

Intelligent Interfaces for Assisting Blind People using Object Recognition Methods

Jamil Abedalrahim Jamil Alsyayadeh^{1*}
Hasvinii Baskaran⁴

Department of Electronics & Computer Engineering
Technology, Fakulti Teknologi Kejuruteraan Elektrik &
Elektronik (FTKEE), Universiti Teknikal Malaysia Melaka
(UTeM), Melaka, Malaysia

Irianto²

Department-General Education, Faculty of Resilience,
Rabdan Academy, Abu Dhabi, United Arab Emirates

Maslan Zainon³

Department of Electrical Engineering Technology
Fakulti Teknologi Kejuruteraan Elektrik & Elektronik
(FTKEE), Universiti Teknikal Malaysia Melaka (UTeM)
Melaka, Malaysia

Safarudin Gazali Herawan⁵

Industrial Engineering Department
Faculty of Engineering, Bina Nusantara University
Jakarta, Indonesia 11480

Abstract—Object recognition method is a computer vision technique for identifying objects in images. The main purpose of this system build is to put an end to blindness by constructing automated hardware with Raspberry Pi that enables a visually impaired person to detect objects or persons in front of them instantly, and inform what is in front of them through audio. Raspberry Pi receives data from a camera then processes it. In addition, the blind will listen to a voice narration via an audio receiver. This paper's key objective is to provide the blind with cost-effective smart assistance to explore and sense the world independently. The second objective is to provide a convenient portable device allows users to recognise objects without touch, having the system determine the object in front of them. The camera module attached in Raspberry Pi will capture image and the processor will then process it. Subsequently, the processed image sends data to the audio receiver narrating the detected object(s). This system will be very useful for a blind person to explore the world by listening to the voice narration. The generated voice narration after processing the image will help the blind to visualise objects in front of them.

Keywords—Object recognition method; computer vision; blind people; image processing; Raspberry Pi; Pi camera; smart assistance; portable device; voice narration; visualise

I. INTRODUCTION

Physical movement is a challenge for the blind. Visual disability stands out from the many extreme obstacles affecting a person. This is clearly explained in this research study about the visually impaired patients. They face physical and social constraints accrued from their visual loss, and they need to improve on their health and independence [1][2][3]. Designing a gadget to help the blind is not something new. Various technologies exist to help the visually impaired, and as innovations increasingly propelled, ideas appear to provide intriguing measures to help the visually impaired. In any case, designing a device to aid blindness comes with a price and does not come easy, as it is often regarded as a so-called luxurious item in most developing nations.

As indicated by the World Health Organization (WHO) research study, [4][5] it has been estimated that over 1.3 billion people around the world have some form of vision impairment. Roughly, 80% of all kinds of vision debilitation are viewed as avoidable. Additionally, the WHO also mentioned that most visually impaired adolescents would require visual recovery intercessions for self-improvement. Nevertheless, it is most often that visual recovery treatments come with substantial hospital expenses, and with 90% of disabled people living with financial difficulties, visual restoration is not the best alternative for all. For full psychological improvement and better independence without bearing costly bills, blind people require an assistive device that helps them with their daily activities. There exist many assistive devices for visually impaired people and it became the inspiration in the background of this research study. Smart assistance such as a smart and autonomous walking stick, smart glasses/spectacles, or prosthetics [6][7][8][9]. The assistance from another individual is not always accessible, and are unfavoured by visually impaired individuals that search for freedom, without having to bear the cost of such expensive smart assistance equipment as well. We propose an audio receiver for blind people that uses real-time smart assistance interfaces and object recognition technologies as a solution to this occurring issue. Our system mainly consists of two components, Raspberry Pi and Pi camera. The smart assistance audio guidance was developed to assist users to determine objects in front of them and help them visualise the environment around them. The camera in the processor will capture image, then processes that image, and a voice narration will be sent through audio receiver.

The main objective is to provide the blind cost-effective smart assistance to explore and feel the world independently. This enables the blind to visualise their surroundings and afford current technologies. Additionally, we also aim to provide a portable device which is easy to utilise and permits them to recognise objects without touching, and describes the surroundings in front of them.

*Corresponding Author.

