Smart Head-mount Obstacle Avoidance Wearable for the Vision Impaired

Peijie Xu, Ron Van Schyndel *, and Andy Song

School of Computing Technologies, RMIT University, Melbourne 3000, Australia
peijie.xu@student.rmit.edu.au, andy.song@rmit.edu.au

Abstract. Obstacles present serious risks and dangers for individuals who are blind or visually impaired (BVI), especially when they are not accompanied by a companion or assistant. In this study, we propose a head-mounted smart device to address this challenge. This study aims to establish a computationally efficient mechanism that can accurately detect the presence of obstacles on the path and provide warnings in real-time. The learned obstacle warning model needs to be reliable and small in size so that it can be embedded in the wearable device and run without consuming too much energy. Moreover, it must be able to deal with natural head turns, which can significantly impact readings from the head-mounted sensors. To determine the most appropriate model that can balance accuracy and real-time performance, we investigated more than thirty models and compared their key metrics. Our study demonstrates that a highly efficient wearable device is feasible and can help BVI individuals avoid obstacles with high accuracy. Additionally, we have collected a large data set that can serve as a benchmark for future studies in this area.

Keywords: Smart wearables · Obstacle avoidance · Vision impaired · BVI · Supervised machine learning.

1 Introduction

Blind and visually impaired people (BVI), disadvantaged groups of our society, are globally estimated to be 43.3 million and 295 million, respectively, according to the 2020 stats [1]. For BVI people, navigating themselves from one place to another is a real challenge. A key part of the challenge is obstacles on their paths, posing a serious risk for a BVI person. Detecting obstacles and other hazards in real-time can greatly improve the mobility of BVI people, reducing their exposure to dangers. To help them, vision researchers and engineers have invested a large amount of effort into this area. In this study, we are to provide a wearable solution as a candidate choice for them. The solution is to be smart and fast. We introduce a low-energy consumption device that can accurately warn of potential risks, e.g., obstacles, but not alarm falsely on harmless objects

* This author passed away prior to the submission of this paper. This is one of the last works of him.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

directly ahead. To be practical for BVI people, the detection must be fast and lightweight, running on the device itself.

The following parts of this paper are organised as such. A review of previous works on obstacle avoidance is presented in Section 2. The proposed head-mount smart device is described in detail in Section 3. Sections 4 and 5 explain our experiments including how sensor data are collected through the wearable device under a controlled environment, how they are labeled, and how learning algorithms are formulated and evaluated. The results of the experiments are in Section 6, while the discussions of this study are presented in Section 7. Section 8 concludes this study with a vision for future work.

2 Literature review

In the past, a large number of approaches have been proposed to provide environment information to a level that can assist a BVI person to navigate around for daily activities [2]. These approaches can be grouped into three categories, which are Electronic Travel Aids (ETA), Electronic Orientation Aids (EOA) and Position Locator Devices (PLD). ETA focuses on perceiving and translating the information of the surrounding environment of the users, while EOA aims to help the BVI person maintain an accurate orientation during travel. On the other hand, PLD is to provide the position information of the person or the target in the scene [2]. In this study, we mainly review the relevant literature in the field of ETA as that is the aim of this study, although our proposed methodology could be transferred to EOA and PLD.

The effort on supplementing or replacing the white cane with ETA started around the nineteen forties [3]. After fifty years of development, electronic travel assistance for BVI people converged much more towards the navigation function [3]. Ran et al. designed a wearable assistance, Drishti, to provide dynamic interactions and adaptability to changes. This approach realised the seamless switching between indoor, outdoor environments and bus station navigation through differential GPS for the outdoor environments. The original equipment manufacturer's ultrasound sensor provides an accuracy of 22 centimetres [4]. Such low accuracy was far from meeting the needs of reliable real-time obstacle avoidance. Bousbia-Salah et al. proposed a navigation aid that relied on memory function with an integral accelerometer, computing and recording walking distance as a type of guidance [5]. Two vibrators and two ultrasonic sensors are mounted on the user's shoulders for obstacle detection. Another ultrasonic sensor was integrated into the white cane. However, this system required a sighted individual to accompany the BVI person, who was also required to carry a cane to complete the first navigation route. In addition, cumulative tracking errors were not well handled and often led to system failure after a period of operation.

Ultrasonic sensors have been popular in sensor-based solutions in the past decades. These sensors have high sensitivity and penetrative ability, hence suitable for obstacle detection tasks [6, 7]. For example, "NavBelt" was capable of scanning a range of 120° by arranging eight ultrasonic sensors on the abdomen.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

The signals from the ultrasonic sensors were then processed by the robotic obstacle avoidance algorithms, of which the outputs were acoustically delivered to users [10]. The limitation of this system is that it cannot reliably detect obstacles above the user's head. Gao et al. created a wearable virtual cane network composed of four ultrasonic sensors on two wrists, the waist, and one ankle of the user. It is designed to detect obstacles as small as 1 cm^2 in size and located 0.7 meters away [9]. A more recent study proposed a wearable assistive device with a glass frame supporting three ultrasonic sensors and a hand band equipped with a LiDAR sensor [8]. A fuzzy decision support system was integrated and achieved better obstacle avoidance performance than the conventional white cane for both outdoor and indoor environments. They reported reduced average walking time and reduced number of collisions in their experiments. However, the participants collided with some small harmless obstacles during the tests [8]. The two studies above both combined the device with the upper limb, but they did not consider the natural swing of the arm during walking.

