LiDAR-based Obstacle Detection and Distance Estimation in Navigation Assistance for Visually Impaired

Bineeth Kuriakose (✉), Raju Shrestha, and Frode Eika Sandnes

Department of Computer Science,
Oslo Metropolitan University,
Oslo, Norway
✉ bineethk@oslomet.no

Abstract. People with visual impairments can face challenges with independent navigation and therefore may use traditional aids such as guide dogs, white canes, or a travel companion for navigation assistance. In recent years, researchers have been working on AI-based navigation assistance systems. Obstacle detection and distance estimation are two of the key challenges in such systems. In this paper, we describe a LiDAR-based obstacle detection and distance estimation technique. A lightweight deep learning-based model called EfficientDet-LiteV4 is used for obstacle detection, and a depth map from the LiDAR is used to estimate the distance to the obstacles. We have implemented and tested the approach with the LiDAR integrated into a Raspberry Pi4 board. The results show good accuracy in detecting the obstacles and in estimating distance.

Keywords: Navigation · Visually Impaired · LiDAR · Deep Learning · Assistive Technology · Obstacle Detection · Distance Estimation

1 Introduction

Navigation or wayfinding for people with visual impairments is a prevailing challenge in the scientific community. Independent navigation could increase the level of independence [1]. However, travelling alone in unfamiliar envrionments can be challenging.

People with visual impairments typically use aids such as guide dogs, white canes, or depend on a travel companion. In addition to those conventional aids, diverse assistance systems and solutions have been proposed in the literature to address issues involved in the navigation of people with visual impairments [2]. Some of the main problems related to such systems are linked to portability and providing real-time environmental information in the immediate vicinity during navigation to avoid obstacles and prevent accidents [3].

In recent years, researchers have been actively exploiting artificial intelligence and machine learning to develop universally accessible navigation solutions [4]. In addition, different technologies and hardware are explored in the development

This is a post-peer-review, pre-copyedit version of a conference proceeding published in HCII 2022: Universal Access in Human-Computer Interaction. User and Context Diversity, 16th International Conference, UAHCI 2022, Held as Part of the 24th HCI International Conference, Proceedings, Part II, which is part of the Lecture Notes in Computer Science book series (volume 13309).
The final authenticated version is available online at DOI: https://doi.org/10.1007/978-3-031-05039-8_35

of navigation assistance systems. It might be inspired by the miniaturization of electronics and the advancement in processing power and sensing capabilities of various devices [1]. Among those, one prominent technology is LiDAR (Light Detection and Ranging) cameras.

LiDAR is a remote sensing technology that uses one or multiple laser beams to estimate distance measurements. LiDAR system sends a pulse of light and estimate distance based on the time it takes for the emitted pulse to return back. Some advantages of LiDAR sensors include high resolution and accuracy in measurements, easy conversion to 3D maps to interpret the environment, performance in low light conditions, and speed as it offers indirect distance measurements that do not need to be decoded or interpreted[1]. Because of these reasons, LiDAR technology has become a useful device for obstacle detection, avoidance, and safe navigation through various environments. LiDAR is commonly used in robotics and autonomous vehicles [5,6].

In this paper, we propose using a miniature LiDAR that can acquire visual image and depth information for accurate obstacle detection and distance estimation in a navigation assistance system for the visually impaired. A lightweight deep learning-based model called EfficientDet-LiteV4 is used for obstacle detection, and a depth map from the LiDAR is used to estimate the distance to the obstacles. We have assessed the performance and compared the results with our previous works, which use smartphone-based object detection and distance estimation methods for navigation assistance.

This paper is organized as follows. Section 2 discusses related works, and section 3 describes our proposed LiDAR-based obstacle detection and depth estimation methods and their implementation. Section 4 describes the experiment involved. Results and discussions are presented in section 5. The paper concludes in section 6.

