IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 November 2023, accepted 27 November 2023, date of publication 30 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3338154

RESEARCH ARTICLE

Obstacle Detection and Distance Estimation for Visually Impaired People

XINNAN LEONG AND R. KANESARAJ RAMASAMY, (Senior Member, IEEE)
Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63000, Malaysia
Corresponding author: R. Kanesaraj Ramasamy (r.kanesaraj@mmu.edu.my)

ABSTRACT In the realm of assistive technologies for visually impaired persons (VIPs), existing solutions such as white canes and guide dogs have limitations in range and practicality. Moreover, current electronic systems often fall short in terms of portability and the ability to estimate distances in real-time. To bridge these gaps, this study introduces a revolutionary wearable device comprising a Raspberry Pi, a camera module, and a pretrained convolutional neural network, all integrated into a pair of smart glasses. These glasses are designed to identify objects and estimate their distances from the wearer, providing real-time auditory or haptic feedback. The development process was rigorous, involving the deployment of machine learning algorithms for object identification and the integration of camera and sensor technology into a lightweight, user-friendly frame. The system's performance was extensively evaluated using quantitative metrics, showing its precision, speed, and usability. Conclusively, this study presents a significant leap in wearable assistive technologies, offering enhanced spatial awareness, autonomy, and quality of life for VIPs.

INDEX TERMS Object detection, IoT, 3D printing, ultrasonic sensor.

I. INTRODUCTION

A. PROJECT OVERVIEW

With the help of a Raspberry Pi and a camera module mounted on a glasses frame, this project aims to create a wearable gadget. The system is intended to recognise objects in the surrounding area and deliver auditory or vibratory feedback showing how far away the identified object is from the camera. A pretrained model is used to achieve object detection and distance measurement capabilities. The pretrained model is trained and tested using a dataset of photos of different objects. The pretrained model takes information from the photographs and categorises the items based on those details. It is designed to be extremely portable and simple to use, making it a useful tool for people who might have trouble seeing or sensing their environment. The following actions will be taken for the project:

1) Configuring the camera module and Raspberry Pi.
2) Making the Raspberry Pi capable of object detection and distance measuring.
3) Including the device in the glasses and evaluating its functionality.
4) Assessing the device's performance in recognising things and giving the user feedback.

The overall goal of this study is to show how wearable technology and pretrained model can improve people's abilities and quality of life.

B. PROJECT STATEMENT

1) PROBLEM STATEMENT 1

Assistive technologies nowadays have limitations and inconvenience. Assistive technologies nowadays, such as white canes and guide dogs, can be helpful. However, they have their limitations, such as white canes can only sense obstacles within their range and guide dogs must feed, house and care for their guide dogs which sometimes might be expensive.

2) PROBLEM STATEMENT 2

Most of the devices that utilise state-of-the-art detection systems are not designed for convenience wearable device. Devices that utilise state-of-the-art detection systems currently out right now are either too bulky or too heavy for VIPs to wear comfortably for everyday uses. Due to the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

computation needed for a CNN-based detection system, the developer tends to utilise a phone as a detection system or use a Wi-Fi-supported ESP32 Camera and send that footage to an external device for detection.

3) PROBLEM STATEMENT 3

The object detection system in the market does not support distance estimation. The current object detection systems available in the market are not capable of estimating the distance of the detected objects. This can be a significant limitation for individuals with visual impairments, as they rely on distance information to navigate and interact with their surroundings. With this information, they may be able to complete daily tasks such as crossing the street or identifying objects in the environment.

Recent advancements in assistive technologies for visually impaired persons have predominantly focused on enhancing navigation and object recognition. However, most of these innovations, like the use of smartphones for object detection or the deployment of non-wearable sensors, have limitations in terms of portability and real-time functionality. For instance, [1] demonstrated a smartphone-based system for object detection, which, while effective, necessitates handheld operation and lacks distance estimation capabilities. Similarly, [2] explored the use of ultrasonic sensors attached to canes, offering some distance information but at the expense of user convenience and range of detection. Contrasting these approaches, our project innovates by integrating object detection and distance estimation into a single, wearable device, combining the convenience of a hands-free experience with the accuracy and immediacy of real-time feedback. This integration represents a significant leap over existing technologies, particularly in terms of usability and practicality for everyday tasks

C. PROJECT OBJECTIVES

Objective 1: To develop a smart glasses using Raspberry Pi for detecting objects and provide auditory or vibration feedback.

Outcome 1: A portable and easy-to-use smart glasses that can detect objects in the environment and provide audio or vibration feedback indicating the distance between the camera and the detected objects.

Objective 2: To design a lightweight and comfortable wearable device which utilise convolutional neural networks-based object detection technology.

Outcome 2: A lightweight and comfortable wearable device that can detect objects using cutting-edge object identification technology, eliminating the bulk and discomfort of existing object detection systems.

Objective 3: To implement a distance estimation feature into the object detections system.

Outcome 3: A device that can provide individuals with visual impairments with the necessary information to navigate and interact with their surroundings more effectively by integrating distance estimating capabilities into the object detection system.

D. PROJECT SCOPES

The scope of this project involves developing the object detection system using machine learning algorithms, designing and prototyping the glasses frame, integrating the Raspberry Pi camera and ultrasonic sensor, and calculating the separation between the camera and the identified object. To correctly recognise and categorise objects in the field of view of the glasses camera, the object identification system will be trained on a sizable collection of object images. Real-time audio or vibration feedback will be provided by the device. The project will also involve testing and assessing the object detection and distance measurement systems' precision, speed, and usability.

Aside from identifying prospective areas for use, the project does not entail developing applications or industries for the object detection system. It also excludes long-term maintenance and updates to the object detection system after the initial prototyping and development phase. The project focuses on creating and implementing the object detection system as well as the distance measurement system itself.

E. DELIVERABLE

1) A wearable device using a Raspberry Pi and a camera module that is housed on a glasses frame and that is capable of detecting objects in the environment and providing audio or vibration feedback indicating the distance between the camera and the detected object.
2) A report documenting the design and implementation of the device, including the machine learning algorithms used for object detection and distance measurement, the dataset used for training and testing, and the performance of the device in various tests.
3) A demonstration of the device's functionality, including its ability to detect objects and provide feedback to the user.
4) A user manual or instructions for operating the device, including instructions for setting it up and using it to detect objects and measure distances.
5) A presentation or poster summarising the key features and contributions of the project and highlighting its potential impact on individuals with vision impairments or other disabilities.

F. ORGANISATION OF CHAPTERS

Chapter 1: Introduction

The project overview, problem description, project objectives, project scope, deliverables, and chapter organisation are all included in this chapter.

Chapter 2: Background Study

The background information on the problem is covered in this chapter, along with details on eye care, vision impairment and blindness, a description of visual assistive technology, earlier research in the area, object detection, hardware, computer vision frameworks, and the suggested solution.

Chapter 3: Requirements Analysis

The system development model, product functionalities, use case diagram, hardware and software requirements, user requirements, and functional and non-functional needs are all covered in this chapter's project requirements analysis.

Chapter 4: Design

The project's design is covered in this chapter, together with the glasses, class diagram, flowchart diagram, dataflow diagram and circuit diagram.

Chapter 5: Implementation

The project's implementation is covered in this chapter, along with its plan, milestones, and phases.

Chapter 6: Conclusion

This chapter summarises the project's findings, including conclusions and key takeaways.

II. BACKGROUND STUDY

The most prevalent challenge that visually impaired individuals have to deal with every day is navigating through areas in their everyday lives owing to their vision being unable to function normally. This is why we believe that this is the problem that we must tackle and assist individuals with visual impairments so that they can walk around any place they desire in comfort. Technologies have grown swiftly, and computer vision has been an industry that has evolved rapidly too. By applying deep learning models to computer vision, we can only detect objects in front of us using a camera. However, more than just identifying them is needed to help visually impaired individuals. We must detect them quickly and precisely to let visually impaired individuals analyse the situation and act accordingly. So this is why picking the proper model for this project is highly critical since the object detection processing time must be fast and also be able to process in real time. To find the proper model for this project, we will have to research and compare the models often used in computer vision.

A. EYE CARE, VISION IMPAIRMENT AND BLINDNESS

Vision is essential for humans to undertake daily activities such as navigation and item or person recognition. Without it, it would be difficult for individuals to navigate safely, so we should always care for our eyes. According to the World Health Organisation, long-lived individuals will have at least one eye problem over their lifespan. The 2018 International Classification of Diseases 11 (ICD-11) says that there are two types of vision problems: those that affect vision at a distance and those that affect vision up close. There are four distinct degrees of visual impairment. Mild visual acuity is between 6/12 and 6/18, moderate visual acuity is between 6/18 and 6/60, severe visual acuity is between 6/60 and 3/60, and blindness is below 3/60.

B. OVERVIEW OF VISUAL ASSISTIVE TECHNOLOGY

There are three approaches to this in visual assistive technologies: visual enhancement, visual substitution, and visual replacement [3]. Visual substitution is like a replacement for a limited vision person's eyes. An image is taken with a camera, the information is processed, and feedback is given through sound, vibration, or a combination of the two. With visual enhancement, on the other hand, the information is given visually, like in virtual reality or augmented reality. In the last step, visual replacement analyses the information and displays it in the cortex of the user's brain; this technique is most frequently employed in the medical profession. In this project, we are investigating and constructing a visual substitution device. Thus, that will be the primary topic of this paper [3].The visual substitution has three subheadings: electronic travel aids (ETA), electronic orientation aids (EOA), and position locator devices (PLD).

The authors have given a few recommendation that we highly agree on which are [4]:

1) The use of effective and appropriate real-time object detection methods is crucial when developing ETAs, as real-time operations are essential to be considered in this scenario.
2) The availability of multiple feedback mechanisms is also one of the aspects of developing quality ETAs, as single-mode feedback might provide liability to the user, such as auditory feedback when in a crowded and loud area. Users should be able to toggle through multiple feedback modes depending on their taste and the scenario.
3) The user-friendly aspect of the device must take into deep consideration as many visual assistive technologies are burdensome to use and learn as it takes much time to get used to and learn.
4) The amount of information projected to the user must also be fine tune as the main point of navigating is to reach their destination safely. If the user is overwhelmed with too much information about what is happening around them, they will not have a good time navigating using the ETAs. During navigation, they may be required to be informed of changes in the surrounding environment, such as traffic jams, hazards, and dangerous environments. During navigation, the appropriate quantity of contextual information should be provided to the user at the appropriate time [5]. For the navigation solution to be effective, it is recommended to emphasise the communication of relevant environmental data [6].
5) More often than not, technical advances cannot convince the visually handicapped to like them. Users of a navigational aid system must feel comfortable and not be embarrassed by using it [7]. The most effective solution is to create a device that is simple to use and does not make people feel awkward when they use it in public.
6) Privacy and security are critical when developing a device that will be used every day by visually impaired individuals. When making a navigation device for the limited vision and visually impaired, it is crucial to think about how to handle private and personal

information. Users who are limited vision should be able to set up their navigation devices to decide what information is needed to run the process and what information is sent over the network. This parameter can be changed based on the user's wants and how the system is used.