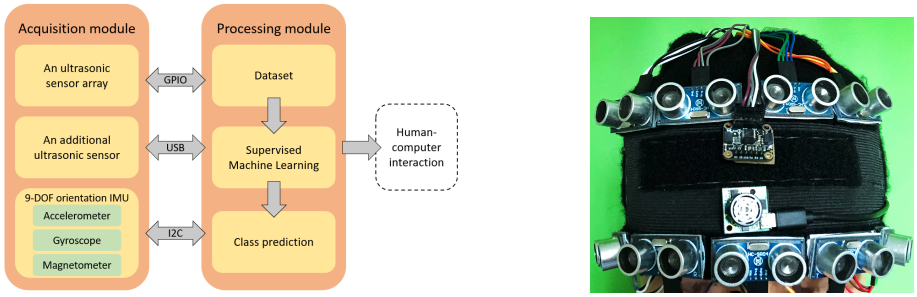
With the proliferation of technologies like edge computing, smart sensors and AI, navigation assistance for BVI people attracts more researchers and engineers, aiming for an ultimate intelligent commericalisable wayfinding and mobile navigation mechanism. Many state-of-the-art techniques such as Radio Frequency Identification (RFID)-based model [12], Augmented Reality (AR) technology [13] or cloud system [14, 15, 17] were gradually adopted into the development of this space. But these techniques often require heavy computation power or external support (e.g., 5G, e-tags etc.). Vision sensors, especially with depth detection function, gradually become the most popular perceptive method in many systems, due to the low cost of these sensors and the fast development in deep learning-based computer vision models. Hicks and colleagues integrated a depth camera, a small digital gyroscope, and an LED display on a ski goggle to help people with poor vision utilise their functional residual vision to navigate around [18]. Yang et al. enhanced close obstacle detection by expanding the preliminary traversable area through a seeded growing region algorithm, which can break the limit of narrow depth field angle and sparse depth map [19]. A research utilised the depth and colour information captured by a consumer-grade RGB-D camera to segment the unobstructed paths in the scene [20]. They reported that the obstacle-free path segmentation algorithm could run at a rate of 2 frames per second (FPS), while the whole system including RGB image and depth data processing and user interface generation run much slower at 0.3 FPS. Lee and Medioni proposed a novel wearable navigation system based on a combination of an RGB-D camera, a laptop, a smartphone user interface, and a haptic feedback vest [21]. The system estimated a real-time ego-motion by sparse visual features, dense point clouds, and the ground plane and create a 2D probabilistic occupancy grid map for dynamic path scheme and obstacle avoidance. The heavy equipment required in this setup however compromises its usability. Other researchers also tried to adopt more accurate vision algorithms such as SSD [23], YOLO [22] on obstacle recognition scenarios with high definition camera to help BVI people [6, 24, 25]. However, the problem with all these works is the large

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

computational cost required to carry out the detection, typically a high-end laptop. Obviously, it is impractical for a BVI person to walk around with a laptop all the time. Our aim is to provide a lightweight yet real-time detection method. Hence cameras are not used in this study.

3 System Design

The system proposed in this study is a head-mount smart device. It is comprised of two main modules: the acquisition module and the processing module as shown in Fig. 1. The former is to perceive the surroundings through ultrasonic sensors and a 9-DOF orientation inertial measurement unit (IMU). These sensors are all connected to the processing module, which is responsible for data acquisition, computation, and decision-making processes. In this study, the module is a Raspberry Pi 4B, which is highly portable and versatile.



(a) System architecture (b) The actual prototype

Fig. 1: The proposed head-mount obstacle avoidance system

The ultrasonic sensor array consists of nine sensors arranged in two rows. Four of the sensors in the top row are to detect obstacles in the upper region, while the remaining five sensors are for lower areas. These sensors are HC-SR04, with an effective detection angle of 15°, a maximum detection distance of 400 cm and a minimum of 2 cm[1]. Their accuracy is up to 3 mm. All these sensors are fully integrated with ultrasonic transmitters and receivers. The proposed system also incorporates an additional ultrasonic sensor, the MaxSonar-EZ1 from MaxBotix [2]. This sensor is placed in the middle to supplement the coverage. It communicates directly with the processing module via a USB port. This sensor has a range of 30 cm to 500 cm and is featured with compensation for target size variations, well-balanced sensitivity and specificity, built-in noise reduction,

[1] HC-SR04 Specs: https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf
[2] HRUSB-MaxSonar®-EZ™ Series: https://www.maxbotix.com

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

real-time background auto-calibration, real-time waveform analysis, and noise rejection. This sensor is expensive but helps validate readings from the sensor array. Note the sensor set position against the subject head is not guaranteed. So the learning algorithm needs to be able to handle such variations.

