, MobileNetV3, using an improved version of the predecessor as the classifier [19] as the feature extraction layer. Considering its accuracy and performance, this network is utilized in some image classification applications [20]. Compared to previous versions, MobileNetV3 introduces several key improvements, including a new architecture with different types of blocks, a more efficient activation function called Hard-Swish, the incorporation of squeeze-and-excitation blocks to emphasize important features, dynamic input resolution, and improved channel pruning. These improvements increase accuracy and efficiency while supporting various devices and introducing new features. At the output of the MobileNetV3 layer, we use a simple neural network model containing a single layer with 256 nodes as the classifier. During the training process, the weights of the MobileNetV3 were frozen. After the training process, a strong classifier model is obtained to determine whether the candidate is a human. The detected candidates that are not predicted to be human will be eliminated. The output of this step is visualized as Figure 3 (b).

III. RESULTS AND DISCUSSIONS

This section describes the results obtained from the experiments with the algorithm. For the experiments, we use the following embedded computer specifications, operating system, and programming tools: Jetson Nano, NVIDIA Maxwell GPU, 2GB RAM, Ubuntu 20.04 LTS, and ROS. We use the RPLidar A3 as the lidar sensor. We placed the lidar sensor on a top of a differential drive mobile robot which later we will use for implement the algorithm. The robot is shown in Figure 4. A test of finding the radius of human legs visualized in a pixel image revealed that the maximum and minimum radii are 5 pixels and 2 pixels, respectively, varying by the distance between the human and the lidar sensor and also varying with different people. The maximum distance of a pair of legs is 10 pixels, varying by distance from the lidar sensor and with different people.

A. Dataset Creation

The candidate's selection from II-B represents all the candidates in the image, but there are still non-human legs selected

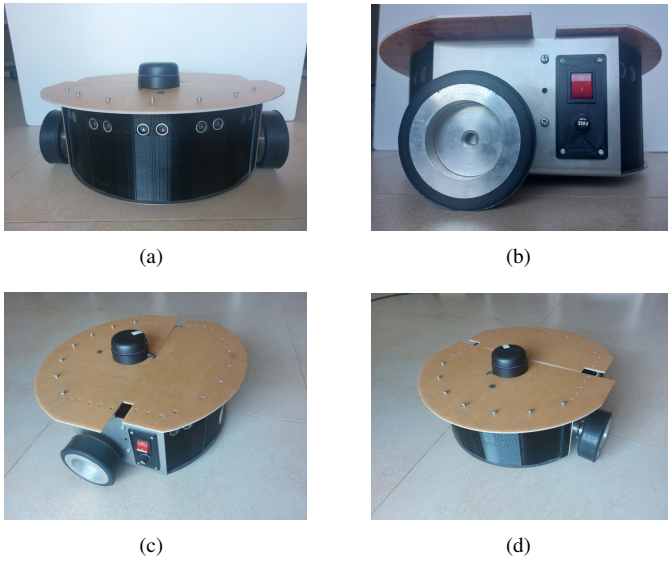


Fig. 4. The placement of the lidar sensor on the robot.

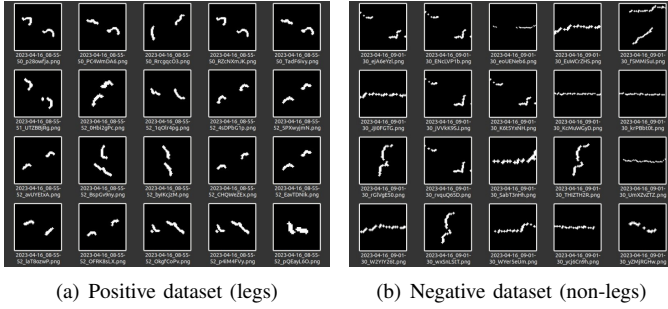


Fig. 5. Result from dataset creation.

at the end of the process. To create the dataset, we utilize this step to crop the candidates and save them in a folder automatically. For the positive dataset, we run this process in a room with many obstacles, and then we ask some people as samples to walk around. The system will crop and record the expected candidates. Moreover, we ran the candidate's selection system for the negative dataset in a room with many obstacles and no humans left. The system will crop and record only the non-human candidates. The result obtained is 10267 images of positive data and 16383 images of negative data. The dataset is shown in Figure 5.

