ISSN (Online) 2456-1290

International Journal of Engineering Research in Mechanical and Civil Engineering (IJERMCE)

Vol 10, Issue 6, June 2023

Performance Analysis of Vision-Based Transparent Obstacle Avoidance for Construction Robots

[1] Siwei Chang, [2] Heng Li, [3] Haitao Wu, [4] Xing Fang

[1] [3] [4] Ph.D. Candidate, Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University
[2] Chair Professor, Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University
Corresponding Author Email: [1] siwei.chang@connect.polyu.hk, [2] heng.li@polyu.edu.hk, [3] Haitao.Wu@connect.polyu.hk, [4] fang.xin@connect.polyu.hk

Abstract*— Construction robots are receiving more and more attention as a promising solution to the emerging shortcomings of the conventional construction industry. The development of intelligent control techniques for obstacle avoidance is crucial for guaranteeing the adaptability and flexibility of mobile construction robots in complex construction environments. Most of the existing obstacle avoidance algorithms are based on processing high-precision point cloud data collected by laser sensors to ensure operation fluency. However, because of the limitations of the laser sensors, those algorithms are invalid when detecting transparent obstacles that frequently appear in building environments. Therefore, this study aims to introduce a vision-based process for mobile construction robots to avoid transparent obstacles. To do so, a monocular camera is mounted on the testing robot platform, Turtlebot3 Burger, to collect visual inputs. A convolutional neural network is trained to compute the received videos and recognize transparent obstacles. The vision programs are coded in the Robot Operating System (ROS) to control the robot's motions. On-site validations are conducted to prove the efficiency of the vision-based obstacle avoidance process. Different from avoiding obstacles using lidar inputs, the vision-based strategy successfully controlled the robot to avoid transparent obstacles. The findings contribute to paving a novel method for robotic obstacle avoidance by combining visual signals and deep learning, which is more efficient for avoiding collisions with transparent obstacles.

***Index Terms*—Construction robot, Obstacle avoidance, Computer vision, Transparent obstacle.**

I. INTRODUCTION

By automatically imitating, assisting, or even replacing various manual construction behaviors, construction robotics is deemed a promising way to realize a fully automated construction process and thoroughly alleviate the shortcomings of conventional construction, such as labor shortage. According to ResearchAndMarket.com [1], the global construction robotics market is expected to grow at a compound annual growth rate (CAGR) of 12.7% from 2020 to 2025, indicating the increasing adoption of these robots in the construction industry. Therefore, a number of construction robotics have been developed, and some of them have begun to be adopted on-site. For instance, mobile robotic platforms were developed for on-site logistic purposes, such as moving packages around indoors [2-3]. The wheeled mobile robot is designed for the automated quality inspection of pipelines [4], concrete bridges [5], or pavement [6].

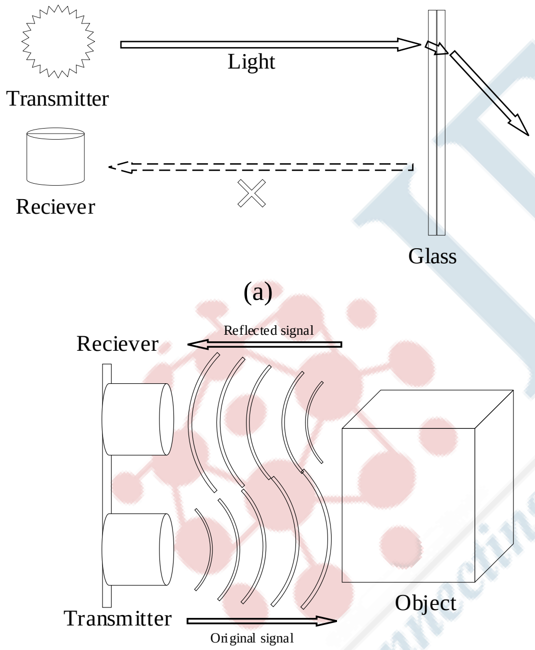
Obstacle avoidance can be regarded as one of the most significant challenges for controlling mobile robots, which realize autonomous movements in dynamic environments by ensuring safe and effective navigation without colliding with obstacles [7]. Without the ability to avoid obstacles, a robot risks becoming stuck, facing damage, or even causing harm to its surroundings [8]. Therefore, it is essential to equip construction robots with intelligent obstacle avoidance algorithms to ensure smooth travel in complex construction environments.

