Aya Al-Qasab

The concept of visual future: smart glasses using Raspberry Pi

Bachelor's thesis
Information Technology

2019

South-Eastern Finland
University of Applied Sciences

Author (authors)	Degree	Time
Aya Al-Qasab	Bachelor of Engineering	October 2019

Thesis title

The concept of visual future: Smart Glasses using Raspberry Pi

58 pages

Commissioned by

South-Eastern Finland University of Applied Sciences (Xamk)

Supervisor

Timo Hynninen

Abstract

The main goal of this thesis project to provide a more sophisticated way to replace the lost senses when people with special needs. The project aims to build smart glasses that contain specialized object detection applications based on the voice assistant system. The work is divided in two major logical parts: theorietical and practical. Each part represents different viewpoint on the process of Internet of Things in theory and production.

The theory part of this thesis described all the technologies which were used in the process of final product creation, their basics and working principles. These included descriptions of technologies used in the project such as Raspberry Pi device and programming languages alongside other supporting tools. It also contains a description of the work projects related to several institutions with different ideas based on the same basis but with different results, most of these products rely on the system of the voice assistant to access several applications such as calendar or contact feature, telephone and maps.

The practical part consisted of a chronological description of the smart glasses development. The whole development cycle from the initial idea of the operating systems to the final state of the IoT product was described. The implementation part was focused on the creation of TensorFlow object detection applications. The applications were designed in order to provide a solution for visually impaired and deaf people with an open-source machine learning library for research and production. The main responsibilities of applications such as the discovery of a blind application object are transferred to the person who has lost this meaning through the camera image as an alternative eye, also the application translated by the languages of the deaf and mute.

The final result of this research was a smart glasses product with different integration features and implemented software. The product was delivered in the requested timeframe with all the requirements satisfied.

Keywords

Information technology, augmented reality, Python, mobile application, Raspberry Pi , logistics, Pi camera, IoT

CONTENTS

1 INTRODUCTION

In the modern, digitally connected world, more data is generated, stored, analyzed, and accessed than ever before. This is also becoming true for physical products and items, as the Internet of Things (IoT), smart homes and connected automobiles become more of the normal. Nowadays, when users are trying to buy any device like a used car or even new machines, they have access to more historical data than ever. This is where augmented reality (AR) smart glasses show promise in stepping in to solve the asymmetric information problem that prevents two parties from undertaking a mutually beneficial value exchange.

In some cases its difficult to use smartphones or computers. For example, if we are driving a car, it’s impossible to use the phones. The smart glasses can be the solution by connecting them via Bluetooth to access and display data with high resolution and quality, although we can give commands by using the talking system Alexa voice controller to make calls or answer messages without touching the phone. Also, this kind of glasses can be used to control servers and big data centers remotely, and handle many problems instead of the regular controlling system.

The main aim of this project is to place computers’ and smartphones’ content in the user’s field of view or ears. The content includes augmented reality, talking system, heads-up displays, and secure connection. Instead of looking at a rectangular screen, we will see words, pictures, objects and virtual environments by simply looking around. Many of the devices which is available in the market have almost the same features like access message, calls, tracking locations and provide GPS maps navigation. The project came up with different features that do not exist at least in the market yet, but few people who have been working with it, because it is difficult to find the software sources that can support this kind of project. The main features include image detection, American sign languages and voice assistant.

The practical part of this smart glasses project contains two sections, hardware, and software. The hardware section requires physical equipment, such as a talking system that includes a Raspberry Pi 3 board, micro SD card, mini speaker, USB micro, SD card reader, and jumper wires. A display system will include a mirror, lens, camera, sensors, Bluetooth, GPS, Wi-Fi, CPU, LiPo battery, LCD display and a charge circuit. All of the hardware components will be implemented inside a 3d printed case and glasses. Software section contains the programming codes' files based on the Python language.

The aim of this project to allow users send and answer messages and phone calls, access a phone's applications using a voice system, manage calender and get pop-up reminders at the same time when displaying the operation for users. This project bases on the same idea as other smart glasses, but with special features. The basic idea is to implement the programmable hardware with 3d replacement case, then it can be attached to any kind of glasses (eyeglasses or sunglasses). Also, it can be used by different users, such as adults, kids and the handicapped.

2 BACKGROUND

Smart glasses are products that are worn like normal glasses. They provide the user with information and technological possibilities to take pictures or record video. They use optics technology based on a Heads-Up Display (HUD), a Head-Mounted Display (HMD), and in particular, an Optical Head Mounted Display (OHMD). There is a plastic object at eye-level through which the user can see both an online, digital world and an offline. The added information can be shown visually before the eyes through the display of the glasses, or the user can get instructions, notifications and answers to questions in audio form.(Due 2014a.)

There are generally three types of initial applications for smart glasses: 1) specific job-related applications that include projection of instruction manuals, road maps and various other similar resources at eye level. 2) task-related and professional,

contextual applications but take place in the private sphere. That means applications, which are involved in specific tasks, which could be accomplished more efficiently with the use of smart glasses: e.g. instructions for a DIY enthusiast building a home extension, or as a way of filming and documenting certain important life events such as a ride on a big-dipper, a first dive from the 5-metre diving board or video communication over long distances. 3) lifestyle applications for self-trackers which tackle specific problems, smart glasses are also useful in terms of a wide lifestyle segment of people, who today are known as quantified self'ers (QS'ers). It becomes apparent that there are several task and job related functions that make sense on an intuitive level because smart glasses allow the user to use both hands. These functions include the projection of instruction manuals, road maps and various other similar resources at eye-level.(Due 2014b.)

The technology behind the smart glasses, for instance a camera, is built into the lens, while specs connect wirelessly to a smartphone to transfer data from one to the other. Some are controlled using a small button, while other versions rely on audio commands(Stevens 2018). In general, smart glasses aim to provide life monitoring services, as well as creating a platform for taking more authentic photos and video clips. They can also be equipped with augmented reality technology, interded to assist the users with their daily home and business life. They're still in early stages of development and there are lots of issues to overcome, such as security and privacy.

2.1 Augmented Reality

Augmented reality is an enhanced version of reality where live direct or indirect views of physical real-world environments are augmented with superimposed computer images over a user's view of the real world. It supplements reality, rather than completely replaces it, and it would appear to the user that the virtual and real objects coexisted in the same space.(Azuma 1997.)

AR can be displayed on various devices screens, glasses, handheld devices, mobile phones, and head-mounted displays. The project will be bases on the same structure of augmented reality, it includes cameras and sensors to collect data about users' interactions and sending it for processing. Also projection which takes data from sensors and projects digital content on the surface to view. The glasses contain mirrors to assist human eyes to view virtual images.
It's a double-sided mirror to reflect light to a camera and to the user's eyes to proper image alignment.

2.2 Logic Design

Nowadays, touchscreen input is the primary interaction for smart devices, and these touchscreens are sized from smart wristbands to smartphones. In smart wearables, such as smart glasses, speech recognition is the major input of choice, because these wearable devices do not have a touch screen display that serves as the input device.

Offering smart glasses with better input approaches makes the interaction experience more intuitive and efficient, which enables the users to handle more complicated and visually demanding tasks. The enhanced interaction experience brings smart glasses from their limited usage of micro-interactions to daily use, as has happened with today's smart phones.