This system is used to assist blind people with voice narration, processed by the Raspberry Pi processor. This portable electronic device's purpose is to give voice narration informing what is in front of them. An important objective is to provide a portable device that is simple to use and low cost and affordable smart assistance to blind people. Another goal is to extend the computerised electronic travel aid for the blind by applying real-time object recognition technology. This blind guidance system is solid and financially perceptive. Real-time based smart assistance interfaces the audio receiver for blind people with voice narration by using object recognition methods to provide the blind with cost-effective smart assistance to explore and sense the world independently. Audio guidance helps them to know what is happening around them and it helps them to visualise their surroundings. By using real-time, the system will recognise the objects faster.

The remaining of this paper has been organized as follows: Section 2 discusses the related works. The background of the study is described in Section 3. Section 4 described the system implementation and testing. Section 5 described the results and discussion and finally, the conclusion is described in Section 6.

II. RELATED WORK

There are a lot of assistive devices for visually impaired people to sense the world independently. All these devices rely mainly on ultrasonic sensors and Brailling.

A. EyeCane and EyeMusic

Maidenbaum et al. [10] designed EyeCane and EyeMusic to improve upon, or likely be within the far distant future, to update the traditional white cane. By applying statistics at visually far distances (5 meters) and greater angles, and most significantly by means of discarding contacts among the cane, and the user's surroundings in cluttered or indoor environments. The EyeCane converts point-distance information into aural and tactile signals. The Prototype of EyeCane and EyeMusic is shown in Fig. 1. The tool can provide distance information to the customer from two different directions at the same time: immediately in advance for long-distance perception and detection of waist-height obstacles, and pointing downward at a 45° angle for ground-level evaluation.



Fig. 1. Prototype of EyeCane and EyeMusic.

B. Blitab

Blitab is a device nicknamed "the iPad for the visually impaired". It appears similar to a digital book, however, its screen utilises smart liquids that protrude tactile pixels to show braille letters, making it conceivable for the blind to see entire pages of braille message at once. Perkins-style keyboard application, text-to-speech yield, and touch navigation provide a completely new user experience for braille and non-braille blind individuals. It empowers the fast conversion of any content into braille. Blitab is a platform for all current and future programming applications for visually impaired people, it is not only a tablet. The Prototype of Blitab is shown in Fig. 2.

Blitab is the world's first real tactile tablet designed specifically for the blind and visually impaired. The device's revolutionary smart liquid technology also allows it to display material images for blind people who do not use braille [11].

C. BrainPort V100

According to Grant et al. [12], BrainPort V100 is an oral electronic vision aid that uses electro-tactile stimulation to help profoundly blind people with direction, mobility, and object recognition. The device is used in conjunction with other assistive devices like a normal white cane or a guide dog.

It deciphers digital data from a wearable camcorder into delicate electrical incitement designs on the outside of the tongue. Users feel moving bubble-like patterns on their tongue then they figure out how to interpret or visualise according to the shape, size, area, and movement of articles in their condition. A few clients have portrayed it as having the option to "see with your tongue". What makes it extraordinary is seeing with your mouth may appear to be outlandish at first, yet with at least 10 hours of one-on-one instructional courses, wearers can figure out how to comprehend the shivers and "see" where objects are found, yet additionally, their size, shape and in the event that they are moving. In a clinical preliminary, 69% of members had the option that effectively recognises protests in an acknowledgment test following one year of preparing with the BrainPort. The Prototype of BrainPort V100 is shown in Fig. 3.



Fig. 2. Prototype of Blitab.



Fig. 3. Prototype of BrainPort V100.

TABLE I. COMPARISON BETWEEN EXISTING SYSTEM

System	Devices needed	Cost	accessibility	purpose
BrainPort V100	Headset, Intra Oral Device(IOD)	Expensive (\$10000)	Controller	To provide oral electronic vision aid
EyeCane& EyeMusic	Infrared emitters, web camera, smartphone, headset	Low cost	Infrared sensors	To provide navigation control and identifies colour, shape and location of objects.
Blitab	Touch screen tablet, Braille lines	Low cost(\$500)	Braille display	Displays tactile images to blind people.
Proposed method	Raspberry Pi 3, Pi camera module, Headset	Cheap (\$100)	Text-to-Speech module	The generated narration will be the final output that is transmitted to the user through a headset.

The difference between existing systems is shown in Table I.

