Division of Computing Science and Mathematics
Faculty of Natural Sciences
University of Stirling

A Guiding Hand:

Helping the Visually Impaired Locate Objects

Dan Harvey

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Artificial Intelligence

September 2021

Abstract

Problem: People thrive when given the freedom to live their lives to the fullest with complete independence should they want it. For the blind and visually impaired community this is simply not an option and they will never have these freedoms until medical science invents procedures to restore sight. Life is incredibly difficult and even the simplest tasks become challenging on many levels whilst depriving them of the basic freedoms and dignity that sighted people take for granted.

Objective: The objective of this paper is to establish whether the current crop of assistive technologies for the blind can be extended with novel research to provide enhanced ways of seeing for the blind and providing them with artificially intelligent guidance functionality. The question is asked: Is it possible to detect objects that are in front of a blind user and guide their hand towards it for the purpose of retrieving the object? If so, can this be done and packaged into a demonstrable solution that can be extended upon with future research?

Methodology: A comprehensive review of the strengths and weaknesses of the current leading assistive technologies was carried out to identify how this paper could best add value to future generations of assistive technology. This was followed by a review of the artificial intelligence applications that would underpin any potential solution. Once the opportunity was identified and matched with the relevant application of AI to solve the problem a personal assistant was created through an agile and iterative design process for the purposes of providing the most realistic proxy for real world intent.

Achievements: A viable proof of concept was put forward for evaluation that not only succeeded in satisfying the project aims and objectives of providing a guidance tool but also laid the foundation for a scalable codebase that could be expanded upon in many ways to create a next or enhanced current generation personal assistant that might one day go a long way to helping the blind and visually impaired obtain increasing degrees of freedom in the way that they live their lives.

This is not to say that the solution is perfect as several improvements were identified and expanded upon as part of user testing and the critical evaluation however due to the modular design of the solution it should be relatively straight forward to integrate these improvements once they have been developed and tested.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following:

- The MediaPipe was code obtained from the MediaPipe website as standard code for implementation with Python (https://google.github.io/mediapipe/solutions/hands.html). The code was stripped down to remove excessive looping and converted into functions.

- The pre-trained SSD model was downloaded as course material provided by an educational course delivered by www.computervision.zone. The code was converted into object oriented code and expanded upon heavily.

- The threading code for the camera feed objects was taken from a tutorial by Najam Syed on how to use various multi-threading designs to increase video frame rate. The code was used as a baseline for solving a blocking problem and expanded upon with custom code.

- The Computer Science Vibrating Glove and sample control code was provided by Dr. Kevin Swingler as part of the project brief.

Signature [illegible] **Date** *13th September 2021*

Acknowledgements

As a technology student and IT Professional I would first like to acknowledge and express thanks and admiration to the entire online community who constantly give back by uploading more research papers, courses, tutorials, and videos than can possibly be counted across a wide range of subjects and whose dedication allows knowledge and research to be utilised by all citizens of Earth as was originally intended by the founders of the early web.

Completing this project would not have been possible without the guidance and patience shown by my supervisor Dr. Kevin Swingler for keeping me focussed when I threatened to disappear down many rabbit holes throughout the project. I must also acknowledgement the input of all the lecturers at the University of Stirling who have performed excellently throughout the year providing online teaching during the Covid-19 pandemic which has challenged us all in so many ways. I would also like to extend a special thank you to course director Dr. Deepayan Bhowmik for accepting me onto the MSc.

On a personal note, I cannot leave out nor underestimate the contribution of Dr. Darren Adamson from Beacon Therapy in Northumbria who performed miracles in bringing me back to the light following a series of traumatic events. Without his help I would not have had the desire to return to education and undertake an artificial intelligence program.

Finally, I would like to thank the Welsh Government and Student Finance Wales for financing this MSc program through their higher education grant.

Table of Contents

List of Tables

List of Figures

1 Introduction

World Report on Vision (WHO)

Blindness and visual impairment are among the most freedom depriving conditions a person can face. A report compiled by vision experts for the World Health Organisation [1] estimates that from a global population of 7.8bn people, approximately 2.2bn people possess some form of visual impairment with half of these people suffering from a condition that could have been prevented or is yet to be addressed. It is important to note that visual impairment is defined as when an eye condition affects the visual system or one or more of its visual functions, it does not imply that all 2.2bn people are suffering from a life changing impairment. 81 per cent of people who are blind or who suffer with moderate to severe visual impairment are over 50 years old, that number could triple by 2050 as the global population ages.

1.1 Background

1.1.1 The Motivating Challenge

"I always say that the things you miss the most are the things that give you freedom." [2]

(Molly Burke, early-blind YouTube personality with over 2 million subscribers)

The world as we know it is built by sighted people for sighted people, little thought has been given to the experience of blind or visually impaired people until recent years leaving sufferers at a distinct disadvantage when fulfilling even basic human requirements such living freely and navigating their immediate environments. The problem also extends to quality of life with a study by Brunes et al. [3] in Norway finding that moderate to severe loneliness amongst adults with visual impairment was higher across all age groups than in the general population. This situation is further exacerbated by statistics provided by the European Blind Union [4] which show that the average unemployment rate for working age adults who are blind or visually impaired exceeds 75% resulting in less opportunities for social interaction and therefore increased chances of experiencing loneliness.

There are also social stigmas attached to these conditions with the common perception being that sufferers are not capable of living independently nor are they able to contribute to the workplace environment. This is simply not true and sighted people tend to forget that blind and visually impaired people live with these disabilities every day of their lives and are more than capable of operating as productive members of society in their own way. Mankind as a sighted species, however, is still learning how to come to terms with integrating blind and visually impaired people into our modern societies (as with other disabilities).

To overcome these stigmas and to be able to live and operate with greater freedom and independence is something that all blind and visually impaired people desire. Unfortunately, there are no medical interventions that can cure blindness and whilst the field of neuroscience is making exciting strides with retinal [5] and direct brain [6] implants, these are very long term projects that will be both costly and medically invasive. Artificial intelligence, in particular computer vision, offers hope of a scalable and affordable solution for the present day without the need for medically invasive procedures often performed at exorbitant cost.

Ultimately, there is one incontrovertible truth that even technology cannot erase - life without vision is incredibly difficult and limiting. Take a mildly frustrating task such as finding a set of misplaced keys. Add a blindfold. Add a time limit. Add strangers watching and judging. Now replicate these thoughts across every aspect of your life and you begin to realise how much we rely on the gift of sight and just how difficult the challenges are that face the blind community in every aspect of their lives.

1.2 First Generation Assistive Technology

Life doesn't have to be such a struggle for the blind and visually impaired and thankfully technology can help. These solutions are known as Assistive Technology and are already quite numerous for people with low vision. The first generation of helpful solutions are quite basic and can be considered the low hanging fruit, or simple solutions (by today's standards) however they did have an immediate impact on the lives of blind and visually impaired people.

Most solutions classed as low vision solutions focus on some form of screen reading or screen magnification use case. These can be very helpful for the users however they are often passively consumed, they do not interact with the user. Some examples are given from the Perkins School for the Blind eLearning website [7]:

Technology	**Description**
Audio Description	This is typically an additional narration track that describes visual information and can be played openly or via headphones. It is commonly used in movie theatres, streaming services, plays, museums.
Computers	All modern computers now come with accessibility features such as options to magnify the display, use high contrast colours, larger font, and print sizes. In addition to this many standard classroom materials can be digitised as made widely available in suitable formats for visually impaired students
Device Cameras (Basic Use)	The device camera will become a cornerstone of AI solutions in future sections but is also worthy of inclusion here with the basic concept of taking a picture with or without zoom, and then zooming in closer for a clearer view of the subject matter.
Electronic Books	As the name suggests, these are books with an electronic component that allows for large print or are text-to-speech enabled so that the book can be read to them. This will likely be superseded by companies such as Audible who provide digital offerings in this area.
High Resolution Images	A higher resolution image allows for greater zoom potential without losing clarity of detail.
Image Descriptions / Alt Text	Used by screen readers to provide a user with a description of an image contained in a document or website for example. This solution is dependent however on the creator remembering to provide the information and can be a source of frustration for screen reader users.
Auditory Feedback	This implies a simple use of sound to draw the user's attention in a particular direction. Particularly useful in the missing keys scenario.
Screen Reader	This is a piece of software that literally reads what is on the screen. It is highly dependent on well formatted documents, web sites and apps with a particular dependency on the use of accurate and helpful image descriptions and alt text tags.

Table 1: Summary of low tech assistive vision technology from Perkins School for the Blind website

1.3 Second Generation Assistive Technology

Whilst the second generation technologies are explained in detail in the state of the art review in the next section, a summary of the functionality provided by the leading products is given below to provide an insight into the ways in which modern technology is helping the blind and visually impaired to become more independent. In addition to smart glasses, the technological leap forward with smart phones has provided perfect platforms upon which to deploy and leverage the huge advances made in the fields of computer vision and artificial intelligence in recent years. As we see from the table below, they broadly fall into two categories:

1) A suite of apps and smart glasses used to sense and describe the environment around the user as well as providing the ability to read on their behalf and identify objects, products and faces.
2) Wearable devices focusing on utilising haptic technology to provide navigational assistance to the user as well as collision avoidance capability in the most advanced offering.

Channel	**Seeing AI**	**OrCam MyEye2.0**	**Envision AI**	**Maptic Wearables**	**Tactile Braille Reader**
Delivery Method / Device	App	Glasses	App & Glasses	Wearable Device	Handheld Device
Internet Required?	Partial Yes	No	Yes (for now)	Yes	No
Short Text	Yes	Yes	Yes	No	Yes
Document Reading	Yes	Yes	Yes	No	No
Product Recognition	Yes	Yes	Yes	No	No
Person / Face Recognition	Yes	Yes	Yes	No	No
Scene Recognition	Yes	Yes	Yes	No	No
Currency Recognition	Yes	Yes	Yes	No	No
Image Recognition	Yes	Yes	Yes	No	No
Collision Avoidance	No	No	No	Yes	No
Directional Assistance	No	No	Limited Video Call Assistance	Yes	No
Voice Activated?	No	Limited	Limited	No	No

Table 2: Summary of second generation assistive technology product features

The drawback of these technologies is that there are subtle differences between them and that in order to possess the full range of assistive technology capabilities, the user would be required to carry a smart phone, smart glasses, a wearable device, a braille-reading device and have internet connectivity. The price alone would be prohibitive never mind the inconvenience of carrying and using multiple assistive items.

The gap in the market appears to be a product that combines all of this functionality and uses an interactive personal assistant to manage the transition between the various functions with integration between app, glasses and wearable haptic devices covered by Bluetooth. With modern smart phones possessing an increasing amount of storage capacity it might be possible to produce a fully offline solution contained within a dedicated memory card.

1.4 Scope and Objectives

The previous section has summarised an amazing range of domain leading technologies that allow a blind person to engage with the world in a much deeper way than was ever thought possible. A well-equipped blind person could now have an easier time of finding their missing keys, navigate to their local coffee house whilst potentially avoiding all collisions, use smart reading to locate their favourite newspaper at the news stand. Scan the coffee house for their friend waiting at the table. Order from the menu. Verify the bill and pay with a banknote of sufficient denomination. Then let their friend leave and enjoy having their chosen articles in the newspaper being read to them by their assistive technology before pressing the "get me home" button to end their day of freedom.

This leads to the question of where value can be added in taking these products towards an enhanced second or even third generation level and give blind and visually impaired people even more freedom and independence. It can be seen from the second generation that the leading products have adopted the same theme and broadly bring the same capabilities and constraints. The common omission found in all of the apps and smart glasses is the lack of any directional assistance in terms of actively guiding a user towards their destination, or target object. The products passively advise the user of what is around them and they do this excellently, but they do not actively assist the user in achieving any sort of retrieval task, for example "take me to the chair" or "guide my hand towards the cup".

As this is a computer science project (and not engineering), the production of an actual device will be outside of the scope. Therefore, the scope of this project will be to build a proof of concept that will provide a rudimentary form of active directional assistance to the user of one of these apps or smart glasses. The smart glasses are powered by common artificial intelligence applications such as object detection, object tracking, computer vision and NLP to provide basic user interaction. Some of these apps appear to be a standalone set of functionalities that could benefit from the addition of a personal assistant who could listen to the user and call the necessary functionality in the background so that the user is not expected to manually move between channels. This would combine the best of the app with the best of the smart glasses. Envision AI is already on this pathway.

Providing a blind person with the ability to control all functionality via voice commands in conjunction with the ability to select an object and be guided towards it seems like the logical next step in adding value to the existing solutions. The project is called "a guiding hand" so the focus will naturally be on guiding a hand towards an object that has been detected and selected. The development style should be such that the solution is extendable, scalable, and most importantly, usable. The goal is to further enhance the freedoms of the blind and visually impaired in a practical fashion whilst providing a platform for future research and development.

Finally, the stated scope of the dissertation supports the vision statement of the World Blind Union [8] which strives for a world where blind or visually impaired people can participate fully in any aspect of life that they choose.

1.5 Achievements

The project aim was to prove a concept that the scope of existing assistive technologies could be extended to the point where an AI solution could guide a blind person's hand towards a detected object to assist in the retrieval of said object with control of the user interaction managed by a personal assistant. The solution offered in this document proves that this is possible and lays a solid foundation for future research and enhancements.

The solution blends a variety of common artificial intelligence applications including an NLP based personal assistant, SSD object detection, open source hand tracking and bespoke computer vision guidance to achieve the objectives laid out in the scope section. The deliverable was not an easy task to achieve and required the use of multi-threading and object oriented coding practices to create a coherent and robust codebase. The learning curve was steep and extremely challenging at times which led to a more in depth appreciation of both the individual technologies and how they can be integrated in innovative ways to provide real world value to communities that stand to gain life changing capabilities from these types of solutions in the future.

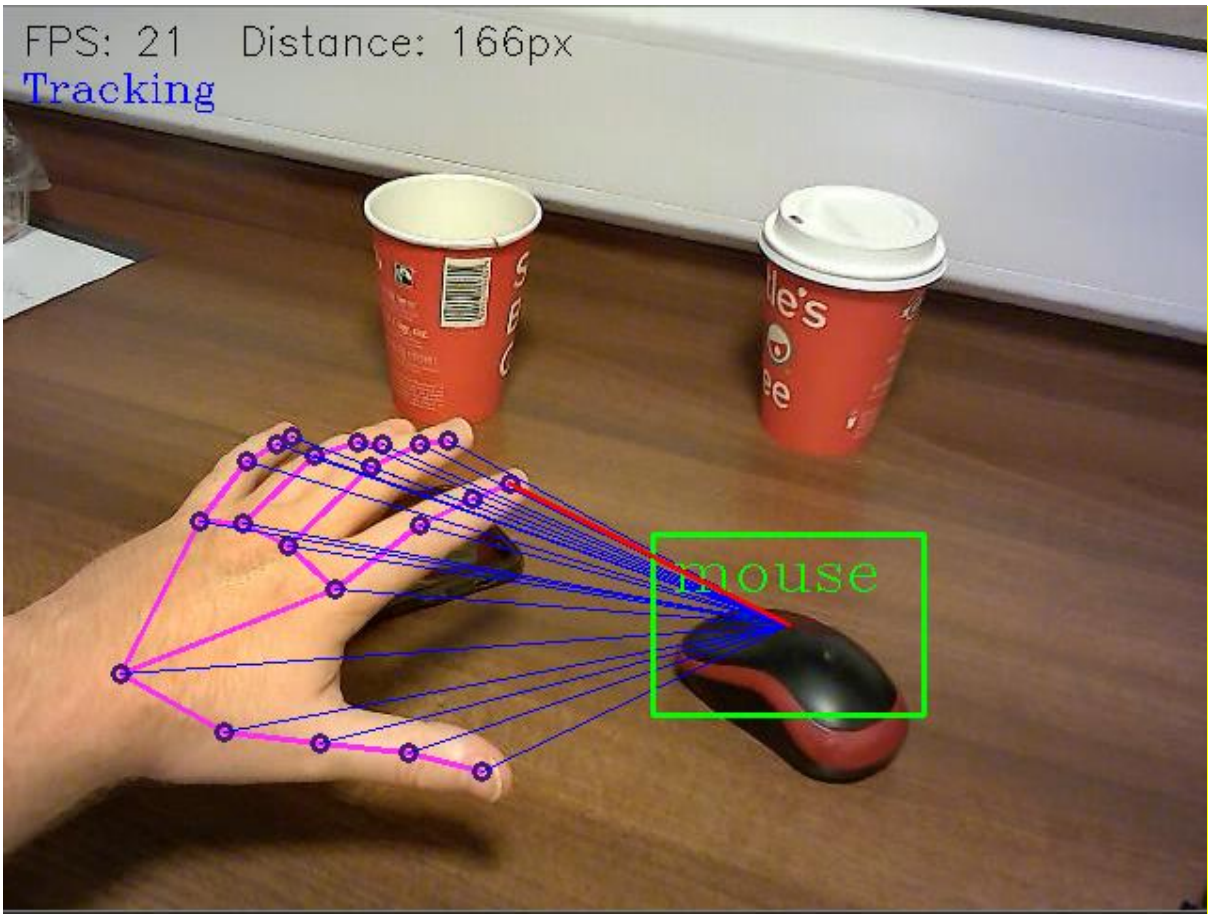


Figure 1: A guiding hand solution in operation with red line denoting closest landmark to target centroid

1.6 Overview of Dissertation

The following sections describe the research and development of the proof of concept solution:

Chapter 2 – State-of-The-Art: This chapter analyses the current developments in the field of assisted vision for blind people and the technologies used to create these solutions.

Chapter 3 – Methodology: This chapter details the processes and decision making constraints that will be used in defining, designing, and guiding the build of the proposed solution.

Chapter 4 – Requirements: This chapter outlines the requirements and use cases that are considered within the scope of the project.

Chapter 5 – Design: This chapter looks at how the application will be created including a high-level view of the proposed architecture before drilling down to more low level details of the individual application components.

Chapter 6 – Implementation: This chapter describes the functional components of the solution and how they are configured and built in line with achieving a successful outcome, as well as documenting decisions around problems and solutions that were encountered during the build process.

Chapter 7 – Testing: This chapter details the testing of the concept with an aim to show that the application is proven as a viable solution against the requirements.

Chapter 8 – Conclusion: This chapter discusses the performance of the solution against the objectives, scope and requirements of the dissertation as well as comparing the work to current solutions and research. An insight into future possibilities is also offered.

2 State of the Art

The start of the art in assistive technologies can be separated into two streams. The first provides a detailed review of the leading second generation assistive products. The second analyses the branches of artificial intelligence that underpin these solutions with a focus on the aspects that could be used to satisfy the scope of the project in creating a next generation solution.

2.1 Second Generation Assistive Products

2.1.1 Seeing AI (Phone App)

Microsoft Seeing AI is a channel based app that allows the user to complete the most comprehensive list of tasks through an app. The app uses the phone's camera to scan the environment and provide feedback to the user. All the user needs to do is switch between various channels in the app to activate the different functions. The table below summarises the channels that are available [9]:

Channel	Description
Short Text	Used for reading short passages of standard text using OCR (not handwritten). The unique element here is that it reads automatically without requiring a prompt.
Document	Detects the edges of the document and guides the user to correctly align the document within the camera frame. It also recognises the formatting within the document (e.g., heading 1). The user can also swipe through the document line by line if required.
Product	Allows the user to hold a product up to the camera which will use a series of beeps to guide the user towards barcode. Once identified it describes the product. The faster the beeps the closer it is to detecting the entire barcode, as per reversing cameras in cars.
Person	Recognises people and gives a description including age and emotion. The app guides the user to centre the face within the frame. This enables blind people to post well taken pictures to their social media platforms which improves their ability to engage with the world via this medium. The app can also be taught to recognise friends by taking photos and entering their name which is then saved and stored.
Scene	This is an experimental channel that is required to be manually turned on in settings. The user takes a photo and the channel attempts to describe the scene within the photo. The scene detection algorithm is constantly improving but the potential is huge.
Currency	Unlike the document recognition channel, the currency channel does not require all four edges to be captured within the frame and provides feedback in real time for new and used currency notes. Currently available for US and Canadian dollars, Pound Stirling, and the Euro.
Recognise Image	This provides the ability to recognise images that already exist and provide scene descriptions as per other channels. This is a great enabling tool as it opens the opportunity for blind people to enjoy commonly shared memes and jokes and fully participate alongside their friends and colleagues as well as understand virtually any existing image.
Others	Other channels not described on the demo page include channels to detect the light levels in the immediate environment, the colour of an object and handwritten text (separate to the short text channel). It's possible that these are also experimental channels that are required to be turned on manually.

Table 3: Each channel on the Seeing AI app performs a different task

The driving force and co-founder behind Seeing AI is a blind software engineer by the name of Saqib Shaikh, whose pedigree includes two other everyday products from the Microsoft stable – Bing and Cortana. Having such a talented and importantly, blind, software engineer leading development has undoubtedly paid dividends in the creation of a sympathetic app proving itself to be extremely useful to the blind and visually impaired community. Many users are treating the app as an essential tool in their quest for greater independence. The app is also free to download and use.

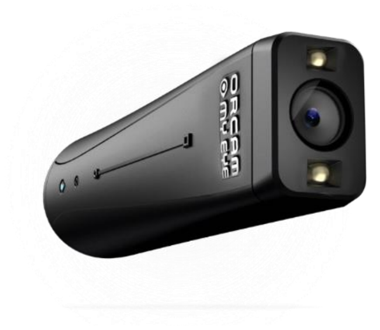
The biggest criticisms of this app are that it is not available on Android, and that it is not voice activated meaning the user is still required to master the use of an iPhone, which is also problematic given how fast phone batteries can drain under heavy use. This can be mitigated somewhat using battery packs.

The language support for this app is rather limited given Microsoft's global dominance in the tech sector. It is only available in 6 languages – English, Spanish, German, French, Dutch and Japanese.

2.1.2 OrCam MyEye 2.0 (Smart Glasses)

OrCam is the idea of Israeli founders Professor Amnon Shashua and Mr. Ziv Aviram, who were also co-founders of Mobileye, a technology that went on to become a system leader in collision avoidance and autonomous driving. They offer two products – MyEye wearable glasses and a laser guided handheld text reader device called OrCam Read. The latter is primarily aimed at people with visual impairment and a range of reading difficulties. It replicates the text reading capabilities of the MyEye product but in handheld form, it is sufficient to recognise its existence without a deep dive.

The MyEye [10] product consists of glasses and a small, elongated device that magnetically attaches to the glasses which acts as the control platform. Priced at around £3000 it is far from being a cheap and affordable solution when compared to Seeing AI but it does take the user closer to a phone free existence and all the freedom and dignity that comes along with that. Once referred to as "talking glasses for the blind" [11], the discrete yet attractive, slimline design is helping to overcome the stigma blind people face in the community and workplace through enabling them to be more independent when interacting with their environment without the device drawing attention to itself.