2.3 Network Connection

In general, the hardware device is the most important section of any project that contains the basics components such as adapters, ports and slots. Most of the devices were including an internal wireless connection and ethernet adapter on the same board like the Raspberry Pi 3. With this feature, it provide a better network with no need to install and plug in to the board an external wireless or Bluetooth chip. Raspberry Pi 3 has a small wireless radio. Its markings can properly seen through a microscope or a magnifying glass. The Broadcom BCM43438 chip provides 2.4 GHz 802.11n wirless LAN, Bluetooth low energy and Bluetooth 4.1 classic radio support. It was built directly onto the board to

keep costs down, rather than the more common fully qualified module. Its only unused feature is a disconnected FM radio receiver. There is no need to connect an external antenna to Raspberry Pi, because its radios are connected to a chip antenna soldered directly to the board in order to keep the size of the device to a minimum. The chip should be more than capable of picking up wireless LAN and Bluetooth signals even through walls(Raspberry Pi 3 Network 2016).

The Broadcom BCM2837 system on the chip includes four high performance ARM cortex-A53 processing cores running at 1.2 GHz with 32 Kb level 1 and 512 Kb level 2 cache memory, a video core graphics processor and is linked to a 1GB LPDDR2 memory module on the rear of the board. The Raspberry Pi 3 features the same 40-pin general-purpose input/output (GPIO) header as all the Pis going back to the Model B+ and Model A+. Any existing GPIO hardware will work without modification, the only change is a switch to which UART is exposed on the GPIO's pins, but that's handled internally by the operating system(Raspberry Pi 3 Network 2016).

2.4 Python Coding Language

On Raspberry Pi, the only library that is not cross-platform for obvious reasons is the RPI.GPIO library which allows Python programs to access the GPIO. This results in the combination of object orientation and microcontroller capabilities into one platform. In the Raspbian operating system, there are a number of Python IDEs installed by default, but the simplest one to use is the official Python 3 IDLE(Python language on raspberry pi 2018).

Python is a handy and a relatively easy to learn programming language. It's flexible to allow users to build a web application as well as an interface with hardware components connected to Raseberry Pi, which makes it the perfect language to start learning on the Raspberry Pi(Python with Raspberry 2018). The Python programming language actually started as a scripting language for Linux. Python programs are similar to shell scripts in that the files contain a series of commands that the computer executes from top to bottom. Like shell scripts, Python can automate tasks like batch renaming and moving large amounts of

files. It can be used just like a command line with IDLE, Python’s REPL (read, eval, print, loop) function (Circuit Basics 2015).

2.5 Google Voice Controller

Google Assistant is a personal voice assistant that offers a host of actions and integrations. From sending texts and setting reminders, to ordering coffee and playing a favorite song, the one million-plus actions available suit a wide range of voice command needs. Google Assistant is offered on Android and IOS, but it can even be integrated with other devices like smartwatches, Google Homes, Android TVs, Raspberry Pi and Arduino.
Actions are the central platform fro developing Google Assistant applications. Actions work with a number of human-computer interaction suites, which simplifies conversational app development. Out of all the platforms, the most popular is Dialogflow, which uses an underlying machine learning (ML) and natural language understanding (NLU) schema to build rich Assistant applications (Google assistant voice controller 2019).

An Action is an entry point into an interaction that is built for the Assistant. Users can request action by typing or speaking to the Assistant. To start a conversation, the user needs to invoke an Action through the Assistant. Users say or type a phrase like "Hey Google, talk to Google IO". This tells the Assistant the name of the Action to talk to. From this point onwards, the user is talking to the Action and giving it input. This conversation continues as a two-way dialog until the user's intent is fulfilled or the conversation is finished (Google Actions 2019).

2.6 Related Work

This subsection is a comprehensive review of all the common smart glasses products available in the market. Phone-based augmented reality has started regular showing up on apps like youtube, snapchat and some games, and virtual reality can be used in gaming, film festivals and arcades. Augmented reality

glasses are a rare sight in daily life, but they hold an outsized place in the imagination. AR headset have not made the jump to consumers but they are still a full-fledged industry. The arrival of 5G, followed by advancements in augmented reality and the need for a more efficient workforce is driving the popularity of smart glasses (Draper 2018). Their popularity is rising so rapidly that some experts believe smart glasses could kill smartphones in the next few years. Some companies like Vuzix, Bose, North and Snapchat are offering smart glasses that look so stylish that everyone would like to have one. Several popular examples will be compared below by advantages, disadvantages, costs, hardware and software (Nobel 2019).

2.6.1 Epson Moverio BT-300 Augmented Reality Glasses

The Epson Moverio BT-300 is a pair of augmented reality smart glasses made by Epson, a manufacturer from Japan. It was designed to complement DJI drones such as Mavic Air. Epson BT-300 is suitable for hobbyists as well as professionals. It enables users to monitor their drone’s statistics while experiencing the flight in FRV (First-Person View).

The main features of Epson BT-300 include that it enables users to visualize information while keeping an eye on their drones and it is possible to add prescription lenses, sun-shield(ing) lenses and other options. Also, it has an internal long battery life which may last up to six hours per charge. A five-megapixel-high resolution front camera enables users to take photos and videos, and there is a dedicated controller that includes a GPS and micro SD card slot. Epson’s proprietary content platform allows users to download specialized drone apps, games and other applications (Aniwaa Epson 2019).

The Epson Moverio glasses can scan real world items such as a QR code to bring up relevant information right ontop of the QR code itself. It can also be useful for other technological activities such as flying a compatible drone in first person view, viewing virtual blueprints or just watching Netflix. The display tech

used is OLED, which means that the display will disappear completely, when not in use.There are two ways to control Epson glasses, through head movements or with a wired controller. It's much easier to use the head movements but for gameing purpose and browsing web, the controller will make for an easier controlling experience.

Figure 1. Epson-Moverio BT-300 Augmented Reality Smart Glasses

It can connect to a variety of WiFi hotspots as it supports wireless B, G and N. Comparing to features, it is available at the manufacturer price of USD 699 .

2.6.2 Everysight Raptor Glasses

These are the first augmented reality glasses to integrate data directly into the display. An OLED projector projects visuals onto a half-mirrored screen on the glasses' lens to display the information it collects, which provides an infinity-focused display in the rider's field of view.

The Raptor setup is an ease with an easy to use app, Android or IOS, and the software proved more polished than expected from a startup brand. It presents helpful cycling info such as directional, cadence and heart rate in front of the users, so that they can keep their eyes on the road (Lee 2018).

The Raptor glasses have built-in GPS, HD camera, microphone, speakers for music, WiFi and Bluetooth and a head-up display showing the ride data on the screen right in front off the eyes. All this tech comes at one of price about USD 499, but their performance is shapely impressive and they certainly feel like a simple-to-use, finished product (Puzak 2018).

Figure 2. Everysight Raptor Augmented Reality Cycling Glasses

The advantage of Raptor is that it has a head up display, easy to use once users get uset to it, but the camera hammers battery life with a big cost.

2.6.3 Google Glass Enterprise Edition Smart Glasses

The enterprise edition of smart glasses comes with powerful components and potentially wider deployment opportunities. The glasses come with a thick frame that is dust and water-resistant to store the technical components of the glasses. The new hardware of google glasses charges faster, has better battery life, better WiFi and Bluetooth, a better camera and it supports Android Enterprise Moblie Device Management (D’Angelo 2019).