III. BACKGROUND OF THE STUDY

Object Recognition is a method used in image processing to recognise real objects. This method is clearly explained in a

research study about the importance of process that will help blind people to identify their daily items that are commonly used. Our system provides some kind of visual aid that recognises objects dynamically [13]. The algorithm used in this system analyses the object. For instance, a blind person is sitting on his dining table. He has multiple objects in front of him such as bottle, chair, dining table, etc. Therefore, our system will help him by narrating what is in front of them. Text-to-Speech module is used to convert text to speech. The text that is written in text file is the output of object detection. Google API is used for conversion of Text-to-Speech dynamically, provided that the internet connection is stable. This has been studied from a research that explains about Google API that is used for text-to-speech [14]. For example, if the camera captures a book in front of it, it detects the book and converts it into text from the image captured. The text will be written in a text file and then converted to speech by using Google Text-to-Speech. The architecture of this proposed system is the Raspberry Pi board. Raspberry Pi controller controls the system and activates the output and sends the instructions. The detailed specifications of Raspberry Pi 3 B+ consists of: four USB ports, an Ethernet port, forty GPIO pins, SD card slot, SOC (system on a chip), a DSI display interface, HDMI port, LAN controller, audio jack, CSI camera interface, RCA video socket, and 5V micro USB connector [15].

The Block diagram of the object recognition process is shown in Fig. 4.

The Pi camera is connected to a CSI camera interface of Raspberry Pi processor. The processor has an operating system named Raspbian, which processes the image, voice narration and other conversions. The headset will connect to an audio jack for audio output. Once the system components activate, the camera module will begin a video stream of its front view, and the image in video will be processed. Before this process starts, the Raspberry Pi will create a video frame, activates "cv" environment, and runs the python script to activate the system. Thereafter, the processed image undergoes object detection for image classification and recognition. Hence, the image in the video will detect through real-time object recognition, and the label of each object will be printed in a text file, which is used for voice narration. The labels in the text file use Google Text-to-Speech for voice narration. The generated narration will be the final output that is transmitted to the user through a headset.

The flowchart of the object recognition process is shown in Fig. 5.

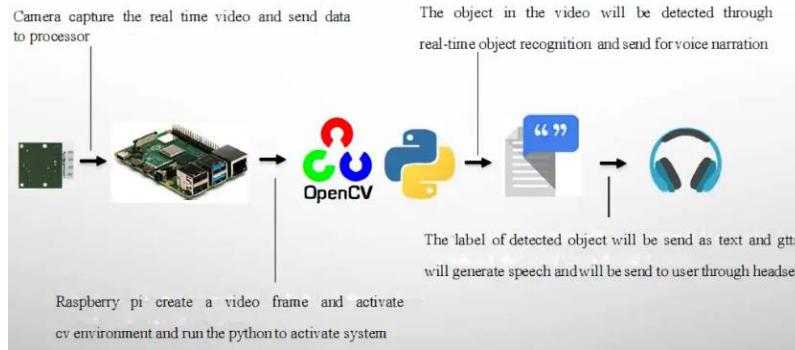


Fig. 4. Block Diagram of the Object Rprocess.