B. Training Process

The dataset from the previous step is used for classifier model training. A total of 8214 images from the positive dataset and 13107 images from the negative dataset were used for the training process. The rest of the datasets were used for validation data. The dataset is split into two, for the train data and the validation data, with a ratio of 8:2. The training was done in 100 iterations. The accuracy score of the training process is 0.99, and the validation accuracy is 0.98. The graph

during the training process is shown in Figures 6 and Figure 7.

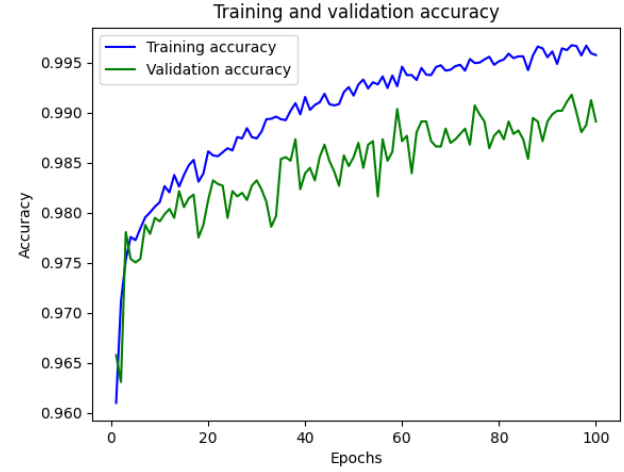


Fig. 6. The accuracy score during training process.

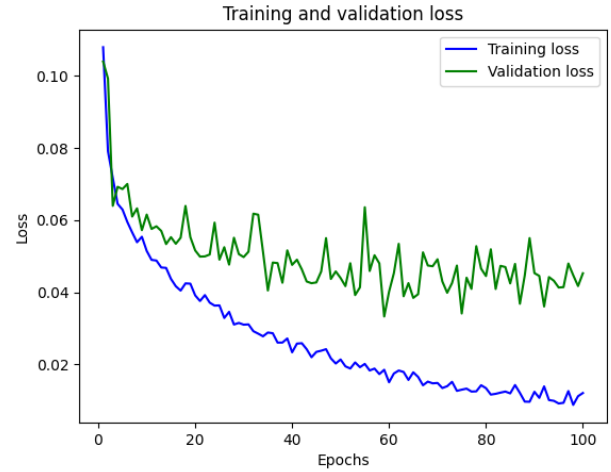


Fig. 7. The loss value during training process.

C. Real Test

The model's success in achieving a good accuracy result shows that the model is ready for the real test. We defined two scenarios for the real test. The first is in a hallway, with the peoples walking along the hallway. And the second is in a wide hall with peoples walk around it. The real setup is shown in Figure 8.

There were several experimental scenarios. During all the experiments, the detected positions of the people are recorded with a certain sampling rate. A camera was set up for records the people's positions around the sensor as the ground truth during the experiment. Some of the captured frames from the obtained ground truth are shown in Figure 9. Moreover, the scanned data is shown in Figure 10.

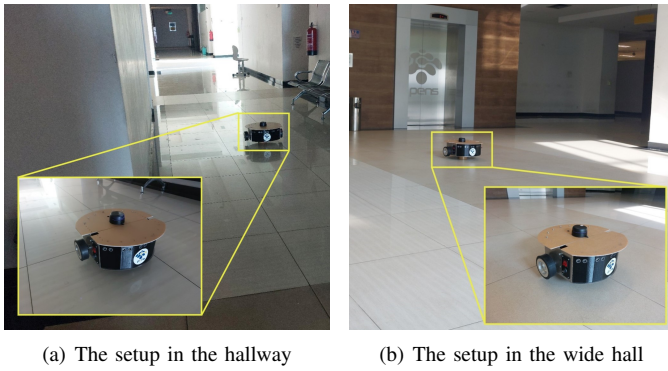


Fig. 8. The Real Experiment Setup.

The system has brought promising results in real experiments. There is some miss detection when there is a non-legs object shaped like a legs object. For example in Figure 9 (d) and Figure 10. When the wall is covered by human legs, the formed line of the wall is split, so the shape is formed, likely human legs. But, there is also another case when the system is able to differ the human legs with the wall even if the line of the wall split, such as in Figure 9 (a) and Figure 10 (b). This is probably happening because the shape has been covered by the dataset created before, and for the miss in Figure 10 (d), the case is still not or is less covered by the created dataset. There is another limitation. Due to the rules in the object selection algorithm, only a pair of human legs will be a candidate. So, the system cannot detect humans when only one leg is scanned by the lidar, for example, in Figure 9 (f) and 10. After all, The result from the whole experiment showed a good performance.

