Smart Hat for the blind with Real-Time Object Detection using Raspberry Pi and TensorFlow Lite

Matshehla. Konaite
Department of Computer Systems
Engineering
Tshwane University of Technology
Pretoria, South Africa
matsobanekgomotso@gmail.com

Pius A Owolawi
Department of Computer Systems
Engineering
Tshwane University of Technology
Pretoria, South Africa
owolawipa@tut.ac.za

Temitope Mapayi
Department of Computer Systems
Engineering
Tshwane University of Technology
Pretoria, South Africa
tmapayi@yahoo.com

Vusi Malele
Department of Computer Systems
Engineering
Tshwane University of Technology
Pretoria, South Africa
vusimalele@gmail.com

Kehinde Odeyemi
Department of Computer Systems
Engineering
Tshwane University of Technology
Pretoria, South Africa
kensonics@yahoo.com

Gbolahan Aiyetoro
Department of Computer Systems
Engineering
Tshwane University of Technology
Pretoria, South Africa
g.aiyetoro@ieee.org

Joseph S. Ojo
Department of Physics
Federal University of Technology,
Akure (FUTA)
Akure, Nigeria
josnno@yahoo.com

Abstract

The purpose of this paper is to introduce a system that uses the combination of working aid blind techniques to help blind people to have better navigation against objects, also to be aware of road and traffic lights signs using real-time image-processing. This study aims at improving the processing speed of object detection by introducing the latest Raspberry Pi 4 module, which is more powerful than the previous versions. As a result, the Single-Shot Multibox Detector MobileNet v2 convolutional neural network on Raspberry Pi 4 using TensorFlow Lite 2, is employed for object detection. A model called SSD MobileNet v2 320x320, which is trained on the COCO dataset from the TensorFlow model zoo, which is used in this system. Also, the traffic light signs classification is achieved via machine learning techniques. With all modules attached on the hat, video frames captured from the piCamera module are analyzed and searched for

known objects by the Raspberry Pi module and playing the pre-recorded sound of the detected object(s) at an approximate distance away from the user. MobileNet uses 3 x 3 depthwise separable convolutions, which uses between 8 to 9 times lesser computing resources compared to the standard convolution at a small cost in accuracy. Thus, this makes the device suitable for the resource constraint devices such as the Raspberry Pi. The model achieves about 5 frames per second on the Raspberry P 4.

CCS Concepts: • Computing Methodologies → Neural Networks; • Computer Vision; • Computer systems organization → Embedded systems;

Keywords: Blind, Hat, Image-processing, MobileNet, Object Detection, Raspberry Pi, Pi Camera

ACM Reference Format:

Matshehla. Konaite, Pius A Owolawi, Temitope Mapayi, Vusi Malele, Kehinde Odeyemi, Gbolahan Aiyetoro, and Joseph S. Ojo. 2021. Smart Hat for the blind with Real-Time Object Detection using Raspberry Pi and TensorFlow Lite. In *International Conference on Artificial Intelligence and its Applications (icARTi '21)*, December 9–10, 2021, Virtual Event, Mauritius. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3487923.3487929>

1 Introduction

According to the World Health Organization (WHO) [11], at least 2.2 billion people have a near or distance vision impairment. In at least 1 billion of these cases, vision impairment

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

icARTi '21, December 9–10, 2021, Virtual Event, Mauritius

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8575-6/21/12...\$15.00

<https://doi.org/10.1145/3487923.3487929>

could have been prevented, or has yet to be addressed [3]. Thus, there is a need for advanced and cost-effective technology solutions to assist blind people to be more aware of their surroundings. The ability to see plays an important aspect of everyday life. It gives us a clear view of our surroundings so we can better interact with the surroundings. Blind people face difficulties when it comes to being fully aware of certain objects and their surroundings. The purpose of this smart hat is to assist blind people in detecting objects, and giving the audio output of the detected objects.

A lot of tools and techniques have been introduced and applied, which helps blind people to navigate around much easier. This paper proposes an efficient system that uses real-time object detection to help blind people better navigate through their surroundings. Section 2 reviews related works in building smart object detection systems for the blind. Section 3 describes the methods that were used to produce the smart hat. Section 3 describes, in detail, different frameworks and different portions of the systems that were used in the overall system. Our system is tested in the different scenes, and results are discussed in Section 4. Section 5 closes with a summary and conclusion.