2 Related Works

Obstacle avoidance is vital during navigation for visually impaired users. Obstacle or object detection involves identifying and locating obstacles in the environment, enabling a safe navigation. This section discusses related literature and notable developments in three areas: obstacle detection, distance estimation, and some literature reported on miniature hardware-based navigation systems and RGBD-based obstacle detection systems.

2.1 Obstacle Detection

Typically, there are two machine learning-based approaches used for obstacle detection in a navigation assistance system. In traditional machine learning (ML) based methods, computer vision techniques are used to look at various features of visual input data (typically image or video), such as the color histogram or edges,

[1] $www.leddartech.com/why-lidar/$

to detect and identify objects. On the other hand, modern deep learning-based methods employ convolutional neural networks (CNNs) to perform end-to-end object detection, in which features do not need to be defined explicitly but rather extracted automatically [7]. Because of this and the availability of ever-increasing computational capabilities required by deep learning models, researchers most recently tend to use deep learning models over traditional ML models.

A deep learning-based object detection model typically has three major components: a *backbone network* that extracts features from a given image; a *feature network* that has the backbone as the input and a list of fused features that denotes salient characteristics of the image as the output; and the *final class/box network* that uses the fused features to predict the object class and location of each object in the image.

Most of the popular object detection models belong to the Region-Based Convolutional Neural Network (R-CNN) family. This includes the models R-CNN, Fast R-CNN, Faster-RCNN, Mask R-CNN, etc. Over the years, they have become both more accurate and more computationally efficient [8]. One of the limitations of such models is their larger size and need of high computational power which limit their use in edge devices. Hence models belonging to the single-shot family are being started to be explored by researchers. Examples includes MobileNet+SSD [9], You Only Look Once (YOLO) [10] in several variants, SqueezeDet [11], etc. SSDs make great choices for models destined for mobile or embedded devices [4,12]. In this work, we use a relatively new, lightweight, and efficient object detection model, called EfficientDet-LiteV4 model [13]. Section 3.1 describes the model in more details.

2.2 Distance Estimation

In earlier times, most navigation assistance prototypes that provide distance information used ultrasonic sensors such as SR04 [1]. Ultrasonic sensors measure the distance of a target object by emitting ultrasonic sound waves and converting the reflected sound into an electrical signal. Typical disadvantages of conventional ultrasonic sensors include limited range, inaccurate readings, and inflexible scanning methods [14]. RGB-D cameras have started to be used in navigation systems to acquire depth information along with the color image. Major limitation in the RGBD depth-sensing technology is that it fails to capture depth information in four critical contexts: (1) distant surfaces (>5m), (2) dark surfaces, (3) brightly lighted indoor scenes, and (4) outdoor scenes with sunlight [15]. Furthermore, another limitation of currently existing RGB-D cameras is their size factor, which is comparatively more extensive, making it inconvenient to use in a portable navigation system. Still, researchers explored the option of RGB-D cameras for depth estimation in their navigation assistant prototypes [16,17]. Various smartphone-based distance estimation methods applied in navigation systems for visual impairments can be found in the literature [18]. In this work, we use an RGB-D camera that utilizes LiDAR technology to estimate the distance to the obstacles.

2.3 Miniature Hardware-based Navigation Systems

There are several navigation assistance systems reported in the literature which use various modes to process information about the obstacles, their type and/or distance. These systems use hardware such as Raspberry Pi, Arduino, Jetson, smartphones, or even a laptop connected with necessary components such as a camera for data processing and computation.

Rahman and Sadi [19] proposed a Single Shot Detector (SSD) model with MobileNet to recognize indoor and outdoor objects. The system consisted of a laser sensor that helps the user to identify directions. The system sends information collected to a remote server for processing. However, the authors did not explicitly mention the usage of such an arrangement and how they deal with privacy issues since the data might contain private and personal information such as images of people. Moreover, the model used for the obstacle detection was comparatively heavy-weighted, which could take a long execution time. Hence, it would not work as a practical solution in a real-time navigation environment. In a similar attempt, Joshi et al. [20] explored a Jetson nano-based system using MobileNet-SSD. The system provides only an overview of identified obstacles to the user without providing other relevant details such as distance to obstacles that are helpful during navigation.