To avoid obstacles, the robots first perceive the obstacles' location, size, and shape in an environment using attached sensors, such as lidar and ultrasonic sensors. After obtaining the sensor inputs, path planning algorithms, such as fuzzy logic, compute the input data and generate a navigation path for the robot to avoid obstacles. By emitting laser beams and measuring the time it takes to bounce back off objects in the environment, the lidar sensor provides highly accurate 3D maps of the environment, which can be utilized to detect and avoid obstacles in real time, making it the most popular and commonly used sensor for robot obstacle avoidance [9-12]. However, because the laser pulses cannot be reflected back effectively, lidar cannot be used to detect transparent obstacles [13], such as glass curtain walls. Therefore, this study aims to introduce a transparent obstacle avoidance method for construction robots based on visual information processing, which conveys messages more efficiently, directly, and flexibly.

II. LITERATURE REVIEW

A. Transparent obstacle avoidance using range sensors

Obstacle avoidance in robotics is the process of achieving a control goal while adhering to non-intersection or non-collision position constraints based on sensorial information [14]. Because lidar provides a larger measurement range and accurate and consistent sensing, it has been employed most widely to detect obstacles compared with other sensory systems, such as infrared sensors [15-17]. In lidar, the laser light is sent from a transmitter and reflected from the obstacle objects. Using the time of flight (ToF), the receiver determines and detects the distance between the transmitter and reflector. However, because light cannot be reflected from transparent obstacles, refraction, and penetration occur (as shown in Fig. 1(a)), which results in inaccurate detection. The ultrasonic sensor can detect transparent obstacles differently because it measures the propagation time of sound between sender and receiver instead of recording ToF (as shown in Fig. 1(b)). As such, ultrasonic sensors gained preference to improve the flexibility of avoiding transparent obstacles in building environments, such as glass doors [18], glass walls [19], and windows [20]. However, the poor range accuracy of ultrasonic sensors is also a significant concern [21].



(a)

(b)

Fig. 1 Working principle of lidar and ultrasonic sensor

B. Transparent obstacle avoidance using vision sensors

In addition to range sensors, visual sensors, including monocular or binocular cameras, are also considered efficient alternatives to detect obstacles. Instead of measuring distances based on light or sound waves, vision sensors detect obstacles by capturing images or videos of a scene, which provides rich and detailed information about the obstacles, such as shape, size, and color. Meanwhile, because vision sensors are insensitive to environmental changes and provide stable information, they are highly recommended to avoid particular obstacles, such as transparent obstacles [22-24] and underwater obstacles [25].

For the recognition, classification, or localization of objects, various computer vision algorithms are developed to process the received visual signals. Generally, computer vision algorithms used for robot obstacle avoidance can be divided into two groups. The first group employs image processing-based algorithms [26–27] for enhancement, restoration, and compression to extract the obstacles from the background. Detailed information, such as obstacle locations, can be provided to guide the robot as it navigates over obstacles. However, transparent objects that have no obvious features are hard to segment using image processing methods. For example, it is difficult to distinguish glass information because objects with solid features, such as cups, are in the glass area. To improve accuracy and robustness, deep learning algorithms, such as convolutional neural networks (CNN) or recurrent neural networks (RNNs), are increasingly embraced [28], especially in building and construction environments, which are surrounded by transparent components [29-30]. Therefore, this study trained a CNN network to control the robot and recognize transparent obstacles. Considering the computation limits of CPU-driven robotic control boards, the lightweight CNN network is employed.

III. METHODOLOGY

A. Data collection

To train the lightweight CNN model, a monocular camera is mounted on the front of the Turtlebot3 Burger to form training and testing datasets (as shown in **Fig. 2**).