IMU is to measure the user's movement e.g., acceleration and rotation. Our system uses Adafruit BNO085[3], which integrates accelerometer, gyroscope and magnetometer. It communicates with the processing module via the I2C bus, transmitting three axes of linear acceleration data, three axes of gravitational acceleration data, and three axes of magnetic field sensing readings. The raw data are converted into the form of a four-point quaternion or rotation vector. That can reduce the computing burden on the processing module and minimise the impact of drift errors.

4 Data Collection

The data for the study is collected in a 15-meter-long corridor under temperature and humidity control, as depicted in Fig. 2. This study focuses on the indoor environment. Outdoor environment has a great amount of uncertainty and variations, and will be addressed in our further studies. One side of the corridor is a clean wall, while the other side is furnished with long sofas. The floor is marked 1-meter and 1.5-meter parallel to the walls, with 0.5-meter and 0.75-meter spacings between them. These marks are to guide participants walking at different paces and speeds. Additionally, angles of 75 and 60 degrees relative to the wall are marked to guide participants walking while looking sideways [4].

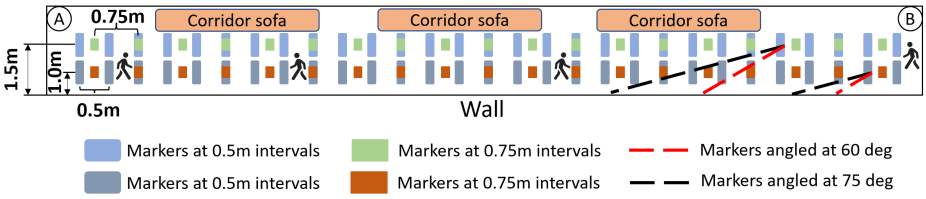


Fig. 2: The setup for data collection

Three healthy participants were invited to assist with our data collection. Firstly, they put on the head-mount device shown in Fig. 1, and adjust it to a comfortable position. We then check whether the device is functioning properly or not, and make sure the orientations of these sensors are aligned correctly. Data are collected while participants complete the following actions:

[3] Adafruit 9-DOF Orientation IMU BNO085 https://cdn-learn.adafruit.com

[4] The experiments are conducted under human research ethics approval. Following the Declaration of Helsinki, participants are informed about the details of the study, including the purpose, potential risks, and obligations.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

1. walking from Point A to Point B, then returning to Point A, at different speeds: 0.5 m/s; 0.75 m/s; 1.0 m/s;
2. walking from Point A to Point B, then returning to Point A, at different speeds: 0.5 m/s; 0.75 m/s; 1.0 m/s, but with other pedestrians same walking pass in the same direction or the opposite direction;
3. walking from Point A to Point B, then returning to Point A, at different speeds: 0.5 m/s; 0.75 m/s; 1.0 m/s, with the head pointing/looking to the side;
4. walking from Point A to Point B, then returning to Point A, at different speeds: 0.5 m/s; 0.75 m/s; 1.0 m/s, with the head kept turning side to side (e.g., looking around with head turns);
5. walking from Point A or Point B towards the wall at different angles: 60 deg; 75 deg.

4.1 Data processing

All sensory data are collected with timestamps. These data are not synchronised due to the different sampling frequencies of these sensors. We manually go through timestamped data to check the validity of these data points. Redundant and duplicate data samples are removed. In total 10,234 entries of valid data are collected from three participants. That formulates our data set for the next stage of our study. Each entry contains 20 separate attributes, which are three axes of linear acceleration, three axes of gravitational acceleration, four attributes from quaternion, one distance reading from the MaxSonar ultrasonic sensor, and nine distance readings from the ultrasonic sensor array. On average, one second of movement generates 30 data points. That is about the sampling frequency of the ultrasound sensor array.

4.2 Data labeling

To facilitate subsequent learning, all data entries are labeled with "0" and "1" to indicate the absence or the presence of obstacles that may be a risk for a BVI person. The labeling is determined according to the literature in BVI studies [10, 29]. An obstacle on the pathway of the person within a distance of 1.5 m is considered positive, e.g., a risk. An obstacle that is more than 1.5 m away or is not on the trajectory of the person, e.g., on the side or way above, is labeled as zero. The processed data set contains 5,174 valid positive entries and 5,060 zero cases. That is to avoid class imbalance which may lead to bias during learning.

