IoT-Enabled Smart Mobility Devices for Aging and Rehabilitation

Nafisa Mostofa[†], Kelly Fullin[‡], Sharare Zehtabian[†], Safa Bacanlı[†], Ladislau Bölöni[†], Damla Turgut[†]
[†]Department of Computer Science, University of Central Florida
{nafisa.mostofa, sharare.zehtabian}@knights.ucf.edu, {sbacanli, lboloni, turgut}@cs.ucf.edu
[‡]Department of Computer Science, Ashland University
kellyfullin@gmail.com

***Abstract*— Many elderly individuals have physical restrictions that require the use of a walker to maintain stability while walking. In addition, many of these individuals also have age-related visual impairments that make it difficult to avoid obstacles in unfamiliar environments. To help such users navigate their environment faster, safer and more easily, we propose a smart walker augmented with a collection of ultrasonic sensors as well as a camera. The data collected by the sensors is processed using echo-location based obstacle detection algorithms and deep neural networks based object detection algorithms, respectively. The system alerts the user to obstacles and guides her on a safe path through audio and haptic signals.**

Index Terms **IoT, walker, obstacle detection, aging, rehabilitation**

I. Introduction

According to the World Health Organization, there are an estimated 1.3 billion people globally living with some form of visual impairment [1]. Particularly among older individuals, this results in a high chance of co-morbidity with a physical impairment requiring the use of a mobility aid, such as a walker. While these individuals might be able to use these devices to navigate around familiar environments such as their home, unfamiliar environments could be more difficult. Also, most conventional mobility aids are not suitable for visually-impaired individuals. With aids such as a walking cane, the visually-impaired individual has to make many stops while they are moving to check their surroundings, resulting in a slow and tedious process. Most conventional visually-impaired aids are also not suitable when used in conjunction with mobility devices or require a network connection that is sometimes not available.

In this paper, we propose a smart walker for visually-impaired individuals that will convey information about the obstacles in the user's path and navigate the user safely away. While designing the walker, we had in mind the following design objectives:

1) Design an approach that works in various environments, specifically outside in a sunlit environment.
2) Provide feedback to the user about what direction to move the walker in order to avoid collision with the obstacle currently in its path.
3) Provide feedback to the user if there is a massive obstacle, such as a door or stairs, so that the user can be safer.
4) Implement a more stable haptic and an audio feedback system.
5) Ensure that the system is affordable.

We implement two approaches for detecting and navigating through obstacles. The first approach uses HC-SR04 ultrasonic sensors powered by a Raspberry Pi, a low-cost, low-power computing device. Once a sensor detects an obstacle, the sensors directly to the left and right will aim to detect an obstacle. If there is no obstacle detected, then that obstacle-free path will be conveyed to the user. The second approach uses Google's TensorFlow Object Detection API to classify obstacles and depending on the placement of the obstacle on the screen, the user will be verbally warned of the obstacle in their path.

The remainder of the paper is organized as follows. In Section II, we review previous work similar to our research problem. In Section III, we describe the overall design of the walker. We provide the details of the ultrasonic sensors and TensorFlow's Object Detection approaches to detect and navigate obstacles in Section IV. The evaluations of the proposed approaches through various scenarios are given in Section V. We conclude and discuss future improvements in Section VI.

II. Related Work

There are several existing systems designed to help with navigation of the elderly through various environments as well as systems to assist with rehabilitation after major surgery.

The smart walker proposed in Zehtabian et al. [2] provides feedback to users on proper walker usage for rehabilitation and assists physicians in checking their patient's rehabilitation progress. In Khodadadeh et al. [3], the walker's data stream is processed by a deep neural network based classifier, which learns to detect unsafe usage of the walker that could hinder a patient's rehabilitation. This classifier can detect in real time if a user is operating the walker in safe or unsafe patterns.

A similar project [4] has been conducted on a rollator which uses audio feedback to communicate the presence of obstacles to the user. This device, called PAM-AID, has two modes: a) the audio feedback only and b) the feedback while also using motors to align the wheels in an obstacle-free direction. Garrote et al. [5] proposed a smart walker (ISR-AIWALKER), which was implemented using a utility decision and safety analysis procedure with user intent adjustments learned by reinforcement learning to help guide a user away from an obstacle.