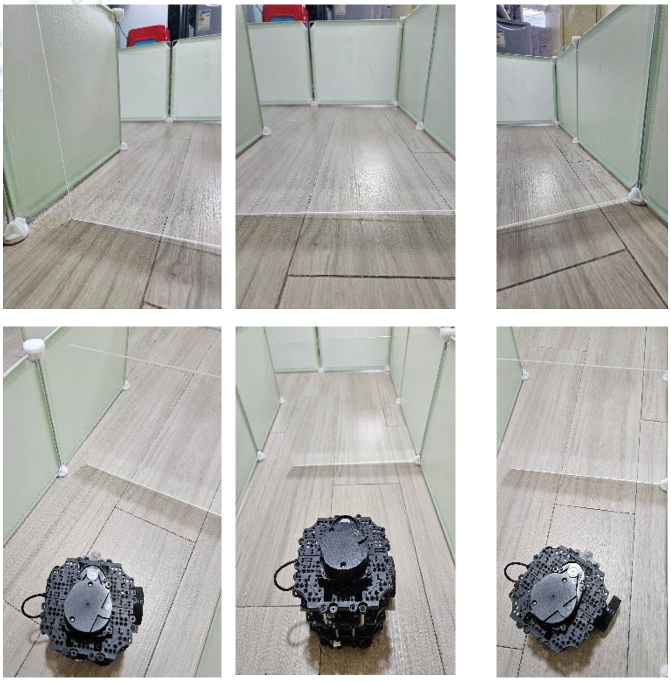
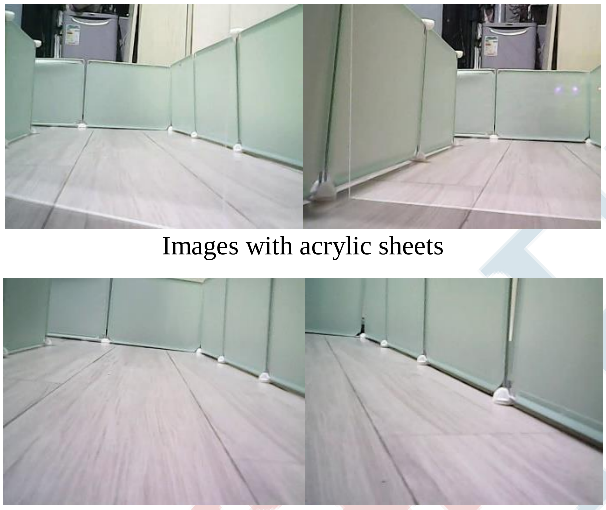


Fig. 2 Data collection scenario

Acrylic sheets were placed accordingly to simulate transparent obstacles in buildings, such as glass curtain walls. The robot is placed in front of the acrylic sheet within 30cm from different angles to record the scenarios with and without acrylic sheets. A safe distance of 30cm is set considering the length and width of the Turtelbot3 burger to leave a certain space for reaction. As a result, 2501 and 634 images with and without acrylic sheets (transparent obstacles) are collected and stored in the datasets. The 3135 images are randomly split with a ratio of 80/20 for the training and testing datasets, which contain 2508 and 627 images, respectively. Examples of the collected images are shown in **Fig. 3**. The collected images are resized to the shape of 224 × 224 to simplify CNN computation. The images with and without acrylic sheets are labeled 0 and 1, respectively, to enable the robot to classify the transparent obstacles and safe paths.



Images with acrylic sheets

Image without acrylic sheets

Fig. 3 Example of collected datasets

B. Data processing

The MobileNet, a lightweight CNN model, is trained to enable the robot to classify transparent obstacles. To be specific, the training images are first transformed into a tensor matrix and input to the MobileNet model. The architecture of the MobileNet is shown in **Table. 1**. The backpropagation algorithm is employed to compute the weight and bias matrix. To obtain fast training, and avoid overfitting and underfitting problems, the learning rate of 0.0001, and batch size of 32 are set after several fine-tunings. The outputs are the image classifications of "place with transparent obstacles" and "place without transparent obstacles."

The confusion matrix is used to examine the model's performance. In the confusion matrix, TP means the number of images predicted as the regions with transparent obstacles, and it is also the transparent obstacle in reality. TN means the number of images that are predicted as regions without transparent obstacles, and it is also the non-transparent obstacle surface in reality. FP means the number of images predicted as a transparent obstacle, but the actual classification is not. FN means the images are predicted as a region without transparent obstacles, but the actual classification is the transparent obstacle. Therefore, all the correct predictions are shown in the diagonal of the confusion matrix. The bigger the values on the diagonal, the more accurate the prediction. The network evaluation index of accuracy, precision, error, recall, and F1-score are then computed using equation (1) to equation (4):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \qquad \text{Equation (1)}$$

$$Precision = \frac{TP}{TP + FP} \times 100 \qquad \text{Equation (2)}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \qquad \text{Equation (3)}$$

$$F1-score = 2 \times \frac{precision \times recall}{precision + recall} \times 100 \qquad \text{Equation (4)}$$

The programming is conducted in the Kaggle platform to ensure computation efficiency, which is equipped with T4×2 GPU.