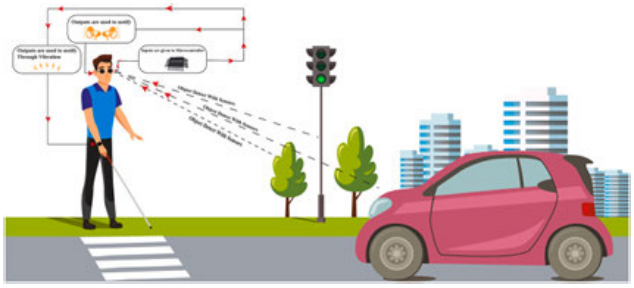

FIGURE 1. Application of smart glass and smart cane in real life. [8].

C. PREVIOUS WORK OF VISUAL ASSISTIVE TECHNOLOGY

1) HEAD WEAR

A Camera-Based Mobility Aid for VIP

Schwarze et al. [9] describe a wearable assistance system for the limited vision that uses a stereo camera system to sense the surroundings and gives the user intuitive sound feedback about obstacles and other things. It is intended to supplement more conventional forms of aid. The authors describe the fundamental head-tracking techniques, sonification, and scene interpretation. They experimentally demonstrate how these techniques enhance users' capacity to navigate new metropolitan surroundings safely.

Limitaion: Only works in an outdoor environment

Wearable Travel Aid for Environment Perception and Navigation of Visually Impaired People

FIGURE 2. Prototype of Schwarze et al.'s device. [9].

For this research, an inertial measuring unit (IMU) was coupled to a camera, a smartphone, and an earpiece for commands and feedback. A consumer Red, Green, Blue, and Depth (RGB-D) camera was also attached to a pair of eyeglasses. The system can be used both indoors and outdoors. Computer vision technologies were incorporated into this device's routing and detection capabilities since they offer much information, are lightweight compared to other sensors, such as ultrasonic and LiDAR sensors, and are less

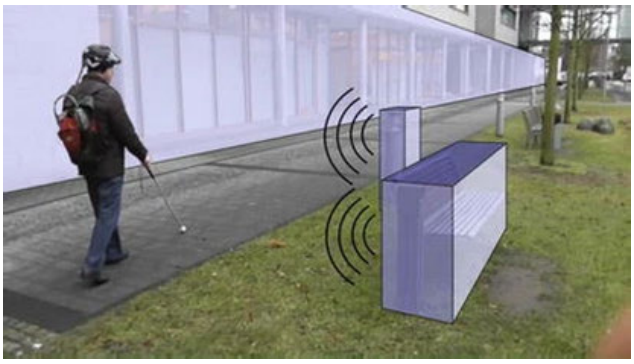

FIGURE 3. Illustration of the fundamental concept of a navigational aid for the limited vision. [9].

expensive. The smartphone handles detecting and routing, and the feedback is delivered to the user's ear via an earphone connected to the smartphone [10].

Limitation:

- Weak in detecting small-size obstacle.
- Staircase detection is not implemented.

Let Limited Vision People See: Real-Time Visual Recognition with Results Converted to 3D Audio

A system was developed by [11] that accepted video input from a handheld camera. They used the You Only Live Once (YOLO) model and streamed it to a server for real-time picture recognition processing. The location and dimensions of the bounding boxes used by the object detection method are used to determine the 3D location of the detected object. The linked wireless earphones will receive 3D audio from the Unity game engine. The sound output interval will begin if a different object is recognised before a few seconds have passed. The YOLO algorithm with an improved wireless transmitter allowed the solution to carry out accurate real-time objective detection with a live feed rate of 30 frames per second in 1080p resolution. The system's data flow pipeline is depicted in Figure 4 of the article. The video is recorded and submitted to the YOLO algorithm for object detection. The unity engine is then used to send the identified item to the earphones. The device of this paper's prototype is shown in Figure 5.

Limitation:

- Can only accurately detect and classify object within 2 to 5 meters away.
- Surrounding ambient will be block when using earbuds.
- Too much information will be sent to user when camera detect multiple objects.

2) SMART CANE

Smart Electronic Stick for Visually Impaired using Android Application and Google's Cloud Vision Bharatia et al. [12] developed the e-stick module to take the place of the crucial simple navigation stick that visually impaired people generally use. It is integrated with a voice-controlled Android application. The e-stick is small, light, and easy to hold, comparable to a regular stick, but it also offers additional functions. These features are inexpensive and

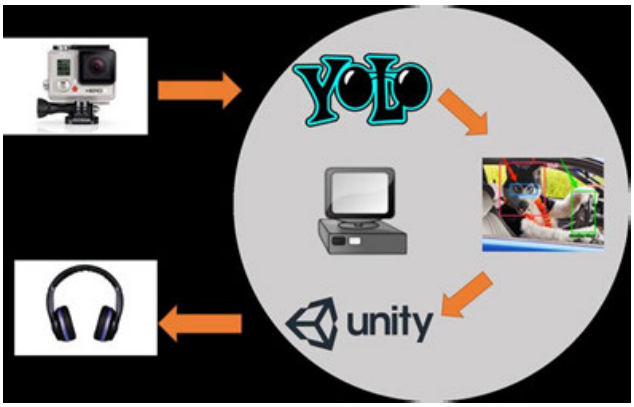

FIGURE 4. Current Data flow pipeline of the system.

FIGURE 5. Camera device and image streaming device.

attainable because of effective natural language processing (NLP) features. When travelling rearward, the e-stick uses ultrasonic sensors to detect low-lying, knee-level obstructions like potholes. Buzzers will be used to offer active feedback on the presence of impediments. The stick's circuit must be Bluetooth-connected to the user's phone for GPS navigation. In unavoidable situations, visually impaired people who live whereabouts will be sent to the closest aid centre or their family for support. Face detection will be introduced to help users identify who is attempting to speak with them. Traffic signals and roadside signs can also be translated using the cloud vision API to help the limited vision navigate. It will be simple for people to read books, documents, newspapers, and other printed items thanks to text recognition technology that can also be used for images.

A stick-tracking gadget has been created in case the user misplaces their stick. This functionality will be made possible through hardware (a smart stick) and a software module (an Android application). The user will use voice commands to instruct these modules using Natural Language Processing technology. Using a rechargeable circuit, the e-stick will be charged as required. The interaction of the technologies employed in this work is depicted in Figure 6.

Limitation:

- Coverage of obstacle detection is short as it is using sensor
- Only suitable for indoor

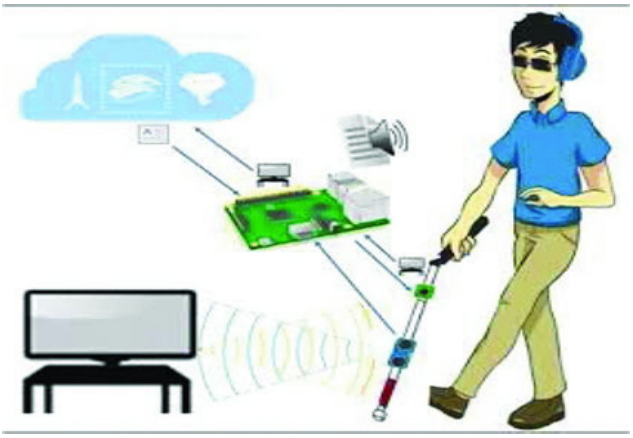

FIGURE 6. Smart e-stick using android application and cloud vision API.

WeWalk

A non-profit organisation called YGA created the smart cane named WeWalk (WeWALK Smart Cane - Smart Cane for the Visually Impaired, 2020). WeWalk resembles a conventional cane but includes a built-in touchscreen in the handle. Through the reputable programme, users can use a cane to browse, save, and find new locations. The mapping services include layers for better user-friendly navigation. Users can receive voice feedback via Bluetooth or the built-in speaker. In order to control their phone, users can also Bluetooth-pair their phone with an intelligent cane. The surface-level An above-ground obstruction is found using the integrated ultrasonic sensor. Depending on the user's choices, the detected obstacle is transmitted back to the user as either audio or vibration. The Wewalk app allows users to browse their transportation choices, including local bus stations and the schedule, and then direct themselves to the selected stop. A voice assistant is also built-in for more straightforward navigation within the application. If users misplace their phone or cane, they can play a sound on to find where it is. The WeWalk Smart Cane is depicted in Figure 7 and is currently available.

Limitation:

- The device is very expansive, with the price of 500 USD
- Rain or snow might cause malfunction on the smart cane and the speaker
- The tip of the cane is loud when navigating rough surface sidewalk

Development of an Intelligent Cane for Visually Impaired Human Subjects

An intelligent white cane that can detect obstacles within 450 metres and calculate their distance was developed by Asati et al. [13]. The sensors' 450-metre range allows them to identify objects above head height. A buzzer responds to the warning signal by beeping and instructing the user to act immediately. The intelligent technique is used to recognise and categorise objects. The web camera records the images so that they can be categorised. To convert them into text

FIGURE 7. WeWalk smart cane.

and an audio signal for text-to-speech, they will be changed. The intelligent cane described in this paper is prototyped in Figure 8.

Limitation:

- Cost of building the system is high.
- Unable to identify pot holes.
- Detection under rainy weather is not tested.

FIGURE 8. Developed intelligent cane.

3) HANDHELD

Android Application for Object Recognition Using Neural Networks for the Visually Impaired

An Android software created by Dosi et al. [14] uses the phone's camera to recognise objects in real-time and pronounce the thing to the visually impaired user as feedback. They chose a convolutional neural network-based deep learning technique for better recognition and quicker reaction times. Because MobileNets is excellent for mobile and embedded vision applications, it is used. Figure 9 displays the outcomes of the object recognition application's detection of the object.

Limitation:

- Only works offline.
- Unknown or untrained objects will be predicted using existing images in the database.
- Have to retrain model for untrained object.

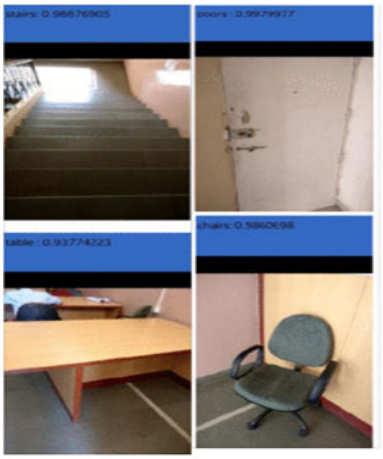

FIGURE 9. Interface of application.

Design and Implementation of an Embedded Real-Time System for Guiding Visually Impaired Individuals

To help the limited vision perceive their surroundings, Duman et al. [15] created and used a portable device that can detect objects and precisely calculate their distance. The system uses a single device connected to a Raspberry Pi board to implement YOLO, a real-time identification method based on convolutional neural networks. Visually impaired users will hear an audio projection of the estimated object distance. This discovered distance estimation has a 98.8% accuracy rate. The video is initially recorded with a portable Raspberry Pi camera. The object detection module then uses YOLO for real-time object recognition and needs to extract the bounding box size for people. The bounding box's dimensions are given to the distance estimate module to calculate how far away the detected person is. The labels of things that are detected and the broad range of any individuals that are caught are briefly kept. Text-based saved results are converted into audio warnings via an audio-generating module that visually impaired users can hear using headphones. In order to lessen noise and uncertainty, alerts are played at regular intervals. Figure 10 depicts the system block diagram from the study.