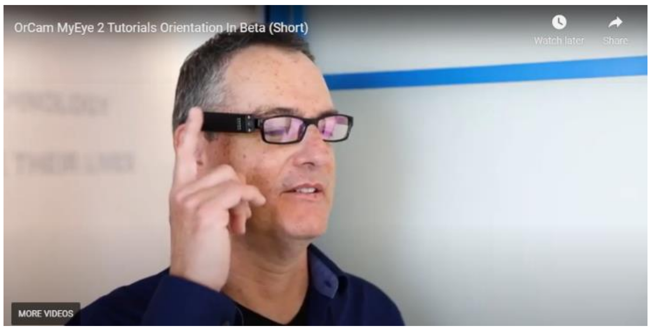


Figure 2: OrCam MyEye 2.0 magnetically attaches to a pair of glasses and is touch enabled

In addition to the appealing looks, it is remarkably lightweight (22.5g), approximately the size of a finger and houses some of the most advanced assistive technology on the market, powered by artificial intelligence and machine learning and a 13 megapixel camera to sense the environment. The device works offline in real-time and can be voice, gesture or touch activated. It is currently used in 50 countries and 25 languages.

The exact activation protocol for each function varies slightly but with commonality between functions where practical. It is interesting to see that the user has the possibility to tap the device to take a picture of what they are looking which gives them more control of their environment and the specific moment at which they want the product to describe their environment. For example, on the product demonstration page an office worker walks into a meeting room, clicks the glasses to take a photo and asks, "What's in the room?" The glasses proceed to perform a scene analysis and relay the results to the user. Being able to approach any scene and gain an idea of what is going on around you is quite an empowering advancement.

In addition to all the functionality provided by Seeing AI, they also offer some gesture recognition such as holding up your arm as if dramatically attempting to tell the time. The gesture is recognised and the time (and optional date) is relayed to the user. Another advantage of certain functions such as text recognition, is that it employs auto detect giving the blind or visually impaired person a hands free experience. There is also the option to point at specific text for more precise actions. Smart reading offers the ability to specific words, amounts, date, etc., within a body of text. This could allow the user to verify the total amount very quickly on a shopping or restaurant bill. It can even identify headlines and articles within a body of text, convenient for reading newspapers.

When dealing with specific detection actions such as product and currency note recognition, adding new products and faces to the memory there is a specific requirement to hold the object steady at a specific distance of 30cm from the camera at a specific orientation, and a distance of 1 metre for facial recognition training. Seeing AI does not require such strict alignment in these scenarios.

2.1.3 Envision AI (Phone App & Smart Glasses)

Envision's [12] offering comes in two forms – an app and a pair of smart glasses. Once again, the app is very similar to Seeing AI and the glasses share a lot of commonalities with OrCam's offering. The Envision AI app was awarded Best Accessibility Experience by Google Play Store in 2019 and in May of 2021, they successfully raised €1.5 million to scale production of the smart glass solution [13].

A product review and comparison with Seeing AI by the American Foundation for the Blind [14] tells us that the Envision short text reader is faster than Seeing AI but that its scene detection algorithm is not as accurate. Envision's document reader also allows the user to scan multiple pages at once which gives a small advantage over its rivals. One area where Envision innovates is that the user is able to select an item or person from a pre-defined list and use the camera to scan the immediate surroundings for that object. When the object is found the phone will sound a beep to let the user know that the object has been found, and when a person is found it will speak their name. Other than that, both Envision and Microsoft seem to be on a par in regard to performance and functionality. Envision AI does not list currency note identification as a feature.

Two nice innovations by Envision smart glasses are scanning for specific objects/faces and the ability to make a video call and receive assistance from a human operator. The operator receives a direct stream from smart glasses (connectivity dependent) and sees the environment from the device POV. From a user support perspective this is a great idea and incorporates the concept of Be-My-Eyes [15], a volunteer community who perform similar tasks for the blind community through live video call.

On cost, it is comparable to MyEye and retails at £3268.91. The app disappointingly comes with a range of subscription fees (Seeing AI is free) – 14 day free trial followed by subscription costs that are hidden until the end of the trial, according to the Play Store. The company website only directs you to the Play Store where it is difficult to find limited information however options are given as monthly, annual or lifetime. No doubt being able to access the app and smart glasses from the one provider is a bonus for some customers.

The 8 megapixel camera is inferior to the OrCam product and at 46g weight is twice as heavy.

2.1.4 Maptic (Wearable device)

Maptic have taken a different direction and have produced a very chic, wearable product that connects to a voice controlled iPhone app to provide navigational via GPS and a unique collision avoidance capability to the user [16]. In fact, it has been so successful that even sighted individuals are now taking an interest in the product. Each component is shaped for ease of identification.

The product is the brainchild of Emilios Farrington-Arnas [17], a graduate of Brunel University Engineering department and user experience is at the forefront of this innovation. The device is exceptionally appealing and avoids the usual ugly design usually associated with assistive technology products but enhances safety by providing vibrational feedback via a series of “hard ticking vibrations resembling sonar” to the left or right side of the user’s body. This approach gives the user a hands free experience and makes them less likely to be a target of opportunistic thieves who have in the past targeted blind people using smart phones in public. Importantly, the vibrations do not distract from the user’s sense of hearing, which becomes the dominant sense [18] once sight is lost.

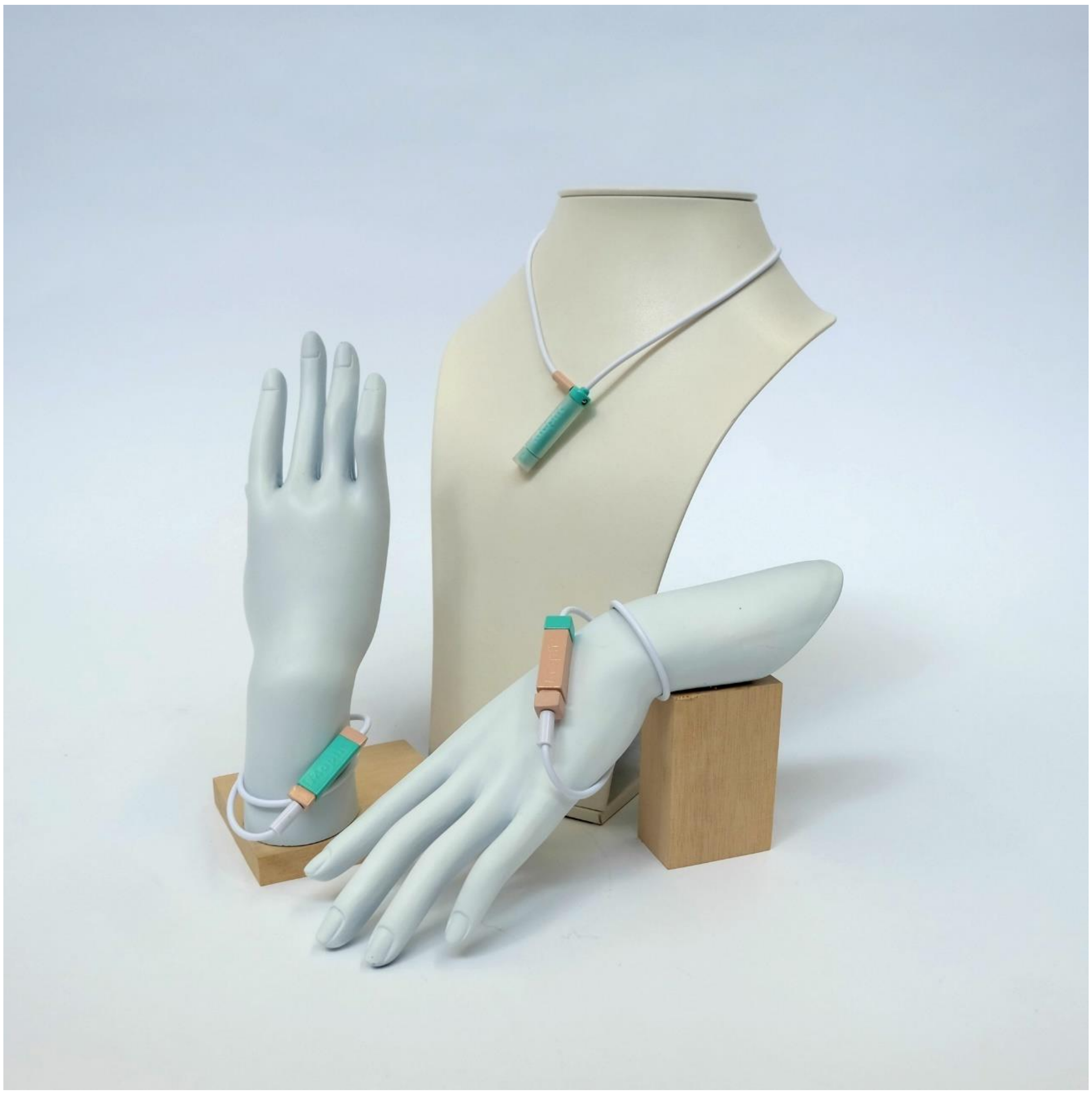


Figure 3: Maptic’s stylish wearables design breaks the stigma of ugly assistive technology products

In terms of collision avoidance, the wearable necklace uses LIDAR to create an awareness of approaching obstacles, something that the traditional blind cane cannot achieve as it has a limited reach of approximately 1-1.5m and is rarely effective at detecting obstacles above knee height.

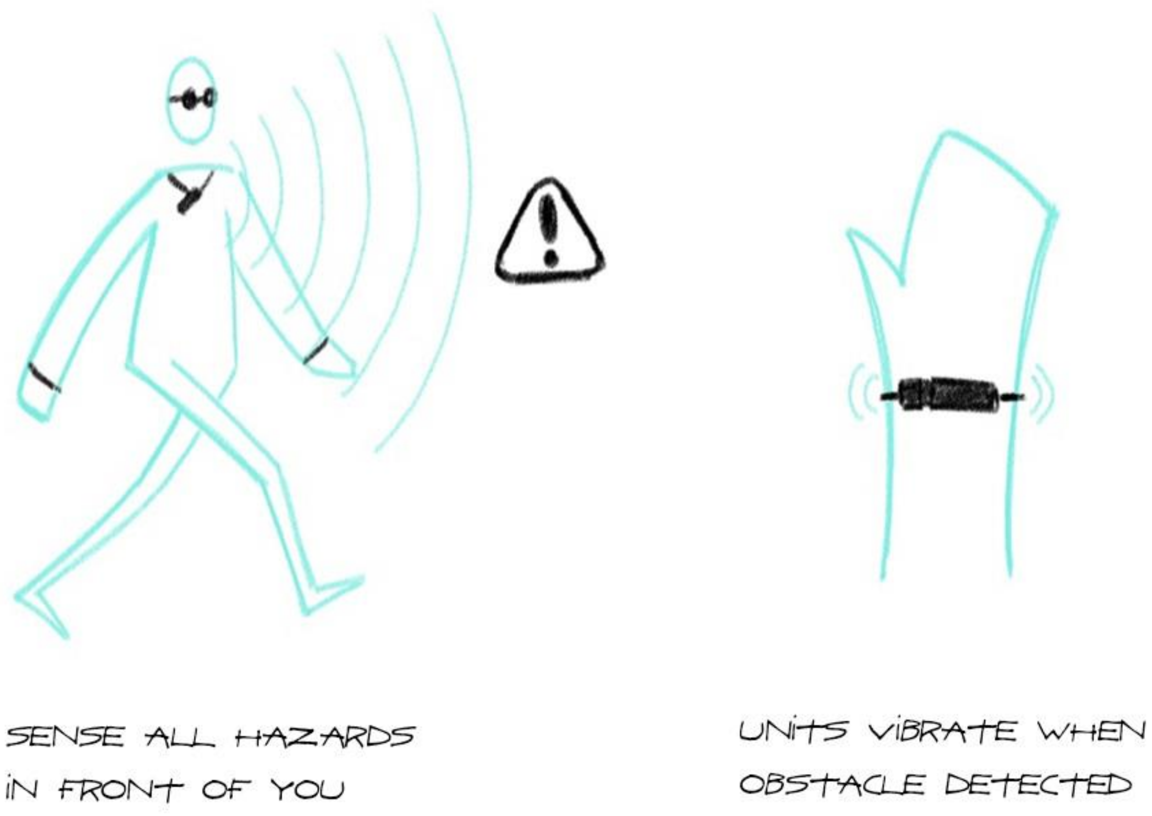


Figure 4: Maptic uses LIDAR to create an innovative collision avoidance capability

In another smart move, the device utilises the smart phone's GPS and Google maps API to minimise the product size and avoid issues with battery life. The downside to this design is that there is now a dependency on having a smart phone, sufficient battery life, and a reliable internet connection which limits its effectiveness in remote, shielded, or underground locations.

As a testament to the extensive user experience research carried out for this product, it comes with a very helpful one touch "get me home" function which makes this wearable truly stand out as a product that places practicality of navigation and user safety as a prime concern.

2.1.5 Tactile

In 2016, an all-female team of engineering students from MIT won first prize in a student technology hackathon [19] by building a prototype of a real-time text-to-braille translator, the first of its kind, and earnt themselves entry to Microsoft's patent program, a program that assists talent in the legal aspect of filing patents. The solution addressed the problem that less than 1% of printed material is currently translated into braille.

They built a device that is slid over printed text such as paper or a menu. The device takes a picture and uses text recognition to convert the text into braille by using a series of moveable pins to create the braille translation on the surface of the device. The ultimate aim is to reduce the size down to that of a chocolate bar (5 inches x 2 inches) and show up to 36 braille characters. The device is intended to be portable, standalone, and ideally costing around £100 (one of the biggest challenges).

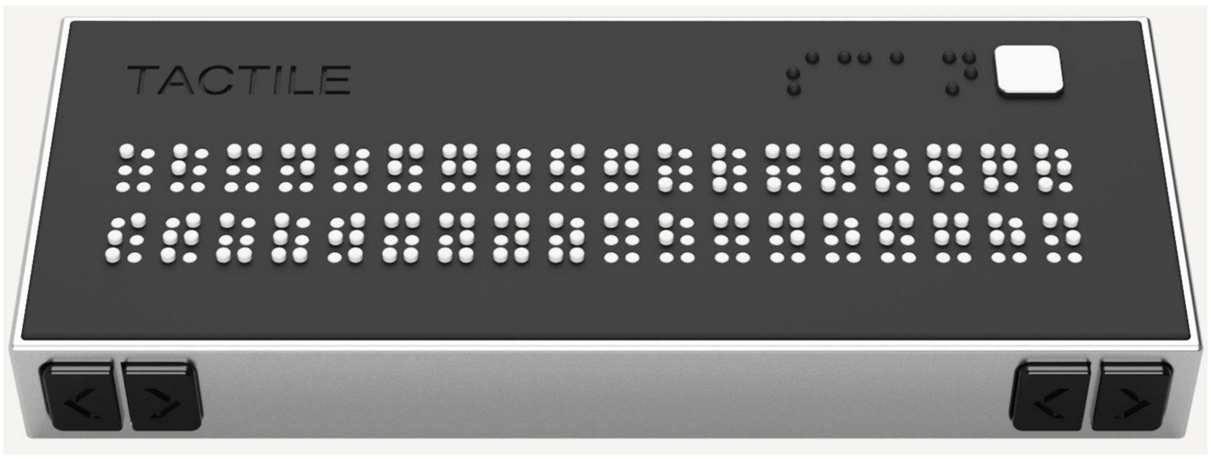


Figure 5: CAD representation of the Tactile text-to-braille product

Whilst not in the same league as the previous products the project does have important implications. Existing braille readers retail from £2000-£12000 making them unsuitable for mass deployment and are often not portable. The prototype was created using a 3D printer which could facilitate the creation of advanced products at affordable prices. When you compare the previous smart glasses that retail at around £3000 you understand how the MIT team's approach to reducing cost and making products affordable becomes relevant to the wider market. Unfortunately, the most recent press release was from 2019 so it is unclear as to the current state of this project.

2.2 Underlying Technology

The focus of this section will be on object detection and natural language processing (NLP) as two technologies that provide the ability to make sense of environments and objects detected by camera, and to provide a human computer interface (HCI). Two hand tracking techniques will be investigated to see how this is achieved as well as a short review of haptic technology and how it is being increasingly used to provide alternative forms of non-verbal instructions to users. All of these methods would be required in creating an enhanced or next generation solution for this project.

2.2.1 Object Detection

Object detection models are crucial in creating assistive technologies for the blind and partially sighted. They are used to act as the eyes of the individual to ascertain what objects are present and where they are within a camera's field of view. Any model worthy of consideration must be accurate, fast enough to be deployed into a real-time environment and small enough to exist on a portable device that may have limited processing power. Deep learning and GPUs have enabled huge advances in recent years. The three leading models are reviewed below.

2.2.1.1 R-CNN Family

When Girshick et al. proposed R-CNN [20], it was one of the first deep learning based object detectors that utilised region based methods such as selective search to produce a large number of potential bounding box objects that were individually passed into a convolutional neural network (CNN). Whilst highly accurate it came at a cost of speed with a frame rate of 5 FPS on a GPU [21], largely due to running up to 2000 individual proposals through the net.

The second iteration known as Fast R-CNN [22] set out to fix a lot of the problems with the original R-CNN. The entire image is first processed by the network to produce a feature map. Then for each proposed region of interest (ROI) a feature vector is extracted and fed into fully connected layers to obtain two outputs - classification and bounding boxes. This approach was estimated to be 9 times faster in the research paper.

This was followed up by Faster R-CNN [23] in the same year (2015) which removed the selective search bottleneck seen in the previous versions and used a Region Proposal Network (RPN) to become a true end-to-end neural network. The RPN is a full CNN that merges with the R-CNN into a single network capable of producing a lower number of high quality region proposals that tell the CNN where to look for the objects. With as little as 300 proposals per image (compared to up to 2000 for R-CNN) the Faster R-CNN was able to produce state of the art results on a CPU and became the bedrock for future benchmarking against the R-CNN family.

Despite its name as Faster R-CNN it is still not that fast as it is routinely quoted as providing a steady 7 frames per second under optimal conditions yet retains a sterling reputation for its high degree of accuracy and is still extremely popular today when accuracy is greatly preferred to speed.

2.2.1.2 YOLO

You Only Look Once (YOLO) [24] proposed by Redmond et al is a self-descriptive approach to object detection as a one-step detection model. It is a popular real-time detection algorithm due to its fast speed. The baseline model clocks 45 frames per second (FPS) with Fast YOLO achieving 155 FPS albeit with the aid of a GPU. The speed is achieved by implementing a more efficient detection architecture. YOLO first resizes the image into an SxS grid before running it through a CNN and then applying non-max suppression to discard low confidence detections and was the first to treat object detection as a regression problem. There are two big drawbacks of YOLO [25] and they are that it does not handle small objects nor objects close together particularly well, leading to a loss of accuracy. The reason for this comes from how YOLO divides the image into an SxS grid where an object prediction is made for each grid. If multiple small objects are contained within each grid, it will naturally struggle to make the correct identification.

The latest version of official YOLO is v3 [26] however v4 [27] and v5 [28] do exist but they are not considered "official" YOLO models as they are not created nor endorsed by creator Joseph Redmond.

2.2.1.3 SSD & MobileNet

SSD was proposed by Wei Liu et al. [29] and is based on the VGG16 architecture created by Simonyan & Zisserman [30] at Oxford University. The key speed gain over Faster R-CNN came from removing the bounding box proposals and then apply a series of filters through a small CNN which was able to retain a high degree of accuracy. It is one of these filters, a multi-scale feature map, that SSD uses to handle aspect ratio issues seen in other detection models. The detection area is resampled into a number of different aspect ratios which are then used to produce a high accuracy prediction of the object. The original version of SSD in the research paper was reported as both faster than YOLO (59 FPS vs. 45 FPS) and more accurate than Faster-RCNN (74.3% mAP vs. 73.2% mAP) on the VOC2007 dataset however the general consensus is that it lies somewhere between the two for the most part.

SSD MobileNetv3 is an implementation of the SSD model on Google's MobileNet backbone. This provides a streamlined and relatively lightweight implementation of object detection specifically designed for mobile and embedded systems. SSD is now arguably the preferred real-time detector where accuracy is preferred over speed with many reviewers stating that they only use YOLO when they absolute need speed above all other considerations.

2.2.1.4 Spatial CNN

Spatial CNN (SCNN) [31] is a new kind of RNN proposed by Pan et al. to explore the possibilities of learning a deeper level of pixel level semantic relationships. To highlight its effectiveness at preserving the integrity of long thin spatial relationships they tested their model on the challenging traffic lane detection and Cityscape datasets with impressive results. The SCNN model outperformed RNN by 8.7% winning 1st place on the TuSimple Lane Detection challenge with an accuracy of 96.5%.

The SCNN appears to implement a number of convolution channels equivalent to the number of available classifications in the model to remove the current iterative approach used by the standard Markov Random Field (MRF) [32] or Conditional Random Field (CRF) [33] models. Instead of all pixels receiving information from all other pixels, the SCNN takes slices and forward propagates the messages through the CNN in four directions (downward, upward, rightward, and leftward) making it more computationally efficient than previous models as well as more flexible, as the layers can be easily inserted into a CNN. The researchers are optimistic that this model could help push forward research into autonomous driving capabilities.

This model would be quite appropriate for any future solution that attempted to guide blind people along public footpaths or along roadsides as a lane management application to avoid the user accidentally stepping into harm's way.

2.2.2 Hand Tracking

The two main representations of hand tracking techniques that exist are virtual reality and artificial intelligence. Here, the technologies behind the leading contenders from each domain are discussed.

2.2.2.1 Virtual Reality

Hand and body tracking has been around for quite a while in the guise of Virtual Reality (VR) and Motion Capture (MOCAP) techniques and they are most prevalent in the gaming and film sectors however it is still a relative newcomer to artificial intelligence with limited resources as yet available. Clunky and difficult to master VR controllers are being replaced by mid-air hand and gesture recognition software and is increasingly being used in many sectors from engineering, medical and product designers. Ultraleap [34] has developed one of the most, if not the most, advanced hand trackers in the world using hands free headsets. These headsets are also far more hygienic with a widely reported [35] study in 2013 by UNICEF and Unilever finding that the average gaming controller held over 5 times the level of bacteria as the average toilet.