An autonomous walker that can guide users in navigating in an indoor environment was explored in Kashyap et al. [6]. This smart walker uses voice commands to navigate the user through various indoor environments by using simultaneous location and mapping(SLAM) and an integrated fall detection system. The walker recognizes simple phrases spoken by the user and can guide a user to their desired location while avoiding obstacles. However, this system is not ideal in an outdoor environment.

Radar sensors are being used in many applications. Since they do not depend on environment conditions, they can be very useful for interactive applications such as human activity recognition [7]. Similarly, the use of ultrasonic sensors in aid devices for the visually impaired has been previously investigated. In Bhatlawande et al. [8], a system is designed to detect and identify the distances of obstacles in front of a user through ultrasonic sensors. These ultrasonic sensors are attached to a belt and a pair of glasses that the user would wear. Similar to our design, if there is an obstacle present, the other sensors attached to the belt will then search for an obstacle-free path and provide audio feedback to the user about what direction to take in order to avoid that obstacle. However, this system does not convey any information about the distance of the obstacle to the user. The device also cannot detect elevation changes such as stairs or a curb and cannot be used by individuals with mobility issues.

In Dey et al. [9], ultrasonic sensors are attached to a walking cane. The sensors are placed to detect obstacles in front, left, and right of the user with a haptic feedback system in place to warn the user of an upcoming object. However, this work can only detect objects within a 5-35 centimeters range and does not have a navigational guidance system in place.

Hybrid systems have also been implemented in order to create a mobility device for the visually impaired. In Sahoo et al. [10], a walking stick comprised of SRF08 ultrasonic sensors and Funduino water level sensors is implemented. While the ultrasonic sensors detects any obstacles in the user's path, the water sensor is able to detect water accumulation or puddles. There is also a Global Positioning System (GPS) module embedded in the walker to provide geolocation and time information. However, this system is quite difficult and costly to implement. In De Silva et al. [11], a personal assistive robot was created to navigate indoor environments. This device relies on the IoT devices already placed in an indoor environment and uses the received data from those devices to react to various obstacles around it. A scalable multi-layered context mapping framework was developed to use these IoT sensors and process the data received from them. While this will be able to indicate if an area is void of obstacles or not, it only conveys this information visually. The technology is also not able to provide a path to navigate the user around obstacle-filled areas.

Youm et al. [12] used the TensorFlow framework to develop tools for helping individuals in emergency situations. The tools used pose recognition to determine if the user was hurt, and used this information to sound an alarm.

The use of TensorFlow framework in wearable aids for visually impaired has been briefly investigated. In Mulfari's work [13], a pair of glasses with a camera attached is controlled by a single board computer running TensorFlow. The camera captures the user's surroundings in real time and processed by TensorFlow's image classification. Once the objects in the user's surroundings are classified, an audio feedback is provided to the user describing those objects through an earphone. This system does not inform the user about the distance of these objects or how to avoid them. It also does not detect any obstacles close to the ground.

In Nishajith et al. [14], a hat is embedded with a NoIR camera that use TensorFlow to detect objects. Once the objects are detected, an audio feedback to the user is be conveyed. However, this system does not inform the user on how to avoid these obstacles.

The proposed work builds upon our previous work [15] where we designed a walker for visually impaired individuals that was modified with an XBOX 360 Kinect and a haptic feedback system to detect obstacles. It uses the camera to capture depth images and averages the depth values across the rows then finds the slope down the averaged column. Another approach uses the depth image from the camera to generate a point cloud which is then analyzed for the largest plane parallel to the z-axis to detect obstacles. However, the use of a depth-imaging camera was not useful in detecting obstacles in sunlight.

III. Design

For both of our approaches, the walker is a standard four-wheeled rollator. In the first approach, there are seven HC-SR04 ultrasonic sonar distance sensors attached to the lower half of the walker and angled towards the floor. A Raspberry Pi operates these sensors and processes the data collected through the sensors. Two vibration motors are attached to the walker, one on each handlebar, providing navigational feedback to the user. In the second approach, obstacle detection and navigational feedback are performed by TensorFlow on a laptop computer with a Logitech C270 HD Webcam attached to the lower half of the walker. The navigational system in this approach is conveyed through an audio feedback by the laptop.