Limitation:

- Only detect humans.
- No design for any wearable option.

Real-time object detection and face recognition system to assist the visually impaired

A real-time object and face identification android app was created by Anish Aralikatti et al. [16] utilising OpenCV, the You Only Live Once (YOLO) algorithm, and FaceNet. The user will hear the detection of items and people in an audio format. Computer vision tasks in real-time are

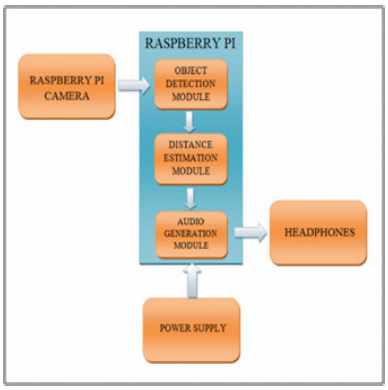

FIGURE 10. Block diagram of the proposed system.

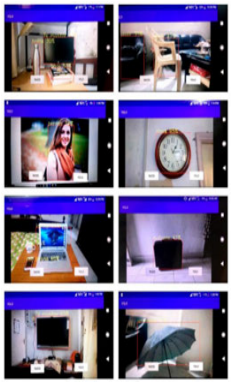

FIGURE 11. Sample output for object detection taken from Android App on mobile phone.

FIGURE 12. Sample output of face recognition on mobile.

performed using OpenCV. They chose Tiny YOLO because it is a lightweight YOLO framework ideal for embedded and mobile devices and integrated into an android phone. FaceNet is used for face recognition systems because it can extract detailed facial traits. The sample outputs from the Android application, when used on a mobile device, are displayed in Figure 11. The android app's facial recognition capability is depicted in Figure 12.

Limitation: Less accuracy than YOLO as Tiny YOLO model is smaller The complete list of all previous work is included in Table 1. VAT comes in three different forms: handheld, headwear, and E-stick. Five projects use computer vision techniques to identify objects, two use ultrasonic sensors, one combines an ultrasonic sensor with the Google Cloud Vision API, and one uses sonification techniques. All of the preceding ones are capable of detecting a single object, but only five of them are also capable of detecting multiple objects. Only one of the feedback systems from the prior work supports vibratory feedback, while all support auditory feedback.

D. OBJECT DETECTION

1) CONVOLUTIONAL NEURAL NETWORK BACKBONES (FEATURE EXTRACTOR)

This research aims to illustrate the importance of deep learning and convolutional neural networks in a particular object detection challenge [18]. VGG16, AlexNet, GoogLeNet, ResNet, Inception-ResNet-V2, and DarkNet-19, which are the building blocks of object detection models, have been chosen for analysis. For this study, the following object detectors have been chosen: HyperNET, PFPNet-R512, YOLOv1, BlitzNet512, and CoupleNet. The ImageNet database was used to evaluate the Top-1 and Top-5 accuracy rates of several CNNs. ImageNet is one of the largest databases available today, containing over 14 million images from various categories. Object detection is evaluated using the Pascal VOC 2007 and 2012 datasets and the Common Object in Context (COCO) datasets.

AlexNet is a convolutional neural network (CNN) developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012 [19]. It was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, a competition that evaluates the performance of computer vision models on a large dataset of images. It comprises eight layers, five convolutional layers, two fully connected layers, and one final output layer. AlexNet uses a combination of convolutional and max pooling. Normalization layers extract features from images and then classify the images using the fully connected layers. It uses the ReLU activation function in the hidden layers and the softmax function in the output layer. AlexNet was a significant breakthrough in deep learning and computer vision. It showed that deep neural networks could achieve state-of-the-art results on large-scale image recognition tasks, which inspired much subsequent research on CNNs and contributed to developing more advanced architectures such as VGGNet, GoogLeNet, and ResNet.

The Visual Geometry Group at the University of Oxford created the convolutional neural network (CNN) architecture known as VGG16. In a 2014 publication, Andrew Zisserman and Karen Simonyan introduced it [20]. It achieved state-of-the-art performance in the ILSVRC-2014 competition. The deep design of the architecture, which consists of 16 layers, including 13 convolutional layers and three fully connected layers, is well-known. VGG16 employs a deep stack of layers and tiny convolutional filters to learn detailed feature representations. Additionally, it employs max pooling layers to minimise the feature maps' spatial extent and guard against

TABLE 1. Summary of related works.

No	Related Works	Real-time Object Detection	Detection Algorithm	Detection Type		Detection Range (M)	Feedback		Weight	Size	Type of Device
				SO	MO		Audio	Vibration			
1	[9]	Y	Sonification	Y	Y	10-20m	Y	N	Li	La	Hw
2	[11]	Y	YOLO	Y	Y	2-5m	Y	N	LI	La	Hw
3	[14]	Y	MobileNets	Y	N	30m	Y	N	Phone Weight	Phone Size	Hh
4	[12]	Y	Ultrasonic Sonic/Google Cloud Vision API	Y	Y	not stated	Y	N	Not Stated	M	Es
5	[10]	Y	PeleeNet + SSD + Depth-Based Object Detection	Y	Y	not stated	Y	N	Not Stated	S	Hw
6	[15]	Y	YOLO	Y(Only Human)	N	2-5m	Y	N	Phone Weight	Phone Size	Hh
7	[13]	Y	HR-SO4 Ultra-sonic Sensor	Y	N	450m	Y	N	Li	M	Es
8	[17]	Y	Ultrasonic Sensor	Y	N	80-165cm	Y	N	H	M	Es
9	[16]	Y	Tiny YOLO	Y	Y	not stated	Y	N	Phone Weight	Phone Size	Hh

Note: Y:Yes, N:No, H:Heavy, A:Average, Li:Light, S:Small, M:Medium, La:Large, Hh:Handheld, Hw:Headware, Es:E-Stick

overfitting. The ImageNet dataset, which has over 1.2 million images and 1000 classifications, is used to pre-train the model. VGG16 significantly improved the performance of CNNs on image classification tasks. It has served as the foundation of numerous different models and architectures.

GoogLeNet, also known as Inception v1, is a convolutional neural network (CNN) architecture developed by Google in 2014. It was introduced in a paper by Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, and Jeff Dean [21]. It achieved state-of-the-art results in the ILSVRC-2014 competition and won first place in the classification and detection tasks. The architecture is unique and known as Inception, composed of multiple parallel convolutional layers and pooling layers organized in a modular fashion. This allows the network to learn features at multiple scales and resolutions, which improves its ability to detect objects at different scales and orientations. GoogLeNet also uses 1×1 convolutional layers, reducing the number of parameters and computational costs, and making it more efficient than other architectures. The model is pre-trained on the ImageNet dataset, which contains over 1.2 million images and 1000 classes. GoogLeNet set a new standard for the performance of CNNs on image classification tasks. It was used as a backbone for many other models and architectures.

ResNet, which stands for Residual Neural Network, is a convolutional neural network (CNN) architecture developed by Microsoft Research in 2015. It was introduced in a paper by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun [22]. It achieved state-of-the-art results in the ILSVRC-2015 and COCO-2015 competitions, winning first place in the classification task. The architecture is unique; it is composed of residual blocks that allow the network to learn the residuals (differences) between the input and output of each block, which improves its ability to detect objects at different scales and orientations. ResNet also uses shortcut connections, which allow the network to skip one or more layers, making it easy to train very deep networks and solve the problem of vanishing gradients. The model is pre-trained on the ImageNet dataset, which contains over 1.2 million images and 1000 classes; it can be used for many computer vision tasks such as object detection, image classification, and segmentation. ResNet set a new standard for the performance of CNNs on image classification tasks. It was used as a backbone for many other models and architectures.

Inception-ResNet-V2, is a convolutional neural network (CNN) architecture introduced in a paper by Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi in 2016 [23]. It is an improvement to the Inception-ResNet architecture, combining the Inception and ResNet architectures. The Inception architecture is composed of multiple parallel convolutional layers and pooling layers that are organized in a modular fashion, which allows the network to learn features at multiple scales and resolutions. The ResNet architecture is composed of residual blocks that

allow the network to learn the residuals (differences) between each block's input and output, which improves its ability to detect objects at different scales and orientations. Additionally, Inception-ResNet-V2 uses shortcut connections, which allow the network to skip one or more layers, making it easy to train very deep networks and solve the problem of vanishing gradients. The model is pre-trained on the ImageNet dataset, which contains more than 1.2 million images and 1000 classes. It can be used for many computer vision tasks, such as object detection, image classification, and segmentation. It also achieved state-of-the-art results in the ILSVRC-2016 competition and won first place in the classification task.

DarkNet-19 is a version of the open-source neural network framework, Darknet, which is built with 19 layers. It is a convolutional neural network (CNN) trained on the ImageNet dataset. It is known for its small number of parameters, fast performance, and ability to be used for real-time object detection and classification tasks. Its architecture is based on the VGG-16 CNN architecture. However, it has fewer filters in each convolutional layer, which leads to a smaller number of parameters, reducing computational complexity and memory requirements. The final layer of the architecture is a fully connected layer used for classification. It can be applied in various computer vision tasks such as object detection and image classification. It is also a base model for other architectures, such as YOLO and Tiny-YOLO, which are used for real-time object detection tasks.

Network	Params(M)	MACs(G)	Top-1 accuracy	Top-5 accuracy
AlexNet	61.1	**0.72**	56.55	79.09
VGG-16	138.36	15.5	71.59	90.38
GoogLeNet	6.62	1.52	69.78	89.53
ResNet-50	25.56	4.12	76.15	92.87
ResNet-101	44.55	7.85	77.37	93.56
Inception-ResNet-V2	55.84	13.22	**80.3**	**95.1**
DarkNet-19[a]	–	–	72.9	91.2

FIGURE 13. Network's performance on the ImageNet 1-crop accuracy rates [18].

Detector	Backbone	Data	mAP@.5	mAP@ [.5,.95]
PFPNet-R512 [32]	VGG-16	trainval35k	**57.6**	35.2
BlitzNet512 [18]	ResNet-50	trainval35k	50.9	32.5
CoupleNet [21]	**ResNet-101**	trainval35k	57.5	**36.4**
Faster R-CNN G-RMI [23]	Inception-ResNet-V2	trainval	55.5	34.7
YOLOv2 [27]	DarkNet-19	trainval35k	44.0	21.6

FIGURE 14. MS COCO test-dev 2015 detection results(%) [18].

Detector	Backbone	Data	mAP
PFPNet-R512 [32]	VGG-16	07++12	80.3
YOLOv1 [16]	GoogLeNet	07++12	57.9
CoupleNet [21]	**ResNet-101**	07++12	**80.4**

FIGURE 15. Comparative results on Pascal VOC 2012 test set (%) [18].