Afif et al. [21] used the RetinaNet model for object detection in their proposed navigation system. Even though the model is claimed to provide high accuracy, the experimental evaluation showed high inference time, rendering it unsuitable for real-time operation. In another work [22], the authors used camera and time-of-flight sensors as its primary system components. The system's accuracy was low, and it was not intended for outdoors.

The system reported in [23] consisted of an ATmega328 microcontroller embedded with an Arduino Uno. An HC-SR04 ultrasonic sensor was used to identify obstacles. The primary limitations associated with the system were its inability to recognize types of obstacles and the use of ultrasonic sensors, which were not accurate compared to other modern distance estimation sensors.

Anandan et al. [24] described an outdoor and indoor navigation system for the visually impaired using Raspberry Pi. The system used SURF (Speeded Up Robust Features) algorithm for obstacle identification and ultrasonic sensors for distance estimation. The main limitation of the system was in the accuracy in detecting the obstacles.

2.4 RGBD-based Obstacle Detection Systems

Researchers also explored the potential of RGB-D-based cameras in navigation assistant systems for people with visual impairments. The Navigation assistance for visually impaired (NAVI) system proposed by Aladren et al. [16] used a consumer RGB-D camera to acquire both depth and visual information. The system uses RGB-D system to fuse range information and color information to detect obstacle-free paths. But it does not give much information such as the type of obstacle.

The authors in [25] put forwarded an indoor navigation system that uses a wearable RGBD camera mounted on head to construct a 2D map for the surrounding environment. An optimal path is generated fro the 2D map. The system also used an ultrasonic sensor to detect obstacles along the path. A Raspberry Pi 3 B+ board was used as the central processing unit. Even though the work mentioned path planning in detail, it did not explain how the RGB images captured from the camera were used for obstacle identification.

Lee and Medioni [17] also investigated the potential of an RGB-D camera in a navigation system. The RGB-D camera was placed in the user's eye position to capture scenes. A laptop was used for data processing. The major limitation of the system is the portability and inconvenience associated with the system due to the carriage weight of all the hardware [1]. The system creates indoor maps which guide the users to navigate. Like in [16], the system was also incapable of giving information about obstacles such as its type.

3 LiDAR-based Obstacle Detection and Distance Estimation

With the development of technologies, more and more miniature LiDAR cameras that can acquire both high-resolution color image and depth information simultaneously are available in the market. In this work, we use such a high-resolution miniature Intel RealSense LiDAR Camera L515[2] (see Fig. 1) for accurate obstacle detection and distance estimation. The camera can detect obstacles up to 9 meters and weighs only 100 grams. The low weight and small form factor make it suitable for specific applications such as navigation.

The Intel RealSense SDK 2.0[3] and other tools such as Intel RealSense API enable configuration, control, and access to the streaming data. It's extensive language support including Python makes it easy to implement the proposed solution.

The RGB image acquired with the LiDAR camera is sent to an object detection model to detect objects there in and their bounding boxes. The model and the methods we used for obstacle detection and distance estimation is described in the following subsections. After that, the preceding subsection describes the implementation done for testing and evaluation of the proposed methods.

3.1 Obstacle Detection

Our previous works explored the pre-trained MobileNet+SSD [12] and YOLOv5m [4] as obstacle detectors. Even though these models provided reasonably good accuracy, we look for a better alternative as new models are introduced to the scientific community to achieve more accurate detection results at the same time with minimum inference time.