TABLE I
EXPERIMENT RESULTS

Metric	Value
True Positives (TP)	431
False Positives (FP)	47
False Negatives (FN)	120

The result of accumulation with different scenarios is shown in Table I. In evaluating the performance of our object detection system, we considered several vital metrics: True Positives (TP), False Positives (FP), and False Negatives (FN). These metrics provide insights into the system's ability to identify and classify objects within an image correctly. True Positives (TP) represent the instances where your system correctly detected and classified objects. These are the cases where both the ground truth and our model's prediction align. False Positives (FP) occur when our system mistakenly identifies an object that isn't present in the image. These are instances where your model produces a positive prediction, but it doesn't match the actual ground truth. False Negatives (FN) indicate the cases where your system fails to detect an object that is present in the image. These are instances where your model produces a negative prediction, but it doesn't align with the actual ground truth.

We calculated two additional metrics further to evaluate our object detection system: Recall and Precision. Recall measures the effectiveness of a model or system in identifying relevant items from a larger set [21]. It is calculated by dividing the total number of correctly classified instances (TP) by the total number of correct positive predictions and false negative prediction (TP + FN). It is commonly used in information retrieval, classification, and binary classification tasks to assess the model's performance in terms of identifying true positive cases. Precision, on the other hand, focuses on the model's ability to minimize false positives [21]. It is calculated by dividing the number of true positives (TP) by the total number of correct positive predictions and false positive prediction (TP + FP). Precision provides insights into the reliability of our system when it makes positive predictions.

While recall and precision are valuable metrics, they do not fully capture the balance between false positives and false negatives. That is where the F1 score comes into play. The F1 score combines precision and recall (equivalent to sensitivity) into a single metric. It is the harmonic mean of precision and recall, calculated using the formula:

$$F1_{score} = 2 * \frac{precision * recall}{precision + recall} \quad (6)$$

The values of precision and recall were calculated from the values of TP, FP, and FN. Moreover, the results are 0.902 and 0.782, respectively. Using Equation (6), we calculated our model's F1 score, which obtained a value of 0.837. An F1 score of 0.837 indicates an excellent overall performance of our object detection system. With an F1 score of 0.837, our model demonstrates a substantial trade-off between precision and recall. Our system achieves a relatively high level of precision and recall, indicating accurate and reliable object detection. It balances correctly identifying objects (precision) and avoiding missed detections (recall). Such a performance is commendable and suggests that our model effectively classifies objects accurately while minimizing false alarms and missed detections.

The F1 score provides a balanced measure of our model's performance, considering false positives and negatives. It ranges between 0 and 1, where 1 represents perfect precision and recall, and 0 indicates the worst performance. Higher F1 scores indicate better performance in terms of both precision and recall. In summary, by considering metrics such as True Positives, False Positives, False Negatives, Accuracy, Precision, and the F1 score, we gain a comprehensive understanding of our object detection system's performance, including its ability to correctly detect objects and minimize both false positives and false negatives.

IV. CONCLUSION

We developed a human detection system based on the foot shape formed by a 2D lidar sensor. The main challenge with this problem is that the scanned shape of the human foot is different. It also depends on what pants the human