5 Methodology

Real-time obstacle avoidance faces three challenges, which are the delay in processing the sensory input in the ETAs system, the delay in presenting warning signals or recommendations to the BVI user, and the delay in the user's response to the signals. This study mainly deals with the first two as the third is not in

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

the scope of computing. In order to satisfy the requirement for real-time performance, tasks such as data acquisition, pre-processing, and decision-making need to be accomplished within a very short period of time. Hence we adopt parallelled multithread processing in our system architecture. The control of data acquisition of each sensor is separately managed as a thread. The controls of different sensors do not interfere with each other but run in a synchronised way. Due to the hardware characteristics of these sensors, they operate at different frequencies, including sampling frequency, response frequency, and feedback frequency. It would be detrimental if a certain sensor reading was blocked by the task scheduled in front of it, which may occur often if the parallel mechanism is not in place. Multithreading can effectively avoid these problems [26]. Also, multithreading facilitates resource sharing during the process. The threads share the memory and CPU time, which are scarce on the devices, e.g., Raspberry Pi that we use for this study. More specifically, the processing unit is a 64-bits Raspberry Pi OS (Raspbian) with Linux kernel 5.10.

5.1 Learning models

To build a good obstacle detection model, we investigate a range of machine learning methods, more specifically a set of classification algorithms, that include (1) ZeroR, (2) C4.5 decision tree, (3) Naive Bayes, (4) k-nearest neighbour algorithm (kNN) and (5) Multilayer Perceptron. These methods are the most well-known and arguably the best-performing classifiers. In addition, we engage ensemble algorithms which can combine multiple classifiers together. Such approach often can improve the performance in comparison with an individual classifier. In this study we use (1) Bagging, (2) Random Forest, (3) AdaBoost, (4) Vote and (5) Stacking. These models are all evaluated through ten-fold cross-validation, a systematic way of repeated holdout that can pragmatically reduce the variance of the estimate [27]. Another factor to consider is the hyper-parameters as most of these learning methods are associated a set of hyper-parameters, for example, the size of leaf nodes in C4.5, the type of connections in multilayer perceptron. To optimise the performance of these algorithms, we conducted a set of prior empirical study on each model to find out the best combination of hyper-parameter for the model. All algorithms we used are from the Weka package, which supports a wide range of learning methods and the ten-fold cross-validation evaluation. Based on our empirical pre-study, the batch size of all the algorithms is set as 100 to ensure the learning quality.

To further compare the performance, we introduce shallow and deep learning models as well in this study, as deep learning has demonstrated its superb capability in many complex machine learning tasks such as machine vision and natural language processing. Hence this study also includes deep learning to verify its suitability for fast obstacle detection. Two binary deep neural network classifiers are therefore established. One is a 20-(10)-1 network and another is a 20-(10-10)-1 network. This means both have 20 input nodes and one output node. The former has one hidden neural layer with 10 nodes and the latter has two such hidden layers. Note, too many layers would seriously slow down

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

the process, hence are not used in this scenario. The loss function of the network is binary cross-entropy loss, which has a strong coupling with the Sigmoid function. Stochastic gradient descent optimisation with a fixed learning rate is used to control the amount of change in weights and biases. The ten-fold cross-validation is also used here to evaluate the performance of this network model on our data set. The realisation of this model is based on PyTorch, one of the mostly used deep learning frameworks. It enables us to train our deep learning models through GPUs and CPUs in an optimised tensor library [28][5].

5.2 Model evaluation and analysis

The aforementioned models are compared using several important metrics, including accuracy, precision, recall, F1-Score (F-Measure) and mean absolute error (MAE). Accuracy measures the proportion of the correctly classified instances (true positives (TP) and true negatives (TN)) among the total predictions (the sum of TP, TN, false positives (FP) and false negatives (FN)) of the data set. Precision computes how close the true predictions are to the positive situation, the sum of TP and FP, e.g., $TP/(TP + FP)$. It means how many of those classified as obstacles are actual dangers ahead. On the other hand, recall evaluates the correct accuracy of prediction over the actual positive instances in the data set, which implies the ratio of the prediction classified as obstacle ahead over all the dangerous instances, $TP/(TP+FN)$. F1-Score is the harmonic mean of the precision and recall, which reflects the balance of the precision and recall, $2TP/(2TP + FP + FN)$. MAE measures the number of misclassification in the model, $\Sigma_{i=1}^{n}|error_i|/n$, where $error_i$ is the deviation from model predictions.

Another set of key indicators are efficiency related, including the model construction or training time, the model execution time, and the size of the model, as the real-time performance of obstacle avoidance is as critical as the detection accuracy. When measuring the model execution time, the trained models are applied to make predictions on a data set with 500 entries on the device, e.g., the Raspberry Pi. The average time of ten executions is calculated as the model execution time. Model sizes are also measured as large models often consume significantly more computational resources, which is not desirable due to the limited resources available on a device for data gathering and computing. Hence, a good model is considered to be high in accuracy, precision, recall and F1-Score, low in MAE, model construction time, execution time, and model size.

