Designing an Obstacle Detection and Alerting System for Visually Impaired People on Sidewalks

Sude Pehlivan and Mazlum Unay
Department of Biomedical Engineering
Izmir Kâtip Celebi University
Izmir, Turkey
sudepehlivan35@gmail.com, unaymazlum@gmail.com

Aydin Akan
Department of Biomedical Engineering
Izmir Kâtip Celebi University
Izmir, Turkey
aydin.akan@ikc.edu.tr

***Abstract*— It is known that being visually impaired is one of the most challenging experiences in life and several people have been facing this situation. Since computer vision and machine learning have started to be growing fields, designing an assisting system have become simpler for both researchers and engineers. There are two fundamental elements in obstacle detection systems that are software part and hardware part. In recent years, operating systems have been reshaped around deep learning algorithms, open source libraries and programming platforms. There are several deep learning algorithms and open source libraries and in this project, we focused on programming the system with Python and using Tensorflow models to create a classifier on a tiny computer called Raspberry Pi. We used a model called ssdlite_mobilnet_v2_coco to detect 9 different objects which could present on sidewalks. Since the auditory sense is the most important sense for visually impaired people and must not be blocked, we proposed a combination of auditory and tactile senses to alert the user. A system that is called eSpeak was used to alert the user via headphones by reading the name of the detected object. At the same time, 3 different vibration sensors were located at 3 different positions as of right, middle and left. When an obstacle was detected inside one of the pre-defined bounding boxes, the related vibration sensor was activated with the name of the object.**

Keywords — Tensorflow; Raspberry Pi; obstacle detection

I. INTRODUCTION

According to the tenth revision of World Health Organization (WHO), when visual acuity is between 6/18 and 3/60 or angle of vision is less than 20 degrees, this situation is defined as low vision. On the other hand, blindness is the situation of having a visual acuity less than 3/60 or having an angle of vision less than 10 degrees. It is important to understand that visual impairment encompasses both blindness and low vision [1]. Recently, since computer vision and bioengineering have become more efficient, researchers from both fields have become closer to find a solution for visual impairment. There are a considerable amount of studies both cell-based and electronic-based to encourage visually impaired people to maintain daily activities without being in need as individuals. Among electronic-based approaches, obstacle avoidance systems are one of the most studied areas which are based on the idea of recognition of an obstacle and alerting the user [2].

The basic definition of the image is that it is a two-dimensional matrix that can be shown as coordinates (r, c) with a piece of specific information called intensity [3]. When this physical context wanted to be represented in a computer screen, it has to be converted to digital data. This digital data can be used to extract features of specific objects in order to detect and classify them by using feature descriptors such as Histogram of Gradients (HOG), Speeded Up Robust Features (SURF), and Scale Invariant Feature Transformation (SIFT) and there are 20 obstacle detection aids that are based on computer vision according to a paper published in 2016 and most of them were using these basic feature extractors [4].

On the other hand, there is a more efficient way of classifying the objects which is called Deep Learning (DL). It is the subset of Machine Learning (ML) and the most common DL algorithm is the Convolutional Neural Networks (CNN) and it uses multiple layers and neural networks. There are three layers as convolutional layers to create a feature map, pooling layer to decrease the dimensions of the key points and fully connected layers to modify feature maps to the one-dimensional vector. To find the output of a layer, the output from the previous layer is used as input of the current layer and it is subjected to an activation function [5].

In most common DL algorithms, supervised learning is used where the desired output must be obtained for a given input by adjusting the weight factors of the system by training and labeling. Adjustments of the weights are implemented by calculating error and error is minimized by moving in the opposite direction of the gradient of the cost function [5].

In order to make tasks easier, there are DL libraries that consist of several special functions such as Torch, Theano and Tensorflow [6].

II. MATERIALS

A. Raspberry Pi

Raspberry Pi is an economical and tiny computer that can perform from a simple task like blinking a led to a complicated one such as image processing. In this project, Raspberry Pi 3B+ is used. It has a 32 bit Advanced RISC Machines (ARM) chip processor, a 3.5 mm jack analog audio output, a 5 V micro Universal Serial Bus (USB) to connect the card to power, a

978-1-7281-2420-9/19/$31.00 ©2019 IEEE

Camera Serial Interface (CSI) connector to connect camera module, General Purpose Input Output (GPIO) pins, and SD card channel to reserve data inside it since Raspberry Pi does not store any data [8].