(a) Scene 1 from experiment in the corridor



(b) Scene 2 from experiment in the corridor



(c) Scene 3 from experiment in the corridor



(d) Scene 4 from experiment in the corridor

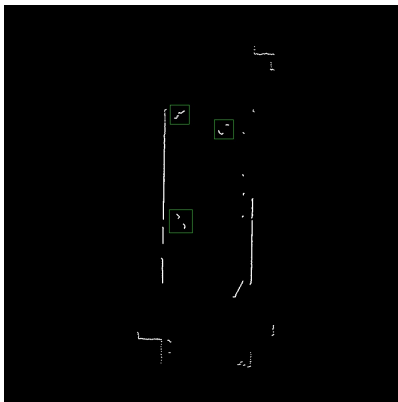


(e) Scene 5 from experiment in the lobby

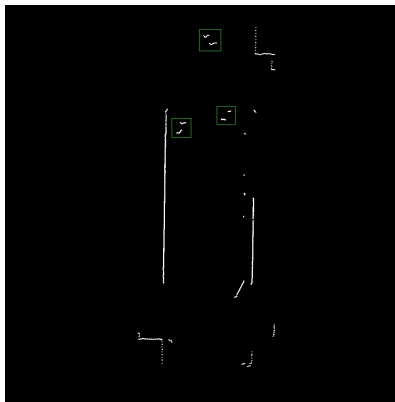


(f) Scene 6 from experiment in the lobby

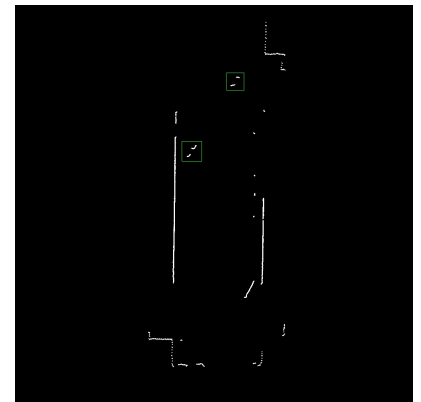
Fig. 9. The Real Experiment Scenes.



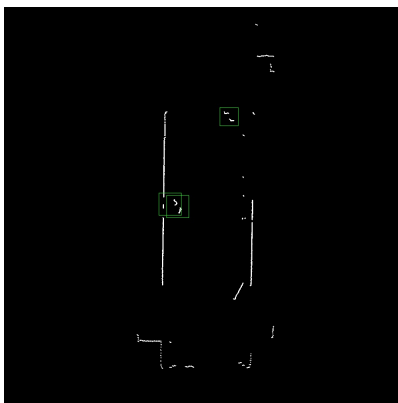
(a) The detected human from scene 1



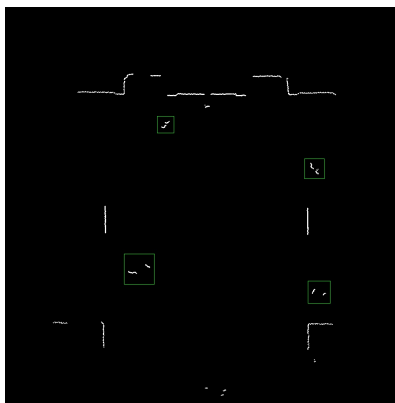
(b) The detected human from scene 2



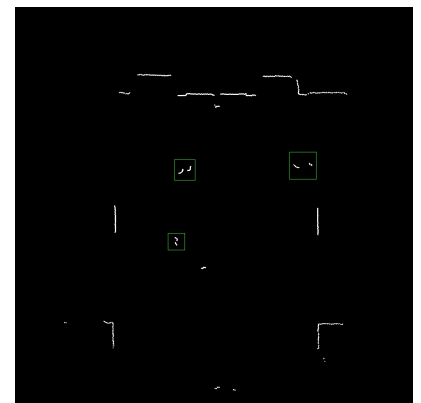
(c) The detected human from scene 3



(d) The detected human from scene 4



(e) The detected human from scene 5



(f) The detected human from scene 6

Fig. 10. The Detected Humans from the real test.

is wearing. As a feature extractor, we implement the well-known and influential deep learning object recognition model, mobilenetv3, and a neural network model as a classifier in the output of the mobilenetv3 model. The system created still requires improvement in several areas. For example, when a chair has rounded legs, the shapes of the chair legs and human legs differ by the same distance. However, at different distances, for example, if the chair legs are seen from a close distance, they will look like human legs at a long distance. As a solution, we recommend adding object distance as an additional parameter in future studies. This solution will allow the system to accurately distinguish between objects with similar shapes but different sizes, even at different distances from the sensor. Despite its shortcomings, this study gave satisfactory results, obtaining an F1 score of 0.837.

ACKNOWLEDGMENT

We want to express our heartfelt gratitude to all the Social Robotics and Smart Systems Applications research group members for their exceptional contributions and unwavering support throughout this research project. The collective expertise, dedication, and collaboration of our research group members have been invaluable in shaping the success of this endeavor. From the guidance of our research group leader to the enthusiasm and creativity exhibited by our fellow researchers, the contributions of each individual have played a pivotal role in driving our research forward. We are immensely grateful for the fruitful discussions, insightful feedback, and tireless efforts that have greatly enriched our work. Lastly, we express our deep appreciation to all the participants who volunteered their time and cooperation, as their involvement was essential in gathering the necessary data.