2 Related Works

There are different systems that are used in object detection to help blind people to be more aware of objects and their surroundings. One of these systems is called Advanced Indoor and Outdoor Navigation System for Blind People Using Raspberry Pi [1]. This system uses a camera and Infrared sensors to get input from the surroundings. The camera is used to capture a video of the surroundings and sends it to the Raspberry Pi 3 module, where image procession detects certain objects in each frame. The system also has an Infrared sensor attached to the stepper motor, which scans for nearby objects all-around and produces vibration using the vibrating motor on the left and the right-hand sides. However, this system has processing power limitations in the Raspberry Pi 3 module, which is limited to the processing power of 1.4GHz CPU and 1GB LPDDR2 SDRAM, and therefore having limited processing power.

The Smart Cap – Wearable Visual Guidance System for Blind is made on a Cap that uses a NoIR camera and Raspberry pi 3 and is programmed in Python and TensorFlow library [7]. This system captures video in real-time and, using the `ssd_mobilenet_v1_coco_11_06_2017` model, objects are detected and results are stored in a text format. The text is then to voice output using the Speak, and played through the headsets. Since this system utilizes the Raspberry Pi 3 module, which has limited processing power, it is greatly affected in terms of processing speed. An indoor navigation system has been proposed for the blind where Yang and Sanii have used augmented reality markers to mark certain locations in the household [11]. This helps blind people when

navigating inside the house. The drawback for the system is that is efficient for indoor use only, and it is not useful for outdoor applications.

Priya et al [8] made the Real-Time Text Detection and Shopping Assistance for the Visually Impaired system. The system uses a camera that captures images of products, and then the image was processed by a computer to recognize the text on the label and the name of the brand of the product using optical character recognition. The recognized characters of the text are merged together to form a word, and output is given in the form of audio or speech. The drawback with this system is that it only recognizes products in the shop, and lacks mobility as the processing is done on a laptop or a computer. Another system uses deep-learning and machine vision techniques and robot operating system (ROS) has been proposed which is wearable and consists of ultrasonic sensors attached to the smart glass to detect objects ahead in real-time and feed them to the Raspberry Pi module for analysis of whether the object is closing in fast and if so, produces vibrations in the recognized direction as a warning [9]. With an added GSM feature, the system is able to make calls in an emergency situation. The drawback with this system is that it does use a camera, thus, it is not able to precisely identify objects according to the names.

Motivated by the above works, the proposed system in this paper uses the pre-trained SSD MobileNet v2 convolutional neural network (CNN), which was trained on the COCO dataset [4] and has 90 different types of objects taken from our everyday life. This network is loaded into the Raspberry Pi 4 module, which is coded in Python for processing. This Python code, using the Single-Shot Multibox Detector (SSD) model [5], detects and identifies objects from the video input obtained from the Pi camera. Once an object, or multiple objects, have been detected, the pre-recorded sound of the object detected will be played through the speaker. This system is easy to wear and is rechargeable as it uses a power bank.

3 Methodology

Under normal circumstances, blind people use walking sticks, which assist them to navigate around and avoid obstacles. However, the limitation of this method is that it cannot help the blind person detect objects that are further away, and cannot alert the blind person of other objects. This system is designed to help the blind person to be more knowledgeable of other objects and their estimated distance, as shown in Figure 1 below.

3.1 System Operation Overview

This smart cap is based on TensorFlow Lite and speech output. The SSD-MobileNetV2 used in this system can detect multiple objects that are present in one image or frame. The model detects the position of each image and outputs the

name of the object and its bounding boxes. This pre-trained SSD-Mobile model can classify 90 different objects. This can be a problem when outputting the results if too many objects are detected at the same time as it will take too long to notify the user of each object. To avoid this, the output has been fine-tuned to output only the most common objects, such as a person, car, television, computer, dog, cat, stop sign, and traffic lights.