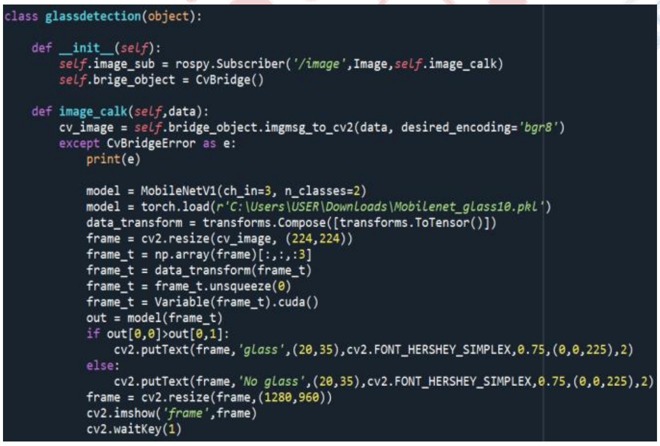
Table. 1 Architecture of MobileNet [31]

Type/Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5 × Conv dw / s1, Conv / s1	3 × 3 × 512 dw, 1 × 1 × 512 × 512	14 × 14 × 512, 14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512

Type/Stride	Filter Shape	Input Size
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

A single computer board, the Raspberry Pi 3b, is used to realize the expected motions of the robot. The robot operating system (ROS) is installed on the Raspberry Pi to program the CNN computation and motion commands. The messages of Image from the sensor_msgs package are employed to receive the images and video stream from the monocular camera. The CvBridge package in ROS is employed to convert ROS image messages to OpenCV format, which is required for the CNN computation. Details can be found in the sample code (as shown in **Fig. 4**). The Twist message from the geometry_msgs package is employed to control the robot's movements by sending linear and angular velocity commands.

To validate the performance of the vision-based transparent obstacle avoidance method, an obstacle avoidance experiment was also conducted. To accurately target the forehead obstacles, interval ranges of [150°~180°], [60°~120°], and [0°~30°] are split for the lidar to represent the ranges of left-front, front, and right-front, respectively. The LaserScan message from the sensor_msgs package is employed to receive the distance data from a 360° lidar. The above-mentioned programs are coded using the Python language on an Ubuntu 16.04 system.



```
class glassdetection(object):

    def __init__(self):
        self.image_sub = rospy.Subscriber('/image',Image,self.image_calk)
        self.brige_object = CvBridge()

    def image_calk(self,data):
        cv_image = self.bridge_object.imgmsg_to_cv2(data, desired_encoding='bgr8')
        except CvBridgeError as e:
            print(e)

            model = MobileNetV1(ch_in=3, n_classes=2)
            model = torch.load(r'C:\Users\USER\Downloads\Mobilenet_glass10.pkl')
            data_transform = transforms.Compose([transforms.ToTensor()])
            frame = cv2.resize(cv_image, (224,224))
            frame_t = np.array(frame)[:,:,:3]
            frame_t = data_transform(frame_t)
            frame_t = frame_t.unsqueeze(0)
            frame_t = Variable(frame_t).cuda()
            out = model(frame_t)
            if out[0,0]>out[0,1]:
                cv2.putText(frame,'glass',(20,35),cv2.FONT_HERSHEY_SIMPLEX,0.75,(0,0,225),2)
            else:
                cv2.putText(frame,'No glass',(20,35),cv2.FONT_HERSHEY_SIMPLEX,0.75,(0,0,225),2)
            frame = cv2.resize(frame,(1280,960))
            cv2.imshow('frame',frame)
            cv2.waitKey(1)
```