The enterprise cost for Google glasses is about USD 999, but the hardware isn't sold directly to customers. The main benefits of Google glasses are the ability for manufacturers to integrate instructions for their work processes in real-time. Glass Enterprise offers workers instructions above their natural line of sight. Other capabilities include sending and adjusting instructions in real-time, live streaming to troubleshoot issues with projects and sending pictures of equipment or data to other members of the team (D'Angelo 2019).

Glass Enterprise design makes integrating the glasses into various levels of production practices. For manufacturers, the glasses can function as safety glasses that are already required, and glasses' pod can be removed and applied to any lenses that are compatible with Google glass. Also, the lightweight model can be applied to industries such as healthcare and logistics (Wang 2019).

The glasses can be controlled by voice commands or by swiping the side of the frame. They features three beamforming microphones and a multi-touch gesture touchpad, it also include an on-head detection sensor and an eye-on screen sensor to save battery. This lets users keep both their hands engaged with their active project. While voice commands offer a convenient way to progress to a different program, touch assistance is a good option when voice commands are unnecessary (D'Angelo 2019).

Figure 3. Google Glass Enterprise Edition 2019

Google Glass appears to be having a strong impact on manufacturing, healthcare and logistics. Also, it appears to be proving its worth as a tool for workers to make their jobs more efficient and easier. It’s expensive, but its worth for business.

2.6.4 Kopin SOLOS Smart Glasses

Kopin SOLOS AR smart glasses enhance sports experiences. They provide measurements of the user’s performance like elapsed time, speed power, cadence or heart rate. With Kopin, it’s possible to communicate with team members of cyclists, take calls, chat, listen to music and access most AR apps (Aniwaa, Solos 2019).

The world’s smallest optical module for Mobile Augmented Reality smart glasses offers a high-resolution display for clear and readable performance in variant lighting conditions. The performance information is streamed on the sharp display, which blends seamlessly with the cyclist’s field of view. Kopin had designed with aspiring and elite cyclists and coaches in mind, the performance eyewear allows users to customize their riding performance data and stream it to the integrated head-up display, allowing them to adapt training or racing effort at a quick look in real-time.

The main features include visual data displays that provide real-time information such as cadence, speed, heart rate and percentage of the achievement of the target. Also, it enables communications via calls, listening to music and monitoring of the device with voice control. It can access a large variety of AR applications. The user has a comfortable experience while running or cycling. Solos are built for cyclists, triathletes, runners and other outdoor enthusiasts. The system delivers critical assistance such as turn-by-turn route directions and information meant to keep an athlete focused and aware (Essential 2019).

Figure 4. Kopin SOLOS Smart Glasses

The glasses remove the need to look down, breaking stride and losing focus. They allow users to get the data they need right when they need it through a display.

2.6.5 Meta 2 Smart Glasses

The Meta glass brings the user inside the frame, inside games, movies or social apps and whatever screen-based world a user always experienced at a remove then AR's is turning the whole world into the frame. The Meta is one of the AR headsets, at USD 1495, that poses an interesting threat to the far more expensive Microsoft Hololens. Its field of view is significantly larger than that of the Hololens and lack of a pixel-based display. There are twin LCD panels but they reflect off the inside of the visor which means that visuals appear far sharper

at close range than VR users might be used to with more readable text and clear image (Rubin 2017).

The Meta uses an array of outward-facing sensors and cameras to map physical surroundings and then uses it as a backdrop for everything users do in the headset. It means that if the user pushes a window all the way behind, it should not effectively disappear, fill by the real-world object. The main feature of Meta glasses, it’s inconsistent at best. The mouse pointer would sometimes simply disappear, never to return until the company pushed a software update. The headset refused to acknowledge a hand, if the user wears a watch. The headset’s software interface called Workspace is a bookshelf of sorts, populated by small bubbles (Rubin 2017).

Figure 5. Meta 2 Smart Glasses

Each represents a Chrome-based browser window or a proof-of-concept demo experience and maybe third-party apps. To launch them, a user reaches out by hand, closing the fist around it and dragging it into free space.

2.6.6 ODG R-7 Smart Glasses

Osterhout Design Group (ODG) is the most advanced augmented reality smart glasses, featuring stunning ultra-transparent 3D stereoscopic display and packed with innovative technologies, is seeing major traction across key markets including healthcare, energy, transportation, warehouse, logistics and government (CES 2016).

R-7 is a next-generation computing platform that gives users the power and performance of a tablet, completely hands-free. It delivers an unparalleled user experience of display clarity, on-board processing power and connectivity, with no external computing or other support needed. The R-7 is shipping with a based price of USD 2750.

ODR is rapidly becoming the preferred platform for developers and partners building applications for smart glasses on ReticleOS-ODR's Android-based operating system that supports apps and delivers an intuitive user experience designed to see-through smart glasses. Some of the applications that display on ODG are Augumenta is a gesture control and navigation capability, Blippar is an AR and visual marketing content. Dysonics is an interactive 360 degrees's audio that creates more immersive AR/VR experiences. Medweb is a collection and recording of vital signs for healthcare use.

Figure 6. ODG R-7 Smart Glasses

OpTech4D is for training and assisted reality designed for oil and gas companies and complementary industries such as nuclear, aerospace and construction. Paracosm is for 3D mapping and AR capability. Scope AR is an AR maintenance functionality app and Vital Enterprises is a telepresence and remote assistance solution.

2.6.7 Toshiba dynaEdge AR100 Viewer Smart Glasses

Toshiba dynaEdge AR smart glasses are a wearable, hands-free augmented reality solution designed to help large enterprises improve efficiency, quality and operating flexibility. It offers a complete solution, incorporating both hardware and software in one package. The dynaEdge enables multiple usage scenarios, including see-what-I-see, remote expert document retrieval, workflow instructions and real-time data capture (Dynabook 2019).

DynaEdge glasses are the first wearable AR solution running Windows 10 Pro, it’s seamlessly integrated into organizations existing infrastructure and IT security standards. Also, battery life is vital, with removable, rechargeable battery pack and an optional four-port battery charger for continuous operation. The Dynabook AR glasses offer a variety of methods of input and navigation, including a touchpad and programmable buttons on the HMD and directional buttons on the

waist-mounted processor. Advanced software options enable voice and gesture capabilities.

Figure 7. Toshiba dynaEdge AR100 Viewer

DynaEdge can connect to the corporate network, send and receive data, stream live video and track assets. It has the ability to record and live stream video, take photos and scan barcodes. Also, it's configurable for use with eye, the HMD's micro-display along with dual microphones, speaker, sensor array and camera allows users the ability to ensure their hands are free to complete their job.

2.6.8 Vuzix Blade Smart Glasses

Blade glass is AR and VR wearable system. It's meant to be used in conjunction with the smartphone and promises the same features as a smartwatch, except that it places on the face instead of the wrist. The Blade was designed to look and feel like a normal pair of glasses. They appear to complement some face shapes more than others and it comes only with one style. Vuzix does provide a couple of customization options. The users can add prescription lenses as well as different nose bridges for a better fit (Prospero 2019).

The Blade uses a proprietary operating system called Blade OS, which is based on Android. The launcher interface is simple, with an array of shortcut icons on a bottom rails and a larger preview icon on top. The users need to use the touchpad on the glasses' right side to swipe and tap around and navigate. The glasses can be used without a phone but with a WiFi connectivity. It's a lot more useful when paired with an Android or iPhone via a companion app. When

connected, the Blade receives all the usual smartphone notifications such as calendar reminders and incoming messages.