With the ultrasonic sensors and Logitech Webcam, the IoT-enabled walker is able to detect and navigate through obstacles in various environments, specifically outside in a sunlit environment. Whereas other sensors, such as the Microsoft Kinect uses an infrared sensor to process the distance; resulting in the infrared rays of the sun to hinder these results. The use of the HC-SR04 sensors and TensorFlow framework contribute towards an affordable design that costs on the order of tens of dollars.

After an obstacle is detected, the navigational guidance system determines which direction to move the walker in order to avoid collision with the obstacle currently in its path. In order to determine the most efficient navigational guidance system, haptic and audio feedback systems are designed and tested. The haptic feedback system conveys information about obstacles in three different categories: close, mid-range, and far. For each category there is a specific vibration intensity assigned to it: high vibration for close range to low vibration for far range. When an obstacle is detected, the direction of the obstacle-free path is determined and the corresponding vibration motor vibrates. For example, if the path to the right of the walker is more obstacle-free, the vibration motor attached to the right handlebar will start vibrating indicating for the user to move in that direction.

The audio feedback system conveys information about obstacles by warning the user that an obstacle is approaching and suggesting the user the direction in order to avoid it. For massive obstacles such as doors, stairs, cars, and trucks, the verbal warning also specifies what obstacle is approaching and how to avoid it. For example, if a door is in the user's path and there is an obstacle-free path to the left, the audio feedback system will verbally convey "Door approaching, Move Left." This provides an extra layer of caution for the users and allows them to safely move through these environments. Currently, the audio feedback is conveyed through the laptop computer's speakers.

IV. Proposed Approaches

A. Background

The HC-SR04 [16] sensor is an ultrasonic sensor that generates a sound wave at a frequency above the range of human hearing. The sensor transmits this wave forward as a trigger. If there is an object in the sensors path, the sound wave bounces back to the sensor as an echo.

HC-SR04 has four pins: VCC, Trig, Echo, and Gnd. The VCC pin powers the sensor with +5V. The Trig pin is the input pin, which needs to be set to a high state for 10 microseconds to generate the sound wave. The Echo pin is the output pin, which is set to a high state equal to the amount of time it takes to receive the returned sound wave. The ground pin is connected to the ground of the system.

A Raspberry Pi is a highly efficient, low-cost, small computing device that consists of General Purpose Input/Output (GPIO) pins. This allows the Raspberry Pi to interface with the ultrasonic sensors.

TensorFlow is an open-source platform for machine learning models, developed by Google. TensorFlow Object Detection API is a framework and uses deep neural networks for object classification [17]. TensorFlow can also be used to train object detection models that can identify what set of objects may be present in a video stream and provide the locations of those objects.

TensorFlow represents these models using dataflow graphs. A graph consists of nodes connected to each other as inputs and outputs. Each node represents a specific operation. The weights of these nodes are also necessary for the graphs; however, they are not stored inside the same file. Instead, they are in separate checkpoint files, where the Variable ops in the graph will load them after being initialized [18]. In order to be more efficient, the graph definition and checkpoint files are frozen together in a single file, which is used for object detection.

B. Approach 1: HC-SR04 Ultrasonic Sensors

The use of HC-SR04 ultrasonic sensors to detect obstacles was similar to Dey et al. [9]. According to the HC-SR04 data sheet, these sensors can provide 2-400 centimeters non-contact measurement function up to a 3 millimeter accuracy with a measuring angle of 15 degrees. Since the measuring angle is small, we attach seven sensors across the width of the walker and angle them slightly to the ground to detect obstacles close to the ground, so every obstacle can be detected in the walkers path. In order to detect obstacles that are parallel to the walker, we attached one sensor each to the left and right side of the walker and angled them outwards (see Figure 1). To eliminate possible interference issues, each sensor transmits a sound wave and calculates the distance in sequential order. We store these distance measurements in an array for further use.