Fig. 4 Sample code for importing CvBridge

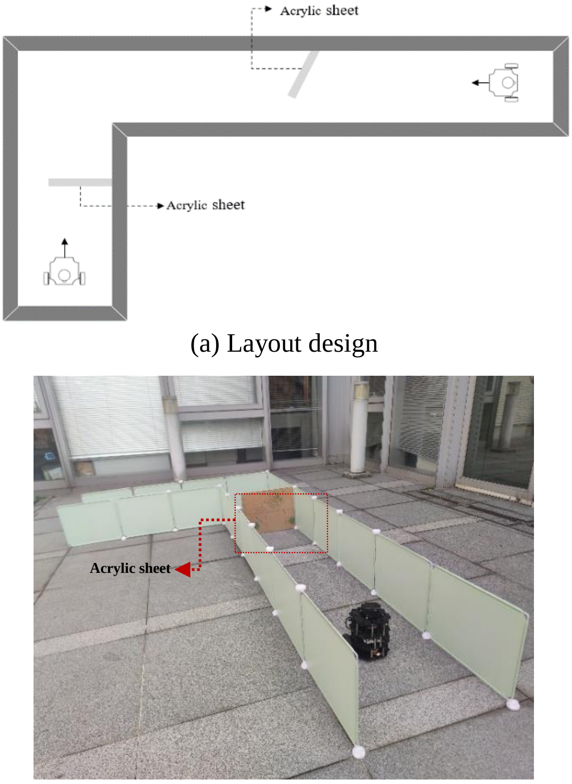
C. Validation experiments

The mobile robotic platform, Turtlebot3 Burger, is employed as the testing machine to implement the experiments. As depicted in Fig. 1, the Turtlebot3 Burger is assembled as a three-layer octagon-shaped platform with a size of 138mm×178mm×192mm. The robot's maximum translational and rotational velocities reach 0.22m/s and 2.84 rad/s, respectively. By sending linear and angular velocity commands to the DYNAMIXEL motors, which are attached to the right and left wheels, the velocities of the wheels can be controlled automatically to enable the basic behaviors of "go straight," "turn left," "turn right," and "stop." According to the principles of kinematic dynamics, the velocities of the right wheel (V_r) and left wheel (V_l) can be calculated using equation (5).

$$\begin{cases} \dot{V}_r = \dfrac{2V + \omega L}{2R} \\ \dot{V}_l = \dfrac{2V - \omega L}{2R} \end{cases} \qquad \text{Equation (5)}$$

Here: the values of L and R are 160mm and 33mm, respectively, which refers to the distance between the left and right wheels and the radius.

A typical L-shaped building layout design is referenced to establish the onsite validation experiment environment. As shown in **Fig. 5**, the experiment places cover both regions with and without transparent obstacles. Transparent acrylic sheets are placed straight and inclined on the traveling paths, respectively to thoroughly examine the obstacle avoidance performance.



(a) Layout design

(b) On-site environment

Fig. 5 Validation experiment environment

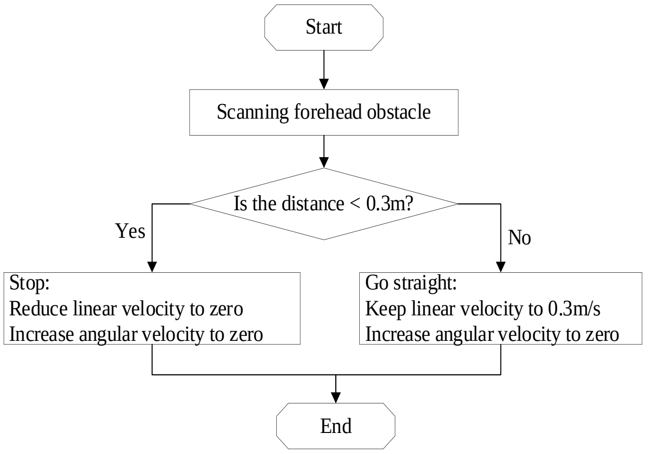

Start
Scanning forehead obstacle
Is the distance < 0.3m?
Yes
No
Stop:
Reduce linear velocity to zero
Increase angular velocity to zero
Go straight:
Keep linear velocity to 0.3m/s
Increase angular velocity to zero
End

Fig. 6 A simple movement flow

As shown in **Fig. 6**, a simple movement logic is employed to test whether the robot can successfully detect forehead-transparent obstacles and avoid them. Specifically, when the monocular camera and the lidar detect the forehead transparent obstacles positioned less than 30cm away, the robot is expected to stop on the ground slowly to avoid obstacles by setting the linear velocity to zero and the angular velocity to zero.

IV. RESULTS

A. Network performance

Fig. 7 shows the variation in loss and accuracy in both training and testing datasets. It can be seen that the loss and accuracy in the training dataset converged at a fast speed. The training loss decreased from 0.26 to 0.001 at the earlier 5th epoch and maintained the value below 0.001. Similarly, the training accuracy increased from 0.90 to 1.00 at the earlier 6th epoch and stayed the same at 1.0. The change in loss and accuracy in the testing dataset is in accordance with the changes in the training dataset, which proves there is no overfitting or underfitting problem with the trained model. The testing loss decreased from 0.02 to 0.0001 at the 4th epoch, while the testing accuracy was maintained at 1.0 from the 2nd epoch.