2) LIGHTWEIGHT NETWORK

Researchers at DeepScale, the University of California, Berkeley, and Stanford University created SqueezeNet [24]. SqueezeNet reduces the network's parameters and computational expenses while still attaining acceptable accuracy on standard image classification benchmarks by combining 1×1 and 3×3 convolutions, depth-wise separable convolutions, and aggressive downsampling. Due to this, SqueezeNet is exceptionally well suited for applications that require a small amount of processing power, like those running on embedded systems or mobile devices.

MobileNet was developed by researchers at Google and is specifically tailored for mobile and embedded devices, where computational resources are limited [25]. MobileNet uses a combination of depth-wise separable convolutions, which reduce the number of parameters and the computational cost of the network, and a specialized network architecture that is efficient on mobile devices. This makes MobileNet well-suited for real-time object detection and classification applications on mobile devices.

MobileNetv2 is an updated version of the MobileNet architecture. Like its predecessor, MobileNetV2 is designed to be lightweight and fast, making it well-suited for mobile and embedded devices applications [26]. MobileNetV2 improves upon the original MobileNet architecture by using inverted residual blocks with linear bottlenecks, making the network more efficient and allowing it to perform better on a wide range of visual recognition tasks. MobileNetV2 also introduces the concepts of the "width multiplier" and the "resolution multiplier," which allow the network to be easily scaled up or down depending on the application's specific requirements.

MobileNetv3, like its predecessors, is designed to be lightweight and fast, making it well-suited for mobile and embedded devices applications [27]. MobileNetV3 introduces several new network architecture components, including a new way of performing downsampling, which allows the network to better balance accuracy and computational cost. MobileNetV3 also introduces the concept of "network architecture search," which allows the architecture of the network to be automatically tuned for specific tasks or applications. This allows MobileNetV3 to be even more efficient and effective than previous versions of MobileNet.

MnasNet was developed by researchers at Google and is part of a more prominent family of neural network architectures known as "MobileNets" [28]. MnasNet uses a combination of depth-wise separable convolutions and architectural improvements to reduce the number of parameters and computational cost of the network while still achieving good accuracy on standard image classification benchmarks. This makes MnasNet well-suited for applications with limited computational resources, such as mobile devices or embedded systems. MnasNet was designed using a technique known as "network architecture search," which automatically tunes the architecture of the network for specific tasks

TABLE 2. Main added features in the convolutional neural networks [18].

Network	Main Feature
AlexNet	Rectified Linear Units (ReLU) can be used to introduce nonlinearity.
VGG-16	Deeper than AlexNet by about twice as much.
GoogLeNet	Instead of stacking convolutional layers, use dense modules.
ResNets	Batch normalisation and skipping connections are employed
Inception-ResNet-V2	• Swapping out inception modules for residual inception blocks. • Following the stem module, add the inception module (Inception-A). • Making use of more inception modules.
DarkNet-19	A single model that combines Darknet extraction, Network In Network, Inception, and Batch Normalization.

or applications. This allows MnasNet to be more efficient and effective than other MobileNet architectures.

PeeleNet is a lightweight convolutional neural network (CNN) designed for real-time object detection and classification tasks. It was developed by researchers at the University of Toronto and presented in a paper titled ''Pelee: A Real-Time Object Detection System on Mobile Devices'' [29]. PeleeNet is designed to be faster and more efficient than many other CNNs, making it well-suited for mobile devices and other resource-constrained platforms. It achieves this efficiency through several design techniques, including depthwise separable convolutions and smaller filters in the first few layers of the network. PeleeNet has performed well on various object detection and classification benchmarks, including the COCO and PASCAL VOC datasets.

ShuffleNet was developed by researchers at Megvii Technology and used a technique called ''channel shuffle'' to improve the efficiency of the network [30]. ShuffleNet uses a combination of 1×1 and 3×3 convolutions and pointwise group convolutions to reduce the number of parameters and computational cost of the network. The channel shuffle allows the network to reuse features more effectively, which helps improve its accuracy on standard image classification benchmarks. This makes ShuffleNet well-suited for applications with limited computational resources, such as on mobile devices or embedded systems.

ShuffleNetv2 was developed by researchers at Megvii Technology and used a technique called ''channel shuffle'' to improve the efficiency of the network. ShuffleNet uses a combination of 1×1 and 3×3 convolutions and pointwise group convolutions to reduce the number of parameters and computational cost of the network. The channel shuffle allows the network to reuse features more effectively, which helps improve its accuracy on standard image classification benchmarks. This makes ShuffleNet well-suited for applications with limited computational resources, such as on mobile devices or embedded systems.

OFA, short for ''once-for-all,'' is an object detection architecture that aims to minimize the number of parameters and computation costs while maintaining high performance. It is achieved by breaking down the complex object detection task into simpler subtasks and training a single network to handle them. The architecture comprises a shared backbone network and several task-specific heads. The backbone network learns a feature representation that is common among all subtasks. In contrast, task-specific heads learn task-specific features. The OFA model can be trained on a large dataset and applied to multiple tasks like object detection, instance segmentation, and keypoint detection with a single model. This allows the network to learn more general and robust feature representation and improve performance on multiple tasks. OFA is a relatively new architecture. It has shown promising results in several object detection benchmarks, but further research is needed to evaluate its potential fully.

MobileViT-S is an object detection model that is optimized for mobile devices. It is a variant of the Vision Transformer (ViT) architecture, a transformer-based model adapted for object detection. MobileViT-S is lightweight and efficient, making it suitable for deployment on mobile devices. It is built on top of the MobileNetV3 architecture. It uses the same convolutional block design, which reduces the number of parameters and computation cost. Additionally, it uses a lightweight transformer module that is applied to the feature maps of the backbone network to learn contextual information. MobileViT-S has demonstrated state-of-the-art performance on several object detection benchmarks while maintaining low computational costs, making it an attractive option for mobile device deployment. It is a recent model proposed by the Google AI team in 2021. It has shown promising results as a good option for object detection on mobile devices.

As using embedded or edge devices for object detection have become popular these days, as evidenced by counting how many people come into the mall using small devices, lightweight convolutional networks have been researched by lots of researchers in recent years. Networks such as SqueezeNet, MobileNet, MNasNet, PeeleNet, and much more have been proposed by multiple researchers and have been proven to have promising results in image classification. Figure 16 shows the comparison of lightweight networks based on the ImageNet Top-1 classification extracted from Zaidi's paper [31]. They are compared with their accuracy, latency, number of parameters, and complexity in MFLOPs. OFA is currently ranked first in this comparison. However,

they have a drawback: the computed cost is expensive due to the sampled model training, which uses neural architecture search. MobileNetV3 is the most promising model for implementation into an embedded system like the Raspberry Pi as the accuracy of its network is good, and its MFLOPs are the lowest compared to other networks.

Table 5. Comparison of lightweight models.

Model	Year	Top-1 Acc%	Latency (ms)	Param. (mil.)	FLOPs (mil.)
SqueezeNet	2016	60.5	-	3.2	833
MobileNet	2017	70.6	113	4.2	569
ShuffleNet	2017	73.3	108	5.4	524
MobileNetv2	2018	74.7	143	6.9	300
PeleeNet	2018	72.6	-	2.8	508
ShuffleNetv2	2018	75.4	178	7.4	597
MnasNet	2018	76.7	103	5.2	403
MobileNetv3	2019	75.2	58	5.4	219
OFA	2020	80.0	58	7.7	595
MobileViT-S	2021	78.4	-	5.6	-

FIGURE 16. Comparison of lightweight networks [31].

3) LIGHTWEIGHT OBJECT DETECTOR

SDDLite with MobileNetv2/v3

Lightweight object detectors are still a new topic within computer vision research. One of the state-of-the-art lightweight object detectors is SSDLite, introduced in the MobileNetV2 paper by Sandler et al. [26]. SSDLite has also been implemented into TensorFlow Lite, which is easier to implement in any project that uses TensorFlow Lite as the object detection framework.

Figure 17 shows that using MobileNetV2 as the backbone, SSDLite can achieve 0.5 more mAP than YOLOv2 with just 4.3 million parameters and 0.8B MAdd. This shows that more prominent and computation-hungry networks will have a higher mAP and that SSDLite with the MobileNetv2 backbone can perform as well or sometimes even better than a more extensive network. Figure 18 shows that using the newer version of MobileNet, MobileNetv3, as the backbone, SSDLite will have the same mAP as MobileNetv2. However, it will only use 3.22 M of parameters and 0.51 B of MAdds, which is significantly smaller and more suitable to implement into an embedded or edge device.

Network	mAP	Params	MAdd	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	**4.3M**	**0.8B**	200ms

FIGURE 17. Comparison of MobileNetV2 + SSDLite and other real-time detectors on COCO dataset object detection task [26].

Backbone	mAP	Latency (ms)	Params (M)	MAdds (B)
V1	22.2	228	5.1	1.3
V2	22.1	162	4.3	0.80
MnasNet	23.0	174	4.88	0.84
V3	22.0	137	4.97	0.62
V3†	22.0	119	3.22	0.51
V2 0.35	13.7	66	0.93	0.16
V2 0.5	16.6	79	1.54	0.27
MnasNet 0.35	15.6	68	1.02	0.18
MnasNet 0.5	18.5	85	1.68	0.29
V3-Small	16.0	52	2.49	0.21
V3-Small†	16.1	43	1.77	0.16

FIGURE 18. Object detection results of SSDLite with different backbones on COCO test set [27].

ThunderNet with SNet

ThunderNet is a lightweight real-time object detection system proposed by Qin et al. [32]. ThunderNet is a two-stage detector that uses a modified ShuffleNetv2 called SNet. There are three backbones for SNet: SNet49 for faster inference, SNet535 for better accuracy, and SNet146 for a better balance between speed and accuracy. Comparing ThunderNet with the state-of-the-art detectors out there right now, it achieves good performance despite using significantly less computation power, either in MS COCO or PASCAL VOC datasets.

Figure 19 shows that ThunderNet outperforms most lightweight one-stage detectors in the VOC 2007 tests with less computation power. ThunderNet with SNet49 outperforms MobileNet-SSD with just 21% of the FLOPs. ThunderNet using Snet146 surpasses Tiny-DSOD by 2.9 mAP with 43% of the FLOPs. Using the SNet535 model, ThunderNet can achieve 6.5 more mAP than Tiny-DSOD with similar computing resources. Lastly, ThunderNet can obtain superior performance to the state-of-the-art large object detectors such as YOLOv2, SSD300, SSD321, and R-FCN. It has the same performance as DSSD321 but with significantly lower computational costs.