As illustrated in Figure 3, the working of the systems is initialized by the powering of the Raspberry Pi 4 micro-controller, and therefore the microcontroller powers on the camera module. The camera module captures a real-time video and feeds the frames into the Raspberry Pi. The Raspberry Pi, with TensorFlow Lite running the Python code, is loaded with the pre-trained SSD-MobileNet-V2 model to detect objects and output the results of the detected object and its estimated distance through the speaker.

Figure 1. System operation overview

Figure 2. Block Diagram

The output of each object depends on the threshold, which is set at 0.5. Any object detected with a higher level of confidence is listed in part of the output, and any object with less than 0.6 level of confidence is discarded. The bounding box and the results can be displayed when the raspberry pi is connected to a display. The speaker, or headsets, are connected to the audio jack of the Raspberry pi 4 used to output the sound results of the detected object and their estimated distance to the user.

3.2 Systems Description

3.2.1 TensorFlow Lite API. TensorFlow Lite is an open-source deep learning framework designed for resource constraint devices such as the Raspberry pi modules and mobile devices. TensorFlow Lite makes it possible to run TensorFlow models on mobile, embedded, and IoT devices. It enables on-device machine learning inference with low latency and small binary size. Therefore, it improves latency, and power consumption gets reduced.

3.2.2 Single-Shot Multibox Detector (SSD). A Single-Shot Multibox Detector is an architecture that has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. The SSD makes 8732 predictions for better coverage of location, scale, and aspect ratios, more than many other detection methods. Many predictions will contain no objects, therefore, any predictions with class confidence scores lower than 0.01 will be discarded [5]. For 300×300 input image, SSD achieves 74.3 per cent mean average precision (mAP) on the VOC2007 test dataset at 59 FPS on an Nvidia Titan X and for 512×512 input, SSD achieves 76.9 % mAP, outperforming a comparable state of the art Faster R-CNN model [5]. Even with smaller input image size, SSD has much better accuracy, even with a smaller input image size when compared to other single stage methods such as YOLO [5].

Table 1 shows the MobileNet overall architecture, where t is the expansion factor, c is the number of output channels, n is the repeating number, and s is the stride. 3×3 kernels are used for spatial convolution. Similar to the SSD MobileNet V1 model, softmax is used as a classification method in the SSD MobileNet V2 model. More information on the implementation of the architecture can be found in [2].

Figure 3. Flowchart showing working of the system.

3.2.3 OpenCV. OpenCV is an open-source library for real-time computer vision. It assists us in drawing and displaying the bounding boxes and labeling the detected objects visually on each video frame.

Table 1. MobileNetV2 Overall Architecture

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
224×224×3	conv2d	-	32	1	2
112×112×32	bottleneck	1	16	1	1
112×112×16	bottleneck	6	24	2	2
56×56×24	bottleneck	6	32	3	2
28×28×32	bottleneck	6	64	4	2
14×14×64	bottleneck	6	96	3	1
14×14×96	bottleneck	6	160	3	2
7×7×160	bottleneck	6	320	1	1
7×7×320	conv2d 1×1	-	1280	1	1
7×7×1280	avgpool 7×7	-	-	1	-
1×1×1280	conv2d 1×1	-	k	-	-

3.3 SSD-MobileNetV2 Model

Single-Shot Multibox Detector (SSD) [5] models are faster than R-CNN models because they eliminate proposals for bounding boxes. The reason behind the selection of the SSD-MobileNetV2 model has been the processing speed of detection and size of the model. As shown in Table 2, the model takes inputs images of 320x320 and takes only 19 milliseconds to detect objects and their locations in that image, whereas other models take more time. The second fastest model, called the *ssd_mobilenet_v1_coco*, takes longer to classify objects in an image, at 0.3 milliseconds, and the third fastest, *ssd_mobile_net_v2_coco* takes 31 milliseconds, and so on. The SSD MobileNet v2 320x320 is the latest MobileNet model configured for Single-Shot Multibox detection, which is optimized for speed at a very small cost in mean average precision (mAP) of only 0.8, as compared to the second-fastest model, SSD MobileNet v1 coco [10]. OpenCV is an open-source library for real-time computer vision. It assists us in drawing and displaying the bounding boxes and labelling the detected objects visually on each video frame.