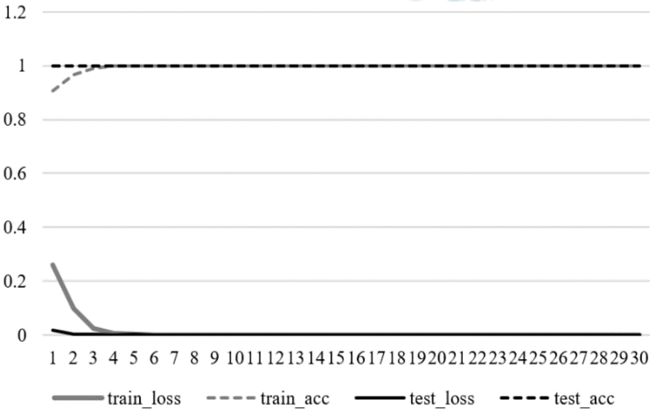


Fig. 7 Loss and accuracy in training and testing datasets

Table. 2. Confusion matrix and evaluation index

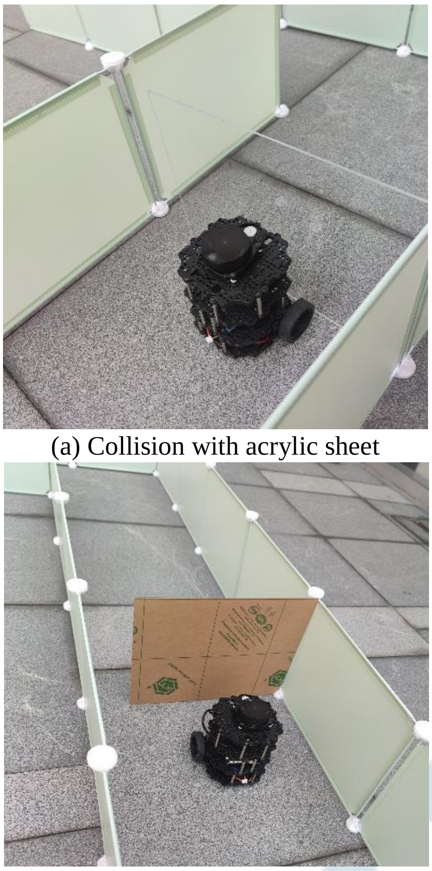
Dataset	TP	FN	FP	TN
Training	2009	1	1	497
Testing	490	1	1	135
Dataset	**Accuracy**	**Precision**	**Recall**	**F1-score**
Training	99%	99%	99%	99%
Testing	99%	99%	99%	99%

The confusion matrix and evaluation index are shown in **Table. 2**, which demonstrate the outstanding performance of the trained MobileNet model. First, the lightweight model shows high accuracy. The accuracy score reached 99% and 99% in the training and testing datasets, respectively, outperforming most existing CNN models [32]. Meanwhile, the precision score reached 99% and 99% in the training and testing datasets, respectively, higher than most CNNs [33]. It can be indicated that the trained MobileNet is applicable in both regions with and without transparent obstacles.

The high recall scores of 99% and 99% in the training and testing datasets are higher than the existing pre-trained models for transparent object detection. For example, the trained model in [34]. A high recall score indicates the accurate recognition of transparent obstacles. Specifically, nearly all the 2010 images with transparent obstacles in the training dataset are correctly recognized. In the testing dataset, nearly all the transparent obstacle images are correctly recognized. The high F1-scores of 99% and 99% in the training and testing datasets prove the high robustness of the trained MobileNet model. The F1-score is higher than most existing CNN models, including the CNN models for transparent object detection, with F1-scores of 91.56% [35].

B. On-site validation

To highlight the effectiveness of the vision-based transparent obstacle avoidance method, the experiment of lidar-based obstacle avoidance is first conducted as the comparison object. As shown in **Fig. 8(a)**, because the light wave penetrates the transparent objects, the robot fails to detect the acrylic sheet when employing lidar to receive point cloud data. It is worth noting that lidar is still more accurate and faster than vision-based processing when detecting solid obstacles, as shown in **Fig. 8(b)**.

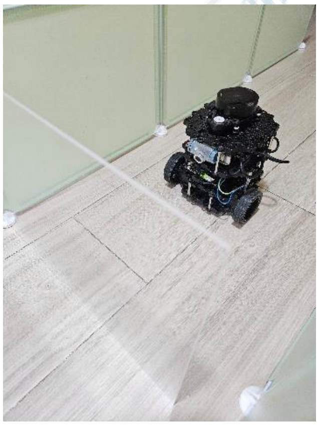


(a) Collision with acrylic sheet

(b) Avoiding a solid obstacle

Fig. 8 Scenes of collision and avoiding obstacles using lidar

Differently, the vision-based obstacle avoidance method successfully controlled the robot to avoid transparent obstacles by processing the input video stream with the MobileNet model. As shown in **Fig. 9**, the robot moves straight with a linear velocity of 0.3 m/s and an angular velocity of zero at first. The linear velocity turns to zero when the MobileNet outputs the object label as "0", which controls the robot to stop on the ground to avoid forehead transparent obstacles. The method is effective no matter the robot's forward angles because the training images are captured by placing the robot in front of the acrylic sheets at different angles.