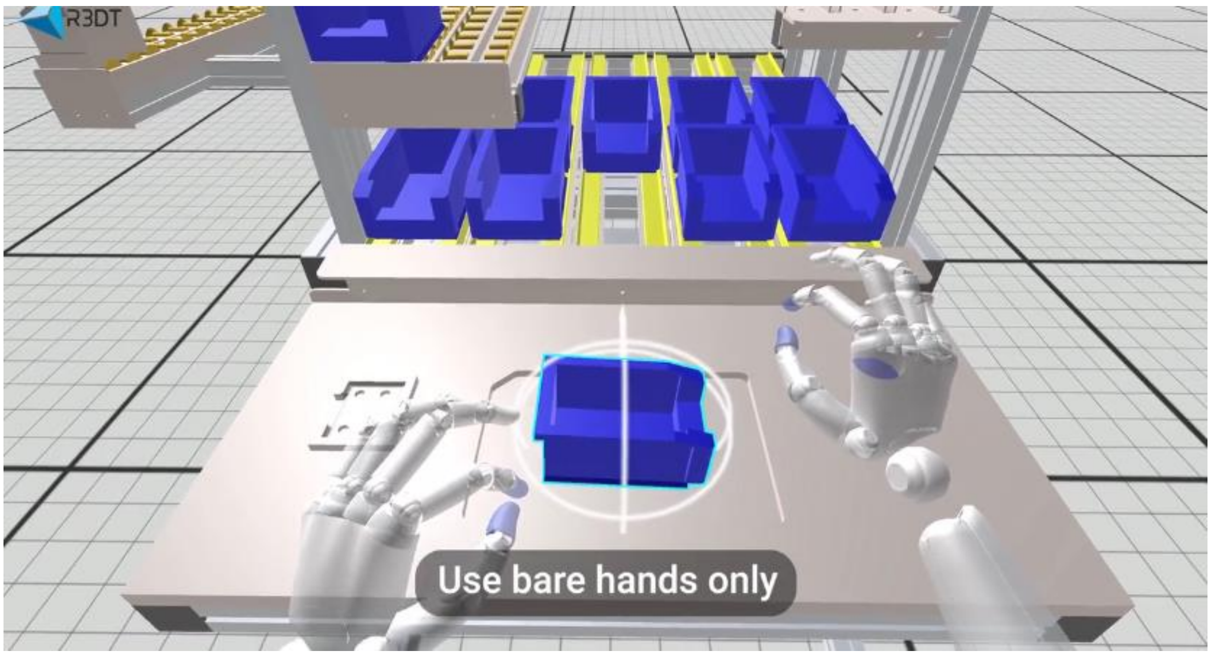


Figure 6: Ultraleap's VR headsets give impressive design capabilities

In the case of Ultraleap's flagship offering, the solution is based upon innovative software and hardware solutions that were a decade in the making and spanning 5 generations of iterative development. The tracking hardware uses a state of the art infra-red stereo vision camera combined with integrated sensors to capture raw data on the position and velocity of the hands. The software side of the house takes this raw data and uses it to create real-time digital models of the user's hands with the VR experience housed on their fifth generation engine called Gemini. The Gemini engine is compatible with Unity, Unreal Engine and OpenXR making it a scalable enterprise level product.

For hardware, the state of the art infra-red camera uses some novel techniques to improve the quality of the hand tracking software. It uses infra-red LEDs to light up the hands for detection whilst filtering out wavelengths that are outside of the 825-875 nanometre range. To get the clearest image of the hands the brightness is automatically adjusted to provide optimal detection conditions. If the hands are moving slowly the camera also slows down the frame rate to improve tracking accuracy and use less power.

It is clear that some advanced thinking has gone into this product but it is not practical for a blind person to walk around with large stereovision headsets just to track their hands in pursuit of picking up their knife and fork at a restaurant. Nor is it a visually appealing solution with many blind people reticent to use something that draws excessive attention to themselves. Whilst artificial intelligence has come a long way in recent years there is only one framework than can match the level of VR technology seen in the Ultraleap headset and that is provided by Google. Additionally, it would be far easier to integrate the supporting AI requirements into an AI solution instead of into a VR solution.

2.2.2.2 Artificial Intelligence

Artificial intelligence has the twin benefit of being more accessible and affordable than VR headsets with equally impressive capabilities albeit with a separate set of use cases. It is not yet at the level to be used to train surgeons as with VR but can be used for gesture recognition such as those gestures used in sign language, or in the case of German company Algoriddim [36] who used Apple's CoreML framework to train an algorithm that allows users to control a set of DJ decks with gesture recognition.

Google MediaPipe is arguably the leading open sourced Machine Learning platform that is rapidly expanding in terms of its solutions and uptake in the AI community. In 2019 at the Computer Vision and Pattern Recognition (CVPR) conference; the MediaPipe framework was used to implement a new approach to hand perception that went on to become known as MediaPipe's hand tracking algorithm [37]. The technique can infer 21 hand landmarks from a single frame. To increase the efficiency the detection is made with the first frame and the ML algorithm then tracks the hands until such time that the track is lost and it needs to make a new detection.

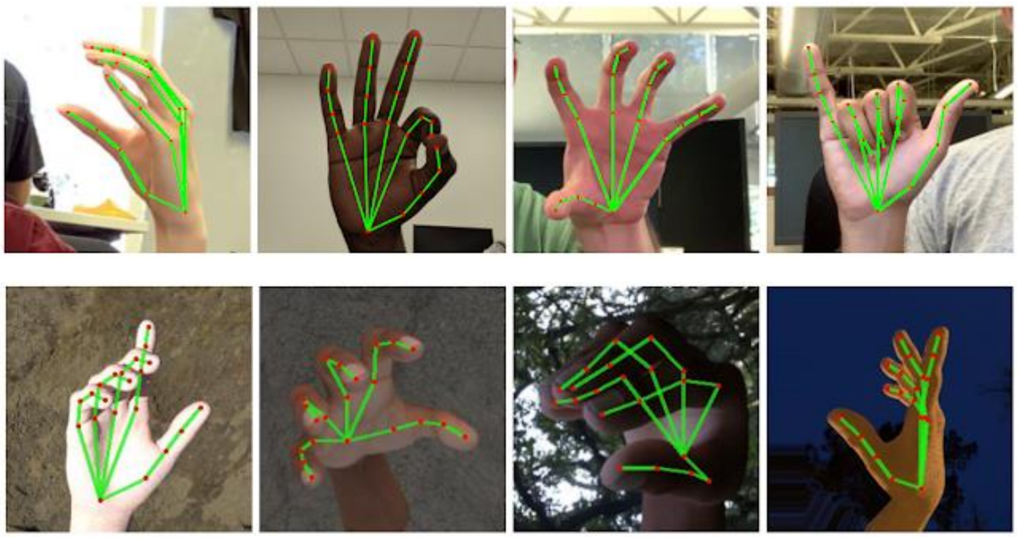


Figure 7: Example detections from the cutting edge MediaPipe Hand Tracker website

The high degree of accuracy is provided using a technique [38] that first using the MediaPipe BlazePalm single shot detector coupled with non-max suppression to accurately detect the bulk of the hand using simple square bounding boxes. Then an encoder/decoder pair is used to provide awareness of the bigger scene around the hand. These techniques combined with a single shot approach to minimise the focal loss, as proposed by Lin et al. [39] allow for a palm detector with an average precision of 95.7% against a baseline of 86.2% when using basic cross entropy loss and no encoder/decoder. Once the palm detection has occurred the hand landmark model performs localisation on the 21 hand landmarks including the palms and knuckles of each finger/thumb. Approximately 30,000 hand images were labelled with the required hand landmarks to obtain a comprehensive set of ground truths with which to train the model.

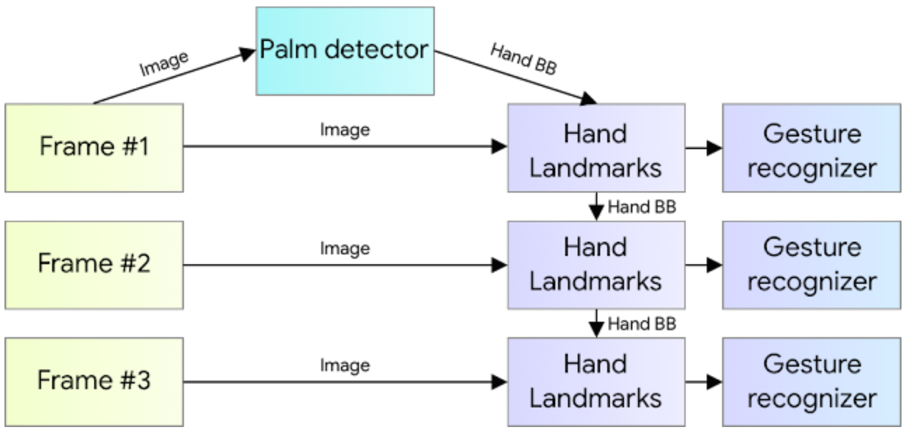


Figure 8: Architecture of the two stage Media Hand Tracker [38]

To extract maximal performance during training the development team found that a combination of real world and synthetic data was optimal:

Dataset	**Mean Regression Error (Normalised by Palm size)**
Only real-world	16.1%
Only synthetic	25.7%
Hybrid (real-work + synthetic)	13.4%

Table 4: Training set results for obtaining the best accuracy for the predictor [38]

MediaPipe, as a whole, is platform independent however as the product is still in its alpha stage with version 1.0 yet to be released there is no official support for Windows leading to reports that it can be somewhat confusing [40]. Performance wise it is a huge step forward in the domain of pose detection. It is written in C++ and utilises multi-threading, GPU acceleration and graphs making it relatively straight forward to achieve 30 frames per second on a modern laptop which is remarkable when compared to previous leader OpenPose, which is considered lucky to achieve 7 frames per second [41].

2.2.3 Natural Language Processing

Natural Language Processing has many forms from the text readers and character recognition used by the leading products reviewed for this project to automated chat bots and personal assistants. This review will focus on speech recognition and synthesis that would be required for developing a personal assistant and the pros and cons of online versus offline functionality where relevant.

2.2.3.1 Speech-To-Text

Automatic Speech Recognition (ASR) is one of the key components required in building a personal assistant. It has been seen that speech recognition it used by some of the products already reviewed that allow the blind user to interact with the apps and smart glasses through vocal command. In simple terms the user's speech is converted to text by the application which then triggers a specific action linked to something within the speech-to-text translation. According to an article by SmartAction [42], the Word Error Rate (WER) is frequently used as the go-to metric for assessing the quality of speech recognition algorithms however it is far from perfect as a metric. The metric is given as:

Word Error Rate = (Substitutions + Insertions + Deletions) / Number of Words Spoken

The terms are defined as:

- Substitution: When a word is replaced (for example, "shipping" is transcribed as "sipping")
- Insertion: When a word is incorrectly added (for example, "hostess" is transcribed as "host is")
- Deletion: When a word is omitted (for example, "get it done" is transcribed as "get done")

Many of the models that rely on this metric are not only trained and tested on the same corpus but are tested under laboratory conditions that are not replicated in the real world. In real world situations there are a number of variables that impact the voice recording that is used for transcription by a service [43]. In addition to the quality of the microphone there are also considerations for the distance of the speaker from the microphone and the level of background noise that could be picked up. Individual speakers have their own distinct accent variations and cadences within group dialects that would require a huge amount of data for a machine to generalise accurately. As such error rates are still not at the level of human transcriber performances (4% WER) with Microsoft and Google claiming leading word error rates of 5.1% and 4.9% respectively [42].

Popular offline speech recognition providers such as Kaldi and Pocketsphinx are capable of producing accurate results but are frequently reported as having large lag times between the user finishing their sentence and the text being transcribed due to the lack of computing power available on smaller devices. A modern mobile device could handle it [44] however it would not be much faster than sending it to a cloud service and the extra processing would drain the battery fairly quickly.

Google has developed a promising offline solution [45] that appears to solve both of these problems in a mobile device. In a paper by developers Yanzhange et al. [46] they describe how their new model, which uses a Recurrent Neural Network Transducer (RNN-T), outperforms conventional Connectionist Temporal Classification (CTC) models on both latency and accuracy whilst coming in at 80MB in size. The downside is that it is only available on a Google Pixel phone with a GBoard keyboard and in American English.

The popularity of online solutions lies in the fact that they are vastly superior in almost all ways, the exception being that an internet connection is required. The leading online speech-to-text providers achieve their superior levels of accuracy with Natural Language Understanding (NLU) engines. These engines not only improve the accuracy available to standard speech-to-text services but can also go one step further and imply intent creating an infinitely more natural interaction between human and machine where sentences can be spoken aloud and meaning inferred without having to speak a solo keyword or include a keyword within a sentence. Whilst the full suite of NLP-NLU capabilities is not required for this project, a summary of leading provider capabilities is given below:

	Facebook	**Google**	**Amazon**	**LUIS**	**IBM Watson**
Training Module	Yes	Yes	Yes	Yes	Yes
Module Import/Export	Yes	Yes	No	Yes	Yes
Recognise User Intent	Yes	Yes	Yes	Yes	Yes
Pre-built Entries	Basic	Basic Plus	Huge List	Basic	Basic
Pre-built Intent Domains	No	~ 35	No	~ 170	No
Speech Recognition	Yes	Google Speech	Yes	Bing Speech	IBM Speech
Third Party Integration	No	Yes	Yes	Yes	No
Supported Languages	132	20	5	11	12
Limits for API Calls	Unlimited	Unlimited	Trial	Free 10k/Month	Free 1k/Month
Pricing	Free	Free	Trial	Free -> Varies	Free -> Varies

Table 5: Selected comparison extract of leading NLP capabilities from AI Multiple [47]

2.2.3.2 Speech Synthesis

Heavily tied to the previous section in terms of the capabilities provided, text-to-speech or speech synthesis is the second key component in the ability to communicate back to the user. Once again deep neural networks (DNN) have enabled a massive step forward in this domain for the cutting edge applications.

Offline options for text to speech are somewhat limited and tend to provide a somewhat robotic sounding voice. One of the most popular is Python Text To Speech (pyttsx) [48] and is platform independent however this tends to suffer from a long standing issue of the voice sounding robotic and unnatural. There are very limited options for configuring the voice, the user can select the language, gender, and speed of the voice. For a language such as C++ there is a package based on hidden Markov models (HMM) called hts_engine [49], as well as Festival and Flite (Festival Lite). Flite was developed as a lightweight version of Festival in response to criticism that Festival was too big, slow, and not portable enough to be used on small devices [50]. Being written in C++ it not easily customisable at run-time and requires a greater programming overhead which would not suit agile projects.

The best options are once again provided in an online environment due to the large processing requirements of creating a very human sounding voice. The leading contenders can once again be chosen from the table at the end of the preceding section and can be seen in some of the most famous human computer interfaces (HCI) on the market today, namely Alexa, Siri, and Cortana, who provide personal or virtual assistant capabilities beyond anything previously seen in the home.

These assistants often merge NLU and ASR technologies to provide cutting edge capabilities in intent recognition. Using the example of Microsoft Azure's LUIS (Language Understanding), a chatbot or personal assistant can be easily developed to extract meaning and intent from a user utterance before responding with an appropriate response from its language model. This gives the user more freedom to interact in a natural fashion whilst giving the developer an easy way to leverage advanced text to speech models in a cloud environment and deploy high quality solutions without excessive investment into the process.

The reason why many speech synthesis solutions sound so robotic [51] is that they rely on concatenated text to speech (C-TTS) where sentences are constructed from piecing together fragments of recorded speech and recombined on the fly to produce sentences. The result is often not pretty and clearly identified as a machine talking. Google Cloud has recently introduced its DNN WaveNet model which is regarded as the most advanced speech synthesis model available today. In a paper by developers Van Den Oord et al. [52] (DeepMind team) they describe how users rate the resulting voice as significantly more natural than other speech synthesis providers in both English and Mandarin. WaveNet is an auto-generative and full probabilistic DNN based on Pixel-CNN [53] also part developed by Van Den Oord which also makes use of gated activation units and both residual and skip connection.

2.2.4 Haptics

Haptics are defined as the use of technology to stimulate the sense of touch and motion to create sensations that are felt by the user, such as vibrational feedback in a PlayStation gaming controller. For blind people, the use of haptic technology can be a subtle way to convey information without the need for vocal feedback ensuring that they retain the use of their primary sense [54] and are able to be guided in a particular direction in a stealthy and dignified manner.

2.2.4.1 Bone Conduction

The Future Cities Catapult and Microsoft [55] have partnered to create a special headset in conjunction with a Chinese sports headphone company called Aftershokz [56] who mastered bone conduction technology whilst making military grade headsets. The headsets can be customised to the position of the user's ear canals for the purposes of creating a "3D soundscape" that can trick the brain through bypassing the inner ear canal and projecting low level vibrational frequencies into the cranial bone to create an impression that sound is coming from a particular direction using bone conductive technology. The exact details are sparse but it is hoped that the soundscape can help blind users to orientate themselves in relation to places that they are navigating towards. For example, if they are looking for a bus stop and it is behind them, the vibration will appear to come from behind so they know that they have to turn around. The project is part of a cities unlocked plan to create a more liveable environment for blind and visually impaired people.

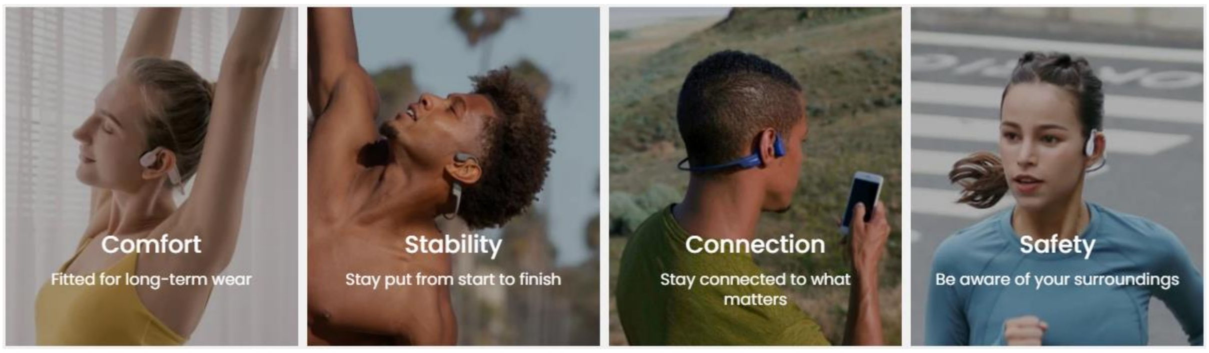

Comfort
Fitted for long-term wear
Stability
Stay put from start to finish
Connection
Stay connected to what matters
Safety
Be aware of your surroundings

Figure 9: Aftershokz produce stylish yet practical bone conduction technology headphones

2.2.4.2 Audio vs. Tactile Methods

In 2015 and 2017 respectively, Flores et al. [57] and Jimenez and Jimenez [58] both found that whilst tactile feedback was slower it did prove to be more accurate with the Jimenez study finding that the tactile test run was more accurate once the user had already completed navigation under audio guidance. As such, Bharadwaj et al. [59] discovered that whilst auditory feedback was superior in quiet environments the advantage diminished to near equal terms in loud, distracting environments. Furthermore, they postulated that the unfamiliarity with tactile feedback devices had an impact on test subject performance and with training the results could equal or better the audio feedback results. Nevertheless, they concluded that tactile navigation methods had great potential.

Researchers from the University of Toulouse [60] specifically looked at using wrist vibrations to guide hand movement using custom designed Arduino bracelets. In two separate experiments involving tracing a route network map onto paper and a straight line corridor walk with blindfolded participants, it resulted in no significant results for tracing a route network but a tendency to complete the task quicker with vibrational feedback. With the corridor walk, it was found that there was a significant decrease in the average deviation and distance travelled (p=0.005) when using the vibrational bracelets.

A study by Durá-Gil et al. [61] looked at where the ideal position of wearing a haptic device might be for running and walking purposes. It was found that a waist belt was the optimal position with suggestions for having vibration pads at the rear for stop, and at the front of the waist for move forward which seems fairly intuitive. The users reported an increased sense of security whilst wearing the belt and the results showed that a single pulse was the preferred vibration pattern.

All of these studies, and many others, point to the fact that tactile feedback mechanisms are not only effective for the purpose of providing navigational assistance for the blind and visually impaired community, but serve to provide the required assistance without compromising their much needed sense of hearing and offer a promising alternative to audio based feedback whilst still empowering the user to be more confident in performing basic navigational functions.

3 Methodology

When implementing a software solution there are many decisions to be made but these can be guided by following an established framework that considers both technical and non-technical constraints.

3.1 Waterfall vs. Agile

When developing software there are a large variety of development frameworks to guide an individual or team through the Software Development Life Cycle (SDLC). They cater for a range of needs that not only serve to inform the speed of development but also the management of the development team and work schedules. Two methodologies considered for this project are Waterfall and Agile.

Waterfall derives its name from the sequential and rather inflexible approach where the project trickles down through the phases when and only when the previous stage has been completed: Requirements -> Design -> Development -> Testing -> Implementation & Maintenance. It is excellent for managing large projects with well-defined and static requirements however this approach can create substantial problems if a change of tack is required during the development or testing phases of a project. In its defence, Waterfall does lend itself to creating well documented solutions.

Agile on the other hand is, as the name suggests, a set of methodologies aimed at moving a project along at a faster and more flexible pace. Work is broken down into short sprints or iterations typically lasting a maximum of 2 weeks. Each sprint can be tested in isolation and therefore issues tend to be identified and rectified quickly and easily since the coding experience is fresh in the mind. This methodology lends itself to smaller scale projects and smaller teams where organisational efficiency is not so much of an issue. A fair criticism is that for complex systems, the methodology does not allow time for reflection as its raison d'être is to facilitate faster and more efficient development cycles. Another criticism is that due to the short timeframe from starting requirements to completing testing it is easy to skip documentation and lose the reasons for why a requirement was developed in a particular way.

3.2 Chosen Framework

For this project the agile philosophy is more appropriate as the development consists of a single developer who needs to iterate quickly without hindrance to create a proof of concept. The project will require several applications of AI to be knitted together therefore a modular approach to design feels like a natural course of action. The concept of each module can be developed independently and brought together into a final solution once all components are working as expected. The coding for this project has the potential to become very complicated in a very short space of time. A modular design completed on an agile sprint basis is a very good way for a single developer to manage this complexity. The managerial aspects of Agile such as daily stand-ups/scrums will not be required as the project team consists of a single individual and will require free-thinking to come up with as many iterative trials as required to create modular code that interacts with other modules. Whilst a Kanban board will not be used as a software tool, the concept will be loosely used to inform the development sequence in order to reach a point where the code can be fully integrated with itself.

3.3 Technical Constraints

The primary technical constraint is that the development laptop that is 7 years old and was neither designed nor optimised for graphics processing. If the aim is to deploy the solution into a wearable device, then the absence of state of the art processing power contained within the latest laptops designed for graphical processing (and AI) might turn out to be a benefit. In other words, inefficient code will be discovered far more easily with the available setup.

For the purposes of demonstration and the efficiency of the object tracking algorithms a high frame rate is a concern. In an ideal world achieving 30 frames per second (FPS) would be the gold standard. From initial explorations into video processing and considering the age and specification of the development laptop, it is more reasonable to hope for a passable frame rate of approximately 15 FPS, with a more cinematic frame rate of approximately 25 FPS being considered as an unlikely top end achievement. A frame rate of less than 12 FPS tends to create a jerky on-screen effect and makes it difficult for the object tracking algorithms to maintain the track and accurately update the bounding box location.