B. Pi Camera

A USB camera or a pi camera can be used to take images or record a video. In this project, a standard 5 Megapixel (Mpix) camera module was used which had the capacity of 1080p at 30 Frame per Second (FPS) and 720p at 60 FPS [9].

C. Tactile Output and Audio Output

In this project, 3 standard vibrators were used to alert the user via tactile sense. Each vibrator was assigned to a certain range of locations as left, right and middle.

In order to get the name of the class that had been detected, a speech synthesizer called eSpeak was used. It performs a procedure called formant synthesis and uses these formants to create a speaking voice which is unfortunately not natural but robotic [10].

D. Python and OpenCV

Python is a programming language which has several open-source libraries and it is supported by Raspberry Pi.

To be able to perform image and video processing, Python needs OpenCV. It is an open source library that allows one to implement different tasks such as face detection and it has some dependencies and most important one of them is NumPy. It is needed to perform numeric tasks and array-related applications in scientific operations. Although they are not dependencies, SciPy and Matplotlib packages must be installed to be able to carry out some specific tasks such as plotting and signal processing [11].

E. Tensorflow

Google has been developed Tensorflow as a library for machine learning and deep learning applications. The system uses tensors as basic elements which are multidimensional arrays and performs several tasks using them [12].

Tensorflow uses the dataflow graphs to define ML algorithms. In a Tensorflow model, there are three basic elements [6] [13].

- *Operations* are represented as nodes in a dataflow graph that perform several tasks on tensors.
- *Tensors* are multidimensional arrays and they are represented as edges in a dataflow graph. Data from one node to another is transferred through the edges. Edges can be a normal edge to carry tensors from one node to another or they can be special edges to control the relationship of two nodes.
- *Sessions* are environments to execute dataflow graphs. To be able to run the code, a session must be created with *tf.session* and then executed with *sess.run* command.

F. A Tensorflow Model: ssdlite_mobilenet_v2_coco

In this project, ssdlite_mobilenet_v2_coco_2018_05_09 model was used. This model is a pre-trained Tensorflow model which is the combination of the Single Shot Multi-box Detector (SSD) model and MobileNet Deep Neural Network (DNN) as feature extractor including Common Objects in Context (COCO) which is a dataset [14].

SSD is based on creating bounding boxes and scores for a detected object using convolutional networks. At the end of every basic network, there are multi-scale convolutional feature layers. Training of the system is performed with ground truth boxes [15].

Depthwise separable convolution is the basic idea of MobileNet. A basic convolution operation is the sum of the products of two matrices by sliding one of them on the other. However, in the case of depthwise convolution, one kernel is convolved with only one channel of N channeled input. Afterward, pointwise convolution is performed where the kernel has a size of 1x1xN [16].

In this model, there are 80 classes and in this project 9 of them were used to perform object detection on a sidewalk. See Table 1.

Table 1 Names and ID Numbers of the Pre-defined Objects

Name of the Pre-defined Objects	ID Number of the Pre-defined Objects
Person	1
Bicycle	2
Car	3
Traffic Light	10
Fire Hydrant	11
Stop Sign	13
Bench	15
Cat	17
Dog	18

III. METHOD

At first, RASBIAN operating system, Tensorflow and was installed along with several packages, tools and *ssdlite_mobilenet* model was downloaded.

In the hardware setup, a keyboard and a mouse were connected to the USB port of the Raspberry Pi and the Pi camera was connected to CSI port. Three vibration sensors were connected to GPIO 4, 16 and 18 along with speakers which were connected to the analog audio output and also to the USB port to provide power. Finally, Raspberry Pi was connected to power via micro USB. See Figure 1.