Fig. 1. The walker configured with ultrasonic sensors

To calculate the distance of the obstacle, we use the formula:

$$\text{Distance} = \text{Speed} \times \text{Time} \quad (1)$$

Speed is the speed of sound in centimeters (34300 cm/sec) and time is the number of milliseconds it takes for the sound wave to reach that obstacle. Using the sensors, we record the time it takes for the Trig pin to send out the ultrasonic wave and for the Echo pin to receive the reflected wave. However, the time recorded is double of the actual time it takes for the sound wave to reach the object. So, this measurement needs to be divided by 2, which results in the correct distance of the object from the walker, as shown in Algorithm 1.

After each sensor calculates the distance that the ultrasonic wave traveled, the sensor that measured the smallest distance will indicate that the closest obstacle is directly in front of it. Using the previously stated array, we check the other sensors distance measurements to the left and right of that sensor that returned the largest distance. If the sensors to the left calculated a larger distance than the sensors to right, then the path to the left is more obstacle free, as shown in Algorithm 2. In order to convey this information to the user, the left handlebar vibrates.

Algorithm 1 Approach 1: Detecting Obstacle's Distance

```
setTrig(High)
LeaveTrigStatefor10ms
setTrig(Low)
while inputEcho(Low) do
    start = startTimer()
end while
while inputEcho(High) do
    end = endTimer()
end while
TimeElapsed = start - end
Distance = (TimeElapsed x 34300) / 2
```

If the sensors to the right calculated a larger distance, then the right handlebar would vibrate.

If all the sensors calculate an obstacles distance within a 5 cm range, then a large obstacle, such as a wall or stairs, is in the walkers pathway. In that situation, the distances of the sensors that are facing parallel to the walker is utilized. Using that distance array, the program compares which sensor recorded the larger distance. If the left parallel sensor recorded a larger distance, that means there is an open path to the left and the left handlebar vibrates. If the right parallel sensor recorded a larger distance, the right handlebar would vibrate, indicating that the path is open to the right.

Algorithm 2 Approach 1: Finding the Safest Path away from the Obstacle

```
sensorDistance = getsDistance();
if sensorDistance < 200 cm then
    AlertUserObstacleDistance()
    if sensorDistanceLeft > 200 cm then
        moveWalkerLeft
    else if sensorDistanceRight > 200 cm then
        moveWalkerRight
    else
        checkParallelSensors()
    end if
else
    moveForward
end if
```

C. Approach 2: Object Detection with TensorFlow

In order to detect obstacles in real time, we implemented TensorFlows Object Detection API with OpenCV. In the TensorFlow environment running on our laptop, we use the model that is already trained on the Open Images data set using TensorFlow. We extract the frozen inference graph of that model [1]. We then create a label map that loads all the labels and maps the indices to the various category names such as door, stairs, person. All of the detection boxes, classes, accuracy scores, and number of detections are defined through tensors, which are generalization of the vectors. The external webcam is then turned on and each frame of the live stream is captured and processed. We run the actual detection through each frame and draw a box around every classified object with a confidence score greater than 70% with the name of the object written on the box as shown in Figure 2.

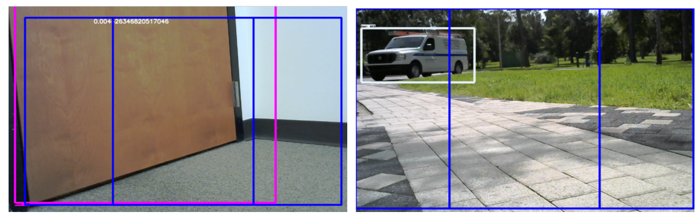

Fig. 2. Object Detection of door and car with TensorFlow.

To get the distance of these classified objects from the walker, we use the formula [13]:

$$\text{Distance} = \frac{\text{Width of Object} \times \text{ Focal Length}}{\text{Pixel Width}} \tag{2}$$

We subtract the x-coordinates of the bounded box to calculate the pixel width of the object. To determine the real-world width of the object, we divide the pixel width by 75, which is the Dots Per Inch (DPI) resolution of the camera. This results in the width of the object in inches which is converted to centimeters. We multiply this measurement by 3 in order to have the calculated width within a 10% error margin of the actual width.