Model	Backbone	Input	MFLOPs	mAP
YOLOv2 [25]	Darknet-19	416 × 416	17400	76.8
SSD300* [19]	VGG-16	300 × 300	31750	77.5
SSD321 [6]	ResNet-101	321 × 321	15400	77.1
DSSD321 [6]	ResNet-101 + FPN	321 × 321	21200	78.6
R-FCN [4]	ResNet-50	600 × 1000	58900	77.4
Tiny-YOLO [25]	Tiny Darknet	416 × 416	3490	57.1
D-YOLO [21]	Tiny Darknet	416 × 416	2090	67.6
MobileNet-SSD [31]	MobileNet	300 × 300	1150	68.0
Pelee [31]	PeleeNet	304 × 304	1210	70.9
Tiny-DSOD [13]	DDB-Net + D-FPN	300 × 300	1060	72.1
ThunderNet (*ours*)	SNet49	320 × 320	**250**	70.1
ThunderNet (*ours*)	SNet146	320 × 320	**461**	**75.1**
ThunderNet (*ours*)	SNet535	320 × 320	1287	**78.6**

FIGURE 19. Comparison of ThunderNet with other state-of-the-art detectors on the VOC 2007 Test [32].

Figure 20 shows that ThunderNet performs admirably on the MS COCO test device. ThunderNet with SNet49 can get the same accuracy as MobileNet-SSD but with 22% of the FLOPs. Using SNet146, ThunderNet can outperform MobileNet-SSD, MobileNet-SSDLite, and Pelee with less than 40% of the computing resources. The comparison shows that ThunderNet is on par with or even achieves better accuracy than the state-of-the-art detector but uses fewer computing resources, which are essential when implemented into mobile devices. As of the current development of

this network, it is only implemented by the author into openmmlab's computer vision object detection framework, called mmdetection.

Model	Backbone	Input	MFLOPs	AP	AP_{50}	AP_{75}
YOLOv2 [25]	Darknet-19	416 × 416	17500	21.6	44.0	19.2
SSD300* [19]	VGG-16	300 × 300	35200	25.1	43.1	25.8
SSD321 [6]	ResNet-101	321 × 321	16700	28.0	45.4	29.3
DSSD321 [6]	ResNet-101 + FPN	321 × 321	22300	28.0	46.1	29.2
Light-Head R-CNN [20]	ShuffleNetV2*	800 × 1200	5650	23.7	-	-
MobileNet-SSD [11]	MobileNet	300 × 300	1200	19.3	-	-
MobileNet-SSDLite [28]	MobileNet	320 × 320	1300	22.2	-	-
MobileNetV2-SSDLite [28]	MobileNetV2	320 × 320	800	22.1	-	-
Pelee [31]	PeleeNet	304 × 304	1290	22.4	38.3	22.9
Tiny-DSOD [13]	DDB-Net + D-FPN	300 × 300	1120	23.2	40.4	22.8
ThunderNet (*ours*)	SNet49	320 × 320	**262**	19.2	33.7	19.7
ThunderNet (*ours*)	SNet146	320 × 320	**473**	**23.7**	40.3	**24.6**
ThunderNet (*ours*)	SNet535	320 × 320	1300	**28.1**	**46.2**	**29.6**

FIGURE 20. Comparison of ThunderNet with other state-of-the-art detectors on the VOC 2007 Test [32].

E. HARDWARE

In this section, we will discuss whether to choose Arduino or Raspberry Pi to run our project, as these two are famous for using embedded and edge devices for computer vision projects.

Arduino is an open-source microcontroller usually used for building small IoT devices. It does not come with an operating system, so what is coded in their IDE is what they will do when they get turned on. It can read data from sensors, compute data, and send them to another device for further computation or output them on attached LEDs or LCD screens. Raspberry Pi is a mini-computer with a Linux operating system loaded on an SD Card. There is an audio out port, an HDMI port, an RCA video output, and an Ethernet port on the board. Raspberry Pi can be used by plugging in a monitor, keyboard, and mouse and getting power from a battery or power outlet.

The comparison between the Raspberry Pi and Arduino is presented in Table 3. While Arduino outperforms Raspberry Pi in terms of power input, general-purpose input output, and price, Raspberry Pi outperforms Arduino in terms of performance, add-ons, programming language support, and functionality.

TABLE 3. Comparison between raspberry pi and arduino.

Feature	Raspberry Pi	Arduiono
Performance	✓	
Power Input		✓
Add-ons	✓	
General Purpose Input Output		✓
Programming Language Support	✓	
Functionality	✓	
COst		✓

F. COMPUTER VISION FRAMEWORKS

OpenCV, PyTorch, and TensorFlow are the commonly used computer vision frameworks for real-time object detection in the current computer vision industry. OpenCV is generally the most popular and widely-used option for this type of project on the Raspberry Pi, as it is a powerful, open-source computer vision library well-suited to real-time object detection tasks. PyTorch and TensorFlow can also be used for real-time object detection on the Raspberry Pi. However, they may require more computational power and memory than OpenCV and may be more difficult for beginners. TensorFlow Lite has a lightweight framework called TensorFlow Lite, a lightweight version of TensorFlow, a popular deep learning framework developed by Google. It is designed to deploy mobile and embedded devices with limited computational resources. TensorFlow Lite uses a novel approach called "flatbuffers" to reduce the models' size and make them more efficient. It also includes several other optimizations to improve performance on devices with limited resources. TensorFlow Lite supports various Android, iOS, and Raspberry Pi platforms. It can be used to deploy machine learning models for various tasks, including object detection, language translation, and image classification. Table 4 shows the key characteristics and descriptions of PyTorch, TensorFlow, TensorFlow Lite and OpenCV.

G. PROPOSED SOLUTION

We propose building and designing a lightweight and straightforward glasses so VIPs can use the devices daily. The glassess will house the camera, recording and streaming live footage of what the VIP sees in front of them. A Raspberry Pi will be housed in the glasses to receive the streamed footage, input it into the object and distance detection software, and finally output it in the form of audio through a connected earphone. The feedback of the detection object can be switched between audio, vibration, or both using a voice command or a button on the side of the glasses. The object detection model we will be using pretrained model that is compatible with the TensorFlow lite support vision library such as efficientdet0 and SSD MobileNetv1, as it is lightweight and achieves good performance when compared with the lightweight and large state-of-the-art detectors in the MS COCO and PASCAL VOC datasets.

III. REQUIREMENT ANALYSIS

A. SYSTEM DEVELOPMENT MODEL

A system development model (SDM) is a structured approach that guides the development of a new system or the modification of an existing one. It outlines the steps and processes required to create a functional system that meets the desired objectives. Several SDMs are available, including waterfall, agile, prototype, spiral, and lean development. The choice of an appropriate SDM for a particular project depends on the project's specific goals, constraints, and the development team's expertise. A system development model based on prototype development is chosen for this project and will involve the following steps:

1) Identify the requirements and objectives of the system through user interviews, gathering feedback, and creating user stories or use cases.
2) Design the initial prototype of the system, including its architecture, user interface, and functionalities.

TABLE 4. Key features of pytorch, tensorflow, tensorflow lite and openCv.

Framework	Description	Key Feature
PyTorch	Facebook created the well-known deep learning framework PyTorch, frequently used for developing and deploying machine learning models.	• Dynamic computation graph • Support for multiple programming languages such as Python, C++ and Java. • Easy to use and intuitive API
TensorFlow	Google's TensorFlow is a well-known deep learning framework frequently used for various machine learning applications, such as object detection and picture categorization.	• Support for a wide range of platforms such as mobile, web, and cloud. • Powerful visualisation tools for debugging and optimisation
Tensorflow Lite	A variant of TensorFlow called TensorFlow Lite is made to be used on smaller, less powerful mobile and embedded devices.	• Optimised for performance on mobile and embedded devices • Smaller model size and faster inference using "flatbuffers" technology • Easy integration with on-device APIs for came and other sensors
OpenCV	OpenCV is a well-known, free, and open-source computer vision toolkit with bindings for many different programming languages, including Python. It was created in C++.	• Support for a wide range of computer vision tasks, including object detection and image processing • Optimised for performance and real-time applications • Extensive documentation and community support

3) Implement and test the prototype to ensure it meets the identified requirements and objectives.
4) Conduct extensive testing and validation of the prototype to gather feedback and identify any issues or problems.
5) Refine the prototype based on the feedback and findings from the testing and validation phase.
6) Build out the final version of the system based on the refined prototype, including the implementation of additional features and integration with other systems.

To make sure the system satisfies user demands and performs as intended, this strategy entails producing a working model of the system to test and improve the design before building the final version.

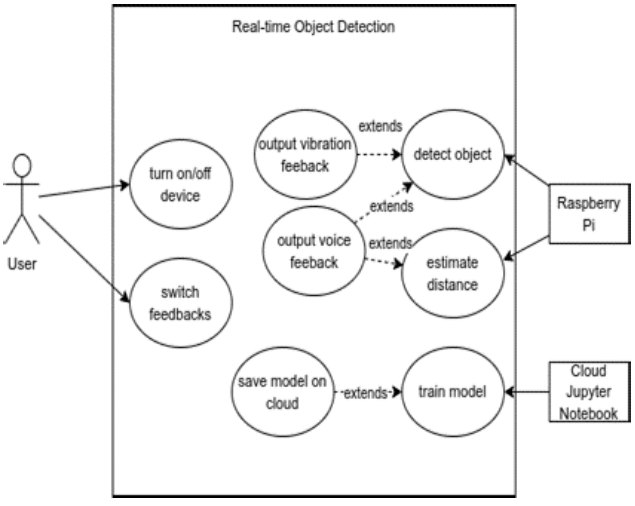

FIGURE 21. Use case diagram of proposed solution.

B. PRODUCT FUNCTIONS

C. USE CASE DIAGRAM

Activity diagram will be used to illustrate the flow of events in each section of the use case diagram.

1) TURN ON/OFF DEVICE

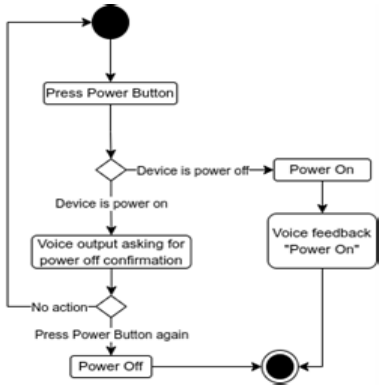

FIGURE 22. Activity diagram of Turn on/off device.

2) DETECT OBJECT

3) ESTIMATE DISTANCE

D. HARDWARE REQUIREMENTS

For our project, we have determined that the Raspberry Pi is the most suitable hardware platform due to its support for popular computer vision frameworks such as OpenCV, PyTorch, and TensorFlow, as well as its ability to run Python code. Additionally, the Raspberry Pi's audio output connector and superior computational power compared to the Arduino make it well-suited for handling image processing tasks involving audio output of identified objects and their

TABLE 5. Product functionality.