6 Experimental results

6.1 Hyper-parameter tuning

Table 1 shows different combinations of hyper-parameter settings in the Decision Tree, the minimum number of instances per leaf varying from 2, 10 to 50, and

[5] All learning is conducted using a desktop computer with an i7-12700K 3.60 GHz CPU, 16.0GB RAM, and NVIDIA GTX 1060 6GB GPU. WEKA version is 3.8.6. PyTorch version is 1.13.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

the confidence factor varying from 0.25, 0.50, to 0.75. It can be seen that the performance increases as the model grows in size. With minimum instances in each leaf being 2, and a confidence factor of 0.5, the accuracy can reach 98.68%. It is worth noting that an increase in confidence factor, e.g., to 0.75, resulted in a doubled training time, especially when the minimum number of instances on each leaf is as small as 2. The model size of the generated trees is however not severely affected. From the above results we can see the optimal setting for the decision tree is 2 and 0.5.

Table 1: Example of hyper-parameter tuning - Decision Tree

Min num of objects	Confidence factor	Accuracy (%)	Construction time (s)	Model size (KB) (# Leaves/Tree size)
2	0.25	98.64	0.14	50 (127 / 253)
2	**0.5**	**98.68**	**0.14**	**53 (134 / 267)**
2	0.75	98.66	0.33	53 (134 / 267)
10	0.25	97.41	0.14	29 (68 / 135
10	0.5	97.42	0.14	29 (69 / 137
10	0.75	97.42	0.26	30 (71 / 141)
50	0.25	94.38	0.13	15 (29 / 57)
50	0.5	94.44	0.13	15 (29 / 57)
50	0.75	94.44	0.17	15 (29 / 57)

Table 2 shows the different settings for the Naive Bayes, using kernel estimator or not, using supervised discretisation parameters or not. The model reaches 91.19% accuracy when using the supervised discretisation only. That took it 0.15 seconds to build the model. When the supervised discretisation is disabled, the training time can be reduced quite significantly. However, the performance and the model size are not ideal. With the kernel estimator, the model size increased more than 18 times, from 38 to 715 KB. When the kernel estimator is off, the accuracy dropped below 80%.

Table 2: Example of hyper-parameter tuning - Naive Bayes

Use kernel estimator	Supervised discretisation	Accuracy (%)	Construction time (s)	Model size (KB)
False	False	79.74	0.04	7
True	False	89.25	0.05	715
False	**True**	**91.19**	**0.15**	**38**

Table 3 shows the settings of kNN, e.g., the nearest neighbour, using three distance measures, Euclidean distance, Manhattan distance and Chebyshev distance; and different k values, 1, 3 and 5. The nearest neighbour algorithm does not require training time. However the execution of the model involves all data instances. In this sense, the model size is constant and undesirably large. With Manhattan distance and $k = 1$ the accuracy can reach 99.46% validation accuracy without much overfitting.

The result for Multilayer Perceptron is presented in Table 4. The training is done by standard backpropagation using the Sigmoid function. In the table, 'a' means the number of perceptrons in a hidden layer is half of the sum of the number of attributes and classes in the data set. Likewise, 'a, a' and 'a, a, a'

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

Table 3: Example of hyper-parameter tuning - kNN

Distance function	k value	Accuracy (%)	Model size (KB)
Euclidean	1	99.11	1,974
Euclidean	3	97.40	1,974
Euclidean	5	96.98	1,974
Manhattan	**1**	**99.46**	**1,974**
Manhattan	3	98.29	1,974
Manhattan	5	98.14	1,974
Chebyshev	1	98.04	1,974
Chebyshev	3	94.16	1,974
Chebyshev	5	93.30	1,974

mean 2 and 3 hidden layers respectively. It is obvious that training a network is time-consuming. When the network adds more layers, the training time rises quite quickly, but not necessarily leads to a better accuracy. Overall the network with one hidden layer is the best performing (97.12%) and most efficient setting.

Table 4: Example of hyper-parameter tuning - Multilayer Perceptron

Hidden layers setting	Number of hidden layers	Accuracy	Construction time (s)	Model size (KB)
'a'	**1**	**97.12**	**13.05**	**27**
'a, a'	2	96.67	22.01	36
'a, a, a'	3	96.53	27.44	44

6.2 Comparative analysis of learning methods

The best hyper-parameter tuned models that are both accurate and fast are listed in Table 5, which shows accuracy, precision, recall, F1-Score, MAE, model construction time (C), model execution time (E) and model size. It can be seen that methods like kNN and some variations of Naive Bayes are less suitable for real-time detection although their accuracies are high.