[2] $www.intelrealsense.com/lidar-camera-l515/$

[3] $www.intelrealsense.com/sdk-2/$

In this work, we used an EfficientDet object detection model [13], which proved to be efficient and it can produce a reasonably good accuracy for detecting objects from an image/video. EfficientDet uses EfficientNet [26] as its backbone network for improved efficiency. EfficientNet is based on a CNN architecture and scaling method that uniformly scales all depth/width/resolution dimensions using a compound coefficient. Combining the new backbone and BiFPN (Bi-directional feature pyramid network), the small-sized EfficientDet-D0 base-line was developed, and then a compound scaling was applied to obtain EfficientDet-D1 to D7. Each consecutive model has a higher compute cost but provides higher accuracy.

EfficientDet-Lite is a derivative of EfficientDet trained on the MS COCO dataset [27], optimized for TensorFlow Lite and designed for mobile CPU, GPU, and EdgeTPU. The accuracy of *lite* models is comparatively less than conventional object detection models, which require high-end GPUs and processors. However, as a tradeoff, the computation time of conventional models is significantly higher than *lite* models. Moreover, while designing a real-time navigation solution, factors such as low inference time, small model size to deploy in a portable device, comparatively good accuracy should be considered [28]. Considering these, we have used the most recent version of the EfficientDet-Lite model, EfficientDet-LiteV4, to transfer learn and train with our custom dataset. The reason for choosing this version is the model's accuracy and size compared to other lightweight object detection models without compromising inference time[4].

Dataset: We have created a custom dataset for testing and evaluating the proposed obstacle detection model. The dataset consists of 15 object classes relevant to indoor and outdoor navigation scenarios, namely, *bed, bench, billboard, cabinetry, chair, door, fire hydrant, kitchen appliance, person, stairs, table, traffic light, tree, vehicle,* and *waste container*. Each of these classes has around 5000 images. The images were collected from various sources, which are publicly available such as Google Open Images [29], ImageNet [30], and images acquired by ourselves. After examining the extracted images, we found that many images require some preprocessing, such as labeling. Those images were labeled using the LabelImg[5] annotation tool. We used the PASCAL VOC data format to save the annotations from the images.

3.2 Distance Estimation

The bounding boxes of the objects on an RGB image detected by the object detection model are projected onto its corresponding depth image acquired by the LiDAR camera. The estimated distance of an obstacle from the camera/user is then calculated by median averaging the depth data within its bounding box.

[4] *www.github.com/google/automl/tree/master/efficientdet*

[5] *www.github.com/tzutalin/labelImg*

3.3 Implementation

TensorFlow Lite Model Maker[6] library was used to train the proposed object detection model using the custom dataset. The Model Maker uses transfer learning to reduce the amount of training data required and shorten the training time.

A Raspberry Pi4 board is used for implementing and testing the proposed methods. Intel RealSense LiDAR Camera is connected to the Raspberry Pi4, and power is supplied from a portable power bank, as shown in Fig. 1. Since a Raspberry Pi4 board is smaller and easier to carry than a heavier hardware device such as a laptop, we considered it a portable and practicable solution for navigation assistance. Only essential components such as the LiDAR camera, which is essentially needed for our application, are included in the experiments.

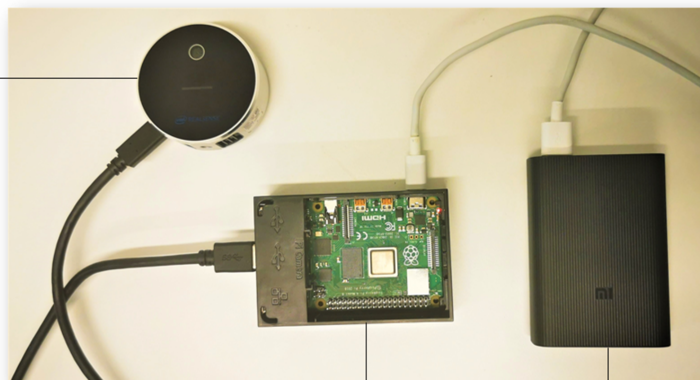

Fig. 1. A photo illustrating Intel RealSense LiDAR L515 depth camera connected to a Raspberry Pi4 board along with a portable power bank.