A second constraint is the camera equipment. It was initially hoped to use the Xbox Connect360 and its range finder camera to facilitate localising an object in 3D however the drivers are not compatible with Windows 10. Wearable spy camera glasses also failed to provide video streaming. As a result, the solution will need to work through a standard web cam positioned to mimic a user wearing the device, unless a workable eye level attachment can be found. Stereovision camera glasses were considered as a wearable product but were too expensive for the requirements of this project.

3.4 Non-Technical Constraints

Due to the Covid situation it will not be possible to engage with the blind community therefore extra consideration will be given to the fact that as a seeing person the design cannot be taken as the best approach possible, ergo the solution will need to be created in a way that allows for upgrades / rewrites to be easily inserted in the future. Where possible, the individual components should be modularised. This built-in flexibility will help researchers who wish to take this solution forward in the future upon receiving feedback from the blind community.

4 Requirements

The most important aspect of any software project is identifying and recording a concrete set of requirements that contain enough high level information to provide a good direction for the design of the solution. The requirements provide the detail behind the scope. They should inform the development team of what needs to be achieved and what functionality must be supplied to the user but stops short of telling the development team how to implement the solution from a technical perspective. The requirements are broken down into functional and non-functional requirements.

The requirements have deliberately been left as high-level as possible to minimise constraints on the creative process as the project objective is to provide a proof of concept solution, not a final product.

4.1 Functional Requirements

The functional requirements describe how the system must respond to user inputs, the actions that the system must take place and define the desired outputs. The functional requirements are defined below along with the prefix GH (Guiding Hand):

Req.	Description
GH_001	The user must be able to verbally interact with a personal assistant.
GH_002	The personal assistant must be able to understand and respond to the user's commands.
GH_003	The user must be able to wake the personal assistant with a keyword.
GH_004	The user must be able to send the personal assistant to sleep
GH_005	The personal assistant must be able to detect objects from a wearable or static camera device.
GH_006	The personal assistant must be able to describe the detected objects to the user.
GH_007	The user must be able to verbally select an object from the list of detected objects.
GH_008	The personal assistant must be able to track the selected object if the camera frame moves or if the object itself moves.
GH_009	The personal assistant must be able to detect the user's hand when it enters the field of view.
GH_010	The personal assistant must be able to provide directional assistance to the user in order to move the user's hand towards the selected object.

Table 6: List of functional requirements describing the required product features and behaviours

4.2 Non-Functional Requirements

The non-functional requirements focus on the requirements of how the system should operate in terms of its performance, usability, and scalability. The requirements are once again prefixed with GH (Guiding Hand) and start at a different number (100) to distinguish themselves.

Req.	Description
GH_101	The system should maintain as high a frame rate as possible to ensure the smooth operation of the object detection and tracking algorithms.
GH_102	The system should provide clear directional feedback in a manner that is as intuitive as possible to a blind user.
GH_103	The system should take care to calculate the distance from the hand to the object in a manner that facilitates the ability to grasp the object and not jab at it with the leading edge of the hand.
GH_104	The user will require the object detection algorithm to be fast enough to operate in real-time but accurate enough that a blind person can have trust in its operation.
GH_105	The personal assistant should be easy to use and sympathetic to provide dignity to the user.
GH_106	The system should be designed in a way that allows functionality to be added, removed, or enhanced in the simplest way possible as a future development platform.
GH_107	Important parameters should not be hard coded but stored in a configuration file where possible to maximise system flexibility and readability.

Table 7: List of non-functional requirements describing the how the system should operate

4.3 Use cases

Some simple use cases are defined to aid development and provide a clearer picture of how the system will be used. The use cases will be leveraged for testing purposes.

4.3.1 Waking the Personal Assistant

- The user will activate the personal assistant with a trigger word similar to “Hey Google”
- The personal assistant will greet the user in the style that blind prefer to be alerted to someone’s presence, usually by saying hello and telling the user who is speaking.

4.3.2 Sending the Personal Assistant to Sleep

- The user will send the personal assistant to sleep (background listening) with a trigger word such as “sleep”.
- The personal assistant confirms they are going to sleep and inform the user to how wake them.

4.3.3 Exit the program

- The user will exit the program with a trigger word such as “exit”.
- The personal assistant confirms they are exiting the program.

4.3.4 Detecting Objects

- Once the personal assistant is awake, the user will ask the personal assistant to scan the camera's field of vision with a trigger word such as "scan".
- The object detection algorithm will perform a detection.
- The personal assistant will store the results and tell the user what was detected.
- The personal assistant will tell the user that they can select an object from the list of detections.

4.3.5 Selecting Objects

- The user will select an object for retrieval by using a trigger work such as "select" and then naming the object.
- If there are no objects detected, the personal assistant will tell the user that they must detect objects first.
- If objects are stored in memory, then the selected object will be selected.
- The personal assistance passes the selected object to the guidance system and informs the user that once their hand is detected directions to the object will be given.
- Control passes to the guidance system.

4.3.6 Guidance System

- The guidance system will calculate the distance from each of the 21 hand landmarks detected by the Hand Tracker to the centroid of the selected object.
- The landmark with the closest distance will be passed to the guidance functions.
- Should the object or camera feed be in motion then the object will be tracked in the frame.
- The personal assistant will guide the user's hand either by voice or vibration (as configured).
- The directional feedback will take the form of 8 cardinal points converted to a user friendly format (up, up-right, right, down-right, down, down-left, left, up-left). Up and Down may be replaced with Forwards and Backwards for usability purposes.
- The personal assistant will direct the user either by voice command or by the Computer Science glove containing vibration pads.
- Once the user's hand has reached the selected object the personal assistant will announce that the target has been acquired.

4.3.7 Tracking an Object

- The user has already detected objects and selected one for retrieval with the help of the guidance system.
- An independent actor will move the object within the camera's field of vision.
- The personal assistant will track the object and provide guidance to the user in relation to object's position at that time as per the previous use case.
- Once the user's hand has reached the selected object the personal assistant will announce that the target has been acquired.

Note: *It is acknowledged that this scenario is unlikely to happen in the real life context of reaching for an object as it would be a pretty cruel person who deliberately moves an object out of reach of a blind person. What is achieves however is proof of concept that the solution could be extended to track and guide an individual to a moving object.*

5 Design Concepts

5.1 Principles of Design Thinking

A topic such as this carries an inherent risk of falling into the trap of thinking like a person with sight. To ensure the requirements of assisting a visually challenged person can be adequately met, it is critical to be able to think like a blind person or at least understand how they perceive the world around them. This is one of the core concepts of design thinking [62]. For a sighted person, grasping an object on a table is a task so simple that you would not even consciously think about your actions. To develop an AI driven solution for the blind and visually impaired, every step must be carefully considered from their point of view. To this end, Schinazi et al. [63] noted that there are many technological solutions that are not used by blind people as the underlying research was constrained by forcing sighted and blind people to use the same spatial awareness strategies for comparative purposes.

The lesson taken from this study and design methodology is that the solution will need to be crafted in a way that allows for individual components to be added or amended without causing a costly failure in the codebase should future testing with members of the blind community reveal better ways of addressing the problem. To achieve this the aim will be to modularise the solution as much as possible.

5.2 Modularised Design

Building a solution that merges together a number of typically disparate AI applications whilst providing the flexibility to facilitate future development and re-work is challenging and complex. The agile framework that was chosen gives a way to manage this complexity and allows the individual components to be built and tested in isolation through a series of short sprints.

Even with a modularised approach there will still be a steep learning curve in creating a solution that can be fully integrated. Most tutorials and classroom sessions focus on providing the code or teaching how to code these applications as standalone units. For this project to be successful this standalone code knowledge will need to be transformed into a series of modules that share information in an efficient way if the goal of achieving a high frame rate is to be achieved in a sub-optimal environment.

The modularised design also allows for a series of skeleton designs to be created to ensure that the main solution file can call each individual component successfully before getting into the hardcore work of implementing a specialised solution over that skeleton. The basic programmatic flow of the solution is given below with an easily identifiable template for the proposed modular approach:

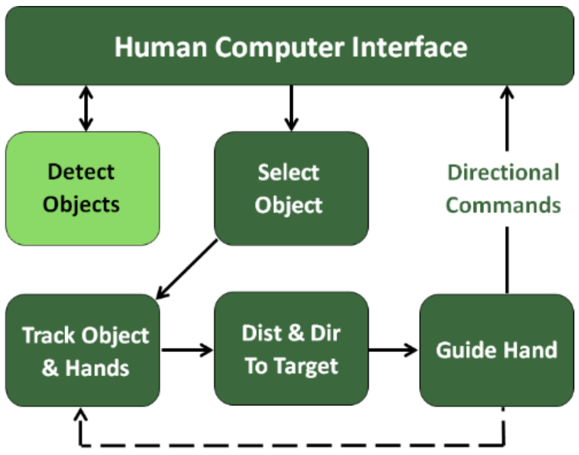


Figure 10: Basic programmatic flow of the solution encompassing a modular design

5.3 High Level Architecture

Having established that a modularised approach will be required, the following diagram shows how the individual AI components are envisaged to interact within the solution:

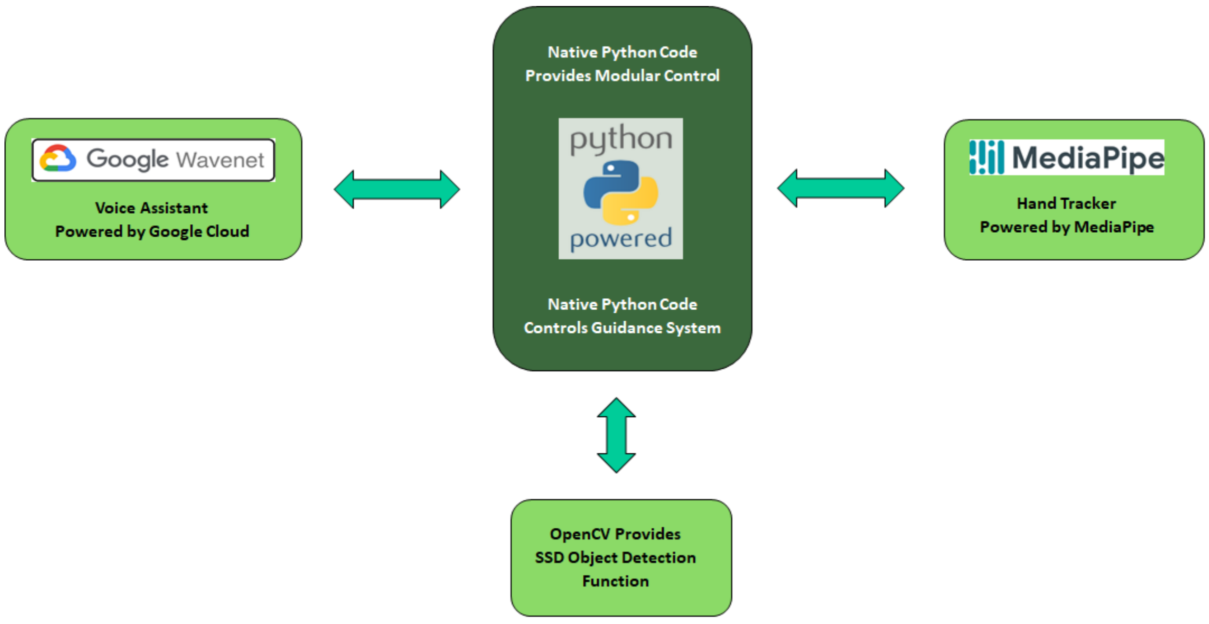


Figure 11: High level architecture components of the proposed solution

A more detailed control flow diagram is provided in section 6.7 of the implementation section where the solution has been fully fleshed out against this high level design architecture.

5.4 Solution Design Decisions

Due to the experimental nature of the design, the decision making process behind each component will be described here, with the lower level details provided in the implementation section.

5.4.1 Configuration File

Building flexibility into a solution often requires the use of a large number of parameters that can be tweaked as and when required. To facilitate any potential requirements to use frequently accessed parameters, a configuration file will be created and used to store any configurable parameters that might benefit from being stored in a such a configuration file.

5.4.2 Main Solution File

The main solution will house the bulk of the control code. Each specific AI component will exist in its own file as per the modularised design requirement. The keyword statements that are used to control the flow of the program will be housed in this file along with the code to control how the individual AI components are called and interact. There may be more efficient and elegant ways to abstract the code further but as the solution is intended as a future research platform it will make more sense to keep some of the logic visible here to facilitate ease of understanding and to encourage developers to consider both the introduction and implications of further abstractions.

5.4.3 Human Computer Interface

The voice assistant module will be powered by Google Cloud for speech synthesis and "speech recognition" for speech recognition. The Google Cloud provides one of the most advanced voice platforms available today and is free. Thought was given to using Pocket Sphinx as an offline solution but it was found to be very difficult to explore its suitability due to installation and code issues. The decision was taken early on to stick with Google for reliability reasons as well as the fact that the voice module whilst being important, is still a supporting facet of this project. It is not the key concern.

5.4.4 Camera Feed

Four options were enlisted for trial in the design of the camera feed. Firstly, the laptop's webcam and a standalone HD webcam designed for streaming were tested. Both of these proved excellent options for development and suitable for use in the proof of concept. The Xbox Connect 360 camera was not compatible with the Windows 10 operating system which put an end to hopes of a realistic 3D solution being developed as part of the design. A pair of low budget wearable spy glasses were purchased but these proved to be unable to support the streaming requirements needed for the solution which was unfortunate as they would have been a good proxy for the OrCam and Envision products.

Stereovision glasses were looked at but these came with either a high price tag or were configured for use with a Raspberry Pi as part of robotics solutions and were therefore unsuitable. Another option would have been to build a crude home-made stereovision system using two webcams attached by a solid length of tubing. The mathematics behind this style of solution was definitely achievable however the build and calibration requirements would have been extremely intensive and quite possibly a project in its own right. It was decided that the project was already sufficiently complex without adding another layer on top and that a workable 2D solution could be extended to a 3D solution as part of a future development without troubling the modularised codebase.

5.4.5 Object Detection

An exploratory attempt at using MediaPipe's 3D object detection was made to see if synergies could be gained from using the same hand tracking and object detection source. The 3D object detection was found to be unsuitable for this project as it yielded 3-4 FPS in isolation and even less when combined with the hand tracking (0-1 FPS). At such low speeds the object detections were poor to non-existent and it was evidently clear that one of the traditional models would be required.

Single Shot Detection (SSD) MobileNet was the preferred model for the object detection algorithm. SSD is a popular and proven choice for real-time object detection and provides a balance between speed and accuracy. From a user perspective (being blind) accuracy is an important factor when considering which model would be best to use. From a technical perspective it is required that the model be as fast as can be expected without sacrificing accuracy as it has to work in harmony with a number of other components. These are the main reasons why SSD was chosen over its faster rival YOLO, and its more accurate but slower rival Faster-RCNN. OpenCV also provides a function call that can be used in conjunction with the SSD model giving an easy way to implement this form of object detection. One other consideration was that the development laptop is under-powered for AI and is likely to provide a comparable level of processing to a modern wearable device. Deploying a super-fast detection algorithm like YOLO would mean carrying a larger architecture than would be required for SSD MobileNet, which was specifically developed for mobile and embedded vision applications.

Another benefit of using the SSD MobileNet model is that it is trained on the popular coco dataset. This dataset contains 91 everyday objects and is perfect for use both as a proof of concept for detecting and retrieving nearby objects, and for use in future iterations where the scope of the guidance system may be reduced, extended to navigating a room or street, or enhanced with transfer learning.

5.4.6 Object Selection

Object selection sounds simple but is slightly more complex than it appears at first glance and is actually a lynchpin component of the entire solution. Having informed the user of the detected objects, the personal assistant will be required to extract the correct object from the user's response when hearing the "select" trigger word. The selected object and its bounding box information will then be made available to the tracking and guidance algorithms so they have a target to work with. This may or may not be implemented as a standalone module, trials will need to be conducted to discover the best method for implementing object selection.

5.4.7 Hand Tracking

Initially the ability to adequately track the user's hand was one of the greatest areas of concern and MediaPipe's unique hand tracking algorithm was an easy decision and a Godsend. Whilst the virtues have already been discussed in the literature review it is worth re-iterating that the ability to calculate the distance to the object from all 21 hand landmarks is invaluable for experimenting with a number of guidance solutions. There is also the option to include up to 4 hands although this could get confusing and add extra layers of complexity in determining whose hand is being detected so will likely be avoided through use of a configurable parameter. MediaPipe provide two options for deploying their code. One is through procedural code provided in a number of standard languages (Python, C++, Java) and the second through the use of graphs. MediaPipe state that learning how to graph efficiently would take approximately 1 month of study. The decision was therefore taken to use the standard code provided as no guarantee could be given that the graph solution would be more efficient, faster, or even successfully learnt to the required standard within an acceptable timeframe.

5.4.8 Object Tracking

Object tracking will form part of the guidance system module and can be covered through use of the standard trackers provided by OpenCV. Options were available to build a custom tracker but as this is a fringe use case and not a key concern it was decided to trial a number of standard trackers and pick one that worked best in the context of the requirements. The tracker will simply take the selected object that is passed to the guidance system and track any movement of its bounding box. It is likely that this will be part of the camera feed component.

5.4.9 Guidance System

The key concern of this project is the guidance system. This will be built entirely by the developer and will utilise data obtained from the object detections, object selection, hand and object tracking algorithms. The solution will focus on a 2D scenario that calculates the Euclidean distance from each of the 21 hand landmarks to the centroid of the object being tracked (selected object). The solution will be configured to calculate the distance from the closest hand landmark to the object.

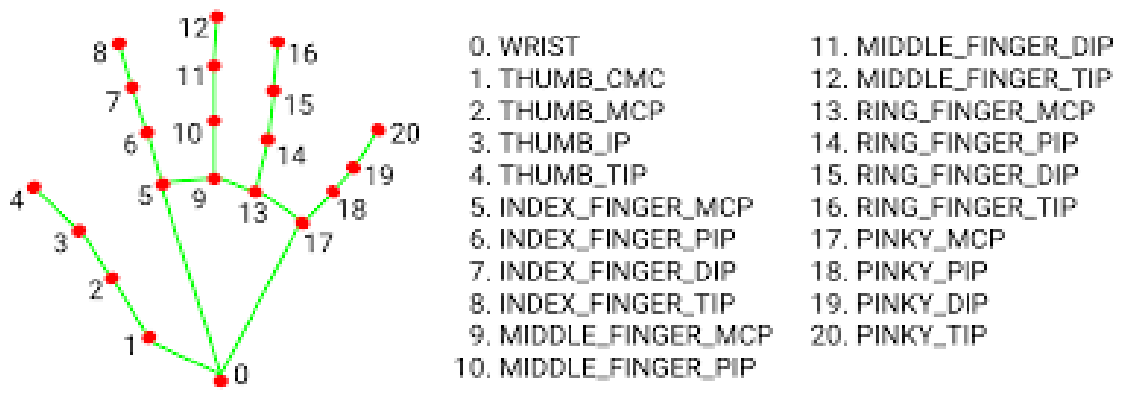


Figure 12: MediaPipe hand tracker provides 21 hand landmarks for innovative solutions

A configurable tolerance parameter will be added so that the distance calculation can be optimised during testing if required. The reason for this is that it may be desirable to guide the hand to a point just before the centroid or just past the centroid to facilitate a smoother retrieval of the object. Bear in mind that future researchers may prefer to use a specific hand landmark instead of the closest one. For example, if using one of the interior hand landmarks, the tips of the fingers would reach the target in advance of the palm so appropriate use of the tolerance parameter allows the distance to be adjusted so the user doesn't blindly knock the object over potentially spilling a hot drink over their fingers.

The (x, y) co-ordinates of the closest hand landmark and the target centroid will be analysed to provide a direction in which to move the user's hand. Once the directional requirements have been identified they will be fed back to the user via voice command, via the CS glove containing vibration pads or a proximity based beep with variable frequency enabling the user to move closer to the object and hopefully retrieve it successfully.

5.5 Functional Requirements Matrix

With the architecture and components conceptually defined at a high level, it is prudent to ensure that all of the functional requirements are covered against at least one of the planned components:

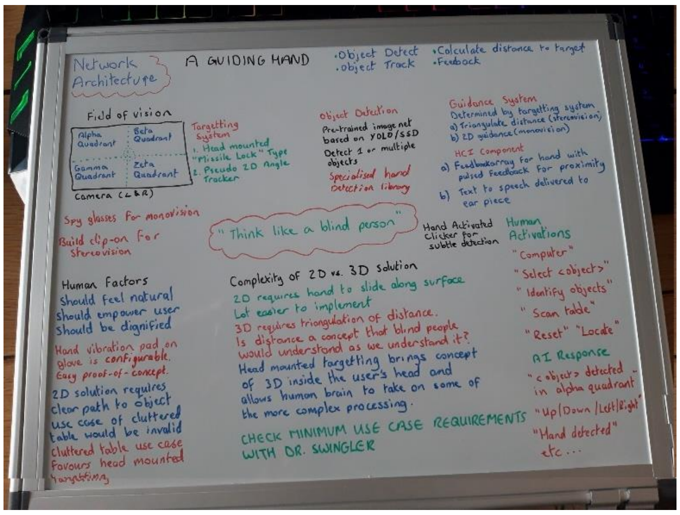
Req.	Short Description	Covered?
GH_001	Verbally interact with a personal assistant.	Human Computer Interface (HCI)
GH_002	Understand and respond to the user's commands.	HCI
GH_003	Wake the personal assistant.	HCI
GH_004	Send the personal assistant to sleep.	HCI
GH_005	Detect objects from a wearable or static camera device.	Camera Feed, Object Detection
GH_006	Describe the detected objects to the user.	Object Detection, HCI
GH_007	Select an object from the list of detected objects.	Object Selection, HCI
GH_008	Track the selected object	Object Tracking
GH_009	Detect the user's hand.	Hand Tracking
GH_010	Provide directional assistance to the user.	Guidance System, HCI

Table 8: Functional Requirements Matrix to ensure requirements are considered in the design

For non-functional requirements, these will largely be satisfied by the performance of the system and will be used to drive the implementation decisions based upon the results of the implementation trials.