Function No	Function	Descriptoin	Actor
F001	Turn on/off device	The user can turn on or off the device	User
F002	Switch feedback of detected object	The user can cycle between audio, vibration, or both as feedback for detected object	User
F003	Detect object	The device can detect object in real-time	Device
F004	Estimate distance of detected object and camera	The device can calculate the estimated distance of detected object and camera	Device
F005	Provide Audio Feedback	Provide Audio Feedback The device can provide detected object name and its estimated distance via audio feedback	Device
F006	Provide Vibration Feedback	The device can provide detected object via vibration feedback	Device
F007	Provide Audio and Vibration Feedback	The device can provide detected object name and its estimated distance via audio and vibration feedback	Device

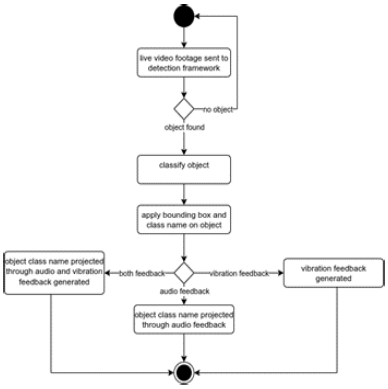

FIGURE 23. Activity diagram of detect object.

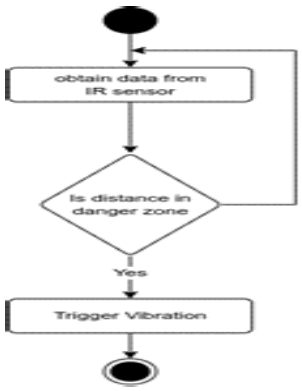

FIGURE 24. Activity diagram of estimate distance.

distances. The Raspberry Pi also has a built-in camera port, whereas the Arduino requires the purchase of a separate camera module board to connect a camera. The Raspberry Pi's ability to multitask and its ample RAM make it capable of simultaneously collecting and broadcasting video for image processing and object detection. For these reasons, we recommend using a Raspberry Pi 4 Model B+ or higher for this project. A camera with at least 5 megapixels and the ability to stream at 720p is necessary for live footage and real-time object detection. Additionally, a vibration module and a wired earphone are required for providing feedback on the detected objects and their distances from the camera. A rechargeable battery pack is necessary to power the Raspberry Pi while on the go and make the device portable, with the added convenience of being able to charge the battery to avoid the need for battery replacement. Overall, these components will enable the device to be used in any location desired by the user.

E. OPERATING SYSTEM

For our project, we will be using the Raspberry Pi OS as our operating system. Previously known as Raspbian, Raspberry Pi OS is the official operating system for the Raspberry Pi and is a free, lightweight Linux-based operating system designed for efficiency. It is commonly used for a variety of applications, such as home media centers, retro gaming consoles, home automation systems, and Internet of Things (IoT) projects.

Raspberry Pi OS is available in two versions: a desktop version with a graphical user interface (GUI) and a "lite" version without a GUI. The desktop version includes a variety of software applications, including a web browser, word processor, and educational tools, while the lite version is intended for more advanced users who want to create their own custom operating system.

The Raspberry Pi Foundation, a charitable organization that supports the development of the Raspberry Pi and its ecosystem, provides support for Raspberry Pi OS. The operating system is regularly updated with new features and bug fixes and can be downloaded from the Raspberry Pi website.

F. SOFTWARE REQUIREMENTS

1) PROGRAMMING LANGUAGE

Python will be the main programming language we will be using for developing the object detection system.

2) COMPUTER VISION FRAMEWORK

TensorFlow Lite will be used for our project, as its lightweight nature is suitable to implement our object detection system in the Raspberry Pi.

3) CODE EDITOR

Neo-vim, an improved version of vim, or Visual Studio Code will be used to write the source code of the object detection system.

4) ISO FLASHING UTILITY

Raspberry Pi Imager is used to flash the Raspberry Pi OS into a SD Card to install the operating system that we will be using on the Raspberry Pi.

G. USER REQUIREMENTS

Glasses are capable of:

- Detecting objects in 10 or higher frames per second (FPS)
- Outputting the detected object as audio feedback
- Outputting the detected object as vibration feedback
- Updating the object detection model if the glasses is connected to the Internet

User are capable of:

- Turning on Glasses
- Turning off Glasses
- Selecting audio feedback
- Selecting vibration feedback
- Receiving detected object feedback in real-time

H. FUNCTIONAL REQUIREMENTS

TABLE 6. Functional requirements for the proposed solution.

Req No	Function Requirements
FR001	The user can switch on or off the Raspberry Pi.
FR002	The glasses can detect objects fast and precisely.
FR003	The glasses can operate in low-light scenarios.
FR004	The glasses can operate in low-light scenarios.
FR005	The glasses can provide feedback on the detected object's distance to the user.
FR006	The glasses can track a detected object.
FR007	The glasses can estimate the distance between the detected object and the camera.

I. NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for the glasses are as follows:

- The Raspberry Pi selected must be Raspberry Pi 3 Model B or Higher.
- The ram needed in the Raspberry Pi must have at least 1 GB.
- The size of the glasses must not be bulky.
- The glasses must be lightweight.

IV. DESIGN

A. DESIGN OF THE GLASSES

The design of the glasses has been separated into three parts: the front piece, the left piece, and the right piece. Both the left and right pieces will have a hollow space inside them to house the components; the left piece will house the battery and the power board, and the right piece will house the Raspberry Pi and the camera. The speaker will be implemented at the end of the right piece. The mini vibrating disc motor will be implemented at the end of both the left and right pieces. Figure 25 show the design of the glasses front frame. Figure 26 shows the design of the glasses left piece. Figure 27 shows the design of the glasses right piece. All of these design are done in the TinkerCad Online 3D modelling platform.

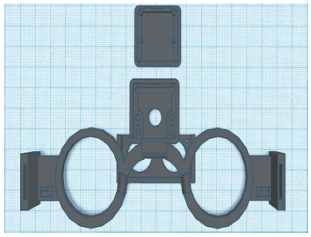

FIGURE 25. Top-down view of the glasses frame.

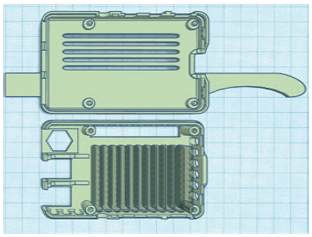

FIGURE 26. Top-down view of the glasses left piece.

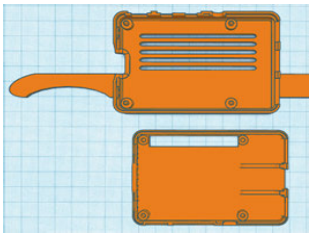

FIGURE 27. Top-down view of the glasses right piece.

B. CLASS DIAGRAM

Figure 28 displays the class diagram for a real-time object recognition and distance estimate system using a Raspberry Pi. It shows the many classes and their relationships, properties, and methods. The system uses a PiZero camera for video recording, an ultrasonic sensor for measuring distance, and the TensorFlow Lite framework for object detection. The key classes are System, PiZeroCamera, Model, Object Dectection, System and Distance Estimator. The graphic shows the relationships, aggregations, and compositions between these classes. It is a valuable tool for comprehending the system's design and structure and how the various classes cooperate to get the desired results.

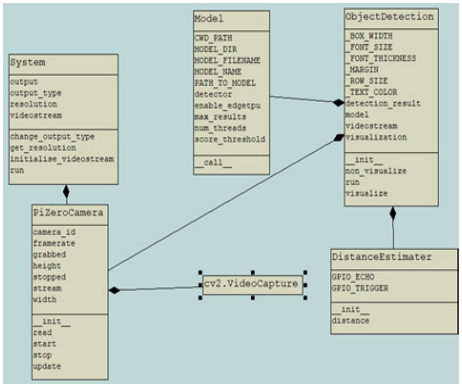

FIGURE 28. Class diagram of proposed system.

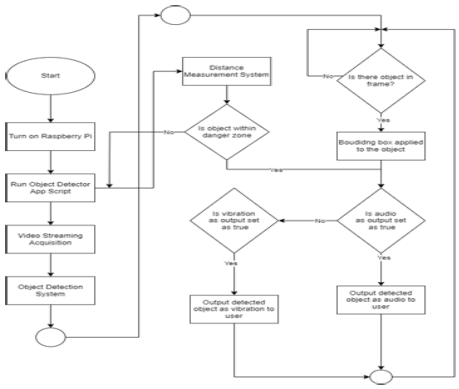

FIGURE 29. Flowchart of proposed solution.

C. FLOWCHART DIAGRAM

A flowchart is a form of graphic that connects standardised symbols and shapes with arrows to show a process or activity. The arrows depict how the steps flow or are arranged, while the symbols and shapes represent different steps or acts in the process. Flowcharts help analyse, communicate, and document complicated processes in industries, including business, software development, and manufacturing. Flowcharts are frequently employed in problem-solving, decision-making, and process development.

The flowchart diagram for the suggested system is shown in Figure 29. The Raspberry Pi turning on initiates the system's flow. The Object Detector Application will be started by the raspberry pi using a starting script. The application will obtain a live video stream from a Raspberry Pi Zero camera for object detection. The object detection algorithm will be fed the live video feed to detect objects. A bounding box will be applied if an object is present in the frame, with its labels located in the top left corner. The system will then look at how the user has configured their feedback. When auditory feedback is enabled, the user hears through a speaker the label and distance of the identified object. The technology will vibrate the side of the glasses to nudge the user if an object gets too close if the feedback is set to vibration. If both options are chosen, the user will be nudged if the object is close by, and the label of the identified object will be output as audio. Until the user switches the Raspberry Pi off, the flow will circle back to the start of the detection procedure.

D. DATAFLOW DIAGRAM

A graphical representation of data flow through an information system that models process elements is called a data flow diagram (DFD). It shows how data moves between the system's inputs, operations, outputs, and storage. The flow of data through a system, from external entities into the system, through system operations, and out of the system to external entities, can be represented by DFDs at any degree of abstraction. DFDs can also show how data moves between or within organisations. They clarify, capturing, and disseminating a system's needs to stakeholders.

1) CONTEXT DIAGRAM

A graphical representation of data flow through an information system that models process elements is called a data flow diagram (DFD). It shows how data moves between the system's inputs, operations, outputs, and storage. The flow of data through a system, from external entities into the system, through system operations, and out of the system to external entities, can be represented by DFDs at any degree of abstraction. DFDs can also show how data moves between or within organisations. They clarify, capturing, and disseminating a system's needs to stakeholders.

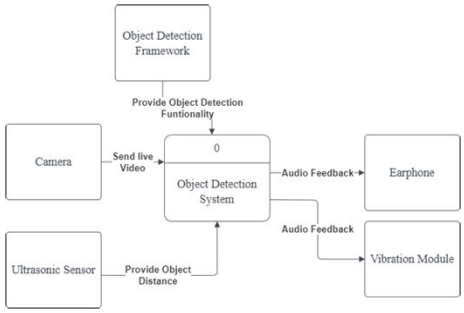

FIGURE 30. Context diagram of proposed system.

2) DFD LEVEL 0

A Level 0 Data Flow Diagram (DFD), a context level or top-level DFD, serves as a visual representation of the system and provides a detailed view of the system's central processes and data flows. This DFD breaks down the high-level process shown in the context diagram into more specific sub-processes, allowing a better understanding of the system's primary functions and how it operates. This DFD serves as the starting point for creating more detailed DFDs, which will further decompose the processes shown on the Level 0 DFD, ultimately leading to a complete DFD model of the system. Overall, the Level 0 DFD plays a crucial role in understanding and modelling a system's data flow. Figure 31 shows the Level 0 data flow diagram of the proposed system.