4 Experiments

This section elucidates how the experiments were conducted for obstacle detection and distance estimation.

4.1 Obstacle Detection

The object detection model was trained on an HP G4 Workstation with an Intel Xeon processor with 32GB RAM and NVIDIA GeForce GTX 1070 GPU. The platform settings of the experiment are TensorFlow-GPU 2.4, NVIDIA CUDA toolkit 11.0, and CUDNN 8.1. The model was trained, validated, and tested by

[6] *www.tensorflow.org/lite/guide/model_maker*

randomly shuffling and splitting the dataset in the ratio of 80:10:10, respectively. The default epochs in the Model Maker library[7] were 50. However, we run 100 epochs. The number was found to achieve better accuracy as determined through hyperparameter optimization using the validation dataset. The default batch size 64 was used. The training model was exported to *tflite* format. The Model Maker library applies a default post-training quantization technique when exporting the model to *tflite* format. This technique can help reduce the model size and inference latency while improving the CPU and hardware accelerator inference speed[8].

4.2 Distance Estimation

To evaluate the performance of the distance estimation, we tested distance measurements on five different types of obstacles, *billboard, chair, waste container, door,* and *table*. The obstacles were placed at different distances, and the actual distance of an obstacle from the camera/user was measured using a measuring tape. The measurement was done to the nearest edge point of the objects. The obstacles with varied sizes were chosen intentionally to check whether the size of the obstacles affects distance estimation. The *waste container* obstacle we considered in this experiment was smaller. We also tried to check if the sunlight affects the distance estimation by placing a *chair* outdoor under direct sunlight.

5 Results and Discussions

Fig. 2 illustrated object detection and depth estimation results where RGB images and depth map images are given with the bounding boxes around the detected objects are marked. The elaborated results from obstacle detection and distance estimation methods are given and discussed in the following subsections.

5.1 Obstacle Detection

Table 1 shows the results from the object detection model in terms of prediction accuracy of the 15 object classes in the custom dataset. The prediction results were reasonably good, with an average accuracy of around 88%. The results show that *cabinetry* and *stairs* are the only two object classes where accuracy is below 80%. The quality of images and annotations could be reasons for the low accuracy in those two classes. We observed that in some cases, the model failed to detect objects correctly because of similarities in some object classes. For example, there were false detections between white *doors*, walls, and long *billboard*. This was probably because of the pattern similarity in those objects.

Even though the class categories in the dataset for obstacle detection are limited (15), we tried to include object classes relevant to the navigation scenario.