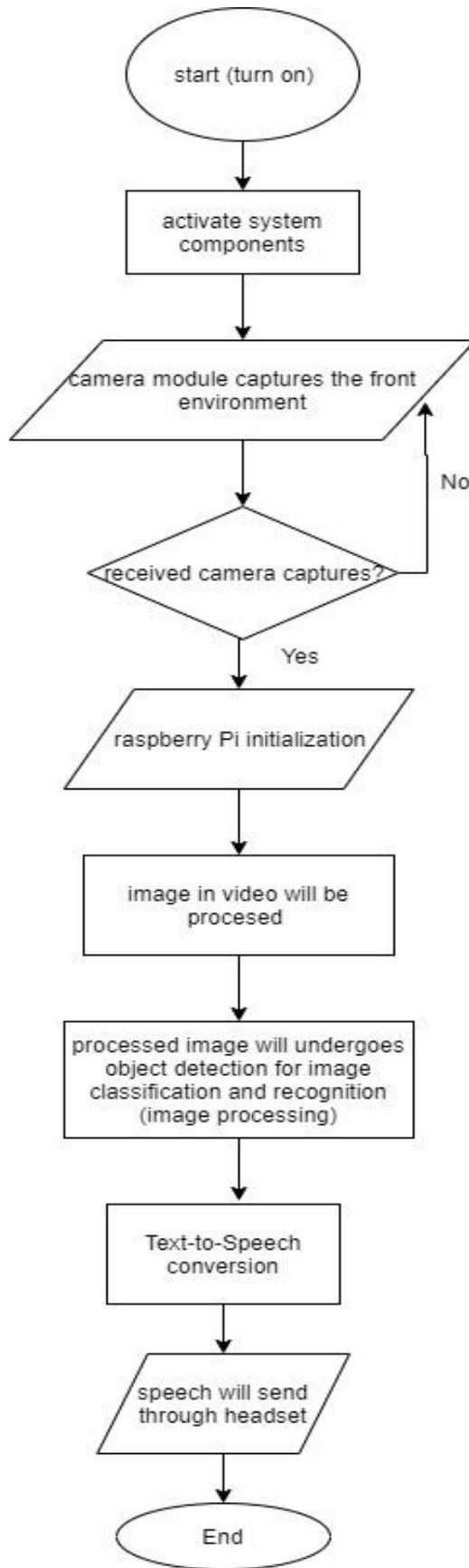


Fig. 5. Flowchart of the Object Recognition Process.

IV. SYSTEM IMPLEMENTATION AND TESTING

A. Hardware Implementation

The necessary components in developing this system consist of a Raspberry Pi and Pi camera. The New Out of Box Software (Noobs) is installed on an SD card to format the Raspberry Pi that will be fixed in the Raspberry Pi, as studied in the manual that was given to study about Raspberry Pi startup [16]. Noobs contain Java SE Platform Products. It is an operating system installer with Raspbian pre-loaded. Once done, this Raspberry Pi will connect to power, start to boot, and be ready to use the operating system, whereas the Pi camera will be configured beforehand. The camera's interfacing option in Raspbian OS will be enabled manually to allow the camera to work with the system. Once the configuration is done, the Raspbian enables the camera. The image captured after configuring the Pi camera is shown in Fig. 6.

B. Software Implementation

The Raspbian operating system is used in the Raspberry Pi 3 model B+ as a platform to run this system; which is, the platform to create, run, and troubleshoot the coding of the software that has been used. Python IDLE software was used to build this system. Python IDLE ran in an OpenCV environment. OpenCV was created to provide a common infrastructure for computer vision applications such as deep learning, optical character recognition (OCR) and object detection, and more as explained in the article [17][18]. OpenCV-Python is the Python API for OpenCV. It's a Python bindings library aimed in solving computer vision challenges. Python has been enhanced with C/C++, enabling programmers to write/express code and develop Python wrappers that can be used as Python modules, as stated in the article named Python. It is packaged as an optional part of the Python packaging with many Linux distributions [19][20]. The actual code will run in the background of the CV environment. To write the necessary codes to run the system, the Python 3.7.3 was used. To capture the image, a Pi camera connected to a Raspberry Pi was used. Furthermore, code will be used to initialise the captured image.

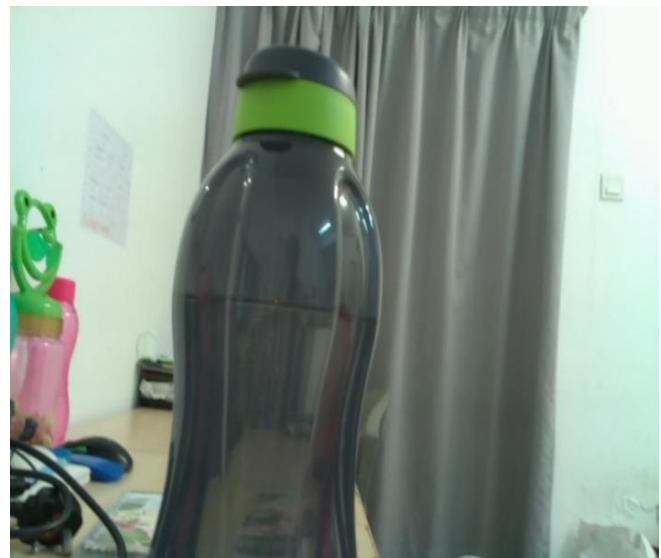


Fig. 6. Image Captured after Configuring the Pi Camera.

V. RESULT AND DISCUSSION

Here is a sample of coding and result for this proposed system. Some discussions are added up as an explanation to understand its function clearly.

A. Coding

Partially applied programming codes are displayed below in Fig. 7 to Fig. 10.

```
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
    help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
    help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
    help="minimum probability to filter weak detections")
args = vars(ap.parse_args())
```

Fig. 7. Construction of the Argument Parse.