Table 5: Evaluation on learning methods

Learning Algorithm	Accuracy (%)	Precision	Recall	F1-Score	MAE	C time (s)	E time (s) (deviation)	Model size (KB)
Decision Tree	98.68	0.987	0.987	0.987	0.014	0.14	0.422±0.022	53
Naive Bayes	91.19	0.915	0.912	0.912	0.091	0.15	0.880±0.038	38
kNN	99.46	0.995	0.995	0.995	0.006	N/A	2.390±0.189	1,974
Multilayer Perceptron	97.12	0.971	0.971	0.971	0.034	13.05	0.839±0.013	27

6.3 Analysis of ensemble algorithms

Five ensemble algorithms are evaluated. Bagging, also called bootstrap aggregating, is a useful statistical estimation technique that can reduce variance and avoid overfitting. Random Forest involves a multitude of decision trees. AdaBoost (Adaptive Boosting) computes a weighted sum of other learning algorithms to provide a boosted classifier. Vote is a simple ensemble algorithm that

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

trains several base estimators and predicts on the basis of aggregating each of them by weighting. Stacking is similar to Vote but with a different way of final aggregation. Four classifiers with the highest accuracy from the previous section, methods in Table 5, are used as the base classifiers in Vote and Stacking. The result of the ensemble algorithms classifiers is shown in Table 6. The Random Forest achieves the highest accuracy at 99.74%, but the size of this model went up to as large as 2,486 KB.

Table 6: Evaluation on ensemble algorithms

Algorithm	Accuracy (%)	Precision	Recall	F1-Score	MAE	C time (s)	E time (s) (deviation)	Model size (KB)
Bagging	99.12	0.991	0.991	0.991	0.042	0.56	0.501±0.059	237
Random Forest	99.74	0.998	0.996	0.997	0.028	1.89	0.998±0.021	2,486
AdaBoost	83.39	0.833	0.830	0.832	0.206	0.26	0.428±0.031	6
Vote	99.40	0.994	0.994	0.994	0.036	11.70	2.261±0.130	2,080
Stacking	99.62	0.996	0.996	0.996	0.006	118.54	2.539±0.131	2,083

6.4 Evaluation on deep learning methods

Two deep networks with different numbers of hidden layers are trained for evaluation. The learning rate, batch size and max epochs are optimised in prior empirical studies. The outcomes of the ten-fold cross-validation for the optimised networks are displayed in Table 7. It shows that the neural network with one hidden layer performs better than the one with two hidden layers. That is similar to the finding in Multilayer Perceptrons. In comparison with Multilayer Perceptrons, the training cost here is much higher although the trained models are smaller and run faster.

Table 7: Evaluation on deep learning methods

Network	Accuracy (%)	Precision	Recall	F1-Score	C time (s)	E time (s) (deviation)	Model size (KB)
20-(10)-1	89.69	0.865	0.928	0.895	1241.75	0.364±0.005	3
20-(10-10)-1	87.76	0.884	0.883	0.881	1342.82	0.398±0.007	4

7 Discussions

7.1 Combined Model Comparison

Following Tables 5 to 7, we further summarise the evaluation results in Fig. 3, which shows the comparison of different learning methods against the three key metrics: accuracy, execution time, and model size respectively. In total, 11 methods are included in the comparison. They are the methods listed in Table 5, Table 6, and Table 7. They are sorted according to the accuracy obtained from the cross-validation. Each method is in a different colour. The same colour means the same method. So the three figures in Fig 3 are in the same order, sharing the same x-axis. For the middle figure, execution time, the measurement is repeated more than 10 runs. The height of each bar here shows the mean,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

while the standard deviation can also be seen on the figure as an "I" on the top of each bar.

The leftmost six methods, Random Forest, Stacking, kNN, Vote, Bagging, and Decision Tree are all with an accuracy above 98%. However, these six methods vary quite considerably in terms of execution time and model size. Three of the six, Random Forest, Bagging and Decision Tree, have execution time lower than 1 second, while the other three need more than 2.2 seconds to execute the trained model on 500 data entries on the test data set. Note, 2.2 seconds of execution time does not mean incapable of real-time performance as that is the time for processing 500 entries. When the system in operation, the number of entries coming through per second is about 30 entries, way less than 500. Nevertheless one would still prefer the models with fast execution time.