One of the most important features of Blade it works with Alexa, to trigger the voice assistant the user most tap the touchoad once. Because it uses Alexa's smart screen SDK, it has an Echo show in the glasses. The Blade also comes with its very own app store. It has already hundreds of developers onboard. Potential apps include video streaming, web browsers and even games.

Figure 8. Vuzix Blade Smart Glaases

Blade is meant to be a smartphone companion. It can be used to respond to messages and to get directions. The Bluetooth connection scales, Alexa is not ready to transfer photos taken by the Blade to computer yet. It has poor battery life, no speakers and IOS users can not respond to texts or see directions, power LED and AR projector cast lights on user's face and it's physically uncomfortable. It costs about USD 999 (Harding 2019).

2.6.9 ThirdEye Gen X1 Smart Glasses

ThirdEye's X1 smart glasses come with the latest sensors, chips and a powerful optics design that allows users to experience an HD augmented reality display. X1 glasses are a fully integrated product and its sleek form factor and UI allow the user to easily wear it while completing tasks. The AR software of the glasses provides a full enterprise augmented reality platform that includes live audio, video and data communication between remote users, such as see what I see app. The application hands-free. The users control it with their head-motion which is based on ThirdEye's proprietary software and provides an intuitive AR interface that allows on-site personnel to collaborate with remotely located experts (Aniwaa, X1 2019).

The X1 glass has the largest battery life of any non-tethered AR smart glasses and dual hot-swappable, replaceable and rechargeable batteries for a full day of usage. It runs on the open-source Android OS and has hundreds of apps in the app store. These apps are easy to create, customizable with existing SDKs and APIs, more secure, manageable and flexible. The glasses can be used to develop powerful AR/MR apps from games and to image recognition apps to positional tracking. It comes with a thirteen megapixel HD camera, three-axis gyroscope, three-axis accelerometer, three-axis compass, light sensor and optional attach to other sensors via the USB-C port (ThirdEye Gen 2018).

Figure 9. ThirdEye Gen X1

The more obvious market for ThirdEye's smart glasses for 5G is in the enterprise and medical areas. AR scenarios for training and work in the field will take off with portable AR glasses, and healthcare will benefit from low-latency bandwidth that lets users bring in remote experts. ThirdEye is more focused on business-to-business and industry solutions (ThirdEye Gen 2018).

2.6.10 Vuzix M300 Smart Glasses

This is a new generation of smart glasses that offers hands-free mobile computing that makes multitasking easier than ever. They are built for the enterprise sector, delivering enhanced functionality for all kinds of commercial applications, and everything from field service to logistics and manufacturing (Flynn 2019).

M300 is a mountable computer with a tiny monitor that can be mounted on a pair of glasses frames over the left or right eye. It has numerous sensors for head tracking, a camera upfront for photo capture, 64 GB of onboard storage, a touchpad for controls, speakers, microphones for voice control and noise-canceling and an 860 mAh battery among a few other sensors and ports.

The applications for the M300 can srange from a UPS driver using the M300 to sort, label, track and verify a delivery of packages, to a mechanic taking apart an engine while cataloging and identifying various components.

Figure 10. Vuzix M300 Smart Glasses

The M300’s built up-to ten megapixels, optically stabilized camera takes in what the user is looking at and adds labels and information on its small eyes mounted screen. All input can be streamed, photographed, videoed 1080p and sent over WiFi. The Android OS is the foundation for the minimal UI which essentially consists of a side-scrolling list of apps and modes.

3 HARDWARE FOR SMART GLASSES

The recent developments in technology have led to new emerging wearable devices such as smart glasses. Smart glasses’ projects solved a major problem have been existing for a long time such as visually impaired and deaf language translators using a Pi camera. With image detection solutions, the visually impaired are able to know what kind of object can appear in front of them by getting alert from the assistant which exists in the application. Also, the glasses included American Sign Language detection to translate signs to letter, which is the easiest way to understand deaf’s language. This project describes the requirements for components, choose the appropriate ones and test them. This will help me to improve the final concept by facing and solving problems during the tests of the proof concept device. All the instructions, schemes and code will be attached to the thesis, so that it will be possible to recreate the device by following the instructions that are mentioned in the project. I described the components and include links to the official documentation. Also, a guide for the device assembly is available in one of the sections below.

The main section in this project is the Raspberry Pi which contained the operating system of the smart glasses. The operating system was installed on SD card and it’s based on Emteria OS. Also, Raspberry was connected to USB speakers’ input/output adapter or connected via Bluetooth speakers, if Bluetooth is configured well. As an independent idea of other smart glasses in the global market, I added some features to make this project different from similar projects. There will be two types of features installed in the operating system: the regular

features such as streaming videos, live, launching maps, access to emails, calls, messages and creating user interface and the special features such as installing a voice assistance to control the rest of features, a sensor which is programmed to inform handicap persons when there is a strange body getting closer. It will inform the user by displaying a message or by the voice assistance, which will be helpful for deaf and blind users.

Thr next step is about displaying the features that I mentioned before. I need to use a 3d printed case to fit in all the components which are required to display. I explained in steps how the components are organized and how they are functioning. In the middle, I based the HDMI LCD and connect it via: Raspberry Pi by cables. Next, to reflect the screen of the HDMI LCD, I had to set a magnifying glass to produce an enlarged image to the transparent glass that will appear in front of the glasses.There will be a side-botton to control a voice controller in case the statup doesn't work, and a sensor. Figure 10 below shows how the final project looks like.