The preceding code demonstrates how to construct an argument parse to parse the arguments. ‘ArgumentParser()’ converts the argument value from a string into some other type. The first line sets up an argument parser, followed by three mandatory command-line arguments. Firstly, the ‘prototxt’ is the path to the Caffe prototxt file which is known as the solver.prototxt, secondly, a configuration file, whereas ‘—model’ is the path to the pre-trained model and thirdly, the ‘—confidence’ is minimum probability threshold when filtering weak detections and it is set to 20% by default.

```
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
    "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
    "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
    "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
```

Fig. 8. Initialisation of ‘Classes’.

These lines of code initialise ‘CLASSES’, class labels, and equivalent COLORS, for on-frame text and bounding boxes. Furthermore, the last line loads the serialised neural network model.

```
for i in np.arange(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with
    # the prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the 'confidence' is
    # greater than the minimum confidence
    if confidence > args["confidence"]:
        # extract the index of the class label from the
        # 'detections', then compute the (x, y)-coordinates of
        # the bounding box for the object
        idx = int(detections[0, 0, i, 1])
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # draw the prediction on the frame
        label = "{}: {:.2f}%".format(CLASSES[idx],
            confidence * 100)
        cv2.rectangle(frame, (startX, startY), (endX, endY),
            COLORS[idx], 2)
        y = startY - 15 if startY - 15 > 15 else startY + 15
        cv2.putText(frame, label, (startX, y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
```

Fig. 9. Looping the Detection.

This part explains how this system is able to detect numerous objects in a single image. First step is to loop over the detections. The chance of each detection will be checked and tallied with confidence. If the confidence exceeds the

threshold, the prediction will be displayed in terminal and drawn on the frame. Detections will undergo loops and its confidence value is extracted in each loop. Therefore, the class label index is extracted if the confidence level is greater than the minimal threshold, as well as the bounding box coordinates surrounding the detected objects that have been computed too, and a rectangle displaying text is created on the detected object. Labels containing CLASS name and confidence build, and displayed as the processed-colored rectangle created around the object. Finally, the system computes the colored text that was generated onto the frame by using the y-value.

```
from gtts import gTTS
from time import sleep
import os
import pyglet

file = open("label.txt", "r").read().replace("\n", " ")
language = 'en'
tts = gTTS(text = 'hi there. i am here to assist you to look the world in front of you. there is' + str(file) + 'in front of you', lang = 'en', slow = False)
tts.save('voice.mp3')
os.system("start /home/pi/Desktop/real-time-object-detection/voice.mp3")
print(file)
```

Fig. 10. Voice Narration.

To enable the Raspberry PI to “talk”, the Google Text to Speech (gTTS) module is used in Python is used and also imported into the Raspbian system. This is used to command the system to read the image classification result that has been written after the real time object detection process. To put it simply, this python coding aims to read the text file and then create a voice narration.

B. Result

The system has been tested and its functionality has been demonstrated as per the design. The system has been able to operate as designed, thanks to the combination of software and hardware components. The system interface with Pi camera will capture the front environment and the data will be transferred to the processor to process the image. Object recognition methods will enable image processing, convert it to text, and use Google-Text-to-Speech to create the voice narration and send it to the user through an audio receiver.

The detected objects in frame are shown in Fig. 11.

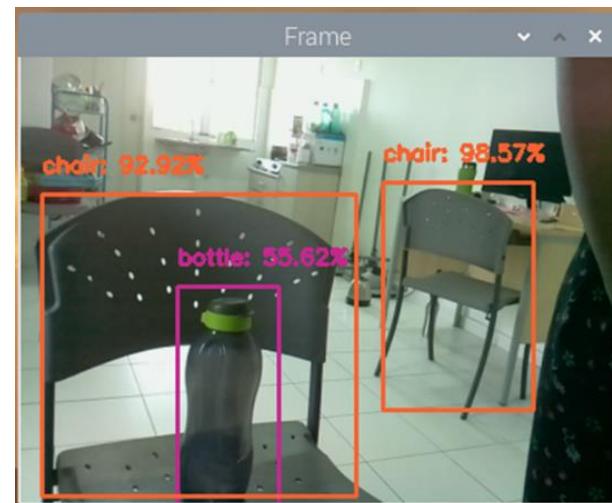


Fig. 11. Shows Objects Detected in Frame and its Confidence Percentage Value of the Detected Objects with its Label of Classes.

```
[INFO] loading model...
[INFO] starting video stream...
table: 57.87%
bottle: 26.65%
person: 99.65%
person: 98.43%
tvmonitor: 99.80%
tvmonitor: 99.46%
chair: 28.39%
person: 96.32%
chair: 54.55%
table: 33.28%
sofa: 51.36%
person: 78.48%
person: 99.10%
person: 97.11%
person: 99.15%
table: 26.27%
person: 49.29%
person: 95.13%
person: 98.91%
```