By referring to Figure 4, a 640x320 pixels image is received by the Python code, whereby the Python code will resize this image to a 320x320 pixel image, which is the size that our model is able to handle during processing. The Python code will pass the resized image to the Single-Shot Multibox Detector model to scan for any objects and, if available, the name(s) of the objects is stored in a variable. The area (height and width) of the object is determined in terms of pixels it occupies out of 320x320 pixels and uses that to calculate the distance of the object away from the user. An object such as the South African stop sign is approximately 0.5 meters in height and width. By a few estimation tests, an object with 0.5 height and 0.5 width occupies approximately

15 percent area of the 320x320 pixel image frame, which is determined when the camera is placed 8 meters away. Then, the following distance constant and distance formula are developed.

Let:

$$\begin{aligned}
 \text{Distance constant for Road Signs} &= C_1 \quad (1) \\
 &= \frac{(\text{Distance of the camera when image was taken})}{(\text{Percentage of area the sign occupies on the image})} \\
 &= \frac{8m}{0.15} = 53.33m
 \end{aligned}$$

Table 2. Models comparison

Model name	Speed (ms)	COCO mAP	TensorFlow version
SSD MobileNet v2 320x320	19	20.2	2
ssd_mobilenet_v1_coco	30	21	1
ssd_mobilenet_v2_coco	31	22	1
Faster R-CNN ResNet50 V1 640x640	53	29.3	2
faster_rcnn_inception_v2_coco	58	28	1

It is also known that objects such as cars and other vehicles are approximately 1.6 meters in height, and about 4.5 meters in width. By a few estimation tests, using a medium-sized car, an object with a 1.6-meter height and 4.5 meters width occupies approximately 24 percent area of the 320x320 pixel image frame, which is determined when the camera is placed 8 meters away. The following distance constant for cars and other vehicles are given as:

$$\begin{aligned}
 \text{Distance constant for Cars} &= C_2 \quad (2) \\
 &= \frac{(\text{Distance of the camera when image was taken})}{(\text{Percentage of area the sign occupies on the image})} \\
 &= \frac{8m}{0.24} = 33.33m
 \end{aligned}$$

Figure 4. Sample of an Input Image

The general distance expression employed is then given as:

$$Distance = d = \frac{Area\ of\ pixels\ occupied\ by\ the\ object}{Total\ Area\ of\ pixels * distance\ constant}$$

$$d = \frac{P}{T}(C) \quad (3)$$

Where,

P = Area of pixels occupied by the object.

T = Total Area of pixels the image has.

C = Distance constant we determined by tests, not precise, but approximated (in meters). The distance constant is equal to 53.33m for small objects and 33.33m for medium-sized objects.

From formula (3), the expression to determine the distances of road signs and small objects is given in (4), and the distances for cars and other medium-sized objects is given (5) as follows:

$$d = 53.33 \frac{P}{T} \quad (4)$$

$$d = 33.33 \frac{P}{T} \quad (5)$$

Figure 5. Smart hat prototype

4 Results and Discussion

The image of the prototype of this system is shown in Figure 5. Figures, 6, 7, and 8 show that the results of our model, which is able to predict and classify objects well, at an accuracy of over 60 percent. The model is also capable of classifying multiple objects of different classes in an image. The model has good accuracy but since it is based mostly on processing speed, some other aspects of object detection are slightly affected. One aspect that is affected, is the fact that some objects are discarded from the output results due to them having lower accuracy than our threshold of 60 per cent. . This can be seen in Figure 6 of model results. The model was able to correctly detect the traffic lights and the cars at above 60 per cent of accuracy in the scene, but it

was not able to detect the minibus and the person on the sidewalk near the trees.