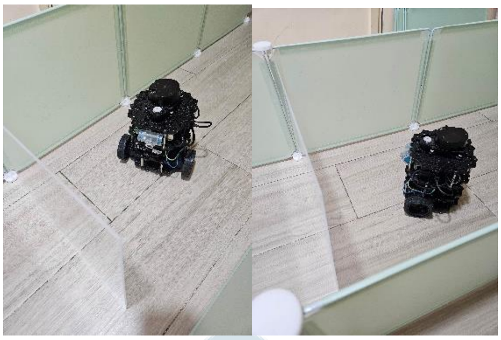


Fig. 8 Avoiding obstacles using the vision-based method

V. CONCLUSION

To achieve transparent obstacle avoidance for mobile construction robots, this research introduces a vision-based obstacle avoidance method to avoid collisions with transparent building obstacles. Specifically, a monocular camera is equipped to capture video streams and form training and testing datasets. The datasets are then input to a trained lightweight MobileNet model. The CNN programs are coded in the ROS platform to control robot's movements. The CvBridge package in ROS is employed to converts ROS information to OpenCV format. Doing so, the robot recognizes the transparent obstacles avoid them by changing linear and angular velocities accordingly when the MobileNet outputs the video objects with the label "transparent obstacles." This research contributes to introducing an effective and novel idea of transparent obstacle avoidance for mobile robots by combining visual signals and a deep learning approach. The performance of the introduced method is proved by conducting an on-site validation and comparing it with a lidar-based transparent obstacle avoidance experiment.

REFERENCES

[1] Tan, M., & Chen, X. "The Development Status and Prospect of Industrial Robots in China.", 2019.

[2] Pikner, H., Sell, R., Karjust, K., Malayjerdi, E., & Velsker, T, "Cyber-physical control system for autonomous logistic robot," In 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), pp. 699-704. April 2021.

[3] C. Follini, M. Terzer, C. Marcher, A. Giusti and D. T. Matt, "Combining the robot operating system with building information modeling for robotic applications in construction logistics", Proc. Int. Conf. Robot. Alpe-Adria Danube Region, pp. 245-253, 2020.

[4] Nassiraei, A. A., Kawamura, Y., Ahrary, A., Mikuriya, Y., & Ishii, K, "A new approach to the sewer pipe inspection: Fully autonomous mobile robot "kantaro". In IECON 2006-32nd Annual Conference on IEEE Industrial Electronics, pp. 4088-4093. IEEE. November 2006.

[5] McLaughlin, E., Charron, N., & Narasimhan, S, "Automated defect quantification in concrete bridges using robotics and deep learning." Journal of Computing in Civil Engineering, 34(5), 04020029.

[6] SHETA, A., & MOKHTAR, S. A, "Autonomous robot system for pavement crack inspection based CNN model." Journal of Theoretical and Applied Information Technology, 100(16), 2022.

[7] Pandey, A., Pandey, S., & Parhi, D. R, "Mobile robot navigation and obstacle avoidance techniques: A review." Int Rob Auto J, 2(3), 00022, 2017.

[8] Salvini, P., Paez-Granados, D., & Billard, A. "Safety concerns emerging from robots navigating in crowded pedestrian areas." International Journal of Social Robotics, vol. 14(2), pp. 441-462., 2022.

[9] Phang, D. R. Y., Lee, W. K., Matsuhira, N., & Michail, P., "Enhanced mobile robot localization with lidar and imu sensor." In 2019 IEEE International Meeting for Future of Electron Devices, Kansai (IMFEDK) (pp. 71-72), November 2019.

[10] Zhang, X., Lai, J., Xu, D., Li, H., & Fu, M. "2d lidar-based slam and path planning for indoor rescue using mobile robots. "Journal of Advanced Transportation, 2020.

[11] Ibrahim, A., Sabet, A., & Golparvar-Fard, M., "BIM-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform." In EC3 Conference 2019, Vol. 1, pp. 182-189, July 2019.

[12] Kim, P., Chen, J., Kim, J., & Cho, Y. K., "SLAM-driven intelligent autonomous mobile robot navigation for construction applications." In Advanced Computing Strategies for Engineering: 25th EG-ICE International Workshop 2018, Lausanne, Switzerland, June 10-13, 2018, Proceedings, Part I 25, pp. 254-269, Springer International Publishing, 2018.

[13] Hutabarat, D., Rivai, M., Purwanto, D., & Hutomo, H., "Lidar-based obstacle avoidance for the autonomous mobile robot," In 2019 12th International Conference on Information & Communication Technology and System (ICTS), pp. 197-202, July 2019.