Fig. 12. Shows Label Classes will be Printed along with the Detection Confidence Percentage Value.

From Fig. 12, the results from the system are collected and the percentage of confidence value obtained, to show the objects detected along with its confidence percentage value. Confidence value is the probability that a bounding box containing an object and it is predicted by a classifier. The object in the bounding box would return many predictions, but out of those, most of them will have a very low confidence value associated. Hence, only predictions above 20% confidence is reported, as fixed in the python coding itself. That is how the object detection algorithm returns values after confidence thresholding, once the video stream starts in our system. As previously indicated in Fig. 11, the objects in the bounding box are correct, due to the quantifying the predictions.

To confirm the predictions, the correctness value of each object detection should be obtained. The measurement that determines the correctness of the bounding box is the Intersection over Union (IoU). IoU [21][22][23] is the ratio between the intersection, and the union of speculated boxes and ground truth boxes. The IoU's calculation is shown below in Fig. 13.

Consequently, the correct detections will be identified, and then its precision and recall will be calculated. To calculate precision and recall, the True Negatives, False Negatives, True Positives and False Positives will be identified. To obtain True Positives and False Positives, IoU will be used and the detection will be identified to determine whether it is correct (True Positive) or not (False Positive). The used threshold is 0.2, if IoU is > 0.2 , it is considered a True Positive. Else, it will be considered as a False Positive. The COCO (Common Objects in Context) evaluation metric suggests measurements through various IoU thresholds.

To calculate the recall, the count of Negatives is required because not every part of the image in the video stream frame detected is an accepted object or is considered a negative. False Negatives will only be measured if the objects detected by our system are missed out. The recall is calculated as the ratio between the number of correct predictions (A) (True Positive) and the missed detections (False Negatives). The correct predictions for each class in the video stream will be commutated after calculation of IoU using the ground truth boxes for each positive detection box that the system has reported. So, with this, the IoU threshold (0.2). Therefore, the formulas will be as below.

$$Precision = \frac{TP}{(FP+TP)} \quad (1)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

Subsequently, the Mean Average Precision (mAP) is calculated in Table II. mAP is used in the domains; Information Retrieval and Object Detection. These two domains have separate ways to calculate mean average precision. Object detection of mAP is formalised in the PASCAL Visual Object Classes (VOC). PASCAL VOC provides a common dataset of images and annotations, as well as a standard evaluation to the vision and machine learning communities [24][25]. The average precision for all object types is shown in the table below. The PASCAL VOC dataset's mAP was found to be 0.665. The best mAP value at the moment is reported to be 0.739.

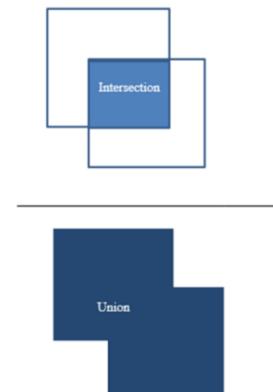


Fig. 13. IoU Calculation.

TABLE II. AVERAGE PRECISION FOR ALL CLASSES

No.	Class	Average Precision
1	Train	0.542
2	Bicycle	0.636
3	Dog	0.818
4	Diningtable	0.534
5	Aeroplane	0.727
6	Chair	0.909
7	Person	0.909
8	Tvmonitor	0.633
9	Bus	0.726
10	Sofa	0.710
11	Bird	0.727
12	Cow	0.632
13	Bottle	0.909
14	Pottedplant	0.359
15	Boat	0.544
16	Car	0.634
17	Cat	0.272
18	Sheep	0.633
19	Motorbike	0.724
20	Horse	0.726