5.6 Brain Storming

The last task before writing experimental implementation code was to fill a whiteboard with "blue sky thinking" ideas to gain some inspiration about where to start and be cognisant of potential pitfalls:



Network Architecture
A GUIDING HAND
•Object Detect
•object Track
•Calculate distance to target
•Feedback
Field of vision
Alpha Quadrant
Beta Quadrant
Gamma Quadrant
Zeta Quadrant
Camera (L&R)
Targetting System
1. Head mounted "Missile Lock" Type
2. Pseudo 2D Angle Tracker
object Detection
Pre-trained image net based on YOLO/SSD
Detect 1 or multiple objects
Specialised hand Detection library
Guidance System
Determined by targetting system
a) Triangulate distance (stereovision)
b) 2D guidance (monovision)
HCI Component
a) Feedbackarray for hand with pulsed feedback for proximity
b) Text to speech delivered to ear piece
Spy glasses for monovision
Build clip-on for stereovision
"Think like a blind person"
Hand Activated Clicker for subtle detection
Human Activations
"Computer"
"Select <objects>"
"Identify objects"
"Scan table"
"Reset" "Locate"
AI Response
"<object> detected - in alpha quadrant"
"Up/Down/Left/Right"
"Hand detected"
etc...
Human Factors
Should feel natural
Should empower user
Should be dignified
Hand vibration pad on glove is configurable. Easy proof-of-concept.
2D solution requires clear path to object use case of cluttered table would be invalid
Cluttered table use case favours head mounted targetting
Complexity of 2D vs. 3D Solution
2D requires hand to slide along surface
Lot easier to implement
3D requires triangulation of distance.
Is distance a concept that blind people would understand as we understand it?
Head mounted targetting brings concept of 3D inside the user's head and allows human brain to take on some of the more complex processing.
CHECK MINIMUM USE CASE REQUIREMENTS WITH DR. SWINGLER

Figure 13: Whiteboard exercise to stimulate creativity through "blue sky thinking"

6 Implementation

Implementing the design concepts required an iterative approach, flexibility and some trial and error in order to get the best solution. The complexity of the project necessitated a massive learning curve and what follows is a description of the implementation, important issues and decisions that were made during the development of a solution that would meet all the identified requirements.

6.1 Development Environment

6.1.1 Laptop Specification

The details of the development environment are given below:

Item	Description
Make	HP Pavilion 15 Notebook
OS	Windows 10 Home on 64-bit operating system, x64 based processor
Processor	AMD A8-6410 APU with AMD Radeon A5 Graphics (2.00GHz)
Memory	8GB installed with 6.94GB usable

Table 9: Recording the specification of the development laptop to give context for performance

6.1.2 Programming Language

Python was selected as the programming language due to its well-known reputation as a preferred language for fast, iterative development. Python was the primary language of the MSc and matches the developer's skillset making it the natural choice. Consideration was given to using C++ with the concerns of achieving a high frame rate with the sub optimal development laptop. The decision was taken not to use C++ due to the fact that it does not facilitate fast, iterative development owing to the complexity of the language and its closeness to the operating system.

6.1.3 Python IDE and Version

The IDE of choice for this project was PyCharm Community Edition 2021.1.1 x64. PyCharm was chosen over Jupyter Notebook for several practical reasons including using a professional standard IDE with access to debug tools, the python terminal and console as well as the ability to easily create a modular design solution. PyCharm installed with version 3.7 but upgraded to version 3.9 during development.

Note: *The code is written and formatted to match the PEP8 checking criteria contained with PyCharm. All warnings were heeded and code reworked until the little green tick appears in the top right of the IDE signifying that the code conforms to PEP8 standards.*

6.1.4 Hardware Requirements

Four items of hardware were required for this solution and are described below. Only two of the four items made the final solution:

Item	Description
HD Webcam	Used to mimic a wearable device. Can be static or used with head attachment.
Computer Science Vibrating Glove	Wearable glove containing 4 vibrating pads controlled by Bluetooth for directional feedback to the user. Pads are positioned around the wrist for directional guidance.
Xbox Connect 360	RGBD camera for facilitating 3D guidance. Not compatible with Windows 10.
Wearable Spy Glasses	Low budget glasses purchased however they did not work with video streaming.

Table 10: Hardware trials for the proposed solution

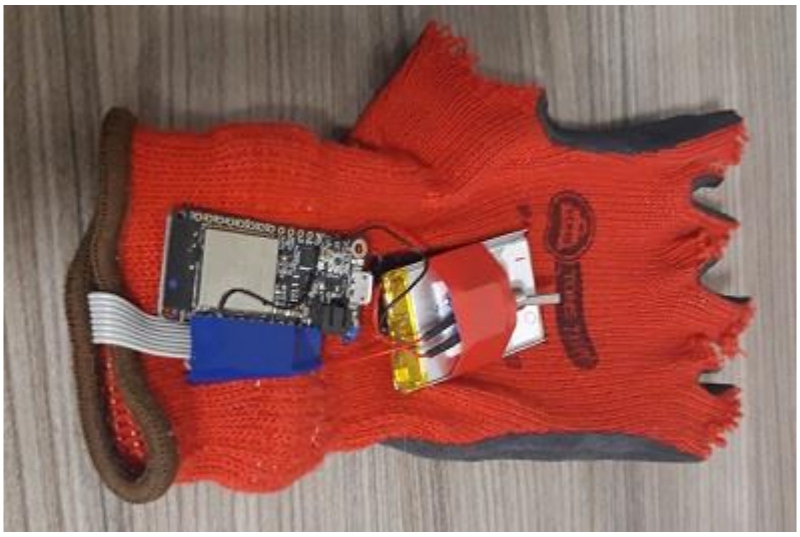


Figure 14: The Computer Science Vibrating Glove connects to the solution via Bluetooth

6.2 Required Python Libraries

The solution uses 13 python libraries with 5 of them pre-installed alongside the python language and can be seen in Appendix A. Two complications arose during the OpenCV version selection that caused some issues of a minor nature:

- Several trackers had been descoped from the latest versions of OpenCV whereas the SSD detection function was not available in early versions of OpenCV. It took some trial and error to find an OpenCV version that had both the desired tracker and SSD object detection function.
- Changing the OpenCV version led to a slight complication with some of the variable data types changing from tuple to list. Although this wasn't a big issue it did highlight an unexpected complication that was initially difficult to track down due to the sub-optimal nature of Python error messages, further validating the inclusion of comprehensive console printing.

6.3 Procedural vs. Object Oriented Programming

The implementation began with procedural code however it became apparent that this would lead to some major problems:

- The codebase would be large and unwieldly making it difficult to read.
- Concerns over the size of data being passed to and returned from function calls
- It proved difficult to pass variables between the different scopes within the codebase leading to a large number of variables having to be initialised in advance of being used. This is generally not considered to be best practice.

The realisation that the code needed an object oriented solution occurred during the early stages of module integration. The re-work required was very manageable and minimally disruptive due to the modular approach taken. The coding changes and testing was able to take place in isolation as per the agile methodology before being made available to the main solution file.

The creation of specific objects (detailed later) allowed certain key data to be stored and retrieved without the need to pass a large number of variables into functions and receive an equally large number of return variables. It also negated the problems associated with variables not being available outside of the inner scopes as well as the tidiness of not being required to initialise a mass of empty variables at the outermost scope level.

The abstraction of code into objects made the code much more readable and allowed for extra object attributes and methods to be added and removed as required during the iterative build process with minimal reworking of existing code. This greatly simplified the codebase and made It far more readable for future developers to pick up and understand.

6.4 Threading vs. Multi-Processing

One of the biggest problems faced was the integration of the cam feed and the voice assistant. Much of this was due to inexperience with threading and the integration of sound and vision. Initial attempts at this integration resulting in failure with the video frame processing freezing whilst the voice assistant was speaking. This is due to the way that python interacts with the Global Interpreter Lock (GIL).

To get around the issues with the GIL, both threading and multi-processing were considered as the laptop has a quad core processor meaning that multi-process solutions were possible. Research into the possible solutions showed that threading was the best solution in this instance due to the shared memory of threading. With multi-processing taking place on separate cores there is no shared memory meaning that the overhead of setting up a queue to transfer data between the cores during function calls would be contrary to the goal of achieving the high frames required (processing time).

The next challenge involved deciding whether to place the cam feed or the voice assistance on the main thread and which to run on the secondary thread. Several experiments to create the code from scratch failed and it was decided in the interest of progress to utilise code from a tutorial that placed both frame loading and processing on separate threads. This had the desired effect and the solution was able to run smoothly as intended. The details will be elaborated upon later in this section.

It is acknowledged that it is likely there are better ways to achieve this however due to the risk of derailing the entire project the decision was taken that progress was more important and a smooth sound and vision integration was paramount to the overall success of the project.

6.5 Achieving High Frame Rates

It was known at the outset of this project that achieving 30 frames per second (FPS) would be impossible given the development laptop specification. It was crucial to test each component in isolation to see where the baseline was for the different components. The voice module was not an issue as the video and speech modules were designed to run on separate threads so that the video frames would load and process continuously as the personal assistant was speaking.

Component	Achieved FPS
OpenCV Standard Baseline (just video)	The baseline frame rate from streaming a standard OpenCV video frame was clocked at 20 FPS. There is an unavoidable bottleneck with the cv2.VideoCapture() function.
Google MediaPipe Hand Tracker	Overlaying the baseline with the Google recommended codebase for the hand tracker resulted in 8-9 FPS.
Google MediaPipe 3D Object Detection	Overlaying the baseline with the Google recommended codebase for the 3D object tracker took the frame rate down to 1-2 FPS and was dismissed as a viable option.
OpenCV SSD Object Detection	Overlaying the baseline with the standard SSD detection codebase yielded a frame rate of 9-10 FPS. Talk about best practice below.
Object Tracker	Integrated last by design, the mosse tracker had a negligible impact on the frame rates achieved by optimising the codebase for the previous components.

Table 11: The baselining of frame rates guided development as the module integration increased

These results gave significant cause for concern as the frame rates were dangerously low in isolation and were not going to improve as each solution layer was added. The first iteration of each module used the recommended standardised code and it was clear from visual inspection that the code was basic and not optimised for speed with multiple occurrences of nested for loops and no pythonic interpretations of the code. Another problem was the realisation that the growing number of variables required to be passed to and from function calls was growing and creating a large memory overhead that was particularly inefficient in terms of heap memory.

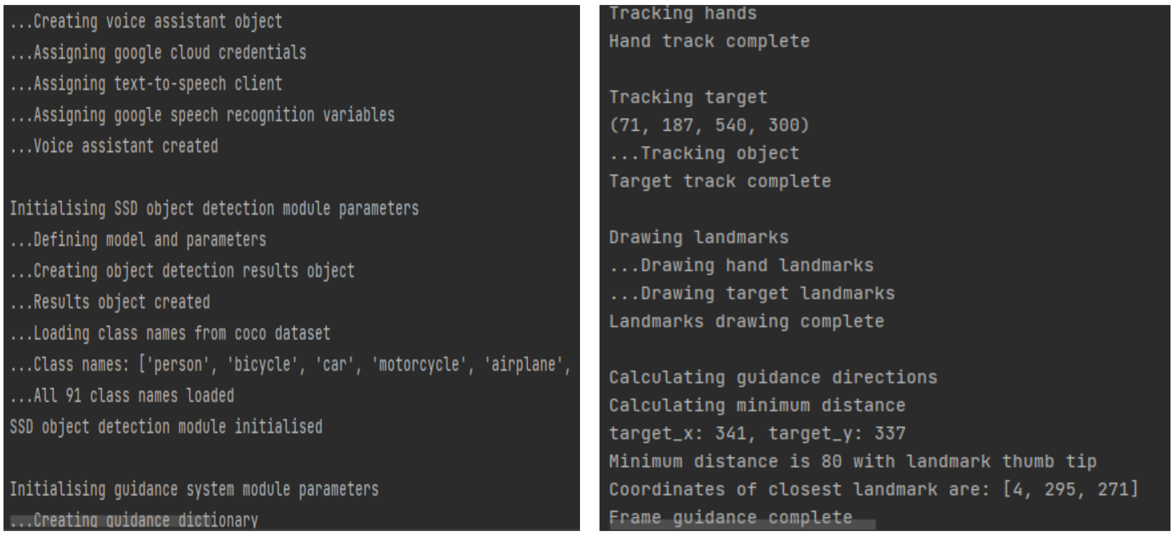
The solution was two-fold and partially unexpected. Converting the procedural code to being object oriented had the benefit of speeding up the code dramatically. The hand tracker for example jumped from 8-9 FP to 14 FPS when converted to an object. Fewer function calls, less data being passed and less variables being declared on each frame facilitated a higher throughput leading to the observed increases on base frame rate. A summary of optimisations is given:

- Converting code to objects enables the data associated with the object to be stored efficiently and was accessible across multiple functions and modules which reduced processing overhead.
- Using objects meant that many of the for loops in the standardised code could be avoided as the standardised code was split into separate functions to be called only when absolutely necessary.
- With the video frames being converted into objects as part of the threading solution the passing of frames between functions was greatly reduced and as frames are data intensive this enabled an even more efficient codebase.
- Experimentation with Nth frame processing further increased the frame rates and culminated in a very acceptable 19-20 FPS by the end of development depending on background laptop processes. With a high, head mounted camera a remarkable 23-24 FPS was noted on occasion.

6.6 Console Logging

Due to the complexity of the project, it was decided to implement console logging as a means to keep track of the development and quickly highlight where bugs were occurring in the code. The console logging is extensive and provides a play-by-play commentary on every step of the solution and is an invaluable tool allowing verification of the process and rapid discovery of issues along with the capability to insert extra logging measures as required at the precise point in the code that they are needed. It was especially useful in testing that the targeting system was operating as intended and the intense level of logging detail has been retained for future verification purposes.

The following screenshots show how the logging details every aspect of the code flow making it easy to pinpoint the line of code that might be failing, or to verify the accuracy of the guidance system calculations:



```
...Creating voice assistant object
...Assigning google cloud credentials
...Assigning text-to-speech client
...Assigning google speech recognition variables
...Voice assistant created

Initialising SSD object detection module parameters
...Defining model and parameters
...Creating object detection results object
...Results object created
...Loading class names from coco dataset
...Class names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane',
...All 91 class names loaded
SSD object detection module initialised

Initialising guidance system module parameters
...Creating guidance dictionary
```

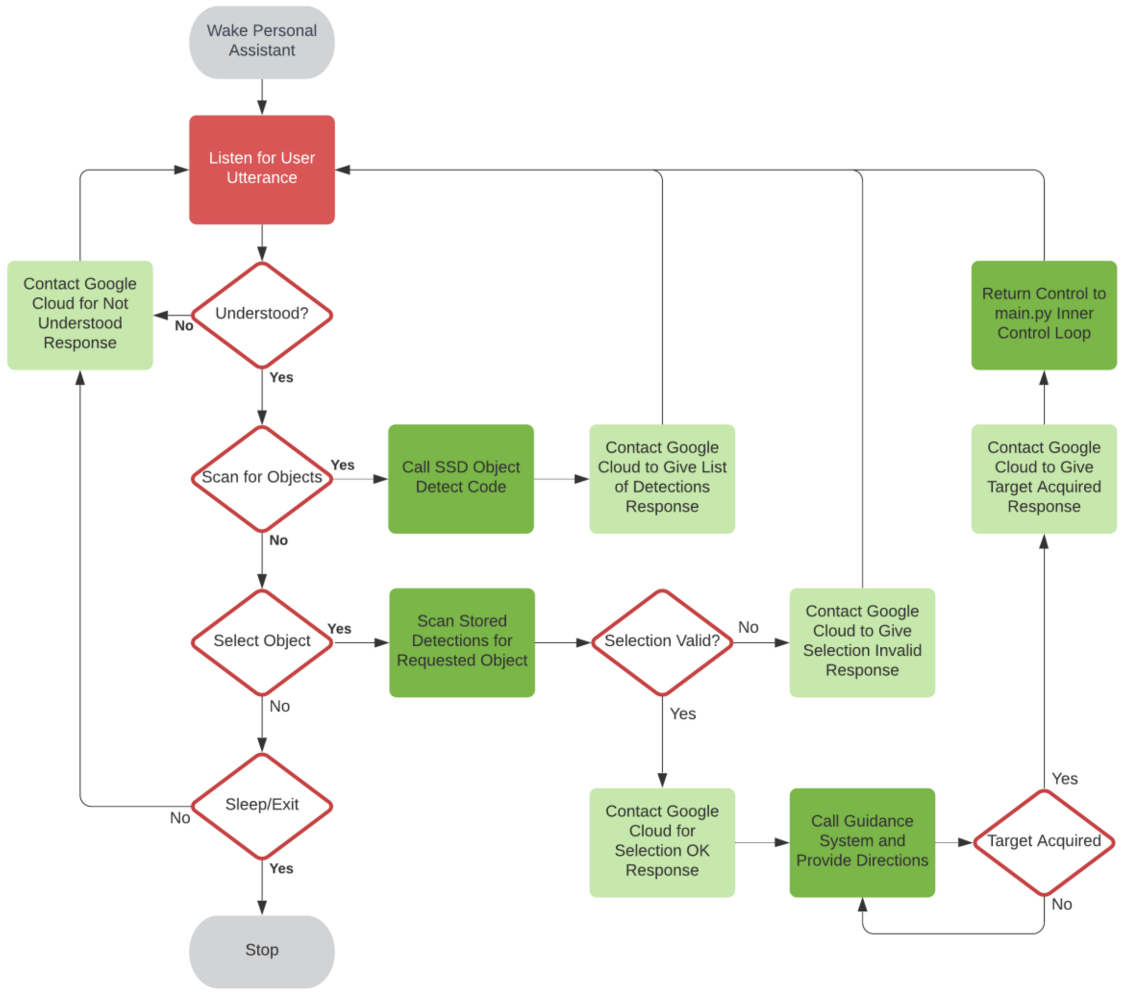
```
Tracking hands
Hand track complete

Tracking target
(71, 187, 540, 300)
...Tracking object
Target track complete

Drawing landmarks
...Drawing hand landmarks
...Drawing target landmarks
Landmarks drawing complete

Calculating guidance directions
Calculating minimum distance
target_x: 341, target_y: 337
Minimum distance is 80 with landmark thumb tip
Coordinates of closest landmark are: [4, 295, 271]
Frame guidance complete
```

Figure 15: Detailed console printing provides ease of error resolution and verifications

Please refer to Appendix D for a sample extract of the console logging.

6.7 Control Flow

The high level flow of the main.py inner control loop is defined below. Note that the "Introduce Yourself" decision node is not included as this was provided for demonstration purposes only.



Wake Personal Assistant
Listen for User Utterance
Understood?
No
Yes
Contact Google Cloud for Not Understood Response
Scan for Objects
Yes
No
Call SSD Object Detect Code
Contact Google Cloud to Give List of Detections Response
Select Object
Yes
No
Scan Stored Detections for Requested Object
Selection Valid?
No
Yes
Contact Google Cloud to Give Selection Invalid Response
Contact Google Cloud for Selection OK Response
Call Guidance System and Provide Directions
Target Acquired
Yes
No
Contact Google Cloud to Give Target Acquired Response
Return Control to main.py Inner Control Loop
Sleep/Exit
No
Yes
Stop

Figure 16: Control Flow of main.py inner control loop

6.8 Development Sequence

To fulfil the agile requirements of spring development the modules were created in the following order in isolation before being knitted together:

Order	Module	Order	Module
1	Voice assistant	4	Guidance system
2	Object detection	5	Video threading
3	Hand tracker	6	Main knitted solution

Table 12: Modular development sequence in according with agile sprint requirements

6.9 Solutions Files

There are 11 files in the final solution. This section describes how they operate.

6.9.1 config.py

The config file merely contains a list of parameters that can be set with greater ease than trawling through the modular code to find a specific parameter. The file is neatly separated into sections by module with comments to assist understanding the meaning of the parameters. The parameters included are simply the parameters that were frequently tweaked during iterative development. Please see Appendix B for the full list of parameters.

6.9.2 main.py

This is the main solution file. This code imports the modules, contains the control flow for the voice assistant and calls the relevant functions as required. A higher level of abstraction is no doubt possible but as previously mentioned parts of the code have been left simplistic for future use purposes.

The first major decision was how to structure the code. Would it revolve around a constant video feed being displayed or around the concept of the personal assistant? The decision was taken to base it around the personal assistant experience. The need for a constant video feed could be justified for demonstration purposes however in reality a blind user would not require this. The development needed to mimic a real world usage as closely as possible to prove the concept. It also had the added benefit of minimising the amount of time that the solution needed to rely on a high frame allowing non-guidance related tasks to be completed without this distraction.

As such, the creation of a personal assistant who could detect objects and relay this information to the user and interact with them for object selection was completed without the user seeing the detections on screen giving life to the design thinking concept of not solving this problem like a sighted person. The video is only seen once the selected object has been passed to the guidance system and the cam window is activated. This focussed development and increased the effectiveness of unit testing in the spirit of agile development.

(For reference, there is heavy use of printing to the console for visibility of what the code is doing throughout the solution at all times).

6.9.2.1 Code Flow

The code in this module flows through the following steps:

Step	Description
1	Import libraries and modules. Modules initialise relevant objects and connections upon import.
2	Check select guidance mode parameter is valid
3	Initialise camera parameters (source, height, width)
4	Initialise voice assistant object
5	Activate background listening for trigger word to awake

Step	Description
6	Once awake enter inner loop for user interactions (see below)
7	Upon exiting the inner loop, close glove connection, destroy objects & cam windows and exit.

Table 13: Step by step guide to the control flow of the main solution file

6.9.2.2 Inner Control Loop

The inner loop of the control flow is how the personal assistant controls the program. It is a simplistic loop that performs basic logic before calling other modules if required. This is where further small scale abstraction is possible but has been left here in an attempt to make the code more readable.

Any user utterances are captured by the voice assistant and compared against a dictionary of keywords. The keywords are embedded into an IF block that contains code to carry out the actions associated with the keywords. The keywords also guide the personal assistant's response to the user. The reasons for the keywork dictionary will be given in the voice assistant module description.

Keyword	Description
Introduce yourself	The personal assistant introduces themselves and gives a brief description of who they are and what tasks they are designed to complete.
Scan	Clears any previous detections from the detections object and performs a new detection on the cameras field of vision. Informs the user of what has been detected.
List	Repeats the list of stored detections. Created to demonstrate ability to retain information and highlight the advantages of using OOP solutions.
Select	Checks that detections are in memory before proceeding. The logic checks that only 1 selection has been made otherwise informs user to only make 1 selection. Each detection in memory is compared to the user utterance. If there is a match then the object is selected. For example, a cell-phone is in memory and the user says, "Please select the cell-phone". As this is a match the cell-phone is selected and the personal assistant passes control to the guidance system. This brings up a cam feed with all the targeting information overlaid on screen as expected for the core part of the project.
Awake	The user can check if the personal assistant is still awake.
Sleep	The user can send the personal assistant to sleep (listen in background mode).
Exit	The user exits the program.
{else:}	Not a keyword but contains code to inform the user that the personal assistant did not understand the user utterance.

Table 14: Functionality currently provided by the inner control loop

Note: *Whilst it would have been more accurate to record the keywords with a space before and after to indicate exact words in the search string (" scan " and not "scan") it would decided not to do this due to issues with the voice recognition. The no space solution meant that other words could be substituted for easier recognition (e.g., "scanner", "scandalous") where vocal clarity was a concern.*

6.9.3 voice_assistant.py

The voice assistant module contains a class object that creates the personal assistant. The assistant provides the ability to perform speech recognition and synthesis tasks to interact with the user.