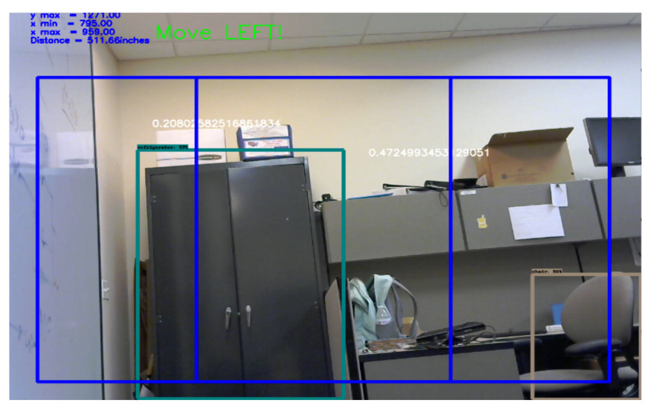

Fig. 3. Example of Region of Interest coordinate system implementation.

We implemented a coordinate Region of Interest (ROI) to create a navigational guidance system for the user. An ROI is a portion of an image that is filtered or an area that another operation is performed upon. We defined the boundaries of the ROI as a three box grid system that stretched horizontally across the width of the screen (See Figure 3). Anything outside of this region is essentially not processed. If an obstacle appears in the ROI, then its distance is calculated and the placement of

[1]Installation instruction is available here: github.com/nafisam/vision_walker-2019/tree/master/TensorFlow

it is processed. If the obstacles bounded box left x-coordinate is greater than the ROI left boundarys x-coordinate, then the obstacle is on the right side of the walker and a verbal warning for the user to move left should be heard. If the obstacles bounded box right x-coordinate is less than the ROI right boundarys x-coordinate, then the obstacle is on the left side of the walker and a verbal warning to go right is given (See Algorithm 3). When the obstacles coordinates are within the ROI, then a verbal warning to stop should be heard, indicating there is a large, unavoidable obstacle in the users path. As an extra layer of precaution, when massive obstacles such as stairs, doors, cars, or trucks are detected within the ROI, a verbal warning with that obstacle name is provided. For example, if a door is detected within the users path, the verbal warning "Door approaching" would be given to the user.

Algorithm 3 Approach 2: Finding the Safest Path away from the Obstacle

```
if DetectedObstacleinROI then
    if ObstacleLeftXCoord > ROILeftBoundXCoord then
        WarnObstacleRight
        moveWalkerLeft
    else if ObstacleRightXCoord < ROIRightBoundXCoord then
        WarnObstacleLeft
        moveWalkerRight
    else if ObstacleWithinCenter then
        stopWalker
    end if
end if
```

V. RESULTS

The different approaches were evaluated on their abilities to detect various types of obstacles in an outdoor and inside environment and determine the distances of these obstacles from the walker (See Table I). These obstacles' actual distance from the walker was measured with a tape measure and compared to the distances measured by the different approaches. The ability to detect an obstacle-free environment was also evaluated.

TABLE I
DISTANCE MEASUREMENT

Obstacle	HC-SR04 Sensors	TensorFlow	Actual Distance
Wall	92.3 cm	82.9 cm	91.9 cm
Door	54.6 cm	40.3 cm	53.7 cm
Person	110.2 cm	120.9 cm	110.3 cm
Stairs	52.3 cm	54.9 cm	51.8 cm
Curb	86.9 cm	100.3 cm	87.3 cm
Backpack	42.6 cm	39.5 cm	42.5 cm
Empty Hallway	No Obstacle	No Obstacle	No Obstacle

An evaluation of each approaches' navigational guidance system was also performed. The test compared both approaches' performance along with a conventional walker/cane in eight different scenarios with an audio and haptic feedback. The first and second scenarios were obstacle-free indoors and outdoor environments, respectively. The user had to avoid collision with walls for the indoor environment and curbs and cars for the outdoor environment. For the remaining scenarios, there were ten obstacles that the user had to avoid. These obstacles were dispersed at three various levels: high, medium, and low. At high levels of dispersion, the obstacles were far away from each other. At medium levels of dispersion, the obstacles were closer to each other and at low levels of dispersion, the obstacles are densely clusters around each other. Each level of obstacle dispersion was evaluated indoors and outdoors. The test subjects were individuals without any visual impairment or mobility issues. Their eye sight was covered with a blindfold during the tests and the average number of objects they collided with was recorded (See Table 2). The time it took to maneuver through these scenarios was also measured (See Table 3).