VI. CONCLUSION

The system's goal of providing intelligent help for visually impaired people through real-time based object recognition has been successfully developed. Most of the important details in the general theory of design and execution have also been introduced throughout this article. From the theory to the practical realisation of this category of smart assistance for visually impaired people, these developments involve a variety of technical and coding details. From the testing and result analysis, the designed system's functionality is advanced and helps the visually impaired people to know what is in front of them. According to the data analysis based on Table II, the average precision for all classes is shown. Occasionally, detecting precision is not as precise as it should be, because the object is detected using values assigned by the system. Additional objects of comparable size or shape may also be detected with incorrect predictions. The strength of this system is users are able to listen to the voice narration audio that informs them what objects are in front of them. The Mean Average Precision (mAP) was calculated. The PASCAL VOC dataset's mAP was found to be 0.665. The best mAP value at the moment is reported to be 0.739.

The limitation of this system is it only has one pi camera interfaced to raspberry pi to capture video stream. So the scope only for blind people since the system included with pre-trained model that used for object detection. The most important recommendation for improvement, is about a future work development by implementing the Non-Maximum Suppression, making the regions more accurate. The object detection algorithm is good but not very accurate sometimes, because the regions reduce the ratio of algorithm. Furthermore, the development of this system should include a more pre-trained model in larger numbers. Lastly, the system can be improved by being cloud based, that way, all the data that had been captured will be saved in the cloud, and it will be easy for the user's guardian to acknowledge the details this system has generated, and can include localisation to know the location of the user travelled.

ACKNOWLEDGMENT

The authors would like to thank Centre for Research and Innovation Management (CRIM) for the support given to this research by Universiti Teknikal Malaysia Melaka (UTeM). We thank also those who contributed in any other forms for providing their continuous support throughout this work.