[7] $www.tensorflow.org/lite/guide/model\_maker$

[8] $www.tensorflow.org/lite/performance/post\_training\_quantization$

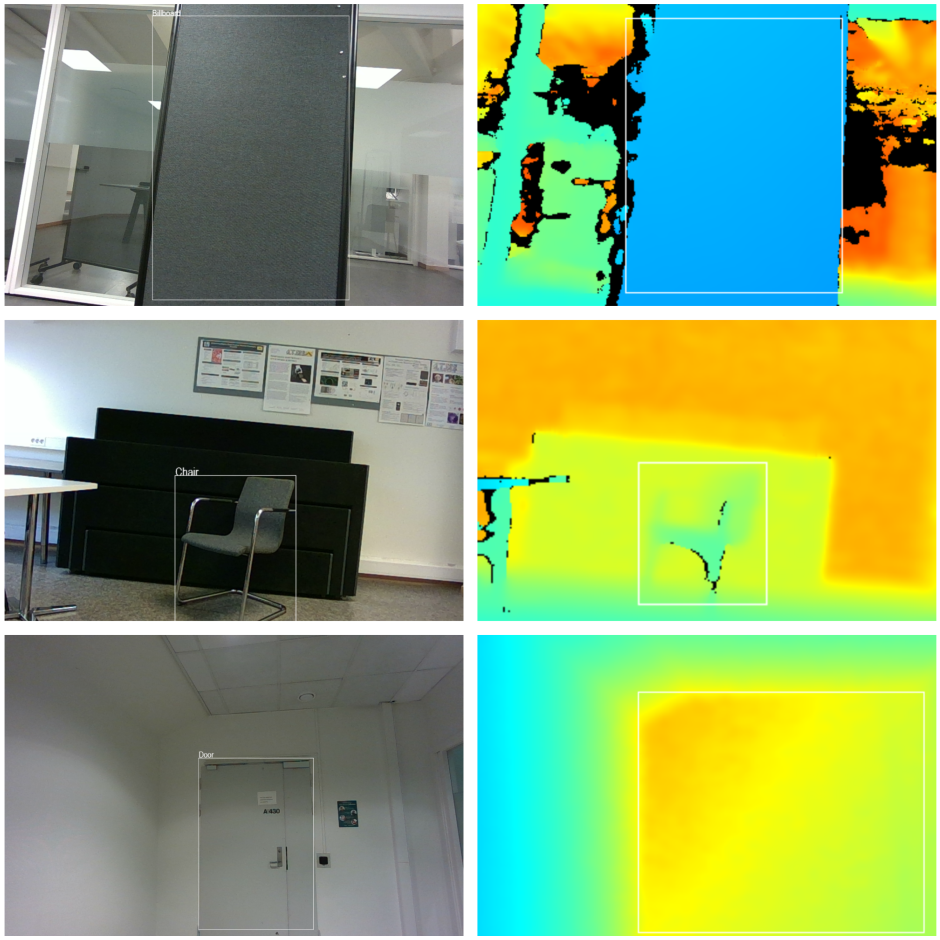

Fig. 2. (a) (Left column) Three different obstacles/objects (*billboard, chair*, and *door*) on the RGB images as detected and marked with bounding boxes by the object detection model. (b) (Right column) Depth map images with the marked bounding boxes around the obstacles after mapping with corresponding RGB images.

Since this is a proof-of-concept, it is possible to extend this with more classes in the future. The average accuracy of around 88% is not the best compared to other heavy-weight object detection models, and it is anticipated from the lightweight model used. Nevertheless, considering the various aspects required for a real-time navigation application (see section 3.1), the proposed obstacle detection model's performance could be considered reasonably good, as low computation time enables real-time environmental information without much delay than conventional models. The results from the model also show it has fewer parameters (29M) and model size (49MB), which makes it possible to deploy in a less-powerful portable device such as Raspberry Pi.

Table 1. Performance of the proposed obstacle/object detection model in terms of percentage accuracy.

Obstacle	Accuracy(%)
Bed	95.4
Bench	93.7
Billboard	84.6
Cabinetry	78.4
Chair	94.7
Door	83.2
Fire Hydrant	88.9
Kitchen Appliance	92.7
Person	84.7
Stairs	79.5
Table	90.9
Traffic Light	83.7
Tree	81.5
Vehicle	94.1
Waste Container	94.6
Average	87.6

5.2 Distance Estimation

Table 2 shows the actual and estimated distance of the four obstacles from our experiment from the proposed method. It also provides estimated distance from the best method, the *Rule of 57*, from among the various smartphone-based distance estimation methods from our previous work [18]. The results show that the proposed method can estimate distance more accurately compared to the *Rule of 57* method.

Table 2. Observations of distance estimation from various obstacles (all in centimeters).