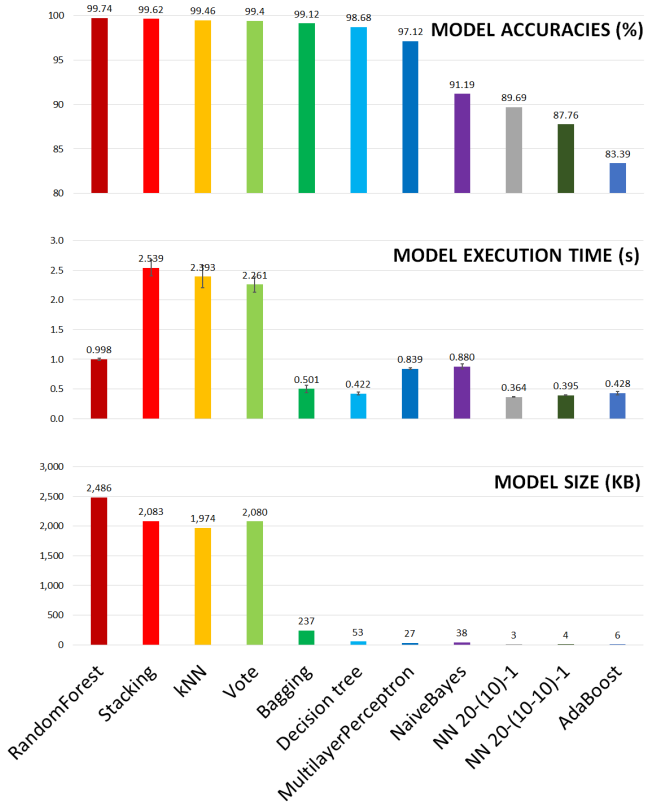


Fig. 3: Model key metrics

In terms of model size, Random Forest is rather disappointing in comparison. Its size is the largest, over 2 MB, not ideal for being stored on a small device. In contrast, the Bagging model is 237 KB and the Decision Tree is only 53 KB.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

Considering their accuracy, which is not the highest, but still amongst the best, Bagging and Decision Tree are the ideal choices. The Decision Tree is in favour, not only because of its high accuracy and low model size but also because it can be easily converted into a set of selection statements so the running time can be further reduced.

The five methods on the right side of Fig. 3, Multilayer Perceptron, Naive Bayes, two deep nets and AdaBoost, are all small in size and run faster than 1 second on the test data set. However, their accuracies are relatively low, ranging from 83.39% to 97.12%. Although fast and small are desirable characteristics for models in our obstacle detection, accuracy still takes priority. Hence in the actual operational prototype system, these models are not selected.

7.2 Real-time detection

To close the loop of this study, we established an operational prototype that can perform detection constantly on sensor input as long as power is on. In the system, a positive detection outcome is transformed into an audio signal. So when an obstacle is detected, the system will produce a slightly prolonged beep sound to warn the user. If the obstacle is still present 0.5 seconds after the beep, another beep will be triggered until there is no further positive detection. Tests on volunteers with the proposed wearable setup produced accurate and timely detection, even the person looked around while moving.

8 Conclusion

This study aims to enable a head-mount smart method to achieve efficient yet accurate obstacle avoidance to help blind and vision-impaired people (BVI) navigate around. We proposed a mechanism that integrates a set of ultrasound sensors and IMU sensors to detect obstacles in front of the person. In particular, the target is only detecting the obstacles directly on the path, excluding the objects on the side, because the head mount sensors could be seriously affected by head turns. Through the study, we show that the proposed method works well. On one hand, it can achieve high accuracy in detection, ignoring the objects directly in front of the person while he/she is looking sideways, but reporting real obstacles on his/her path even when the person is not looking straight ahead. On the other hand, the learned model is small enough to achieve real-time performance on a Raspberry Pi where the computational resource is very limited. Our study reveals that the decision tree prevails over other methods as it has the best combination of speed and accuracy, ideal for our BVI assistant task. In comparison, more sophisticated methods like ensembles and deep learning were sub-optimal due to their high computational cost.

Our future study will further improve the proposed system to extend its applicability and performance. One foreseeable improvement is to include more complex indoor settings, e.g., with furniture scattered around, with moving dynamic objects. In addition, we will investigate more types of sensors e.g., vision

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

sensors and temperature sensors to compensate for variations. We also plan to extend to outdoor environments which are far more complex and to explore some power consumption studies to optimise the system's energy efficiency.