TABLE II
NAVIGATIONAL GUIDANCE SYSTEM (OBSTACLES HIT)

Environment	Dispersion of Obstacles	Walker/Cane	HC-SR04 with Haptic	TensorFlow with Audio
Hallway	Empty	0	0	0
Outdoors	Empty	1	1	0
Hallway	High	2	1	1
Outdoors	High	4	3	0
Hallway	Medium	3	2	1
Outdoors	Medium	2	4	1
Hallway	Low	3	3	2
Outdoors	Low	3	3	2

TABLE III
NAVIGATIONAL TIME

Environment	Dispersion of Obstacles	Walker/Cane	HC-SR04 with Haptic	TensorFlow with Audio
Hallway	Empty	2:31	2:14	2:15
Outdoors	Empty	3:34	3:40	3:30
Hallway	High	2:48	2:33	2:32
Outdoors	High	3:48	4:02	3:50
Hallway	Medium	3:15	3:09	2:57
Outdoors	Medium	3:54	3:59	3:42
Hallway	Low	3:45	4:21	4:19
Outdoors	Low	3:51	4:57	4:32

Based on the results of the various trials, the ultrasonic sensor approach was more successful in identifying the distances of obstacles from the walker. The sensors were able to calculate the distance within 2 centimeters of the actual distance, while the TensorFlow approach calculated a measurement around 20 centimeters of the actual distance. This discrepancy is due to the measurement error of the width of the objects. Both approaches were able to identify an obstacle-free hallway.

The ultrasonic sensor and TensorFlow approaches were more effective than a conventional walker when navigating through clear or spaces with a high dispersion of obstacles. With spaces with a low dispersion of obstacles, the TensorFlow approach with audio feedback was the most successful in its navigational guidance system. This approach was able to guide the user quickly across various environments with different dispersion of obstacles and help them avoid colliding with these obstacles. Compared to the ultrasonic sensors with haptic feedback, the TensorFlow-implemented walker was able to navigate through the various environments faster and more accurately. Due to the guidance system that informed the user which direction was more obstacle-free, there were less obstacle collisions. Overall, the TensorFlow-implemented walker collided with less obstacles than the ultrasonic sensors. The ultrasonic sensors angle to the ground was continuously shifting due to the unstable wooden frame, so many false obstacles were processed such as uneven pavement.

It was observed that the audio feedback was able to provide an extra layer of safety for the user, as it warned them of stairs, doors, cars, and trucks. The user was able to use that information and had more success with avoiding those obstacles as opposed to the haptic feedback where none of the obstacles were classified.

Other studies on cognitive load of audio and haptic feedback in assistive systems such as [19], suggest that blind people prefer haptic feedback over audio feedback for short range navigation tasks, however, they prefer audio feedback for other tasks such as orientation, communication and alerts. Further testing of these feedback systems with different approaches would provide more information about the efficacy of each system.

VI. Conclusion

In this paper, we designed and implemented an IoT-enabled smart walker that was able to detect obstacles in a user's path and provide feedback to the user about which direction to move in order to avoid those obstacles. The walker was capable of calculating the distances of those obstacles and conveying it to the user, along with classifying certain dangerous obstacles for an extra safety precaution. The first approach used HC-SR04 ultrasonic sensors to detect obstacles while the second approach implemented TensorFlow's Obstacle Detection API. Both approaches were fully functional in various environments, including outside in a sunlit environment.

Further improvements to this walker include improving the distance accuracy of the TensorFlow approach, retraining the TensorFlow model to classify more obstacles, specifically different types of walls, improving the sensors' navigational system's accuracy in maneuvering walls, create a more stable holding frame for the sensors, and eliminate the need of a laptop computer to run the TensorFlow approach. Further testing of the different feedback systems would also be beneficial in evaluating the efficacy of both approaches.