6.9.3.1 Class Object: VoiceAssistant

The initialisation process creates a connection the Google Cloud TextToSpeech client (v1 – very import) using a dedicated API key stored in a JSON file as part of the solution package. It checks and removes any response file that might be present from previous incarnations and sets all the required parameters for providing speech synthesis and voice recognition capabilities. It also initialises the keyword response dictionary that controls user interactions. It performs microphone calibration for 2 seconds to account for any consistent background noise and sets the dynamic energy threshold property to true so that it constantly checks the energy levels for optimal voice recognition.

6.9.3.2 self.respond

Used to convert text inputs from the personal assistant's keyword dictionary into speech. Also takes an input variable dictating whether a separate thread is to be used (used in conjunction with the cam feed). The function sends the text via REST JSON to the google cloud service and receives an mp3 file of the response. The response is saved to the project directory, played to the user, and then deleted. The use of a response file was a convenient way to manage the issue of the guidance system providing voice feedback to the user. If there is a response file present the feedback is skipped and the personal assistant avoids forming a backlog queue of feedback which might also overwhelm the system and cause a fatal error.

6.9.3.3 self.thread

Initialises a new thread for the personal assistant to use for speaking without interrupting the video flow. Calls the self.speak function to deliver the response on the new thread.

6.9.3.4 self.speak

Plays the saved mp3 file received from the Google Cloud and then deletes from the OS. This function is called directly from self.respond if no theading, else it is called from self.thread when threading.

6.9.3.5 self.capture_input

Uses the speech recognition library to listen for user utterances. These are converted to text and then used in the main.py file to decide what actions the personal assistant needs to take.

6.9.4 object_detect_nms.py

This module initialises an object that provides a lot of functionality for object detection, selection, and tracking. The object stores all the necessary information in an object dedicated to object detection requirements thereby reducing overhead in managing and manipulating the object detection data. The following steps are executed when the module is imported:

Step	Description
1	Sets the standard recommended parameters for use with the OpenCV SSD object detection call.
2	Creates an object for managing the object detections and subsequent use of that data.
3	Loads and verifies the class names from the coco dataset, as used by the SSD model.

Table 15: Initialisation routine of the object detection module

6.9.4.1 Function: contains_word

Basic one line function to determine whether the object that the user wishes to select is available for selection. An attempt was made to convert this to a lambda function without success.

6.9.4.2 Class Object: Detections

Initialises all variables as blank lists and tuples for future use. Defines the OpenCV tracking algorithm to be used (Mosse tracker). Please see appendix C for a list of variables initialised in this object.

6.9.4.3 self.clear_previous_detections

Self-explanatory function that clears data related to previous detections.

6.9.4.4 self.get_detections

This function makes the SSD detections and performs some data type conversion to make the results usable. Then it performs non-max suppression to improve the results. Lastly it converts the IDs of the coco dataset into text format so the user can receive a human understandable list of detections.

6.9.4.5 self.draw_detections

Draws the bounding boxes and puts the labels on the detected objects. Redundant for the final solution but has been retained as the module allows for code testing in isolation should the detection and non-max suppression confidence thresholds need adjusting.

6.9.4.6 self.validate_object_selection

Validates that the user selected object exists as a single item. For example, if the user selected "person" but the detection has picked up two people if would throw an error (known limitation).

6.9.4.7 self.initialise_tracker

This function initialises the tracker and draws the bounding box around the user selected objected.

6.9.4.8 self.track

This function updates the location of the tracked object and provides an on-screen report of whether the update is successful ("Tracking") or whether the object track has been lost ("Tracking Lost").

6.9.4.9 self.draw_tracker

Draws the bounding box associated with the latest tracked location.

6.9.4.10 self.track_thread

Initialises a new thread and calls the self.track function. Placing the tracker on a separate thread allows the code to continue performing other guidance system calculations without impacting the frame rate.

6.9.4.11 Tracker Selection Trials

OpenCV comes with a number of pre-built object tracking algorithms. Three trackers were selected and trialled to see which one performed best in terms of both its ability to track and the resultant impact on the frame rate. The tracker was one of the last components built and was designed to track the selected object once the object detection had taken place.

Tracker	**Description**
BOOSTING (baseline reference)	This algorithm was not tested but is included to give a reference point for the other trackers. It is based on the same algorithm used to power the well-known Haar cascades and is over 10 years old. It is considered to be slow and not very good.
KCF	Kernelised Correlation Filters. It is faster than the baseline algorithm and had a minor impact on the overall frame rate but requires good visibility of the target object. Partial obstruction of the object did present a slight problem on occasion but this is suspected to be more of a frame rate issue than a tracking issue.
CSRT	Discriminative Correlation Filter. This tracked better than the KCF tracker with less failure however it had a high impact on the frame rate, rendering the solution unusable with the frame rate dropping to 3-4 FPS.
MOSSE **SELECTED TRACKER**	The mosse tracker is considered to be extremely fast but not quite as good at tracking as the other two trackers. Given that the tracker is used to track an object that a blind person would wish to retrieve the speed of the tracked object is not envisioned to be moving overly fast so this was not an issue during testing. There was no noticeable impact on the frame rate and was therefore considered to be the best choice for this project. It also handled partial occlusion of the object without issue.

Table 16: Three trackers were trialled to see which offered the best capability with minimum frame rate impact

6.9.5 hand_track.py

This module imports the MediaPipe library and initialises two variables that define the hand tracker as the required algorithm, and the drawing utilities package provided by MediaPipe were rendering the hand landmarks onto the frame.

6.9.5.1 Function: track_hand

Calls the hand track algorithm and returns the results to the calling code.

6.9.5.2 Function: draw_landmarks

Draws the landmarks on the identified hand and returns a list of landmarks whose co-ordinates on the cam feed have been converted from the MediaPipe normalised format into a usable pixel format based on the configurable height and width defined in the config file.

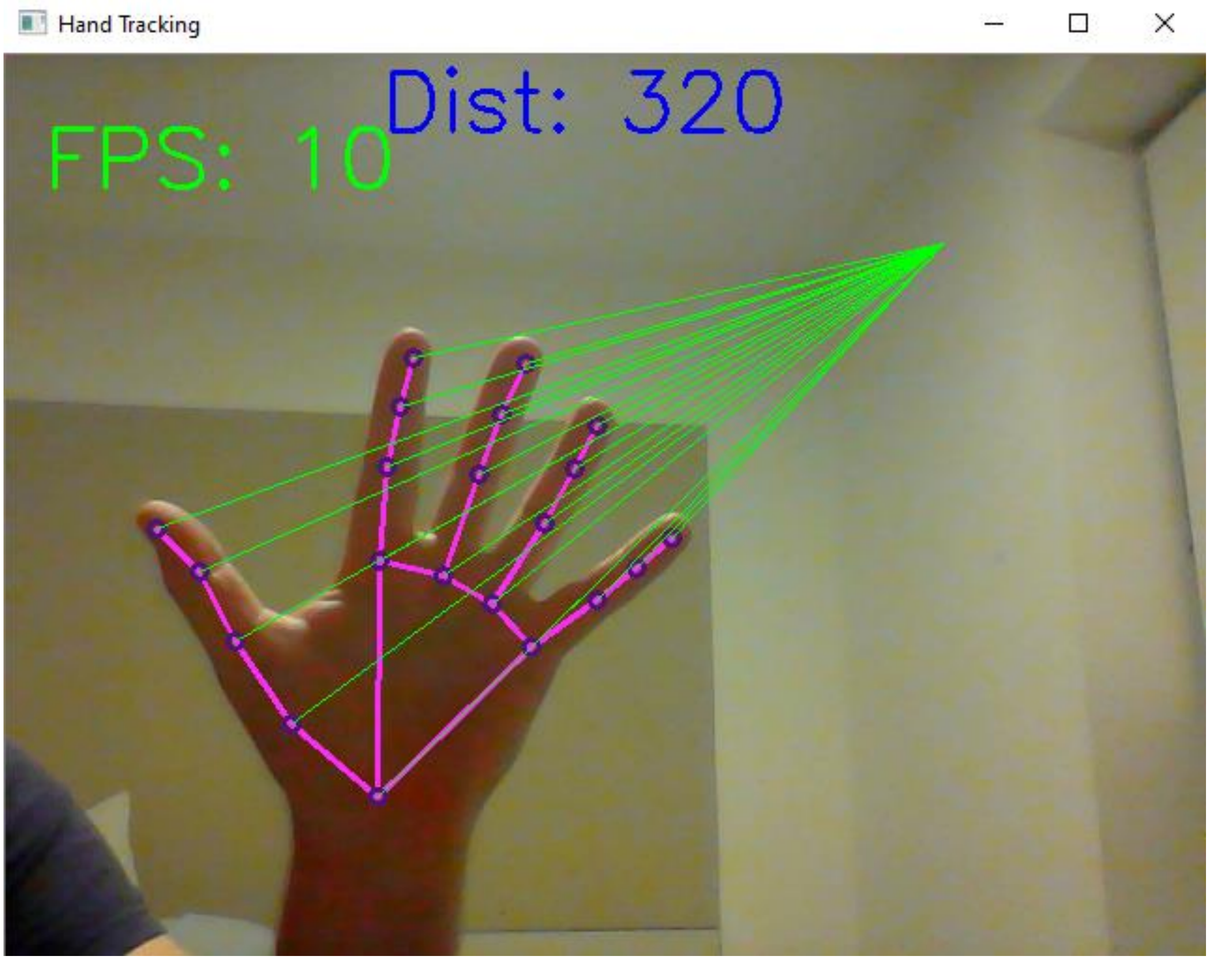


Figure 17: First experiment of standard hand tracker code to a static target during the initial sprint

6.9.6 video_thread.py

This module contains three class objects with two of them threaded. The code was taken from an online tutorial as referenced in the attestation section. This was done as it solved a major problem that threatened progress on completing the project.

6.9.6.1 Function: assign_base_cam

Function to initialise the basic video capture settings. Used as original background settings are wiped during guidance system video threading and is called from main.py on two occasions.

6.9.6.2 Class Object: CountsPerSec

Simple object to store and monitor frame rates. Largely untouched from the tutorial that the code was sourced from.

6.9.6.3 self.start

Records the starting time.

6.9.6.4 self.increment

Increments the frame count.

6.9.6.5 self.fps

Calculates the frame rate in frames per second (FPS).

6.9.6.6 Function: put_iterations_per_sec

Function to draw the frame rate on the cam feed.

6.9.6.7 Class Object: VideoShow

Initialises an object that is used for controlling the cam feed thread during multi-threading.

6.9.6.8 self.start

Starts a new thread for the cam feed.

6.9.6.9 self.show

Shows the current frame as long as the object is not marked as stopped.

6.9.6.10 self.stop

Sets the flag to stop the cam feed.

6.9.6.11 Function: thread_video_show

Control function to start the cam feed thread and control the contents of the frame. The code starts with the necessary threading setup from the tutorial but then moves on to house the control code for the guidance system. Further abstraction may be possible but once again it is left in a simplified format for readability and future use. The following steps are contained within this function:

Step	Description
1	Initialise the threading and set up the variables for displaying the cam feed.
2	On first frame initialise the mosse tracker within the object detections object.
3	Detect hands on every second frame (or Nth frame) due to slow laptop speed.
4	Call the mosse tracking thread to track the selected object. If speed is an issue this can also be pasted into the above step to detect every second (or Nth frame).
5	Call drawing functions to draw the detected hand landmarks and tracked object.
6	Call guidance system functions if hand landmarks have been detected (i.e., only guide the hand if the hand has been identified and is being actively tracked).
6a	Calculate the distance of the hand landmarks to the target centroid and identify which is closest.
6b	Call the guidance feedback function every Nth frame (depending on feedback choice).
7	Increment frame count (within CountsPerSec object).
8	Use keypress functionality to end screen and release video capture.
9	Voice assistant informs user that control is returning to the inner control loop.

Table 17: Step-by-step guide to the cam feed control function

6.9.7 guidance_system.py

The guidance system is the module that uses all of the previously obtained data to make calculations on the relevant positions of the tracked object and the user's hand. Upon import the module checks the guidance mode and opens a connection to the CS Vibrating Glove if required. It also creates the guidance dictionary containing the directional feedbacks used for guidance. The feedback is provided independently of the guidance calculations meaning that the feedback methods can be updated without affecting the rest of the codebase in this module.

6.9.7.1 Function: open_glove

Opens a connection from the COM port to the glove.

6.9.7.2 Function: close_glove

Closes the connection from the COM port to the glove.

6.9.7.3 Function: buzz

Used in glove mode only. Takes an input variable describing the directional feedback and sends a binary string to the glove that tells it which vibration pads to activate. The resulting activations in the glove tell the user which way to move their hand.

6.9.7.4 Function: calc_min_dist

Takes the detected hand landmarks and calculates the distance from each one to the selected object centroid and stores the results in a list. It then takes the shortest distance from that list and writes the distant on the current frame. It returns the closest landmark and the target distance in pixels.

6.9.7.5 Function: dir_to_target

Calculates the direction that the closest landmark needs to move in order to reach the centroid of the tracked object. It is a simple calculation that determines whether the hand landmark is to the left or right of the object, and above or below it (forward or backwards from it). It returns two variables that give the required x and y direction in which to move.

6.9.7.6 Function: guidance_feedback

Using the guidance mode parameter this function provides feedback to the user according to the value of the parameter. It will either send the directional binary string to the glove, give vocal commands, or gives a beeping tone that either raises or lowers in frequency as the hand moves closer or further away respectively.

6.9.8 Supporting Files

There are 4 supporting files in the project solution that were part of the SSD MobileNet download:

Filename	Description
coco.names	Contains the class names of the coco dataset used by the SSD model.
inference_graph.pb	Contains the weights associated with the SSD object detection model.
ssd_mobilenet_v3.pbtxt	Contains the SSD object detection model configuration data.
a-guiding-hand-service-account.json	Contains the API key for the Google Cloud speech synthesis app service

Table 18: Supporting files used within the guiding hand project for SSD and speech synthesis

6.9.9 Google Cloud Text To Speech Console

The Google Cloud console is briefly mentioned for choosing the voice and setting up the API key.

6.9.9.1 Choosing the Voice

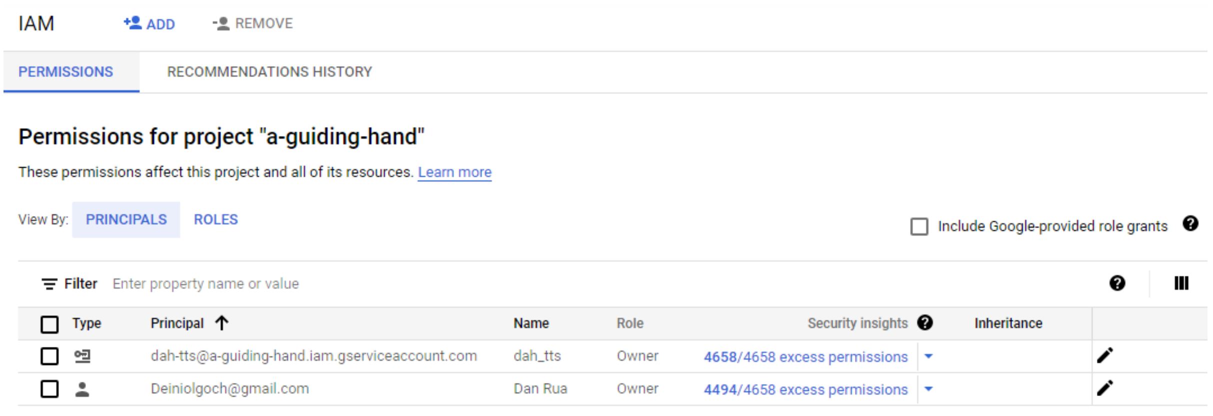
There is a demo page on the cloud: https://cloud.google.com/text-to-speech#section-2

From this page it is possible to trial a number of voice types, vocal speeds and pitches to find the perfect voice. There is also a "Show JSON" button that gives the format for the JSON request that was embedded into the voice_assistant.py code.

6.9.9.2 Setting up API Key

After creating a project: https://console.cloud.google.com/iam-admin/iam?project=a-guiding-hand

Setting up the API key was easy and consisted of following some basic instructions (You can log in from the previous link). Once the project was created it was just a matter of creating some permissions and downloading the JSON file that can be found in the project code.


IAM
ADD
REMOVE
PERMISSIONS
RECOMMENDATIONS HISTORY
View By: PRINCIPALS
ROLES
Include Google-provided role grants
Filter Enter property name or value

Permissions for project "a-guiding-hand"

These permissions affect this project and all of its resources. Learn more

Type	Principal	Name	Role	Security insights	Inheritance
	dah-tts@a-guiding-hand.iam.gserviceaccount.com	dah_tts	Owner	4658/4658 excess permissions	
	Deiniolgoch@gmail.com	Dan Rua	Owner	4494/4658 excess permissions	

Figure 18: Defining API permissions on the Google Cloud Console for Text-To-Speech was a simple task

7 Testing

As with all software developments the testing phase is critical in being able to assess the strength of the development and whether it satisfies the user requirements as laid out earlier in this document.

7.1 Test Set-Up

The original plan was to use a pair of action camera glasses however the glasses did not support video streaming. Instead, a web cam was attached to the user's head as a makeshift head mounted camera.

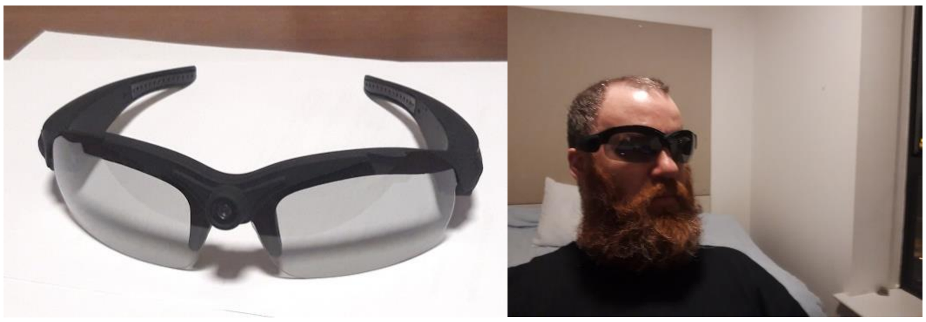


Figure 19: Original plan was to use action camera glasses but they didn't support streaming

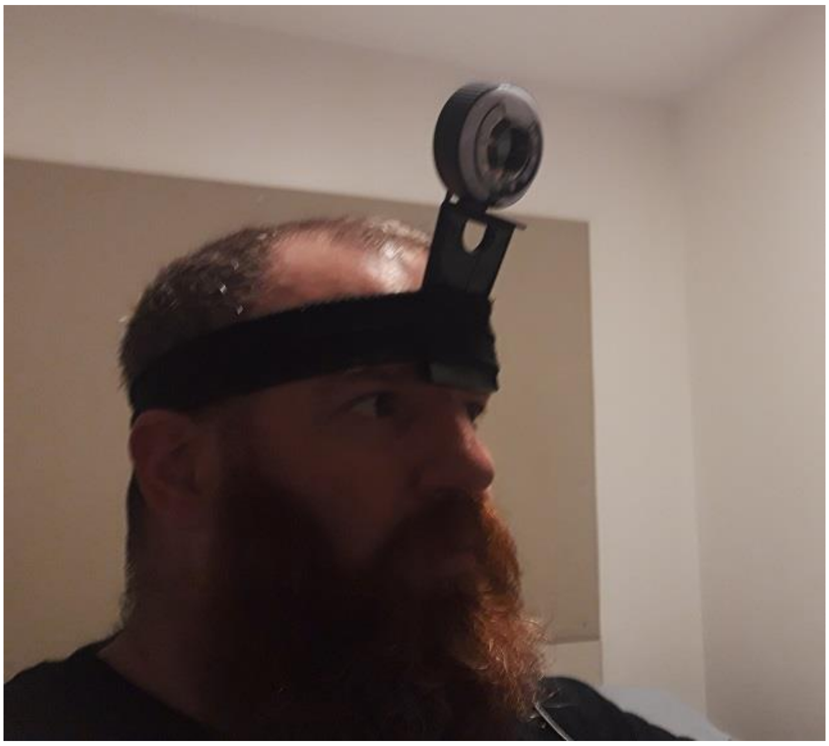


Figure 20: The workaround involved strapping a webcam to the user's head as shown above

7.2 Functional Testing

7.2.1 Summary

In order to test whether the functional requirements have been met a series of tests will be executed against the use cases listed in section 4.3. The tests were executed against the test cases laid out below and were then repeated in the negative form to try and expose flaws in error handling.

7.2.1.1 Waking the Personal Assistant

Test	Pass / Fail?
Attempt to wake the personal assistant by calling their name: Surah.	Pass
The personal assistant wakes and greets the user appropriately.	Pass
Negative testing to expose errors.	Pass

*Table 19: Test results for use cases involving waking the personal assistan*t

7.2.1.2 Sending the Personal Assistant to Sleep

Test	Pass / Fail?
Attempt to send the personal assistant to sleep using the keyword: Sleep.	Pass
The personal assistant informs the user they are going to sleep.	Pass
The personal assistant informs the user how to re-activate them.	Pass
Negative testing to expose errors.	Pass

Table 20: Test results for use cases involving sending the personal assistant to sleep

7.2.1.3 Exiting the program

Test	Pass / Fail?
Attempt to exit the program using the keyword: Exit.	Pass
The personal assistant informs the user the program will end.	Pass
Negative testing to expose errors.	Pass

Table 21: Test results for use cases involving exiting the program

7.2.1.4 Detecting Objects

Test	Pass / Fail?
Attempt to detect objects using the keyword: Scan.	Pass
The personal assistant informs the user that they are performing a scan.	Pass
The personal assistant informs the user what objects they detected.	Pass
The personal assistant informs the user how they can select an object.	Pass
Negative testing to expose errors.	Pass

Table 22: Test results for use cases involving detecting objects

7.2.1.5 Selecting an Object

Test	Pass / Fail?
Attempt to select an object using the keyword: Select + an object that was detected.	Pass
The personal assistant informs the user that the object has been selected.	Pass
The personal assistant informs the user that control is being passed to the guidance system.	Pass
The guidance system activates.	Pass
Negative testing by selecting an object that occurs zero times: Not allowed.	Pass
Negative testing by selecting an object that occurs multiple times: Not allowed.	Pass
Negative testing by selecting an object without first scanning for objects: Not allowed.	Pass
Negative testing to expose other errors.	Pass

Table 23: Test results for use cases involving selecting objects from the list of detected objects

Note: *The video window does not always open as the prime focus and must be manually brought to the top-most visible position. This only affects demonstration visibility.*

7.2.1.6 Guidance System

Test	Pass / Fail?
The guidance system recognises as many hands are as allowed in the config file.	Pass *
The distance to the object centroid is represented by green lines from each landmark except for the closest landmark which is highlighted in red.	Pass
The distance to the object centroid is display on the video frame.	Pass
Move the selected object and the tracker successfully tracks it.	Pass
Repeat tests using voice, glove, and beep mode. Directional feedback is provided as expected on each occasion.	Pass
Manipulate the hand location to ensure that directional feedback is provided for all 8 cardinal points (up, up-right, right, down-right, down, down-left, left, up-left).	Pass
The personal assistant announces "target acquired" once the closest landmark has a distance less than the defined tolerance.	Pass
Negative testing to try and break the guidance system.	Pass

Table 24: Test results for use cases involving waking the personal assistant

Note: *Whilst negative testing passes, it should be noted that the system cannot track an object that leaves the field of view. The hand tracker also struggled with identifying the CS Vibrating Glove.*

7.2.1.7 Tracking an Object

Test	Pass / Fail?
With the guidance system in control, move the selected object around the field of view and verify that the tracker keeps tracking the object.	Pass
Repeat the test with an independent actor holding the object.	Pass
Repeat the test but move the hand and object around each other to ensure that tracking is continuously updating the directional guidance feedback.	Pass
The personal assistant announces "target acquired" once the closest landmark has a distance less than the defined tolerance.	Pass
Negative testing to try and break the guidance system.	Pass

Table 25: Test results for use cases involving tracking the selected object

Note: *Whilst the test with the independent actor passes it should be noted that sometimes the actor's hand is detected before the user's hand enters the frame. This can be remedied by hiding the hand behind the object, obscuring its shape to avoid detection or have the user's hand detected first.*

7.2.2 Specific Issues Noted

Some specific issues are elaborated on below. Note that some are interdependent.