REFERENCES

- [1] M. Glatz, R. Riedl, W. Glatz, M. Schneider, A. Wedrich, M. Bolz, R. W. Strauss. "Blindness and visual impairment in Central Europe", PLoS one. 2022; 17(1):e0261897. <https://doi.org/10.1371/journal.pone.0261897>.
- [2] SR. Flaxman, R.R.A. Bourne, S. Resnikoff, P. Ackland, T. Braithwaite, MV. Cincinelli, et al. Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis. Lancet Glob Health. 2017;5(12):e1221–e34. pmid:29032195.
- [3] TY. Wong, J. Sun, R. Kawasaki, P. Ruamviboonsuk, N. Gupta, VC. Lansingh, et al. Guidelines on Diabetic Eye Care: The International Council of Ophthalmology Recommendations for Screening, Follow-up, Referral, and Treatment Based on Resource Settings. Ophthalmology. 2018;125(10):1608–22. pmid:29776671.
- [4] B. Thylefors, A. D. Negrel, R. Pararajasegaram, and K. Y. Dadzie. "Global data on blindness", Bulletin of the World Health Organization. vol. 73, no. 1, pp. 115–121, 1995. PMID: 7704921; PMCID: PMC2486591.
- [5] A. Hydara, I. Mactaggart, S. J. Bell, J. A. Okoh, S. I. Olaniyan, M. Aleser, H. Bobat, A. Cassels-Brown, B. Kirkpatrick, M. J. Kim, I. McCormick, H. Faal, M. J. Burton. "Prevalence of blindness and distance vision impairment in the Gambia across three decades of eye health programming". The British Journal of Ophthalmology. 2021 Dec;bjophthalmol-2021-320008. DOI: 10.1136/bjophthalmol-2021-320008. PMID: 34949578.
- [6] M. A. Iqbal, F. Rahman, M. R. Ali, M. H. Kabir and H. Furukawa, "Smart walking stick for blind people: An application of 3D printer", Proc. SPIE 10167 Nanosensors Biosensors Info-Tech Sensors 3D Syst., pp. 101670T, Apr. 2017.
- [7] A. Krishnan, G. Deepakraj, N. Nishanth, and K. M. Anandkumar, "Autonomous walking stick for the blind using echolocation and image processing," 2016. 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 13–16.
- [8] J. Bai, S. Lian, Z. Liu, K. Wang, and Di. Liu, "Smart guiding glasses for visually impaired people in indoor environment," IEEE Transactions on Consumer Electronics. vol. 63, no. 3, pp. 258–266, 2017.
- [9] Z. O. Abu-Faraj, E. Jabbour, P. Ibrahim, and A. Ghaoui, "Design and development of a prototype rehabilitative shoes and spectacles for the blind," 2012. 5th International Conference on Biomedical Engineering and Informatics, BMEI, pp. 795–799.
- [10] S. Maidenbaum, S. Hanassy, S. Abboud, G. Buchs, D. R. Chebat, S. Levy-Tzedek, A. Amedi. "The "EyeCane", a new electronic travel aid for the blind: Technology, behavior & swift learning," Restorative neurology and neuroscience. vol. 32, no. 6, pp. 813–824, 2014.
- [11] J. L. Robinson, V. Braimah Avery, R. Chun, G. Pusateri, W. M. Jay. "Usage of Accessibility Options for the iPhone and iPad in a Visually Impaired Population," Seminars in ophthalmology. vol. 32, no. 2, pp. 163–171, 2017.
- [12] Grant, P.; Spencer, L.; Arnoldussen, A.; Hogle, R.; Nau, A.; Szlyk, J.; Nussdorf, J.; Fletcher, D. C.; Gordon, K.; & Seiple, W. "The functional performance of the BrainPort V100 device in persons who are profoundly blind," Journal of Visual Impairment & Blindness. vol. 110, no. 2, pp. 77–88.
- [13] J. Spehr, "Object Recognition. In: On Hierarchical Models for Visual Recognition and Learning of Objects, Scenes, and Activities". Studies in Systems, Decision and Control, vol 11. Springer, Cham, 2015.
- [14] B. Sandeep, S. Palaniappan, "Kinect language translator by using Google API," International Journal of Pharmacy and Technology. vol. 8, no. 4, pp. 20051–20060, 2016.
- [15] J. Campos, S. Colteryah and K. Gagneja, "IPv6 transmission over BLE Using Raspberry PI 3," 2018. International Conference on Computing, Networking and Communications (ICNC), pp. 200–204.
- [16] M. Richardson and S. Wallace. "Getting Started with Raspberry Pi". O'Reilly Media, Inc., Sebastopol, 2012.
- [17] V. Sati, S. M. Sánchez, N. Shoeibi, A. Arora, and J. M. Corchado, "Face detection and recognition, face emotion recognition through NVIDIA Jetson nano," in Ambient Intelligence—Software and Applications, P. Novais, G. Vercelli, J. L. Larriba-Pey, F. Herrera, and P. Chamoso, Eds. Cham, Switzerland: Springer, 2021, pp. 177–185.
- [18] W. A. Indra, A.F.M.F Ismail, Nurulhalim Hassim, M.H. Idris, S.G.Herawan, N.S. Zamzam, F. Zuska., "Feasibility of RF RSL for RF Energy Harvesting : A Case Study of Alor Gajah Area," 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), 2021, pp. 24–28.
- [19] K. D. Ismael and S. Irina, "Face recognition using Viola-Jones depending on Python," Indonesian Journal of Electrical Engineering and Computer Science (IJECS), vol. 20, no. 3, pp. 1513–1521, December 2020.
- [20] N. Hassim, W. A. Indra, A.F.M.F Ismail, M.S.A.A Majid, S.G.Herawan, N.S. Zamzam, F. Zuska, "GSM900 Downlink Power Density Survey in Merlimau Area," 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), 2021, pp. 99–103.
- [21] F. Ahmed, D. Tarlow, and D. Batra, "Optimizing expected intersection-over-union with candidate-constrained CRFs," 2015. The IEEE International Conference on Computer Vision, pp.1850–1858.

- [22] W. A. Indra, S. G. Herawan Industrial, N. S. Zamzam, S. b. Mohd Najib Fakulti, N. b. Hassim Fakulti and F. Zuska, "Development of A Guided Drone Powered by Radio Frequency Energy Harvesting," 2021 IEEE International Conference in Power Engineering Application (ICPEA), 2021, pp. 127-131.
- [23] W. A. Indra, A. I. I. Jurjani, N. Hassim, S. G. Herawan, N. S. Zamzam and F. Zuska, "Digital TV Spectrum Survey for the Scope of Energy Scavenging in Jasin, Melaka," 2021 IEEE 17th International Colloquium on Signal Processing & Its Applications (CSPA), 2021, pp. 35-40.
- [24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," International Journal of Computer Vision. vol. 88, no. 2, pp. 303-338, 2010.
- [25] A. -A. Nayan, J. Saha, K. Raqib Mahmud, A. Kalam Al Azad and M. Golam Kibria, "Detection of Objects from Noisy Images," 2020. 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), pp. 1-6.