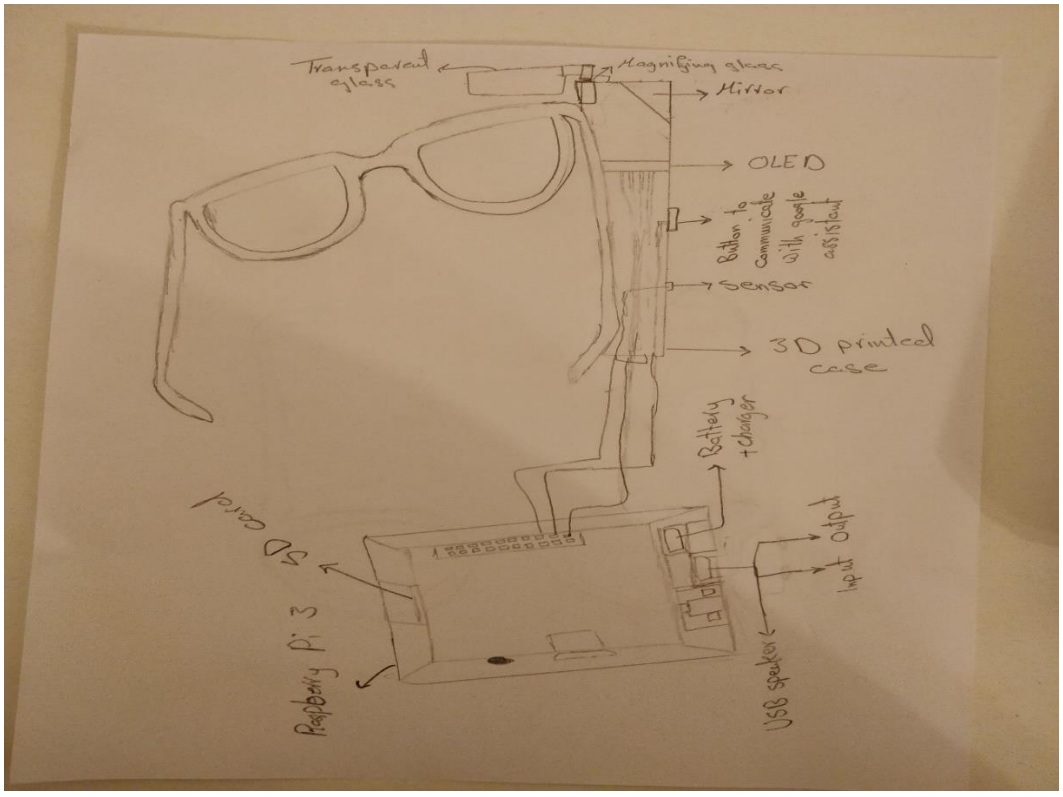

Figure 10. Project's hardware design

The last components used in this project is an eyeglasses. I tried the project on a sunglass but the final result will work well on eyeglasses, it depends on what kind of lense there is and thickness.

3.1 Rapberry Pi Connecting

The components required are:

- Raspberry Pi 3
- A computer (Mac, PC or Linux) with a stable internet connection
- Micro-USB cable
- Ethernet cable
- MicroSD card reader
- 64 GB microSD card
- HDMI LCD
- HDMI adapter
- USB speaker Adapter (I/O)
- Power supply
- Pi camera

Raspberry Pi is the main component of the project, where it has an entrance to insert the card that contains the operating system. The SD card was programmed using a PC with many special programs to install operating systems and format the SD card. The keyboard and mouse were also connected via USB input as well as audio I/O. One of the advantages of smart glasses is the use of a monitor connected via the HDMI port to get a high-resolution image and clear so that the user can see it and because there is no smaller display than expected to match the display ports in Raspberry Pi. A 3.5 LCD screen was used to display it. Because there are programs require to install a camera in the device. The camera was connected using a special display strip connected to the display port on the board. The final step, the device was connected to a power supply to start booting the operating system.

3.2 Smart Glasses design

After completing the process of connecting the components of the device, as well as install the software and settings for smart glasses. The cover is designed to insert the device into it to fit the components connected to the device. It is mounted on a plain glasses body from the right side and the camera is installed on the left side of the glasses as it shown on figure 11.

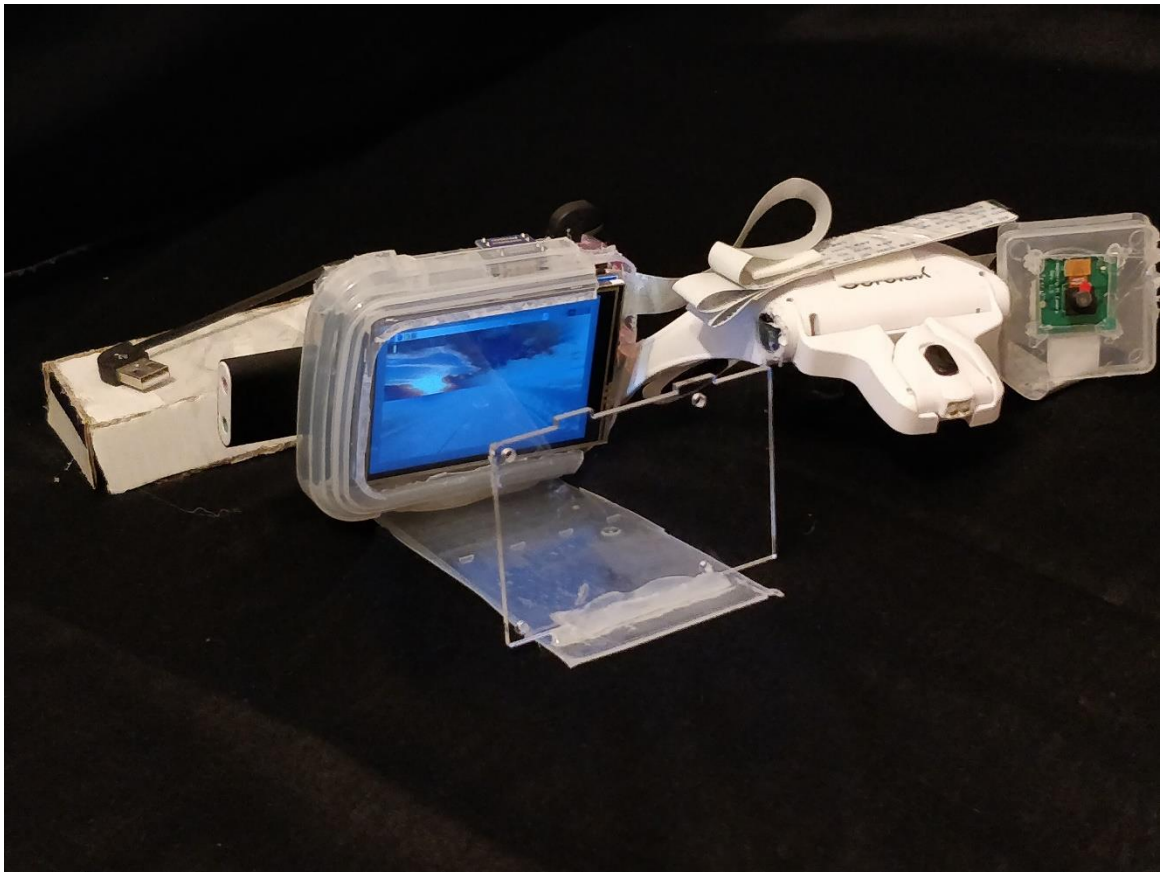

Figure 11. Final result of the smart glasses project

One of the parts that relied on the initial design is to place a reflective glass in front of the screen connected to the device where it is in front of the user's eyes, as it allows the user to see the surrounding environment clearly without obscuring the vision completely.

4 SOFTWARE FOR SMART GLASSES

The major part of the project based on installing an operating system, configuring devices and programming the features. In the reference section, there are a couple of links and guides on how the installation is done and what platforms I needed to use during the whole project. The project contained two operating systems. It based on the Emteria operation system which is the only platform for industrial HMI with a focus on usability. Emteria.OS is the first proper Android release for the Raspberry Pi. Though there is already the RTAndroid, an older solution, Emteria.OS is a full build of Android, and you can get it for free to use as an individual. But because some of the features that needed to be installed on the smart glasses have issues that are not supported by the OS version, I installed another operating system to prove the concept of the project such as the Raspbian operating system.

4.1 Emteria OS

The emteria.OS is released in the form of multiple separate packages, which have to be combined on the storage of the Raspberry Pi. Emteria OS provides a graphical installer that can be executed on the PC or laptop and performs all required steps automatically (Emteria 2019).

The operating system needs to be installed on the SD card. It requires to have Emteria account to download the image file which is cost about EUR 19 as a personal account. The package has full installer tool, it requires only to set up the settings that fit with the device. As soon as the installer starts, processed with the installation of emteria.OS as described below.