Obstacle	Actual Distance	Estimated Distance (Proposed method)	Estimated Distance (*Rule of 57* [17])
Billboard	100	100.0	74.8
Chair	200	200.0	209.6
Waste Container	300	299.5	312.4
Door	500	500.1	485.0
Table	900	898.9	Unable to estimate

Another advantage of using LiDAR cameras for distance estimation compared to smartphones is that LiDARs can detect obstacles at longer distances (900cm in the case of the LiDAR camera used in this work). The smartphone-based method had a distance limit of 500cm. Moreover, the results were not consistent at a 500cm distance. Therefore, the estimated distance in the case of *door* at 500cm from the smartphone-based method was recorded by averaging five readings due to its fluctuating nature. Smartphone-based also failed to report any result when the obstacle was placed at less than 100cm. One disadvantage with the LiDAR camera-based method is that it needs to be connected to a computer (e.g., Raspberry Pi). This could raise portability concerns and cause inconvenience to the user. On the other hand, the portability factor is positive for the smartphone-based method. Therefore, one could note this tradeoff while designing a navigation assistant system while making a design choice.

When the smallest obstacle from the test set, the *waste container*, was placed at 50cm, the smartphone-based method could not detect any result. However, the proposed solution using LiDAR gave the result as 49.5cm. When we experimented with the *chair* obstacle placed outdoor under the sunlight, the obstacle detection model was able to detect the obstacle as a *chair*. However, the distance estimation method failed to estimate the distance well. It estimated a distance of 281.7cm against the actual distance of 300cm. Surprisingly, the smartphone-based method also showed similar results, with an estimated distance of 280.4cm. The performance degradation with the LiDAR method could be because of interference to its depth estimation system from the infrared light from the sun. This issue is cross-checked with the manufacturer's website and found that this LiDAR camera is recommended for indoor environments[2]. However, in other applications such as autonomous vehicles, LiDARs are used together with other devices such as laser reflectors, radars, and stereo cameras to address the influence of sunlight[9]. 3D LiDARs, which are very expensive compared to 2D LIDARs, can also solve the issue to a certain extent[9]. Nevertheless, in application scenarios such as navigation assistance, where portability is also a significant concern, installing additional reflectors or devices only to use outdoors may not be a preferred design choice.

6 Conclusion

The proposed LiDAR-based method, which used an EfficientDet-LiteV4 model, shows reasonably good performance in terms of obstacle detection and distance estimation indicating its potential to be used in a navigation assistance system for individuals with visual impairments. Using a LiDAR camera connected with a Raspberry Pi and a power bank asks for proper camera placement and needs for carrying the hardware, which might raise portability concerns. And the performance degradation of the LiDAR cameras when it is used under bright sunlight could limit their use in outdoor navigation. We believe this research could bring valuable insights to the use of LiDARs in portable navigation assistance solutions for the visually impaired.