7.2.2.1 Large Object Detections

As previously mentioned, detecting large objects has a huge impact on the frame. The picture below shows how the detection of the table surface has reduced the frame rate down to 3 frames per second.

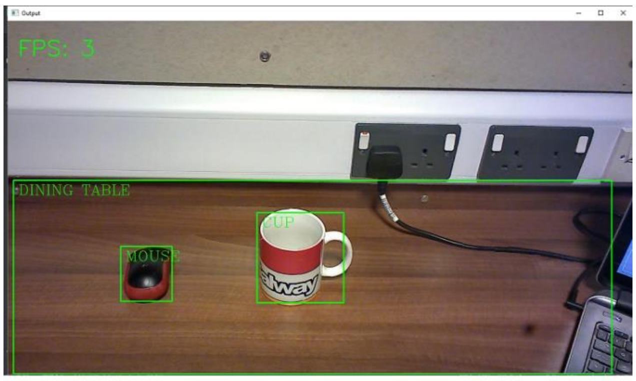


Figure 21: When SSD makes large object detections such as "Dining Table" the frame rate drops substantially

7.2.2.2 Time Delay Between Detect and Select Object

One of the design flaws highlighted through converting from static camera to wearable camera towards the end of testing was the requirement for the user to keep their head completely still from detection to selection as the bounding box co-ordinates are tied to a specific location on the camera field of view.

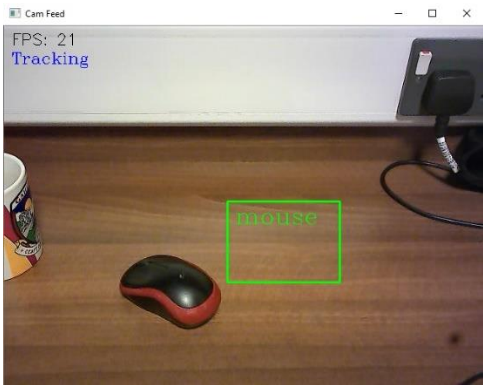


Figure 22: The time delay from detection to selection can cause issues with the bounding box alignment

Whilst the problem could be fixed by adding in a re-detection on the first frame of the guidance system it was decided against implementing the fix as there still exists a separate issue whereby if multiple objects were detected of the same class, it would not know which object to pick for guidance and crash the program. Similarly, the object detection is not infallible and it might also fail to detect the object on the re-scan. Whilst this is a slight frustration, a slightly misaligned bounding box did not break the program nor prevent the tracker from moving with the object in this proof of concept design.

7.2.2.3 Losing the Track

Sometimes the object tracker loses track or track off screen. It could be that the user's head wearing the camera turned too fast, or the object moved too fast, or the head moved too far between detect and select (previous point). When this happens. The tracker can either continue off screen as seen below or revert to guide the hand to a (0, 0) co-ordinate when the track is lost. A simple re-scan on "Tracking Lost" would fix the problem but was not implemented for the same reasons as the previous bug, the selection error handling functionality would need to be expanded substantially.

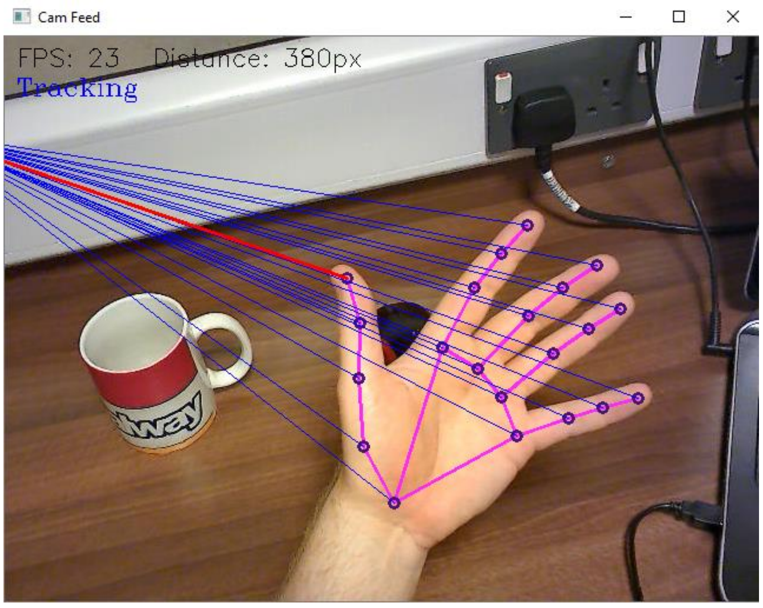
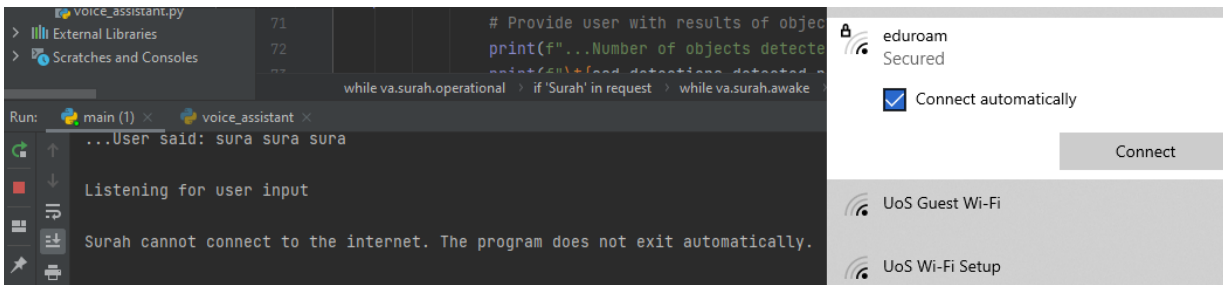


Figure 23: When the tracker loses the selected object the guidance system does not handle the error

7.2.2.4 Losing Internet Connectivity

With the NLP element of the personal assistant requiring an internet connection the solution can sometimes trip over itself or store multiple commands as one phrase in the event that the internet connection is degraded or lost. The picture below shows how "Sura" is stacked up three times into one phrase as the internet connection drops before alerting the user to the fact that the internet cannot be reached (vocally via offline sound file and printed to the console). The right hand side shows the generally poor wireless network strength observed during testing.



```
# Provide user with results of objec
print(f"...Number of objects detecte
```

```
...User said: sura sura sura

Listening for user input

Surah cannot connect to the internet. The program does not exit automatically.
```

Figure 24: Poor internet connectivity disrupts the voice assistant's ability to communicate smoothly

7.3 User Acceptance Testing (UAT)

With functional testing complete feedback was elicited from a small sample of users for their opinions on how easy the solution was to use and what improvements they would suggest for future iterations of the design. The users were selected from among the student populace at University of Stirling and are both fully sighted. Please see Appendix E for images of the ethics forms for each.

Name	Background	Nationality	Accent
Azizah Asadullah	MSc. Criminology	Indian	Indian
Ailin Chen	MSc. Behavioural Health Psychology	Chinese	Chinese with Canadian Influence

Table 26: Two sighted users were selected from the student body to test the solution with varying accents

7.3.1 Speed of solution

It was noted that when an object with a large bounding box was selected, such as a person, the frame rate drops off to below 5 FPS. This would present a severe limitation that requires further investigation in future iterations. When "normal" sized objects such as cups or the mouse were selected the frame rate was seen to be approximately 19-20 FPS with slight variations due to background processing.

7.3.2 Depth Issues

When the user's hand or target object is very close to the camera lens tracking and guidance become difficult as the bounding boxes take up the majority of the frame making it incredibly difficult to accurately guide the hand towards the object or to even track the object. It can be concluded that the solution works best with objects that are at a medium to long distance from the source lens. Similarly, as the angle of attack between the camera and surface decreases the guidance system tends to acquire the target before the hand has reached it when approached from the front due to the lack of depth perception. If resolving the distance vector first (y co-ordinate) it is then acceptable in approaching from the side (x co-ordinate) to successfully retrieve the object.

7.3.3 Speech Recognition

The voice recognition proved challenging under demonstration conditions. Once other people were in the room the AI seemed to develop a temperamental personality and frequently misinterpreted the command statements to the point of being borderline unusable. This goes to show that whilst great strides have been made in voice recognition the basic packages used in this solution still suffer from common legacy issues with background noise, accent, vocal cadence, and clarity of speech all playing a factor. The frequently poor internet connectivity at the test location in Beech Court also played a part in damaging the credibility of the speech recognition as a robust solution.

7.3.4 User Preference on Guidance Mechanism

The users were happy with both the voice and haptic feedback. Azizah expressed a preference for the haptic feedback as being timelier and more accurate whilst Ailin was neutral with no preference stated. One of the circuit wires disconnected on the glove during Ailin's trial leading to feedback that the final solution would need to be more robust to cater for rough handling. Ailin's wrist was a little bit too small to get a good contact with the glove and she had trouble differentiating the "move right" vibration from the "move forwards" vibration. There was no trouble identifying "move forward".

Where both agree is that the pitched tone proximity feedback (beep frequency) was novel and fun but perhaps better suited to a game scenario than the real world.

7.3.5 General Feedback

The overall feedback was excellent. Both subjects were impressed with the ability to guide the hand in a variety of ways whilst being able to interact with the personal assistant. The issues with the speech recognition were highlighted as frustrating and if present in a quiet environment with limited feedback how would it behave in a noisy restaurant or other public settings? This is a valid question and one that would require a deeper analysis before refining the solution. It is accepted that the quality of the microphone and speech recognition programs are not state of the art and perhaps the personal assistant would behave better if it carried the level of quality used by the Alexa or Google assistants.

Both subjects were disappointed that the wearable glasses were not operational as they felt it offered a stylish appearance to the solution and gave the impression that the user was an "active and cool person". This feedback would certainly be welcomed by the blind community who seek to fit in and wear assistive technologies that do not look overly medical in nature. With the increased and high profile use of high tech prosthetics such as those worn by combat veterans and athletes (the so called blade runners), the modern public arena is more familiar with, and more forgiving towards, these high-tech wearable technologies that have the potential to look aesthetically pleasing.

7.3.6 Racial Bias or Glove Bias?

One of the ethical issues around the widespread use of artificial intelligence focuses on machines accidentally learning racial bias or having difficulty identifying races that were not sufficiently included in the training material. One of more regrettable computer vision examples occurred when New Zealand's online passport photo application system accused an east Asian man of having his eyes closed [64].

During integration testing it was noticed that the hand detection algorithm had trouble identifying the hand when the CS Vibrating Glove was used. Occasionally it was recognised but the vast majority of the time it failed to recognise the hand. The glove contains cut-off fingers giving the impression of a two-tone hand (red palm and base of fingers transitioning to a white end of finger for the developer).

An interesting thought became whether it would identify a skin-tight black glove. Testing revealed that the algorithm failed to recognise the black glove 100% of the time. Without access to a larger diversity of test subjects it is not known whether the algorithm is confused by the apparent skin tone or whether it is simply smart enough to recognise that a glove is being worn. Either way, this highlights the importance of remembering to use a diverse test population to ensure that accidental biases do not creep into the solution. In the screenshots below, we see a white hand correctly identified even though it is holding a cup whereas two clearly held up gloved hands are completely unrecognised.

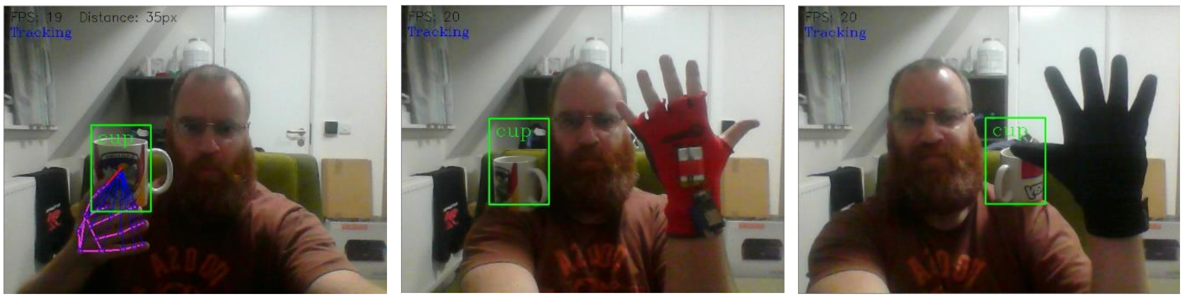


Figure 25: Is the hand tracker algorithm smart enough to detect a glove?

User testing highlighted that the hand tracker had no problem picking up Azizah's darker Indian skin tone therefore on the basis of that test, it could be said that perhaps the hand tracker is in fact smart enough to distinguish between a hand and a glove and is not inherently racist. Whether the ability to ignore a gloved hand was intended by the developers remains to be seen. If the training images did not include gloved hands, then perhaps this is an area for improvement on their part.

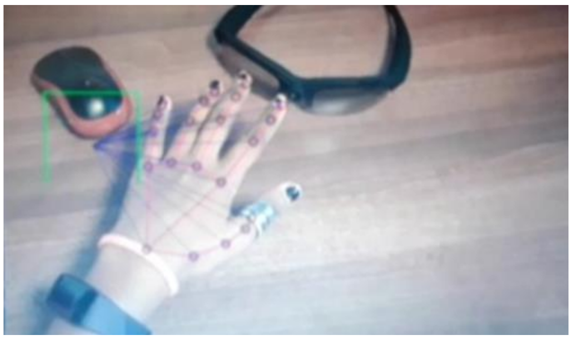


Figure 26: The hand tracker successfully identified the (Indian) hand of Azizah during UAT

8 Conclusion

8.1 Summary

The project achieved the high level goals laid out in the scope and objectives by providing a gateway towards active directional assistance delivered via voice or Bluetooth haptic device (CS vibrating glove) within the framework of solution controlled by a fully interactive personal assistant. Both of which were identified as gaps in the current offerings and the concept shows that it is possible to fill this gap.

The user can request that the personal assistant scans the camera's field of vision to detect a range of everyday objects before selecting one. They can then enter their hand into the field of vision and receive directional feedback that will guide their hand towards the selected object at a very respectable 19-20 FPS (not quite top end but good enough). Whilst the solution may be a crude representation of what is possible, it has proved the concept that artificial intelligence can be used to provide directional feedback to a blind or visually impaired user on a limited 2D basis. With this concept now proven the potential for further increasing the capability of assistive technologies is limited only by the time and resources that are available or willing to be spent on developing more elegant solutions from the platform that this project provides. The goal of taking the blind and visually impaired one step closer to possessing individual freedom and independence is a distinct possibility.

The project was not without its difficulties. The learning curve was very steep with a move from classroom based, standalone applications of artificial intelligence into creating an integrated solution that makes use of several applications inside one project solution. The lack of processing power in the development laptop necessitated the use of creative thinking in maintaining an acceptable frame rate and imposed certain design constraints that had an impact on the look and feel of the final output. The challenges were not in short supply but were embraced and ultimately led to a more robust solution being created that allows for further development without radically altering the core of the codebase whilst providing ample opportunity for improvements to the existing codebase. The complexity of the project led to noticeable mental fatigue that had an impact of how far the solution could be realistically pushed and refined. The ongoing Covid-19 crisis made it difficult to take high quality breaks and come back with fresh eyes, something that is critical in a research project. It is acknowledged that this affects every person in all walks of life at this current point in time.

8.2 Critical Evaluation

Arguably the biggest criticism from a non-technical perspective also stems from the Covid-19 pandemic in that it was not possible to seek the guidance of the blind community and engage in the type of hands-on research that is invaluable. Having said that, the project achieved the broad scope and objectives laid out at the start of this document but there is much that can be improved upon:

- Whilst the speech synthesis files could have been saved to provide offline capability the speech recognition does require internet connectivity. Offline solutions are available but are not at the same level as the online solution provided by Google. This would limit the usage of the solution to areas where internet connectivity is available.
- The directional guidance is provided in 2D which is a major blocker at the moment for real-world applications. Whilst the 2D approach can be useful if the camera has a high enough angle of attack to the background surface where objects are located – looking down onto a table from an angle of greater than 60 degrees – it becomes unusable at lower angles due to lack of depth perception.

- The Envision smart glasses offer the ability to ask the camera to scan for a specific product instead of just selecting from what is available. It was a regrettable mistake to omit this from the design as it would have been an excellent feature to have available and not difficult to add.
- The majority of development testing occurred with a static camera instead of a worn device and this led to a critical oversight that was only spotted during testing. Using the object detection on a single frame in a different part of the inner control loop meant that the user had to keep their head in a fixed position so that the tracker could use the bounding box co-ordinates stored in memory without losing the track due to the time it took to receive the list of detected objects, select an object, and have the co-ordinates passed to the guidance system.
- The OrCam glasses provide a simple description of where the objects are in the frame. It would be relatively simple to add a function to describe which quadrant of the camera's view the objects are in. This was planned but fell by the wayside once the wearable glasses failed to be useful.
- The Computer Science Vibrating Glove identified an issue with the hand tracker algorithm. It would appear that the algorithm struggled to pick up the gloved hand. A simple solution for this project was to wear the glove to receive directional feedback as normal but use the left hand to retrieve the object. It worked but would be better if the glove wearing hand were able to be used.
- During tracking tests with the object in motion an issue was identified where a second person's hand would be correctly detected but the guidance system only calculates the distance for the first hand detected in frame if multiple hands are present. This would need to be looked at in more detail in the future but would not be a major issue to resolve.
- The voice assistant was developed to provide a realistic framework and context in which to ground the guidance system. As such it needed to be more than a trigger word but did not warrant the full blown effort of an NLP solution that could infer desired meaning from a spoken sentence. The voice assistant was developed to be a half-way house that could recognise keywords within a sentence, but only if the sentence contained one keyword. Due to the structure of the IF statement of the inner control loop, any clash of keywords would also be subject to the order in which they occurred in the IF statement as it executes sequentially top to bottom.
- The speech recognition did struggle when background noise was present or when user's spoke with a less than crystal clear voices or with a non-standard accent. The lack of a high quality microphone and use of basic speech recognition packages almost certainly had a negative impact.

The object tracking algorithm performed admirably in relation to its reputation. The mosse tracker is regarded as one of the fastest trackers but is said to be lacking in accuracy. For the limited speeds of the objects in the tests this proved to be a non-issue. One quirk of the code that was discovered during testing is that if the track was lost it could be picked back up by moving to the last place that it was successfully tracked as the bounding box data was stored in the detections object memory. Whilst not the most helpful in this context there is probably some use for this quirk in other scenarios.

As mentioned in the user testing it is unknown at this point whether the Google MediaPipe algorithm is smart enough to recognise a gloved hand and ignore it or whether this is a sign of a potential racial bias that has not yet been picked up by the development team. As such a more diverse user base would be required to ascertain the extent of this issue.

8.3 Conversion to 3D Guidance

The biggest opportunity for future development is to take the solution from providing directional guidance in two dimensions and provide three dimensional guidance. This would be the natural next stone in creating a product that can help blind and visually impaired people navigate their immediate environment and potentially help with outdoor navigational tasks. Some of the available methodologies are discussed briefly.

8.3.1 Stereovision Camera

Stereovision cameras offer a trusted method using simple triangulation techniques. Once the dimensions of the field of view and camera field of vision angles are known, each camera can be set up to calculate Euclidean X, Y and Z distances to the target using simple trigonometry. This method does require a high degree of calibration both from within the programming framework and with the alignment of the camera to adjust for the fisheye effect of the lens and to ensure that the cameras are aligned along a known XYZ axis with appropriate adjustments to the calculations known. Online results posted by amateur stereovision builders highlight limits to the effectiveness of this system near the edges of the cameras field of view, as well as incorrect camera calibrations and alignments.

8.3.2 RGBD Camera

Whilst reviewing literature for virtual reality hand tracking it was learnt that RGBD cameras are generally considered the best camera to use for obtaining accurate depth perception. There are two methods for achieving the depth maps required for constructing a 3D with depth camera – Time of Flight and Structured Light. Kadambi et al. [65] performed an experiment on the different methods used by subsequent Microsoft Kinect versions to ascertain if one was better than the other. The result was that it depends upon the task leading to the conclusion that methods should be analysed further against proposed use cases before choosing one over the other.

8.3.3 Reference Pictures & Bio-inspired Depth Perception

In 2014 the European Space Agency used a student internship [66] to investigate how the monocular vision of jumping spiders utilised image defocus with high degrees of accuracy to serve as a reliable fallback mechanism for future robotic space missions to ensure that rovers could continue to function if one of their camera eyes fail. The principal method of investigation involved using a database of known stereovision images containing accurate distance mappings to provide a ground truth allowing for a monovision system to predict distances using deep learning models. An email newsletter earlier this year revealed that the prototype had been successfully tested aboard the International Space Station. Further details have not yet been released into the public domain on this experiment.

Whilst this methodology could be adapted for “a guiding hand”, it may make it difficult to realistically expand the use cases of the solution without a huge amount of training data available across a wide range of scenarios and variables within those scenarios. For example, in order to trust the solution to guide a blind person down a road with heavy traffic one would want to be extremely confident that the deep learning model has seen enough reference pictures in as many configurations as possible to be sure that the user won’t end up in a serious traffic accident.