Acknowledgement

The support for this work was provided by the National Science Foundation REU program under Award No. 1852002. Any opinions, findings, and conclusions and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] (2018, October) Blindness and vision impairment. World Health Organization. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] S. Zehtabian, S. Khodadadeh, R. Pearlman, B. Willenberg, B. Kim, D. Turgut, L. Bölöni, and E. A. Ross, "Supporting rehabilitation prescription compliance with an iot-augmented four-legged walker," in *2nd Workshop on AI for Aging, Rehabilitation and Independent Assisted Living (ARIAL'18)*, July 2018.

[3] S. Khodadadeh, S. Zehtabian, J. Guilbe, R. Pearlman, B. Willenberg, B. Kim, E. A. Ross, L. Bölöni, and D. Turgut, "Detecting unsafe use of a four-legged walker using iot and deep learning," in *Proc. of IEEE ICC 2019*, May 2019.

[4] S. MacNamara and G. Lacey, "A smart walker for the frail visually impaired," in *Proceedings of IEEE International Conference on Robotics and Automaton*, April 2000.

[5] L. Garrote, J. Paulo, and U. J. Nunes, "Reinforcement learning aided robot-assisted navigation: A utility and rrt two-stage approach," *International Journal of Social Robotics*, Sep 2019.

[6] P. Kashyap, M. Saleh, D. Shakhbulatov, and Z. Dong, "An autonomous simultaneous localization and mapping walker for indoor navigation," in *IEEE Sarnoff Symposium*, September 2018, pp. 1–6.

[7] S. Zhu, J. Xu, H. Guo, Q. Liu, S. Wu, and H. Wang, "Indoor human activity recognition based on ambient radar with signal processing and machine learning," in *IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[8] S. S. Bhatlawande, J. Mukhopadhyay, and M. Mahadevappa, "Ultrasonic spectacles and waist-belt for visually impaired and blind person," in *2012 National Conference on Communications (NCC)*, Feb 2012, pp. 1–4.

[9] N. Dey, A. Paul, P. Ghosh, C. Mukherjee, R. De, and S. dey, "Ultrasonic sensor based smart blind stick," 03 2018, pp. 1–4.

[10] N. Sahoo, H.-W. Lin, and Y.-H. Chang, "Design and implementation of a walking stick aid for visually challenged people," *Sensors*, vol. 19, p. 130, 01 2019.

[11] V. De Silva, J. Roche, X. Shi, and A. Kondoz, "Iot driven ambient intelligence architecture for indoor intelligent mobility," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, Aug 2018, pp. 451–456.

[12] S. Youm, C. Kim, S. Choi, and Y.-S. Kang, "Development of a methodology to predict and monitor emergency situations of the elderly based on object detection," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5427–5444, Mar 2019.

[13] D. Mulfari, "A TensorFlow-based assistive technology system for users with visual impairments," in *Proceedings of the Internet of Accessible Things*, ser. W4A '18, 2018, pp. 11:1–11:2.

[14] A. Nishajith, J. Nivedha, S. S. Nair, and J. Mohammed Shaffi, "Smart cap - wearable visual guidance system for blind," 7 2018, pp. 275–278.

[15] C. Feltner, J. Guilbe, S. Zehtabian, S. Khodadadeh, L. Bölöni, and D. Turgut, "Smart walker for the visually impaired," in *IEEE ICC 2019*, May 2019.

[16] "Ultrasonic Ranging Module HC - SR04," https://www.mouser.com/datasheet/2/813/HCSR04-1022824.pdf, accessed: 2019-10-23.

[17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[18] M. Abadi, M. Isard, and D. G. Murray, "A computational model for TensorFlow (an introduction)," 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3088525.3088527

[19] M. Martinez, A. Constantinescu, B. Schauerte, D. Koester, and R. Stiefelhagen, "Cognitive evaluation of haptic and audio feedback in short range navigation tasks," in *International Conference on Computers for Handicapped Persons*. Springer, 2014, pp. 128–135.