REFERENCES

- [1] J. Bačík, P. Tkáč, L. Hric, S. Alexovič, K. Kyslan, R. Olexa, and D. Perduková, "Phollower—the universal autonomous mobile robot for industry and civil environments with COVID-19 germicide add-on meeting safety requirements," vol. 10, no. 21, p. 7682, number: 21. [Online]. Available: <https://www.mdpi.com/2076-3417/10/21/7682>
- [2] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," vol. 4, no. 2, pp. 610–617, number: 2. [Online]. Available: <https://ieeexplore.ieee.org/document/8606991/>
- [3] V. Moysiadis, N. Tsolakis, D. Katikaridis, C. G. Sørensen, S. Pearson, and D. Bochtis, "Mobile robotics in agricultural operations: A narrative review on planning aspects," vol. 10, no. 10, p. 3453, number: 10. [Online]. Available: <https://www.mdpi.com/2076-3417/10/10/3453>
- [4] F. Gul, W. Rahiman, and S. S. Nazli Alhady, "A comprehensive study for robot navigation techniques," vol. 6, no. 1, p. 1632046, number: 1. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/23311916.2019.1632046>
- [5] B. Patle, G. Babu L, A. Pandey, D. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," vol. 15, no. 4, pp. 582–606, number: 4. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2214914718305130>
- [6] H. Jiang, H. Wang, W.-Y. Yau, and K.-W. Wan, "A brief survey: Deep reinforcement learning in mobile robot navigation," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, pp. 592–597. [Online]. Available: <https://ieeexplore.ieee.org/document/9248288/>
- [7] A. W. Ramadhan, B. S. B. Dewantara, and S. Setiawardhana, "Optimization of fuzzy social force model adaptive parameter using genetic algorithm for mobile robot navigation control," vol. 19, no. 1, number: 1.
- [8] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," p. 3583741. [Online]. Available: <https://dl.acm.org/doi/10.1145/3583741>
- [9] B. S. B. Dewantara and J. Miura, "Generation of a socially aware behavior of a guide robot using reinforcement learning," in *2016 International Electronics Symposium (IES)*. IEEE, pp. 105–110. [Online]. Available: <http://ieeexplore.ieee.org/document/7860984/>
- [10] E. Repiso, A. Garrell, and A. Sanfeliu, "Adaptive side-by-side social robot navigation to approach and interact with people," vol. 12, no. 4, pp. 909–930, number: 4. [Online]. Available: <http://link.springer.com/10.1007/s12369-019-00559-2>
- [11] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dube, "Robot navigation in crowded environments using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5671–5677. [Online]. Available: <https://ieeexplore.ieee.org/document/9341540/>
- [12] R. Algabri and M.-T. Choi, "Deep-learning-based indoor human following of mobile robot using color feature," vol. 20, no. 9, p. 2699, number: 9. [Online]. Available: <https://www.mdpi.com/1424-8220/20/9/2699>
- [13] J. N. Hayton, T. Barros, C. Premevida, M. J. Coombes, and U. J. Nunes, "CNN-based human detection using a 3d LiDAR onboard a UAV," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, pp. 312–318. [Online]. Available: <https://ieeexplore.ieee.org/document/9096075/>
- [14] W. Kim, M. Tanaka, M. Okutomi, and Y. Sasaki, "Learning-based human segmentation and velocity estimation using automatic labeled LiDAR sequence for training," vol. 8, pp. 88443–88452. [Online]. Available: <https://ieeexplore.ieee.org/document/9090135/>
- [15] T. Kim Geok, K. Zar Aung, M. Sandar Aung, M. Thu Soe, A. Abdaziz, C. Pao Liew, F. Hossain, C. P. Tso, and W. H. Yong, "Review of indoor positioning: Radio wave technology," vol. 11, no. 1, p. 279, number: 1. [Online]. Available: <https://www.mdpi.com/2076-3417/11/1/279>
- [16] H. Yoo, H. J. Moon, S.-H. Kim, and S.-I. Choi, "Multi-target tracking with multiple 2d range scanners," vol. 8, pp. 99990–99998. [Online]. Available: <https://ieeexplore.ieee.org/document/8618455/>
- [17] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [18] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," vol. 30, no. 1, pp. 32–46, number: 1. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0734189X85900167>
- [19] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," publisher: arXiv Version Number: 5. [Online]. Available: <https://arxiv.org/abs/1905.02244>
- [20] S. Qian, C. Ning, and Y. Hu, "MobileNetV3 for image classification," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. IEEE, pp. 490–497. [Online]. Available: <https://ieeexplore.ieee.org/document/9389905/>
- [21] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," vol. 10, no. 3, p. 279. [Online]. Available: <https://www.mdpi.com/2079-9292/10/3/279>