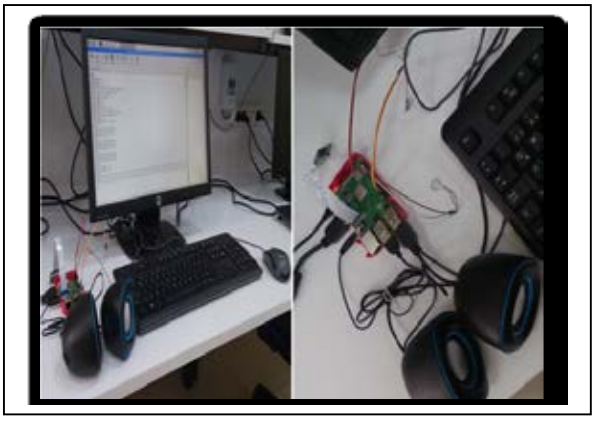

Figure. 1. Hardware Setup

In the software part, necessary packages were imported and appended inside the working directory for python. After the GPIO settings and camera arrangements, label map was loaded to get categories and indexes. In order to run the computational graph, a session was created where images were defined as input tensors and boxes, scores, number of detected objects and classes were defined as output tensors. Three boxes were drawn as left, right and middle and the real detection was performed by running the session. Boxes, scores and detected classes were displayed on the screen if the scores of the detected objects were above the 0.6 threshold. When the detected object had a pre-defined index, the center of the object was calculated and if the center was inside one of the pre-defined boxes, the related GPIO pin was activated to give tactile output. Detected objects were printed on the screen and the names of the objects that were read by eSpeak engine were given as auditory output from speakers.

IV. SUMMARY AND CONCLUSIONS

A. Summary

In this project, it was aimed to design obstacle detection and alerting system on sidewalks for visually impaired people in order to provide them guidance. In the hardware setup, a Raspberry Pi was used along with three vibration sensors to alert the user about the direction and a speaker to alert the user about the name of the object. As programming language Python was chosen because of its speed, compatibility with Raspberry Pi and package availability. A Python package called Tensorflow was used to perform object detection and classification with a pre-trained model

B. Results

In this study, pre-defined objects were detected and results were displayed as both visual and auditory to compare the correctness of auditory findings along with tactile output. It was concluded that auditory results were mostly matched with visual results with a delay on the screen due to the FPS limitations of the camera. The reason behind using both auditory and tactile outputs is that the response of a visually impaired person to the question of whether he wanted to use an auditory or tactile output was the usage of the combination of these sensory outputs.

When an object was recognized, the center of the object was calculated and represented as a pink circle. There were pre-drawn boxes on the screen as red, yellow and blue to represent the direction of the object as left, middle and right. When the center of the object was inside one of the boxes, related vibration sensor was activated. Each class had a different bounding box color to visualize the results and on top of the bounding boxes, names of the objects and scores of the detection were written and since this project was designed in laboratory, to be able to represent the results the class "chairs" was used. See Figure 2.

Figure. 2. Screenshot of the Results

Since a pre-trained model was used, to visualize the accuracy of the model Tensorboard was not available. Instead, the general performance of the models was given in Figure 3. It can be seen that SSD w/MobileNet has the lowest mAP value; however, this model has lower processing time than others. The reason for the usage of SSD lite w/MobileNet in this project was that it had the same amount of mAP as SSD w/MobileNet with lower processing time [17] [18].

C. Limitations and Future Work

In the future, the system can be improved by considering some current limitations of hardware setup such as camera constraints, Graphics Processing Unit (GPU) speed, and the choice of the board. In this project, Pi camera was chosen as a 5 Mpix standard module which can be replaced with an 8 Mpix new model to increase resolution which can affect the precision of the detection. Although FPS of the camera was set as 1, from the written FPS information on the screen, it was seen that it was fluctuating between 0.2 and 2 which affected the detection precision and the response time of the system to the movement.

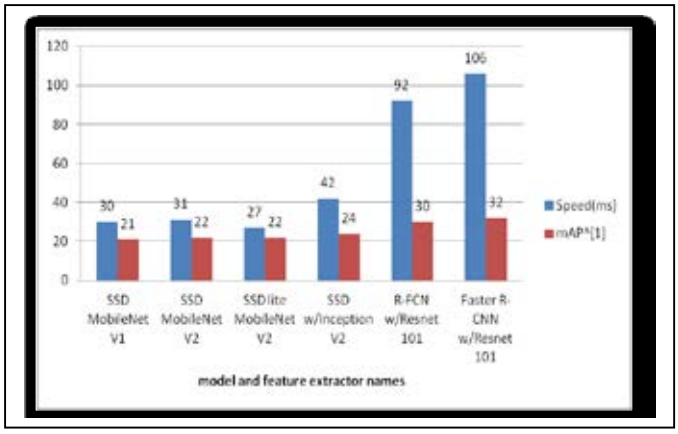

Figure. 3. Mean average precisions (mAPs) and speeds of different model and feature extractor combinations [18]