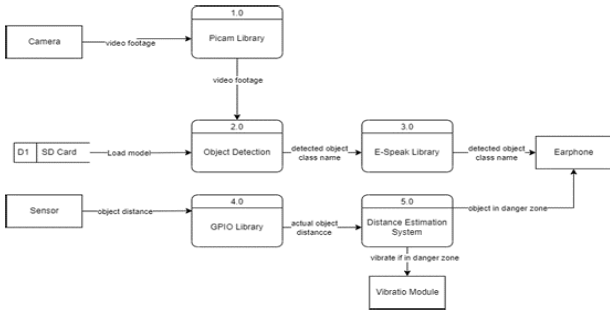

FIGURE 31. Lv0 data flow diagram of proposed system.

3) DFD LEVEL 1

The activities and interactions of the system are depicted in further depth in the Level 1 Data Flow Diagram (DFD). It divides the context diagram's high-level processes into more manageable and focused sub-processes. Three primary processes-Object Detection, Distance Estimation, and Output-are included in the Level 1 DFD for this real-time object detection and distance estimation system using a Raspberry Pi. Subprocesses inside each process deal with specific tasks. The data stores and entities the system uses are also displayed in the DFD. Additionally, it shows how data moves between data stores, processes, and inputs and outputs. This DFD gives readers a comprehensive grasp of the system's functionality and how its many components work together. Figure 32 shows the Lv1 data flow diagram of the proposed system.

FIGURE 32. Lv1 data flow diagram of proposed system.

E. CIRCUIT DIAGRAM

A circuit diagram, often known as an electrical diagram or wiring diagram, is a condensed representation of an electrical circuit. It uses symbols to represent the numerous electrical components found in circuits, including switches, transistors, capacitors, resistors, and more. The graphic shows how various elements are connected and work together to accomplish a particular purpose. Circuit diagrams are used in electrical and electronic engineering to plan, construct, and debug circuits. They are also used to clearly and concisely describe how a circuit operates in technical manuals, educational materials, and other papers. The proposed system's circuit diagram is shown in Figure 33 using the Fritzing application. General Purpose Pins link the Raspberry Pi to the HC-SR04P ultrasonic sensor and vibration motor. The Raspberry Pi's aux port is attached to the device's speakers. The Raspberry Pi's camera port is connected to the Pi Zero camera using the Pi Cam Ribbon Cable. Finally, a Type C cable is used to attach the battery to the Raspberry Pi. A usb microphone is connected to the USB port of the Raspberry Pi for speech recognition functionality.

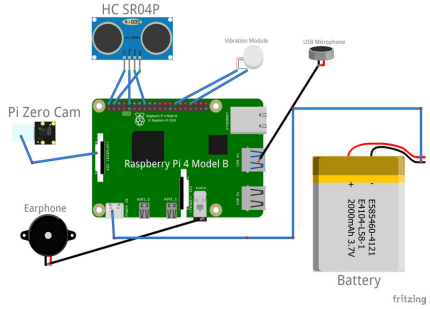

FIGURE 33. Circuit diagram of proposed system.

V. IMPLEMENTATION

A. IMPLEMENTATION PLAN

Hardware and Software Setup:

- Assemble the Raspberry Pi, camera, and ultrasonic sensor.
- Install Raspberry Pi OS on the Raspberry Pi.
- Install TensorFlow Lite and any necessary libraries on the Raspberry Pi.
- Configure the camera and ultrasonic sensor to work with the Raspberry Pi.
- Test the hardware components to ensure proper functionality.

Algorithm Development:

- Develop the object detection system using TensorFlow Lite.
- Develop the distance calculation algorithm using the ultrasonic sensor.
- Test the system separately to ensure proper functionality.

System Integration and Testing:

- Integrate the object detection system and distance calculation algorithms with the hardware components.
- Test the system in various scenarios to ensure proper functionality.
- Optimize the system to improve performance and accuracy.
- Address any bugs or issues that arise during testing.

Output Feature Implementation:

- Implement the output feature in the form of vibration and audio.
- Test the output feature to ensure proper functionality.

Deployment and Support:

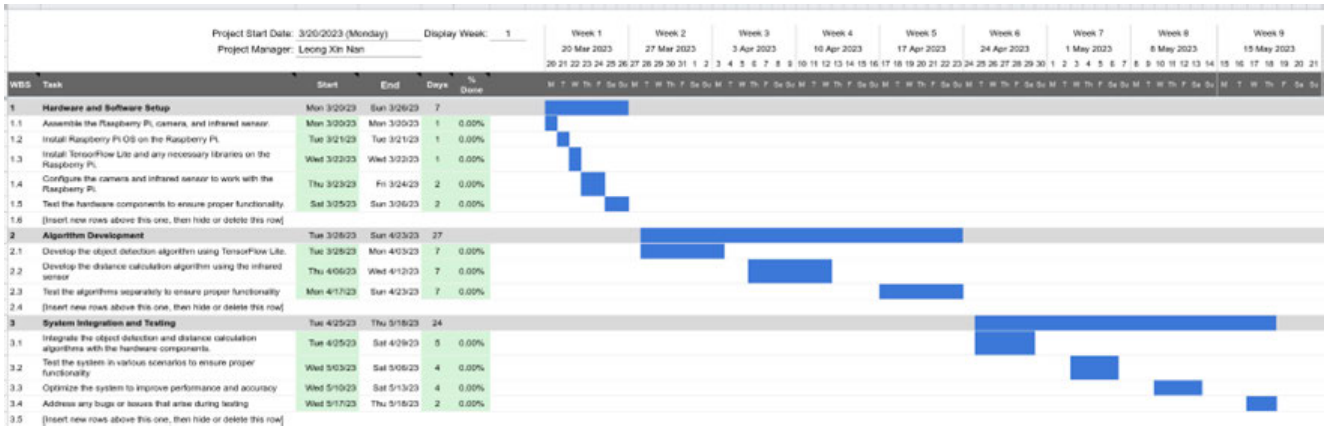

FIGURE 34. Gantt chart of FYP2 for Phase 1 to 3.

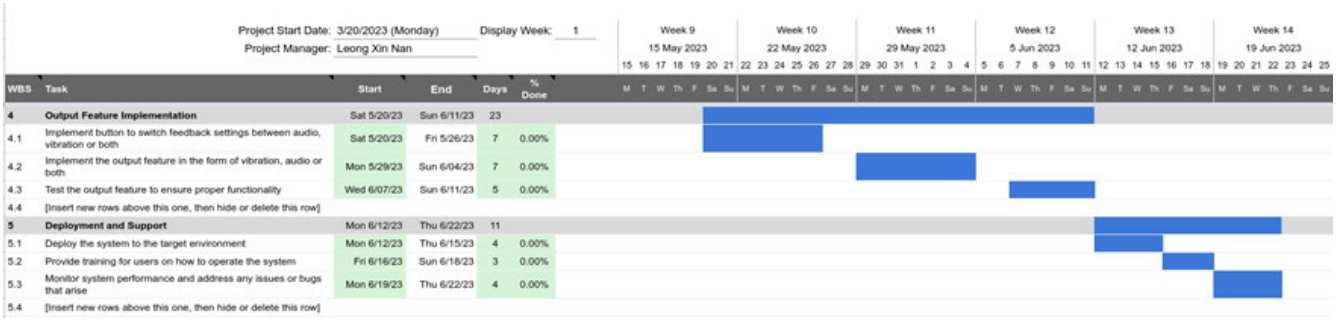

FIGURE 35. Gantt chart of FYP 2 for Phase 4 to 5.

- Deploy the system to the target environment.
- Provide training for users on how to operate the system.
- Monitor system performance and address any issues or bugs that arise.

B. MILESTONES

The milestones section outlines the key milestones that will be achieved during the project. These include:

- Hardware Assembly and Software Installation
- Object Detection Algorithm Implementation
- Distance Estimation Algorithm Implementation
- System Integration and Testing
- Output Feature Implementation
- System Deployment
- User Training

C. PHASES

The phases section outlines the main phases of the project. These include:

- Hardware and Software Setup
- Algorithm Development
- System Integration and Testing
- Output Feature Implementation
- Deployment and Support

Figures 34 and 35 show the Gantt charts for the first three phases of FYP2 and the fourth and fifth phases of FYP2, respectively. The Gantt chart shows chard as lasting 14 weeks. Phase 1: Setting up the hardware and software will take seven days. Phase 2: Developing the algorithm will take 27 days. Phase 3: Integrating and testing the system will take 24 days. Phase 4: Implementing the output feature will take 23 days. Phase 5: Deploying and supporting the system will take 11 days. Each task will be finished two days apart so that we will feel well-rested.

VI. CONCLUSION

In summary, the real-time object recognition and distance estimate system utilising a Raspberry Pi is a difficult and complex project that needs careful planning, in-depth study, and analysis. The project is divided into numerous key steps, such as setting up the hardware and software, developing the algorithms, integrating, and testing the system, implementing the output features, and deploying and supporting the system. To deliver the system effectively, certain activities and milestones must be accomplished in each phase. The finished system will use an ultrasonic sensor to identify things in real time and gauge their distance. The output is then presented as audio, vibration, or a combination. The initiative does, however, face some difficulties and constraints. The system's performance is influenced by the data's accuracy, the algorithms' stability, and the hardware and software limitations. High knowledge and experience in image processing, machine learning, and embedded systems

are necessary for the project's successful completion. The result is a system that may be applied to various tasks, including robotics, security, and surveillance, among other things.

Throughout this project, I have learned that object detection using a computer and camera is a complex task requiring significant technical knowledge. This includes understanding convolutional neural networks (CNNs), their function and how they extract and learn features of a specific class. I needed to gain more knowledge in this field at the start of this project. However, through extensive research and self-study, I am confident in completing this project.

I have also learned the importance of time management skills in project planning. As the adage states, ''if you fail to plan, then you are planning to fail.'' Additionally, many computer vision technologies available on the market can be utilized in various scenarios. For example, TensorFlow Lite can be used for edge and embedded devices like Raspberry Pi or Android devices, and mmdetection is an excellent open-source alternative for those who prefer it.

Furthermore, I have gained a significant amount of knowledge on utilizing the Linux operating system, specifically Debian-based distributions. At the start of this project, I began using Ubuntu daily to familiarize myself with its ecosystem, ensuring that I would not encounter any difficulties using Raspberry Pi for my project. I want to thank my supervisor for providing me with the Raspberry Pi to experiment with.

VII. TESTING

The testing methods that we conduct user testing with individuals who have visual impairments or may benefit from the audio feedback feature. Observe and gather feedback on the usability, effectiveness, and overall user experience of the device. This feedback can inform any necessary improvements or refinements to the system. As my device feedback is via vibration and voice feedback, it is impossible to provide them in this document, so the feedback will be printed to the console instead for testing purpose.

TABLE 7. Test cases.