References

1. Bourne, R., Steinmetz, J.-D., Flaxman, S., Briant, P.-S., Taylor, H., Resnikoff, S., Casson, R.-J., Abdoli, A., Abu-Gharbieh, E., Afshin, A.: Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the Global Burden of Disease Study. The Lancet global health **9**(2), e130–e143 (2021)
2. Dakopoulos, D., Bourbakis, N.-G.: Wearable obstacle avoidance electronic travel aids for blind: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **40**(1), 25–35 (2009)
3. Loomis, J.-M., Golledge, R.-G., and Klatzky, R.-L.: GPS-based navigation systems for the visually impaired. Fundamentals of wearable computers and augmented reality **429**, 46 (2001)
4. Ran, L., Helal, S., Moore, S.: Drishti: an integrated indoor/outdoor blind navigation system and service. In: Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the, pp. 23–30. IEEE, Orlando, FL, USA (2004)
5. Bousbia-Salah, M., Bettayeb, M., Larbi, A.: A navigation aid for blind people. Journal of Intelligent & Robotic Systems **64**(3), 387–400 (2011)
6. Priya, T., Sravya, K.-S., Umamaheswari, S.: Machine-learning-based device for visually impaired person. In: Proceedings of Artificial Intelligence and Evolutionary Computations in Engineering Systems, pp. 79–88. Springer, Chennai, Tamilnadu, India (2020)
7. Dos S., Aline D.-P., Suzuki, A.-H.-G., Medola, F.-O., Vaezipour, A.: A systematic review of wearable devices for orientation and mobility of adults with visual impairment and blindness. IEEE Access **9**, 162306-162324 (2021)
8. Bouteraa, Y.: Design and Development of a Wearable Assistive Device Integrating a Fuzzy Decision Support System for Blind and Visually Impaired People. Micromachines **12**(9), 1082 (2021)
9. Gao, Y., Chandrawanshi, R., Nau, A.-C., Tse, Z.-T.-H.: Wearable virtual white cane network for navigating people with visual impairment. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine **229**(9), 681–688 (2015)
10. Shoval, S., Ulrich, I., Borenstein, J.: NavBelt and the Guide-Cane [obstacle-avoidance systems for the blind and visually impaired]. IEEE robotics & automation magazine **10**(1), 9–20 (2003)
11. Lee, J.-H., Kim, D., Shin, B.-S.: A wearable guidance system with interactive user interface for persons with visual impairment. Multimedia Tools and Applications **75**(23), 15275–15296 (2016)
12. Sammouda, R., Alrjoub, A.: Mobile blind navigation system using RFID. In: 2015 Global Summit on Computer & Information Technology (GSCIT) on Proceedings, pp. 1–4. IEEE, Sousse, Tunisia (2015)
13. Joseph, S.-L., Zhang, X., Dryanovski, I., Xiao, J., Yi, C., Tian, Y.: Semantic indoor navigation with a blind-user oriented augmented reality. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics on Proceedings, pp. 3585–3591. IEEE, Manchester, United Kingdom (2013)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34

14. Yánez, D.-V., Marcillo, D., Fernandes, H., Barroso, J.: Blind Guide: anytime, anywhere. In: Proceedings of the 7th international conference on software development and technologies for enhancing accessibility and fighting info-exclusion on Proceedings, pp. 346–352. ACM, New York, New York, US (2016)
15. Li, G., Xu, J., Li, Z., Chen, C., Kan, Z.: Sensing and Navigation of Wearable Assistance Cognitive Systems for the Visually Impaired. IEEE Transactions on Cognitive and Developmental Systems (2022)
16. Lee, J.-H., Kim, D., Shin, B.-S.: A wearable guidance system with interactive user interface for persons with visual impairment. Multimedia Tools and Applications **75**(23), 15275–15296 (2016)
17. Silva, C.-S., Wimalaratne, P.: Context-aware assistive indoor navigation of visually impaired persons. Sens. Mater **32**, 1497 (2020)
18. Hicks, S.-L., Wilson, I., Muhammed, L., Worsfold, J., Downes, S.-M., Kennard, C.: A depth-based head-mounted visual display to aid navigation in partially sighted individuals. PloS one **8**(7), e67695 (2013)
19. Yang, K., Wang, K., Hu, W., Bai, J.: Expanding the detection of traversable area with RealSense for the visually impaired. Sensors **16**(11), 1954 (2016)
20. 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)
21. 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)
22. 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 on Proceedings, pp. 779–788. Las Vegas, NV, US (2016)
23. 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 on Proceedings, pp. 21–37. Amsterdam, The Netherlands (2016)
24. Suresh, A., Arora, C., Laha, D., Gaba, D., Bhambri, S.: Intelligent smart glass for visually impaired using deep learning machine vision techniques and robot operating system (ROS). In: the 5th International Conference on Robot Intelligence Technology and Applications 5 on Proceedings, pp. 99–112. Daejeon, Korea (2019)
25. Mallikarjuna, G.-CP., Hajare, R., Pavan, PSS.: Cognitive IoT System for visually impaired: Machine Learning Approach. Materials Today: Proceedings **49**, 529–535 (2022)
26. Silberschatz, A., Galvin, P.-B., Gagne, G.: Operating system concepts. 10th edn. John Wiley & Sons, New Jersey, US (2018)
27. Frank, E., Hall, M.-A., Witten I.-H.: The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". 4th edn. Morgan Kaufmann, Massachusetts, US (2016)
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Annual Conference on Neural Information Processing Systems 2019 on Proceedings, pp. 8024-8035. Curran Associates, Inc., Vancouver, BC, Canada (2019)
29. Kassim, A.-M., Yasuno, T., Suzuki, H., Aras, M.-S.M., Jaafar, H.-I., Jafar, F.A., Subramonian, S.: Conceptual design and implementation of electronic spectacle based obstacle detection for visually impaired persons. Journal of Advanced Mechanical Design, Systems, Manufacturing **10**(7), JAMDSM0094 (2016)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_34