In this project, the model that was used was pre-trained with the COCO database and it was the combination of two Tensorflow models which are SSD and MobileNet. Due to the GPU limitations of the Raspberry Pi, used model was chosen as *lite* which is compatible with Central Processing Unit (CPU).

In the future works, if this model is trained with different and larger scaled databases, there may be a higher chance of increment in precision and accuracy. Besides, rather than using the same model, a different Tensorflow model can be chosen to perform object detection. Furthermore, if a better card with a higher GPU speed is chosen, it will be possible to train the model and in this way, the need for a pre-trained model will be eliminated. If GPU limitation is eliminated it won't be necessary to use only Tensorflow models, it is also possible to use different ones such as You Only Look Once (YOLO), Tesla or RetinaNet to improve detection rate.

This study can be used to create a prototype with a suitable design to provide comfort and improved quality of life for visually impaired people all around the world. Design can be placed on the belt with three vibration sensors located at the right, left and middle. Instead of speakers, headphones can be used to not only provide a better understanding of the words for users but also to eliminate noise pollution for the surrounding environment. Camera model can be placed and stabilized on the chest along with the Raspberry Pi on the belt. Further, this design can be improved as locating the camera module into the glasses and vibration sensors onto the gloves. It would be better to insert a Global System for Mobile Communications (GSM) to inform a member of the family in case of an accident and a Global Positioning System (GPS) to navigate the user.

In conclusion, this project was performed to make visually impaired people feel more self- confident with living their lives fully without any concerns. Furthermore, it can be a precursor for other projects that will be performed in the future with boards that have higher GPU speed, cameras with higher resolution and higher frame rate, larger databases and more comfortable and improved designs.

KAYNAKLAR

[1] World Health Organization. (2007). Global Initiative for the Elimination of Avoidable Blindness: action plan 2006-2011.

[2] Dakopoulos, D., & Bourbakis, N. G. (2010). 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.

[3] Gonzalez, R. C., & Wintz, P. (1977). Digital image processing(Book). Reading, Mass., Addison-Wesley Publishing Co., Inc.(Applied Mathematics and Computation, (13), 451.

[4] Kaur, P., & Kaur, S. Aids for Visually Impaired Persons for Obstacle Acknowledgment: A Study and Proposed New Framework. International Journal of Computer Applications, 975, 8887.

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

[6] Goldsborough, P. (2016). A tour of tensorflow. arXiv preprint arXiv:1610.01178.

[7] Bahrampour, S., Ramakrishnan, N., Schott, L., & Shah, M. (2016). Comparative study of caffe, neon, theano, and torch for deep learning.

[8] Richardson, M., & Wallace, S. (2012). Getting started with raspberry PI. " O'Reilly Media, Inc.".

[9] Nguyen, H. Q., Loan, T. T. K., Mao, B. D., & Huh, E. N. (2015, July). Low cost real-time system monitoring using Raspberry Pi. In 2015 Seventh International Conference on Ubiquitous and Future Networks (pp. 857-859). IEEE.

[10] Kaur, R., & Sharma, D. (2016). An Improved System for Converting Text into Speech for Punjabi Language using eSpeak. International Research Journal of Engineering and Technology (IRJET), 3(04), 500-504.

[11] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... & Yu, T. (2014). scikit-image: image processing in Python. PeerJ, 2, e453.

[12] Zaccone, G. (2016). Getting Started with TensorFlow. Packt Publishing Ltd.

[13] Girija, S. S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Software available from tensorflow. org.

[14] Jokela, J. (2018). Person counter using real-time object detection and a small neural network.

[15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

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

[17] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7310-7311).

[18] Fleury, D., & Fleury, A. (2018). Implementation of Regional-CNN and SSD machine learning object detection architectures for the real time analysis of blood borne pathogens in dark field microscopy.