[14] Li, H., Wang, Z., & Ou, Y., "Obstacle avoidance of manipulators based on improved artificial potential field method," In 2019 IEEE international conference on Robotics and Biomimetics (ROBIO), pp. 564-569, December 2019.

[15] Hutabarat, D., Rivai, M., Purwanto, D., & Hutomo, H., "Lidar-based obstacle avoidance for the autonomous mobile robot." In 2019 12th International Conference on Information & Communication Technology and System (ICTS), pp. 197-202, July 2019.

[16] Wang, Y., Goila, A., Shetty, R., Heydari, M., Desai, A., & Yang, H., "Obstacle avoidance strategy and implementation for unmanned ground vehicle using LIDAR." SAE international journal of commercial vehicles, vol. 10(1), pp. 50-56. 2017.

[17] Moffatt, A., Platt, E., Mondragon, B., Kwok, A., Uryeu, D., & Bhandari, S., "Obstacle detection and avoidance system for small uavs using a lidar." In 2020 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 633-640, September 2020.

[18] Wei, H., Li, X., Shi, Y., You, B., & Xu, Y., "Fusing sonars and LRF data to glass detection for robotics navigation." In 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 826-831, December 2018.

[19] Bell, D. A., Borenstein, J., Levine, S. P., Koren, Y., & Jaros, J., "An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance." In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pp. 2018-2022, May 1994.

[20] Arai, K., & Mardiyanto, R., "Autonomous control of eye based electric wheel chair with obstacle avoidance and shortest path finding based on Dijkstra algorithm." International Journal of Advanced Computer Science and Applications, vol. 2, pp. 19-25, 2011.

[21] Cho, G., Kim, J., & Oh, H., "Vision-based obstacle avoidance strategies for mavs using optical flows in 3-d textured environments." Sensors, vol. 19, pp. 2523, 2019.

[22] Kompella, V. R., & Sturm, P., "Detection and avoidance of semi-transparent obstacles using a collective-reward based approach." In 2011 IEEE International Conference on Robotics and Automation, pp. 3469-3474, May 2011.

[23] Guo-Hua, C., Jun-Yi, W., & Ai-Jun, Z., "Transparent object detection and location based on RGB-D camera." In Journal of Physics: Conference Series, vol. 1183, pp. 012011, March, 2019.

[24] Nakamura, S., & Takatori, Y., "Estimation of NLOS Obstacle Position Using Reflected Image on Transparent Surface." In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pp. 1656-1661, October 2022.

[25] Gaya, J. O., Gonçalves, L. T., Duarte, A. C., Zanchetta, B., Drews, P., & Botelho, S. S., "Vision-based obstacle avoidance using deep learning." In 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), pp. 7-12, October 2016.

[26] Talele, A., Patil, A., & Barse, B., "Detection of real time objects using TensorFlow and OpenCV." Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146, 2019.

[27] Wang, S., "Development and implementation of an obstacle avoidance algorithm for an Autonomous Mobile Robot," Doctoral dissertation, Politecnico di Torino, 2022.

[28] Leiva, F., Lobos-Tsunekawa, K., & Ruiz-del-Solar, J., "Collision avoidance for indoor service robots through multimodal deep reinforcement learning." In RoboCup 2019: Robot World Cup XXIII 23, pp. 140-153, 2019.

[29] Xiao, B., Chen, C., & Yin, X. "Recent advancements of robotics in construction." Automation in Construction, 144, 104591, 2022.

[30] Aldao, E., González-deSantos, L. M., Michinel, H., & González-Jorge, H., "UAV Obstacle Avoidance Algorithm to Navigate in Dynamic Building Environments," Drones, vol. 6(1), pp. 16, 2022.

[31] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861, 2017.

[32] Bekhet, S., & Alahmer, H., "A robust deep learning approach for glasses detection in non-standard facial images." IET Biometrics, vol. 10(1), 74-86., 2021.

[33] Pan, Z., Yang, J., Wang, X. E., Wang, F., Azim, I., & Wang, C., "Image-based surface scratch detection on architectural glass panels using deep learning approach.", Construction and Building Materials, 282, 122717, 2021.

[34] Ling, Z., Zhang, D., Qiu, R. C., Jin, Z., Zhang, Y., He, X., & Liu, H., "An accurate and real-time method of self-blast glass

insulator location based on faster R-CNN and U-net with aerial images." CSEE Journal of Power and Energy Systems, vol.5(4), pp. 474-482, 2019.

[35] Bilgin, M., & Mutludoğan, K., "Detecting transparency of glasses with capsule networks based on deep learning." In 2021 6th International Conference on Computer Science and Engineering (UBMK), pp. 157-162., September, 2021.