Figure 12. Emteria.OS installer process

I used the same username and password into emteria account to login. I had to select the target hardware platform I would like to install emteria.OS by clicking ”Show changes” to see the changes for the current version as shown in figure 13 below.

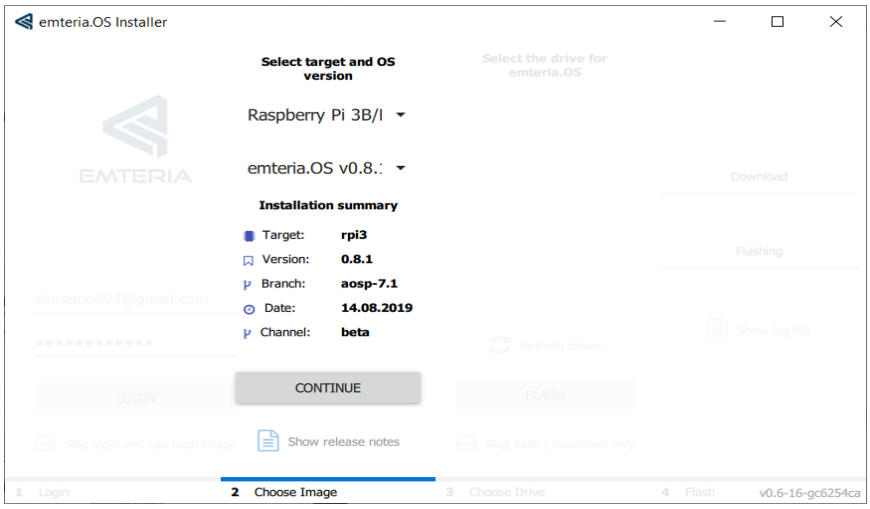

Figure 13. Selecting the board model

By making sure the target data storage or the corresponding internal memory is properly connected to the computer and choosing one of the listed drives to start the installation. The size of packages can vary depending on the device and emteria.OS version. The installer will use the full data storage and partition it accordingly. After choosing the model type and determining the size of memory, I pressed to continue to flash the operating system to the SD card. When it was ready, I took the SD card and inserted to the Raspberry Pi board and turn on the

power. In case if I don’t buy a license and use a free evaluation license, the system reboots after eight hours and an evaluation license notice pop up now and then. The personal license cost about EUR 19, one-time payment for unlimited use of emteria.OS on a single device for personal purposes.

To modify HDMI settings in /boot/config.txt, I mounted the SD-card in Windows bow, edit emteria.OS’s /boot/config.txt ane enabled safe HDMI configuration.

```
hdmi_safe=1

#hdmi_group=2
#hdmi_drive=2
#hdmi_mode=82
#hdmi_cvt=1024 768 60 6 0 0 0
#hdmi_force_hotplug=1
disable_overscan=1

dtoverlay=vc4-kms-v3d,cma-256
avoid_warnings=2

dtparam=sd_force_pio=on
dtparam=i2c1=on
dtparam=i2c_arm=on

dtparam=audio=on
audio_pwm_mode=2

initial_turbo=30
start_x=1
kernel=u-boot.bin
```

Then, saved the changed /boot/config.txt file, unmount the SD card. Alternatively, I might want to configure HDMI parameters manually. This is a tricky process that only worked for me with a dated non-HD resolution display. The next step inserted the emteria.OS SD card into Raspberry Pi connected a display, keyboard, and mouse and then power up. The Raspberry was connected to the internet via WiFi by choosing a WiFi network and enter the password as usually in Android.

The emteria.OS comes without Google Play Store which I need to install required apps and unlock Synthesia. To install Google Play, In the EmteriaOS/Raspberry PI browser, I went to https://opengapps.org/ and select ARM, 7.1, pico in my case

there was a problem to install other settings which are not supported by emteria.OS. When the settings were ready I clicked the download button and wait a while until the download is finished. I had to power off the app icon and then press on Recovery. After Android reboots, I swiped to allow modifications and clicked install. By going to the Download folder and click the downloaded open_gapps-arm-7.1.-pico-201991.zip, swiped to confirm Flash and clicked reboot system after flashing is finished. After reboot, I started Google Play and configured the google account. It will take a while, Android is busy with system optimization in progress that takes a couple minutes and few reboots. Android was rather unresponsive all the time (Yolkhovyy 2018).

4.2 Raspbian stretch OS

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilitities that make the Raspberry Pi run. Raspbian provides more than a pure OS, it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installion on the Raspberry Pi. The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012 (Raspbian 2019).

A Raspberry Pi 3 board will load an operating system from a microSD card. So I needed to have a computer with a SD card reader. After that, I proceed to download a version of Raspbian Stretch latest version 2019 for the Raspberry Pi 3. There were three versions of Raspbian Stretch that are released on 2019: Raspbian Stretch with desktop and recommended software, Raspbian Stretch with desktop and Raspbian Stretch Lite. Since the first option contains a lot of software that are not needed for Python 3 development, I had to choose Raspbian Stretch with desktop (Zach 2019).

The .img file is the operating system image that I used for installing Raspbian Stretch onto a microSD card. I used Win32DiskImager to install operating system images onto a microSD card. After installing the Raspbian operating system onto

the microSD card, I needed to put it into the microSD card slot of the Raspberry Pi 3 board. Next, I connected the HDMI port on the Raspberry Pi 3 board to the monitor with the HDMI adapter. That enables to get visual feedback from the rest of the configuration steps. The Raspberry Pi 3 has four USB ports for external peripherals. To be able to surf the web through the web browser in Raspbian to performing other computing tasks, I had to connect the mouse and the keyboard to two of the USB ports.

Once everything was connected, I had to boot into the Raspbian operating system to perform further configurations. To power up the Raspberry Pi, I needed to plug in the micro USB power adapter to the wall socket. After powering on the wall socket, the Raspberry Pi started to load the Raspbian operating system. The Raspbian operating system had successfully booted up, a program brought through several configurations. I went through the instructions to indicate timezone, a keyboard type, connect to WiFi and get updates (Techcoil 2018).

4.3 Installing Google assistant

Spoken dialogue systems or voice-controlled assistants are devices that can respond to multiple voices, regardless of the answer, can execute several commands or can provide an answer, so imitating a natural conversation. One of the most important parts of such an assistant is represented by speech recognition also called speech to text translation because it transforms human voice into a string of data that can be interpreted by the computer. In recent years, cloud-based speech recognition systems have been developed a lot. All elements of a voice-controlled assistant is placed in the cloud. The most important ones from this category of assistants are Apple, Siri, Google Assistant and Amazon Alexa. They are present in most smartphones and are based on artificial intelligence elements such as deep learning and neural networks.

Google has launched in May 2017, a project called AIY “Artificail Intelligence Yourself” voice kit which contains mainly a hardware on top “HAT” that can be

connected on the GPIO connector of a Raspberry Pi 3 microcomputer, a microphone , a speaker and an Arcade style button. The last three components must be connected to the HAT by suitable wires and connectors. The function of this combo device, voice assistant. By using Google's cloud facilities it can answer questions asked by the user. It can be further improved to execute some commands because the GPIO pins of the Raspberry Pi are available on the HAT so it can be interfaced with other devices as remote controls, actuators, sensors. The software part of this project contains sereral Python files and includes on the image of the Raspbian operating system which is available on the AIY Voice Kit site (Mischie, Matiu-lovan, Gasparesc 2018).