8.4 Future Opportunities

This solution could be expanded in many ways and for many uses if the notion of a personal assistant is retained as the central pillar of the solution. Here are a few that spring to mind:

- The coco dataset already provides object detection capabilities for 91 everyday objects including vehicles, chairs, desks, and other common obstacles. The object detection classes could even be expanded or contracted to be more specialised through the use of transfer learning.
- The Maptic product uses LIDAR to warn of obstacles and provide collision avoidance capability. There is no reason why an object detection algorithm could not be developed to provide a similar function if the conversion to stereovision can be achieved, or RGBD glasses.
- Taking inspiration from self-driving cars it should be possible to create functionality that keeps a blind person in a "safe lane" whilst walking down the street for example with safe being defined as not stepping out into the road or notifying of an approaching kerb / road in front of them.
- Scanning for specific objects and in conjunction with the above, guiding the user towards them if they are not close by.
- Adding the standard recognition suite as per the existing products – faces, products, reading, forms, etc., and integrating them with the guidance system.
- Enhancing existing functionality by integrating the existing products face, product, and text recognition capabilities with the detect object function so that instead of hearing "box, box, person, chair" the user would hear a more useful "Coco Pops, Cornflakes, John, Chair".
- Add map functionality.
- Increasing the object detect frequency to provide an updated bounding box once superior processing power is available.
- Convert the voice assistant module to be fully offline and thereby increase the range of operational to environments where internet connectivity is not possible.
- Italian researchers [67] found that blind people also rely on timing cues to locate the origin of sound. Converting sighted distance metrics into timing cues (or step counts) based on estimation of speed from GPS technology could be very useful in designing novel spatial navigation techniques for the blind.
- The hand tracker provides 21 usable landmarks for manipulation. It is possible to create gesture recognition models and even train sign language or product specific gesture recognition. One could stream the camera feed through display lens on the wearable glasses for the visually impaired, who could then use gestures to zoom in on the cam feed. This could even be deployed to sighted individuals who have a need to see and detect objects further than the eye naturally allows. Obvious use cases could be military or search and rescue missions in open terrain.
- For ultimate usability add a night vision or infra-red camera display.

The corollary of the project seems to be that if an object, of whatever nature, can be detected then it can almost certainly be avoided, visited, or retrieved with a sophisticated enough guidance system.

The freedom and independence of blind and visually impaired people depend on such innovations.

9 References

References were generated using MS Word citation tracker with IEEE style.

[1] WHO, "World report on vision," World Health Organisation, 2020. [Online]. Available: https://www.who.int/publications/i/item/9789241516570.

[2] M. Burke, "How blind people see with sound," Molly Burke, [Online]. Available: https://www.youtube.com/watch?v=08smCjKWNL0.

[3] Brunes et al., "Loneliness among adults with visual impairment: prevalence, associated factors, and relationship to life satisfaction," Health & Quality of Life Outcomes 17, 24, 2019. [Online]. Available: https://doi.org/10.1186/s12955-019-1096-y.

[4] European Blind Union, "Facts & Figures," [Online]. Available: http://www.euroblind.org/about-blindness-and-partial-sight/facts-and-figures.

[5] Chuang et al., "Retinal implants: A systemic review," British Journal of Ophthalmology 98:852-856, 2014. [Online]. Available: https://bjo.bmj.com/content/98/7/852.

[6] R. Juskalian, "A new implant for blind people jacks directly into the brain," MIT Technology Review, 2020. [Online]. Available: https://www.technologyreview.com/2020/02/06/844908/a-new-implant-for-blind-people-jacks-directly-into-the-brain/.

[7] V. Lewis, "A to Z of assistive technology for low vision," Perkins School for the Blind, 2020. [Online]. Available: https://www.perkinselearning.org/technology/blog/z-assistive-technology-low-vision.

[8] World Blind Union, "Who we are," World Blind Union, [Online]. Available: https://worldblindunion.org/.

[9] Microsoft, "Seeing AI," Microsoft Corporation, [Online]. Available: https://www.microsoft.com/en-us/ai/seeing-ai.

[10] MyEye 2.0, "Orcam MyEye," Orcam, [Online]. Available: https://www.orcam.com/en/myeye2.

[11] Staff Article, "Talking glasses for the blind," Orcam, 2019. [Online]. Available: https://www.orcam.com/en/article/talking-glasses-blind.

[12] Envision AI, "Homepage," Envision, [Online]. Available: https://www.letsenvision.com/.

[13] Envision AI, "Blog," Envision, 2021. [Online]. Available: https://www.letsenvision.com/blog/envision-raises-eu1-5-million.

[14] Access World, "Envision AI & Seeing AI: Two multi-purpose recognition apps," American Foundation for the Blind, [Online]. Available: https://www.afb.org/aw/19/9/15059.

[15] Be My Eyes, “Homepage,” Be My Eyes, [Online]. Available: https://www.bemyeyes.com/.

[16] E. Tucker, “Maptic is a wearable navigation system for visually impaired people,” dezeen, 2017. [Online]. Available: https://www.dezeen.com/2017/08/02/maptic-wearable-guidance-system-visually-impaired-design-products-wearable-technology-graduates/.

[17] E. Farrington-Arnas, “Maptic - Tactile navigation for the blind,” Emilios, [Online]. Available: https://emilios.co.uk/work/maptic.

[18] Föcker et al, “The superiority in voice processing of the blind arises from neural plasticity at sensory processing stages,” Neuropsychologia, Volume 50, Issue 8, Pages 2056-2067, 2012. [Online]. Available: https://doi.org/10.1016/j.neuropsychologia.2012.05.006.

[19] T. Ith, “How six scrappy young inventors built a breakthrough text-to-Braille translator device,” Microsoft Corporation, 2016. [Online]. Available: https://news.microsoft.com/stories/people/team-tactile.html.

[20] Girshick et al., “Rich feature hierarchies for accurate object detection and semantic segmentation,” arXiv (1311.2524), 2014. [Online]. Available: https://arxiv.org/abs/1311.2524.

[21] G. Boesch, “Object detection in 2021: The definitive guide,” Visio.ai, 2021. [Online]. Available: https://viso.ai/deep-learning/object-detection/.

[22] R. Girshick, “Fast R-CNN,” ArXiv (1504.08083), 2015. [Online]. Available: https://arxiv.org/abs/1504.08083.

[23] Ren et al., “Faster R-CNN: Towards real-time object detection,” ArXiv (1506.01497), 2016. [Online]. Available: https://arxiv.org/abs/1506.01497.

[24] Redmon et al., “You Only Look Once: Unified real-time object detection,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, [Online]. Available: https://ieeexplore.ieee.org/document/7780460.

[25] A. Rosebrock, “YOLO object detection with OpenCV,” PyImageSearch, 2018. [Online]. Available: https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/.

[26] Redmon et al., “YOLO v3: An incremental improvement,” arXiv (1804.02767), 2018. [Online]. Available: https://arxiv.org/abs/1804.02767.

[27] Bochkovskiy et al., “YOLO v4: Optimal speed and accuracy of object detection,” arXiv (2004.10934), 2020. [Online]. Available: https://arxiv.org/abs/2004.10934.

[28] G. Jocher, “YOLO v5,” Github (No Paper Yet), 2020. [Online]. Available: https://github.com/ultralytics/yolov5.

[29] Liu et al., “SSD: Single shot multibox detector,” Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-46448-0_2.

[30] Simonyan et Zisserman, “Very deep convolutional networks for large scale image recognition,” International Conference on Learning Representations, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556.

[31] Pan et al., “Spatial as deep: Spatial CNN for traffic scene understanding,” 32nd AAAI Conference on Artificial Intelligence, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewFile/16802/16322.

[32] S. Z. Li, “Markov random field models in computer vision,” European Conference on Computer Vision, 2005. [Online]. Available: https://link.springer.com/chapter/10.1007/BFb0028368.

[33] He et al., “Multiscale conditional random fields for image labeling,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. [Online]. Available: https://ieeexplore.ieee.org/document/1315232.

[34] Ultraleap, “Homepage,” Ultraleap Ltd, [Online]. Available: http://www.ultraleap.com.

[35] A. Rivas, “Household items with more bacteria than a toilet seat,” Medical Daily, 2013. [Online]. Available: [31] https://www.medicaldaily.com/household-items-more-bacteria-toilet-seat-have-you-cleaned-your-video-game-controller-recently.

[36] DJ Pro, “Homepage,” Algoriddim, [Online]. Available: https://www.algoriddim.com.

[37] MediaPipe, “MediaPipe Hands,” Google on Github, 2021. [Online]. Available: https://google.github.io/mediapipe/solutions/hands.html.

[38] Google AI Blog, “On-device, real-time hand tracking with MediaPipe,” Google, 2019. [Online]. Available: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html.

[39] Lin et al., “Focal loss for dense object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8417976.

[40] A. Alavi, “A review of Google's new mobile friendly AI framwork: MediaPipe,” Medium, 2020. [Online]. Available: https://medium.com/swlh/a-review-of-googles-new-mobile-friendly-ai-framework-mediapipe-25d62cd482a1.

[41] Vidanpathirana et al., “Tracking and frame-rate enhancement for real-time 2D human pose estimation,” Vis Comput 36, 1501–1519, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s00371-019-01757-9.

[42] H. Chen, “Does word error rate matter?,” Smartaction.ai, 2021. [Online]. Available: https://www.smartaction.ai/blog/does-word-error-rate-matter/.

[43] Raf100steam, “6 problems AI faces in speech recognition,” Medium, 2017. [Online]. Available: https://medium.com/@RAF100STEAM/6-problems-artificial-intelligence-faces-in-speech-recognition-ae705cfa1a72.

[44] D. Coldewey, “Google's new voice recognition system works instantly and offline,” TechCrunch, 2019. [Online]. Available: https://techcrunch.com/2019/03/12/googles-new-voice-recognition-system-works-instantly-and-offline-if-you-have-a-pixel/.

[45] J. Schalkwyk, “An all-Nnural Oo-Ddvice speech recognizer,” Google AI Blog, 2019. [Online]. Available: https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html.

[46] He et al., “Streaming end-to-end speech recognition for mobile devices,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8682336.

[47] C. Dilmegani, “Natural language platforms: Top NLP APIs & Comparison,” AI Multiple, 2021. [Online]. Available: https://research.aimultiple.com/natural-language-platforms/.

[48] N. Bhat, “Documentation page,” PyPi, 2020. [Online]. Available: https://pypi.org/project/pyttsx3/.

[49] HTS, “hts_engine API,” HTS, 2015. [Online]. Available: http://hts-engine.sourceforge.net/.

[50] Open Source, “Festival Lite,” Github, 2021. [Online]. Available: https://github.com/festvox/flite.

[51] Van Den Oord & Dieleman, “WaveNet: A generative model for audio,” Google Deepmind, 2016. [Online]. Available: https://deepmind.com/blog/article/wavenet-generative-model-raw-audio.

[52] Van Den Oord et al., “WaveNet: A generative model for raw audio,” Deepmind, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499.

[53] Van Den Oord et al., “Pixel recurrent neural networks,” Google Deepmind, 2016. [Online]. Available: [https://arxiv.org/abs/1601.06759.

[54] Gougoux et al., “A functional neuroimaging study of sound localization: visual cortex activity predicts performance in early-blind individuals,” NIH National Library of Medicine, 2005. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/15678166/.

[55] D. Howarth, “Headset creates "3D soundscape" to help blind people navigate cities,” dezeen, 2014. [Online]. Available: [52] https://www.dezeen.com/2014/11/06/future-cities-catapult-microsoft-guide-dogs-3d-headset-soundscape-to-help-blind-people.

[56] Aftershokz, “Homepage,” Aftershokz, 2021. [Online]. Available: https://aftershokz.co.uk/.

[57] Flores et al., “Vibrotactile guidance for wayfinding of blind walkers,” IEEE Transactions on Haptics, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7060731.

[58] Jimenez & Jimenez, “Blind waypoint navigation using a computer controlled vibrotactile belt,” Advances in Intelligent Systems & Computing, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-41956-5_1.

[59] Bharadwaj et al., “Comparing tactile to auditory guidance for blind individuals,” Frontiers in Human Neuroscience, 2019. [Online]. Available: https://doi.org/10.3389/fnhum.2019.00443.

[60] Brock et al., “Using wrist vibrations to guide hand movement and whole body navigation,” i-com, vol. 13, no. 3, 2014, pp. 19-28, 2014. [Online]. Available: http://dx.doi.org/10.1515/icom.2014.0026.

[61] Durá-Gil et al., “Analysis of different vibration patterns to guide blind people,” PeerJ 5:e3082, 2017. [Online]. Available: https://doi.org/10.7717/peerj.3082.

[62] MITx: 11.155x, “Design Thinking for Leading & Learning,” Massachusetts Institute of Technology, 2019. [Online]. Available: https://learning.edx.org/course/course-v1:MITx+11.155x+1T2019/home.

[63] Schinazi et al., “Spatial navigation by congenitally blind individuals,” WIREs Cognitive Science, 7(1), 37–58, 2016. [Online]. Available: https://doi.org/10.1002/wcs.1375.

[64] J. Griffiths, “New Zealand passport robot thinks this Asian man's eyes are closed,” CNN News, 2016. [Online]. Available: https://edition.cnn.com/2016/12/07/asia/new-zealand-passport-robot-asian-trnd/index.html.

[65] Kadambi et al., “Computer vision and machine learning with RGB-D sensors. Advances in computer vision and pattern recognition,” Springer, Cham, [Online]. Available: https://doi.org/10.1007/978-3-319-08651-4_1.

[66] A. Nolte, “Lab report: Investigating jumping spider vision,” European Space Agency, 2015. [Online]. Available: https://www.esa.int/gsp/ACT/doc/BIO/AlekeNolte_FinalReport.pdf.

[67] Gori et al., “Temporal cues influence space estimations in visually impaired individuals,” iScience, Volume 6, Pages 319-326, 2018. [Online]. Available: https://doi.org/10.1016/j.isci.2018.07.003.

Appendix A – Python Libraries

The following libraries were used in the creation of the solution:
Note: 'Python' as the version denotes pre-installed alongside the python language

Library	Version	Where Used?
datetime	Python	video_thread.py
google.cloud	2.5.0	voice_assistant.py
keyboard	0.13.5	video_thread.py
mediapipe	0.8.5	hand_track.py
numpy	1.20.3	object_detect.py, guidance_system.py
opencv-contrib-python	4.4.0.46	main.py, object_detect.py, hand_track.py, video_thread.py, guidance_system.py
os	Python	video_thread.py, voice_assistant.py
playsound	1.2.2	voice_assistant.py
serial	3.5	guidance_system.py
speech_recognition	3.8.1	voice_assistant.py
sys	Python	main.py, object_detect.py, guidance_system.py
threading	Python	object_detect.py, video_thread.py, voice_assistant.py
time	Python	object_detect.py, video_thread.py
winsound	Python	guidance_system.py

Table 27: Libraries that are used in the python solution codebase

Appendix B – Configuration File

List of parameters contained within the config file along with default values:

Parameter	Default	Description
cam_src	1	Camera source (0 for laptop webcam, 1 for 1st external cam, etc.)
cam_width	640	Width of camera frame window
cam_height	480	Height of camera frame window
ssd_dataset	->	Filename of the SSD dataset containing class names
configPath	->	Filename of the SSD model configuration file
weightsPath	->	Filename of the SSD model weights
detect_thresh	0.5	Probability threshold for SSD object detection
nms_thresh	0.2	Non-max suppression probability threshold
num_coco_class_names	91	Number of classes in downloaded version of coco dataset
mp_detect_conf	0.75	MediaPipe hand detection probability threshold
mp_track_conf	0.4	MediaPipe tracking probability threshold
max_hands	1	Maximum number of hands detected by hand tracker (max = 4)
guidance_options	->	List of available guidance modes ([‘glove’, ‘voice’, ‘beep’, ‘none’])
guidance_mode	Voice	Selected guidance mode
com_port	COM5	COM port for haptic (glove) Bluetooth connection
target_tolerance	50	Method of adjusting pixel distance to target
beep_duration	200	Duration of beep feedback in milliseconds
guide_hand	1	Boolean - Temp mechanism to end guidance loop
mp_dict	->	Dictionary of hand landmark codes -> description
api_key	->	Filename of the Google Cloud api key for text-to-speech call

Table 28: Frequently used parameters that fell out of the iterative design and development phase

Appendix C – Variables of Class Object: Detections

List of variables initialised for future use in managing object detection data:

Variable	Data Type	Description
coco_names	List	Class name list for SSD model from coco dataset.
detected_ids	List	Object IDs returned from SSD model.
detected_names	List	Conversion of detected IDs into class names.
detections	List	SSD object containing detection results data.
prob	List	Object detection results confidence probability.
bound_box	List	Bounding box data.
selected_object	List	Holds the user selected object for the guidance system.
selected_object_idx	List	Holds the selection index for ease of access to bounding box data.
selected_bound_box	Tuple	Holds bound box info for the selected object.

Table 29: Record of intended uses of the variables initialised by the detections class object

Appendix D – Console Logging Sample Extract

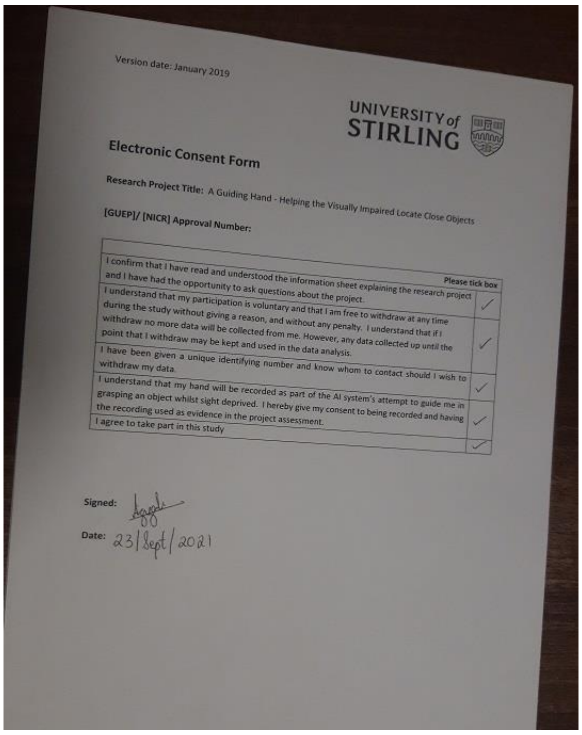
Below is a sample of the level of detail found in the console logging for this project:

```
Initialise voice assistant
...Creating voice assistant object
...Assigning google cloud credentials
...Assigning text-to-speech client
...Assigning google speech recognition variables
...Calibrating microphone
Voice assistant initialised

Initialising SSD object detection module parameters
...Defining model and parameters
...Creating object detection results object
...Results object created
...Loading class names from coco dataset
...Class names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane',
<< REMOVED FOR BREVITY >>, 'toothbrush', 'hair brush']
...All 91 class names loaded
SSD object detection module initialised

Initialising guidance system module parameters
...Creating guidance dictionary
Guidance system initialised

Creating hand tracker object
Hand tracker created

All module dependencies initialised

Checking guidance parameter
Guidance parameter validated: beep selected

Initialising camera parameters
Camera parameters initialised

Startup completed
...Response received from google cloud and saved
...Response delivered to user and file removed
User informed of startup completion

Listening for user input
...User utterance not understood

Listening for user input
...User said: Surah
...Response received from google cloud and saved
...Response delivered to user and file removed

Surah is awake

Listening for user input
...User said: go to sleep

Surah informs she is going to sleep
...Response received from google cloud and saved
...Response delivered to user and file removed
```

```
Listening for user input
...User utterance not understood

Listening for user input
...User said: exit

Surah says goodbye and the program ends
...Response received from google cloud and saved
...Response delivered to user and file removed

Camera windows destroyed
All objects deleted
Program ended successfully

Process finished with exit code 0
```

Appendix E – Signed Ethics Forms for User Acceptance Testing

Both users were required to complete the ethics form before taking part in UAT:



Version date: January 2019

UNIVERSITY of STIRLING

Electronic Consent Form

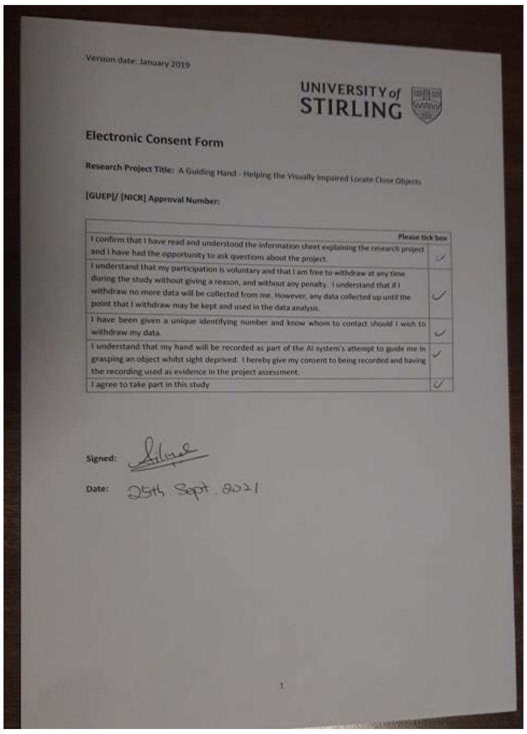
Research Project Title: A Guiding Hand - Helping the Visually Impaired Locate Close Objects

[GUEP]/ [NICR] Approval Number:

	Please tick box
I confirm that I have read and understood the information sheet explaining the research project and I have had the opportunity to ask questions about the project.	✓
I understand that my participation is voluntary and that I am free to withdraw at any time during the study without giving a reason, and without any penalty. I understand that if I withdraw no more data will be collected from me. However, any data collected up until the point that I withdraw may be kept and used in the data analysis.	✓
I have been given a unique identifying number and know whom to contact should I wish to withdraw my data.	✓
I understand that my hand will be recorded as part of the AI system's attempt to guide me in grasping an object whilst sight deprived. I hereby give my consent to being recorded and having the recording used as evidence in the project assessment.	✓
I agree to take part in this study	✓

Signed: [illegible]

Date: 23/Sept/2021



Version date: January 2019

UNIVERSITY of STIRLING

Electronic Consent Form

Research Project Title: A Guiding Hand - Helping the Visually Impaired Locate Close Objects

[GUEP]/ [NICR] Approval Number:

	Please tick box
I confirm that I have read and understood the information sheet explaining the research project and I have had the opportunity to ask questions about the project.	✓
I understand that my participation is voluntary and that I am free to withdraw at any time during the study without giving a reason, and without any penalty. I understand that if I withdraw no more data will be collected from me. However, any data collected up until the point that I withdraw may be kept and used in the data analysis.	✓
I have been given a unique identifying number and know whom to contact should I wish to withdraw my data.	✓
I understand that my hand will be recorded as part of the AI system's attempt to guide me in grasping an object whilst sight deprived. I hereby give my consent to being recorded and having the recording used as evidence in the project assessment.	✓
I agree to take part in this study	✓

Signed: [illegible]

Date: 25th Sept. 2021

1

Figure 27: Signed ethics forms were required before volunteers could complete user testing

END OF DOCUMENT