[2] $www.intelrealsense.com/lidar-camera-l515/$

[9] $www.sevensense.ai/blog/localization$

References

1. Kuriakose, B., Shrestha, R., Sandnes, F.E.: Tools and Technologies for Blind and Visually Impaired Navigation Support: A Review. IETE Technical Review pp. 1–16 (2020)
2. SSMR, U.o.S.: Understanding the Needs of Blind and Partially Sighted People: their experiences, perspectives, and expectations (May 2009), https://www.rnib.org.uk/knowledge-and-research-hub/research-reports/general-research/understanding-needs
3. Kuriakose, B., Shrestha, R., Sandnes, F.E.: Multimodal Navigation Systems for Users with Visual Impairments—A Review and Analysis. Multimodal Technologies and Interaction 4(4), 73 (2020)
4. Kuriakose, B., Shrestha, R., Eika Sandnes, F.: Towards Independent Navigation with Visual Impairment: A Prototype of a Deep Learning and Smartphone-based Assistant. In: The 14th PErvasive Technologies Related to Assistive Environments Conference. pp. 113–114 (2021)
5. Han, J., Kim, D., Lee, M., Sunwoo, M.: Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor. IEEE Transactions on Vehicular Technology 61(3), 971–985 (2012)
6. Peng, Y., Qu, D., Zhong, Y., Xie, S., Luo, J., Gu, J.: The obstacle detection and obstacle avoidance algorithm based on 2-D Lidar. In: 2015 IEEE international conference on information and automation. pp. 1648–1653. IEEE (2015)
7. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object Detection in 20 years: A Survey. arXiv preprint arXiv:1905.05055 (2019)
8. Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X.: Object Detection with Deep Learning: A Review. IEEE transactions on neural networks and learning systems 30(11), 3212–3232 (2019)
9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot multibox Detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
11. Wu, B., Iandola, F., Jin, P.H., Keutzer, K.: Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 129–137 (2017)
12. Kuriakose, B., Shrestha, R., Sandnes, F.E.: Smartphone Navigation Support for Blind and Visually Impaired People-A Comprehensive Analysis of Potentials and Opportunities. In: International Conference on Human-Computer Interaction. pp. 568–583. Springer (2020)
13. Tan, M., Pang, R., Le, Q.V.: EfficientDet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
14. Borenstein, J., Koren, Y.: Obstacle avoidance with ultrasonic sensors. IEEE Journal on Robotics and Automation 4(2), 213–218 (1988)
15. Litomisky, K.: Consumer RGB-D cameras and their applications. Rapport technique, University of California 20, 28 (2012)
16. Aladren, A., López-Nicolás, G., Puig, L., Guerrero, J.J.: Navigation assistance for the visually impaired using RGB-D sensor with range expansion. IEEE Systems Journal 10(3), 922–932 (2014)

17. Lee, Y.H., Medioni, G.: RGB-D camera based wearable navigation system for the visually impaired. Computer Vision and Image Understanding 149, 3–20 (2016)
18. Kuriakose, B., Shrestha, R., Sandnes, F.E.: Distance Estimation Methods for Smartphone-Based Navigation Support Systems. In: Proceedings of SAI Intelligent Systems Conference. pp. 658–673. Springer (2021)
19. Rahman, M.A., Sadi, M.S.: IoT Enabled Automated Object Recognition for the Visually Impaired. Computer Methods and Programs in Biomedicine Update 1, 100015 (2021)
20. Joshi, R., Tripathi, M., Kumar, A., Gaur, M.S.: Object recognition and classification system for visually impaired. In: 2020 International Conference on Communication and Signal Processing (ICCSP). pp. 1568–1572. IEEE (2020)
21. Afif, M., Ayachi, R., Said, Y., Pissaloux, E., Atri, M.: An evaluation of RetinaNet on indoor object detection for blind and visually impaired persons assistance navigation. Neural Processing Letters 51(3), 2265–2279 (2020)
22. Noman, M., Stankovic, V., Tawfik, A.: Portable offline indoor object recognition system for the visually impaired. Cogent Engineering 7(1), 1823158 (2020)
23. Ramadhan, A.J.: Wearable smart system for visually impaired people. Sensors 18(3), 843 (2018)
24. Anandan, M., Manikandan, M., Karthick, T.: Advanced indoor and outdoor navigation system for blind people using Raspberry-Pi. Journal of Internet Technology 21(1), 183–195 (2020)
25. Hakim, H., Fadhil, A.: Indoor Wearable Navigation System Using 2D SLAM Based on RGB-D Camera for Visually Impaired People. In: Proceedings of First International Conference on Mathematical Modeling and Computational Science. pp. 661–672. Springer (2021)
26. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: European conference on computer vision. pp. 740–755. Springer (2014)
28. Kuriakose, B., Shrestha, R., Sandnes, F.E.: SceneRecog: A Deep Learning Scene Recognition Model for Assisting Blind and Visually Impaired Navigate using Smartphones. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 2464–2470. IEEE (2021)
29. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., et al.: OpenImages: A Public Dataset for Large-Scale Multi-Label and Multi-class Image Classification. Dataset available from https://github. com/openimages 2(3), 18 (2017)
30. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009)