In this project, I installed the implementation of a Google Assistant on a Raspberry Pi without the AIY Voice Kit. I used a USB microphone and a speaker for the 3.5 mm audio output of the Raspberry Pi. All the knowledge of the operating system such as accessing the audio devices or knowledge of Python language to modify the initial form of the software part. The first step, I get started with setting up the Google Assistant code on the Raspberry Pi itself, I had to register and set up a project on the Google Actions Console. With my own Google account, I was ready to go to the Google Console Actions dashboard or use the URL below which direct access to the dashboard (Gus 2018).

`https://console.actions.google.com`

Once I logged into the account, I was able to click on "Add/import project" buttons, that will pop up a new page that contains a box asking to enter a project name, country and language. Figure.14 shows the created project.

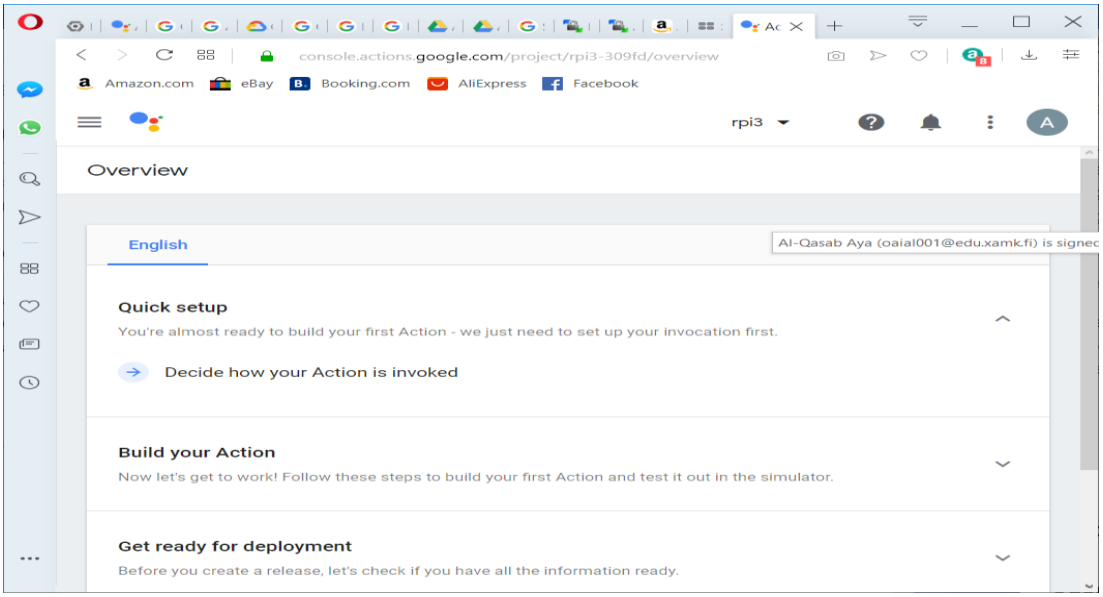

Figure 14. Console actions account dashboard

In a new tab, I went to the Google developers console and enable the Google Embedded Assistant API. But before enabling I needed to make sure that I selected the right project I created in the previous step and then click on the Enable button. Next, the device needs to be registered in Google Console Actions, by pressing on Device Registration to continue. In this step, it required to add a product name, manufacturer name and set a device type. For product name, I just set as a simple descriptor of what I'm using this for, which is a simple that same as the project name "rpi3". The manufacturer name does not matter, so I wrote a short sentence but it's not required. Lastly, I set the device type as a speaker as it felt it matched best what I intend on using the Google Assistant API for on the Raspberry Pi. Once everything was set, I copied the Device Model ID and pressed the Register Model button to continue. After registering the model I was able to be taken to the Download credentials screen. This screen is crucial as the provided credentials file is what I need for the Raspberry Pi based Google Assistant to talk with the server.

To get the credentials file, I clicked the "Download OAuth 2.0 credentials" button. The file was saved and downloaded to my own computer, so I can easily move it to the Raspberry Pi. But before closing the box, I should finish setting up the

model by saving traits. Once everything is done, I was able to have what I need to set up the Google Assistant on the Raspberry Pi board as shown below.

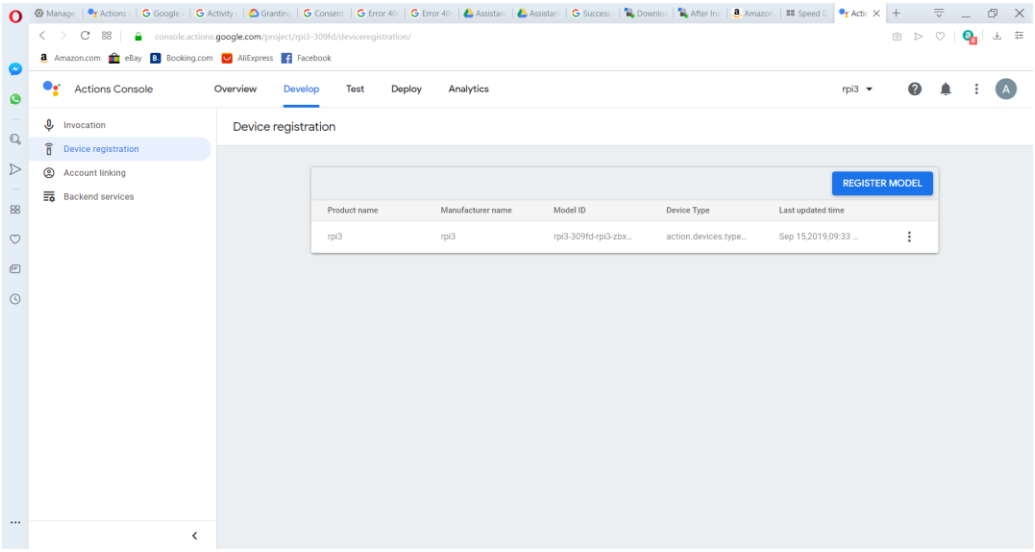

Figure 15. Registered model for Google Assistant API

Also, I needed to configure the OAuth consent screen, without it Google won't let me authorize the Raspberry Pi Google Assistant app. I was able to do it by visiting the API Credentials OAuth consent screen and make sure that the project is selected in the top dropdown box. I checked everything was configured as it required and saved it. It is so important to add an email to OAuth because if there is no email, the error will appear during installation. Finally, I needed to activate the following activity controls to ensure that the Google Assistant API works correctly.

- Web and App Activity
- Location History
- Device Information
- Voice and Audio Activity
-

`https://myaccount.google.com/activitycontrols`

After setting up an account on the Google Actions Console, I was able to start configuring the Raspberry Pi board using the terminal command line. The first set up was to configure the audio for it. The Google Assistant SDK that I will be using during the project has some strict requirements for it to work correctly. To get started with setting up the audio on the Raspberry Pi, I must first obtain the card

and device numbers for various inputs and outputs. By locating the USB microphone by utilizing the following command: arecord -l. Also, I had to locate the speaker by utilizing the following command: aplay -l. The Raspberry Pi's 3.5mm-jack is typically labeled as Analog or bcm2835 AlSA, with the HDMI output being identified as bcm2835 IEC958/HDMI.