Table 3. Models comparison

Model name	Speed (ms)	COCO mAP	N Frames/s
SSD MobileNet v2 320x320	19	20.2	5.26
ssd_mobilenet_v1_coco	30	21	3.33
ssd_mobilenet_v2_coco	31	22	3.23
Faster R-CNN ResNet50 V1 640x640	53	29.3	1.89
faster_rcnn_inception_v2_coco	58	28	1.72

The Model seems to identify objects even better when objects are closer. As shown in Figure 7 of model results, the car has been detected with an accuracy of 69 per cent, and the stop sign, with 71 per cent accuracy, whereas in Figure 8, the results were 75 per cent and 76 per cent respectively. This is the disadvantage of the system, but this does not affect the systems' aim too much, as we would like the system to detect nearby objects mostly. The last column of Table 3 shows that the SSD MobileNet v2 model has the highest frames per second at the value of 5.26. With this, it means that it is possible to process approximately 5 frames in a second, which is ideal for real-time object detection. If we let the response time of the model be given as Speed, the number of frames N the model can process in a second is given as follows:

$$N = \frac{1}{Speed} \quad (6)$$

Figure 6. Model Results 1

Figure 7. Model Results 2

Figure 8. Model Results 3

The number of frames per second can be seen on each results image at the top-left corner as the value of the FPS. The value ranges roughly around 4.5 frames per second. The reason for this is because some of the processing time is consumed by the python code in displaying the results using OpenCV. The time it takes for the output sound to play may affect the speed of detection, therefore, the output sound needs to be shortened to be as short as possible.

5 Conclusion

The purpose of this system was to help blind people to be more aware of their surrounding objects and environment, so that they can better navigate independently and carefully. In this system, Raspberry Pi 4 is employed along with the pi camera and a speaker. The camera is used to record video and send it to the Raspberry Pi module, which is running on TensorFlow Lite 2 framework, which has a Python code that does object detection on each frame. Once objects have been detected, the results are played out in sound through the speaker to notify the blind person of the object(s) that are being detected, along with the estimated distances of the object(s) away from the blind person. The distance of a small object, such as a road sign is calculated by dividing the total area of pixels the object occupies on the 320x320 image, by the total area of pixels of the image, and then

multiplying by the distance constant, 53.33 meters, whereas for medium-sized objects, such as cars, we multiply by the distance constant, 33.33m. If a small object occupies more than 15 percent of the total area of pixels of the image, it is considered to be nearby, otherwise, it is considered to be farther away by more than 5 meters. If a medium-sized image occupies more than 24 percent of the total area of the image, it is considered to be nearby, otherwise, it is considered to be more than 5 meters away. With the aid of the SSD-MobileNet v2 model, the proposed system can detect objects along with their accuracies well. All objects detected by the model, with accuracies of less than 60 percent, are discarded from the output results to avoid the issue of ambiguity. Since our model is based mostly on speed, some other aspects, such as accuracy, are slightly affected.

References

- [1] M Anandan, M Manikandan, and T Karthick. 2020. Advanced Indoor and Outdoor Navigation System for Blind People Using Raspberry-Pi. *Journal of Internet Technology* 21, 1 (2020), 183–195.
- [2] Yu-Chen Chiu, Chi-Yi Tsai, Mind-Da Ruan, Guan-Yu Shen, and Tsu-Tian Lee. 2020. Mobilenet-SSDv2: An improved object detection model for embedded systems. In *2020 International Conference on System Science and Engineering (ICSSE)*. IEEE, 1–5.
- [3] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
- [4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.
- [6] Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE, 342–347.
- [7] A Nishajith, J Nivedha, Shilpa S Nair, and J Mohammed Shaffi. 2018. Smart cap-wearable visual guidance system for blind. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 275–278.
- [8] M Padma Priya, JK Periasamy, P Jayaselvi, and R Aditya Siddharth. 2019. Real Time Text Detection and Shopping Assistance for the Visually Impaired. *International Journal of Psychosocial Rehabilitation* 23, 04 (2019).
- [9] Aswath Suresh, Chetan Arora, Debrup Laha, Dhruv Gaba, and Siddhant Bhambri. 2017. Intelligent smart glass for visually impaired using deep learning machine vision techniques and robot operating system (ROS). In *International Conference on Robot Intelligence Technology and Applications*. Springer, 99–112.
- [10] V Vijeesh, E Kaliappan, B Ponkarthika, and G Vignesh. 2019. Design of Wearable shoe for blind using LIFI Technology. *International Journal of Engineering and Advanced Technology (IJEAT)* 8 (2019), 1398–1401.
- [11] WHO. 2021. World Health Organization. <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment> [Accessed 15 August 2021].