Test Case	Result
A human is in front of the camera and within the range of detection system.	Pass
A human is in front of the camera, within the range of detection system, and within the danger zone.	Pass
A cup is in front of the camera, within the range of detection system, and within the danger zone.	Pass
A cup is in front of the camera, within the range of detection system, and within the danger zone.	Pass

VIII. RESULTS AND DISCUSSION

The results of our testing phase provide valuable insights into the performance and effectiveness of the smart glasses. The test cases, as outlined in Table 7, demonstrate the system's ability to detect and distinguish between different objects, such as humans and cups, and to accurately gauge the distance to these objects.

The successful detection and distance estimation in varied scenarios confirm the reliability of the pretrained convolutional neural network and the hardware integration. The use of Raspberry Pi and a camera module, combined with an ultrasonic sensor, has proven to be effective in real-time object detection and distance estimation, as evidenced by the test results. This marks a significant advancement in assistive technologies, particularly in terms of providing real-time, accurate, and practical solutions for visually impaired persons.

In comparing these results with the objectives set out in the project statement, it is clear that the system meets the criteria of portability, accuracy, and user-friendliness. However, the limitations observed in terms of data accuracy, algorithm stability, and hardware constraints highlight areas for future improvement.

Furthermore, the discussion with users who have visual impairments, as part of our user testing, provided critical feedback on the usability and practicality of the device. The insights gained from this feedback have been instrumental in identifying user-centric improvements, ensuring that the device not only functions effectively but also aligns with the needs and preferences of its intended users.

In the context of existing literature and similar technologies, our system stands out in its integration of distance estimation with object detection in a wearable format. While other technologies have addressed object detection, the addition of accurate distance measurement in a hands-free, wearable device presents a novel contribution to the field. This integration enhances the spatial awareness and autonomy of visually impaired users, thereby contributing significantly to their quality of life.

Overall, the results affirm the potential of this technology in various applications beyond assistive devices for the visually impaired, such as in robotics, security, and surveillance. The scalability and adaptability of the system suggest a wide range of possibilities for future exploration and development.

A. LIMITATIONS AND FUTURE RESEARCH DIRECTIONS

Despite the promising results, several limitations were identified during the testing phase. The accuracy of object detection and distance estimation is contingent upon the quality of the data used for training the machine learning model. Additionally, the hardware limitations of the Raspberry Pi and camera module, particularly in terms of computational power and sensor range, pose challenges for more complex and demanding scenarios.

Future research should focus on enhancing the computational capabilities of the device, possibly through the development of a custom-designed board. Improving the

machine learning model, either by training a custom model or by refining the existing pretrained model with a more diverse dataset, would also contribute to the system's accuracy and reliability. The exploration of advanced distance estimation techniques, such as stereo vision or time-of-flight sensors, could provide more precise and nuanced distance information.

Ultimately, the continual evolution of this technology will depend on iterative testing, user feedback, and integration of advancements in machine learning and hardware design. This iterative approach will ensure that the system remains at the forefront of assistive technology for visually impaired individuals.

IX. FUTURE IMPROVEMENTS

There are several important enhancements that can be made to the smart glasses for object detection and distance measurement:

1) Integration of a separate walking cane: In addition to camera-based detection, incorporating a dedicated sensor or module in the lower portion of the smart glasses can enhance the detection of obstacles below eye level. This will result in a more thorough and reliable detection system.
2) Custom-trained model for high precision: Developing and training a custom model designed specifically for generic object detection can considerably improve the system's precision and dependability. Using a diverse dataset to fine-tune the model will enable more precise and robust object recognition.
3) Enhanced distance estimation with depth estimation: Using depth estimation techniques, such as stereo vision or time-of-flight sensors, can increase the precision of distance measurement. This will provide consumers with more accurate distance information between the camera and detected objects.
4) Upgraded hardware with a custom board: Using a custom-designed board with increased computational resources can allow for more efficient processing and use of sophisticated tools and algorithms. This will improve the smart glasses' overall efficacy and responsiveness.
5) Feedback switching mechanism: By incorporating a button or switch on the smart glasses to toggle between vibration feedback and audio feedback, users will be able to select their preferable method of receiving object detection notifications. This provides flexibility and accommodates the preferences of individual users.
6) Sleek and comfortable design: Enhancing the smart spectacles' ergonomic design will increase user comfort and wearability. Weight distribution, frame material, and adjustability should be considered to ensure a secure and comfortable suit for extended use.
7) Integrated over-the-ear headphones: By integrating over-the-ear headphones into the design of smart eyewear, audio feedback and an immersive sound experience can be provided to users. This streamlines the user experience and ensures consistent audio quality by eliminating the need for distinct headphones.

By incorporating these enhancements, the smart eyewear will provide enhanced object detection capabilities, accurate distance estimation, user-friendly feedback options, enhanced hardware performance, and an overall improved user experience.

REFERENCES

[1] A. Shahdib and B. M. Bhuiyan, "An efficient obstacle detection and distance estimation system for visually impaired people using ultrasonic sensors and deep learning," *IEEE Access*, vol. 9, pp. 143675–143688, 2021.
[2] Y. Xie, N. Bore, and J. Folkesson, "Sidescan only neural bathymetry from large-scale survey," *Sensors*, vol. 22, no. 14, p. 5092, Jul. 2022, doi: 10.3390/s22145092.
[3] W. Elmannai and K. Elleithy, "Sensor-based assistive devices for visually-impaired people: Current status, challenges, and future directions," *Sensors*, vol. 17, no. 3, p. 565, Mar. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/3/565/htm
[4] B. Kuriakose, R. Shrestha, and F. E. Sandnes, "Tools and technologies for blind and visually impaired navigation support: A review," *IETE Tech. Rev.*, vol. 39, no. 1, pp. 3–18, Jan. 2022.
[5] P. Chanana, R. Paul, M. Balakrishnan, and P. Rao, "Assistive technology solutions for aiding travel of pedestrians with visual impairment," *J. Rehabil. Assistive Technol. Eng.*, vol. 4, Aug. 2017, Art. no. 2055668317725993, doi: 10.1177/2055668317725993.
[6] N. A. Giudice and G. E. Legge, "Blind navigation and the role of technology," in *The Engineering Handbook of Smart Technology for Aging, Disability, and Independence*. Hoboken, NJ, USA: Wiley, 2008, pp. 479–500, doi: 10.1002/9780470379424.ch25.
[7] A. Abdolrahmani, W. Easley, M. Williams, S. Branham, and A. Hurst, "Embracing errors," in *Proc. CHI Conf. Human Factors Comput. Syst.*, May 2017.
[8] M. Mashiata et al., "Towards assisting visually impaired individuals: A review on current status and future prospects," *Biosensors Bioelectron., X*, vol. 12, p. 100265, 2022.
[9] T. Schwarze, M. Lauer, M. Schwaab, M. Romanovas, S. Böhm, and T. Jürgensohn, "A camera-based mobility aid for visually impaired people," *KI Künstliche Intelligenz*, vol. 30, no. 1, pp. 29–36, Feb. 2016. [Online]. Available: https://link.springer.com/article/10.1007/s13218-015-0407-7
[10] J. Bai, Z. Liu, Y. Lin, Y. Li, S. Lian, and D. Liu, "Wearable travel aid for environment perception and navigation of visually impaired people," *Electronics*, vol. 8, no. 6, p. 697, Jun. 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/6/697/htm
[11] R. Jiang, Q. Lin, and S. Qu, "Let blind people see: Real-time visual recognition with results converted to 3D audio," Tech. Rep., 2016.
[12] D. Bharatia, P. Ambawane, and P. Rane, "Smart electronic stick for visually impaired using Android application and Google's cloud vision," in *Proc. Global Conf. Advancement Technol. (GCAT)*, Oct. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8978303
[13] C. Asati, N. Meena, and M. F. Orlando, "Development of an intelligent cane for visually impaired human subjects," in *Proc. 28th IEEE Int. Conf. Robot Human Interact. Commun. (RO-MAN)*, Oct. 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8956328
[14] S. Dosi, S. Sambare, S. Singh, N. Lokhande, and B. Garware, "Android application for object recognition using neural networks for the visually impaired," in *Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA)*, Aug. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8697886
[15] S. Duman, A. Elewi, and Z. Yetgin, "Design and implementation of an embedded real-time system for guiding visually impaired individuals," in *Proc. Int. Artif. Intell. Data Process. Symp. (IDAP)*, Sep. 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8875942

[16] A. Aralikatti, J. Appalla, S. Kushal, G. S. Naveen, S. Lokesh, and B. S. Jayasri, "Real-time object detection and face recognition system to assist the visually impaired," *J. Phys., Conf. Ser.*, vol. 1706, no. 1, Dec. 2020, Art. no. 012149, doi: 10.1088/1742-6596/1706/1/012149.
[17] (2020). *WeWALK Smart Cane—Smart Cane for the Visually Impaired*. [Online]. Available: https://wewalk.io/en/
[18] A. B. Amjoud and M. Amrouch, "Convolutional neural networks backbones for object detection," in *Proc. Int. Conf. Image Signal Process.*, in Lecture Notes in Computer Science, 2020, pp. 282–289.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.
[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.
[24] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.
[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018, *arXiv:1801.04381*.
[27] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," 2019, *arXiv:1905.02244*.
[28] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," 2018, *arXiv:1807.11626*.
[29] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," 2018, *arXiv:1804.06882*.
[30] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," 2017, *arXiv:1707.01083*.
[31] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103514. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200422001312#br0030
[32] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, and J. Sun, "ThunderNet: Towards real-time generic object detection," Tech. Rep., 2019.

XINNAN LEONG is currently pursuing the degree with the Faculty of Computing and Informatics, Multimedia University Cyberjaya, Malaysia. He is specializing in software engineering and has a strong aspiration to become a Software Engineer or a Web Developer after graduation. He possesses a strong passion for programming and is eager to explore various programming methods and languages to enhance his skills and knowledge.

R. KANESARAJ RAMASAMY (Senior Member, IEEE) received the Ph.D. degree from Multimedia University. His Ph.D. dissertation was titled, "Adaptive and Dynamic Web Service Composition for Cloud-Based Mobile Application." He is currently a Senior Lecturer with the Faculty of Computing and Informatics, Multimedia University Cyberjaya, Malaysia. He was also awarded a Professional Technologist (Ts) by the Malaysian Board of Technology (MBOT) and Microsoft Office 2016 Specialist, Microsoft Certified Professional and Specialist in both Web Development and Database Technology. His research interests include service oriented computing and the Internet of Things (IoT). He has also published in several conferences and journals. He has nine years of experience in the software industry in both the development and implementation phases. He is also certified by the International Software Testing Qualification Board (ISTQB), which allows him to practice as a Professional Software Tester. He was also involved research project funded by JICA & SASTREPS to develop an early warning system for floods and landslides in Malaysia. In 2018, he was awarded Telekom Malaysia Research Grant as a Project Leader for an IoT project to implement the prototype of an actual building. Besides, a research grant, he is also an IoT Trainer for Telekom Malaysia. Other than IoT training, he also provides training on Mobile Applications for all levels (School students and executives).