Now that I have grabbed the device and card numbers for both the microphone and audio output, I needed to create a file named .asoundrc in the pi users home directory. The .asoundrc file helps by defining which audio devices that the audio driver should be utilizing by running the following command.

```
nano /home/pi/.asoundrc
```

Adding the following script and save it by pressing Ctrl+X then Ctrl+Y.

```
pcm.!default {
  type asym
  capture.pcm "mic"
  playback.pcm "speaker"
}
pcm.mic {
  type plug
  slave {
    pcm "hw:1,0"
  }
}
pcm.speaker {
  type plug
  slave {
    pcm "hw:1,0"
  }
}
```

Before starting to go deeper, I must test the sound to ensure that the audio is working and both input/output are connected to the Raspberry Pi with three commands.

```
speaker-test -t wav

arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw
out.raw

aplay --format=S16_LE --rate=16000 out.raw
```

With the Google Assistant API now configured and set up there are a few things I needed to do. First of all, I have to update the Raspberry Pi's packages list by running the following command.

```
sudo apt-get update
```

Once the Raspberry Pi has finished updating, I can then proceed with setting up everything I need for running the Google Assistant API. On the Raspberry Pi I created a new file to store the credentials I downloaded earlier on the computer by running the following command.

```
mkdir ~/googleassistant
nano ~/googleassistant/credentials.json
```

Within this file, I needed to copy the contents of the credentials file that I downloaded to the computer. I opened the .json file in a text editor and press Ctrl+A then Ctrl+C to copy the contents. Once I have copied the contents of the credentials over to the nano session. I could then save the file by pressing Ctrl+X then Ctrl+Y and finally hitting Enter. Now with the credentials file have been saved safely to the Raspberry Pi, I started installing some of the dependencies I rely on. By running the following command to install Python3 and the Python 3 Virtual Environment to the Raspberry Pi.

```
sudo apt-get install python3-dev python3-venv libssl-dev libffi-dev
libportaudio2
```

When installing was ready, I enabled python3 as a virtual environment variable by running the following command on the Raspberry Pi.

```
python3 -m venv env
```

With that now enabled I could go ahead and ensure that I have installed the latest versions of pip and the setuptools with the command below.

```
env/bin/python3 -m pip install --upgrade pip setuptools –upgrade
```

To get into the new Python environment that I have set up I should run the following command in terminal.

```
source env/bin/activate
```

Now that I have all the packages I need to install the Google Assistant Library, to do this I have to run the following command to utilize pip to install the latest version of the Python package.

```
python3 -m pip install --upgrade google-assistant-library
python3 -m pip install --upgrade google-assistant-sdk[samples]
```

The final step to authorize the system by installing the Google authorization tool to the Raspberry Pi. This package will allow to authenticate the device and give the right to be able to make Google Assistant queries for the Google Account. With the command below, I installed the Python authorization tool.

```
python3 -m pip install --upgrade google-auth-oauthlib[tool]
```

When libraries was installed successfully, I needed to run it by running the command on the Raspberry Pi which will create a URL I needed to go to in the web browser.

```
google-oauthlib-tool --client-secrets ~/googleassistant/credentials.json
\
--scope https://www.googleapis.com/auth/assistant-sdk-prototype \
--scope https://www.googleapis.com/auth/gcm \
--save --headless
```

I was presented with the text “Please visit this URL to authorize this application:” followed by a very long URL. I have to make sure to copy the URL entirely to the web browser to open it. On the screen I logged in to Google account with a set up API key. I was presented with a screen with the text “Please copy this code, switch to your application and paste it there” followed by a long authentication code. I copied the authentication code and paste it back into the terminal session and pressed Enter. The authentication was accepted, and I got the text below in the command line.

“**credentials saved: /home/pi/.config/google-oauthlib-tool/credentials.json**”

The authentication credentials was saved, but Google still requires to agree to some stuff. With this command by replacing the “projectid” with the id of the project I created before. Also, make sure to replace “deviceid” with the device ID that I obtained earlier in the beginning, pressing Enter in the terminal and start speaking an action such as “What is the time”.

```
googlesamples-assistant-pushtotalk --project-id rpi3-309fd --device-model-id rpi3-309fd-rpi3-zbxezr
```

Now finally with fully authorized the Raspberry Pi, I was able to walk through the steps on how to run the Google Assistant software without having to go through the entire steps. To begin with whenever I start a new terminal session I need to put it back into the environment that I set up the Google Assistant software in by running the following command.

```
source env/bin/activate
```

The (env) should appear at the front of each line which means that I can once again make calls to the Google Assistant samples. To start up the push to talk sample I need to run the following command to save time.

```
googlesamples-assistant-pushtotalk
```

The Google Assistant was installed successfully set up but I could not startup with boot. There were some errors during the operation which effect on the software and stopped it of executing.

4.4 Installing OpenCV and Python

Python is a general-purpose programming language started by Guido van Rossum, which became very popular in a short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in

fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives two advantages: first, the code is as fast as the original C/C++ code and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation (Mordvintsev, K 2017).

The support of Numpy makes the task easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy array. Any kind of operation could be in Numpy such as combine it with OpenCV. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems. The OpenCV and Python were used to create image detection and American Sign Language features for the smart glasses (Mordvintsev, K 2017).

By opening up a terminal and started to update and upgrade installed pakages followed by updating the Raspberry Pi firmware (Rosebrock 2015).

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo rpi-update
```

Installing the required developer tools and packages. Both build-essential and pkg-config are likely already installed, but just in care they are not, there must include apt-get to the command. That would take about two minutes to be installed.

```
$ sudo apt-get install build-essential cmake pkg-config
```

Next, installing the necessary image I/O packages. These packages allow to load various image file formats such as JPEG, PNG, TIFF.

```
$ sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
```

To install the GTK development library which is used to build Graphical User Interfaces (GUIs) and is required for the highgui library of OpenCV which allows to view images on the screen.

```
$ sudo apt-get install libgtk2.0-dev
```

Also, it is important to install the necessary video I/O packages which are used to load video files using OpenCV.

```
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

Some of the libraries are used to optimize various operations within OpenCV need to be installed by using the following command.

```
 $ sudo apt-get install libatlas-base-dev gfortran
```

The pip must be installed inside the virtualenv and virtualenvwrapper. Then update ~/.profile file to include the line which contains #virtualenv and vritualenvwrapper, reload the file and create a computer vision virtual environment.

```
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python get-pip.py
$ sudo pip install virtualenv virtualenvwrapper
$ sudo rm -rf ~/.cache/pip
```

~/.profile:

```
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

reload ~/.profile:

```
$ source ~/.profile
$ mkvirtualenv cv
```

It requires to install neither Python 2.7 or Python 3 tools or both. I installed both because in some features it requires to use Python 2.7 tool as default and some needs to install Python 3 tool. By the following command I was able to install Python library.

```
$ sudo apt-get install python2.7-dev
```

I also needed to install NumPy since the OpenCV Python bindings represent images as multideimensional NumPy arrays.

```
$ pip install numpy
```

It require to download OpenCV and unpack it. Also, setup the build and compile OpenCV.

```
$ wget -O opencv-2.4.10.zip http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.10/opencv-2.4.10.zip/download
$ unzip opencv-2.4.10.zip
$ cd opencv-2.4.10

$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D BUILD_NEW_PYTHON_SUPPORT=ON -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON  -D BUILD_EXAMPLES=ON ..

$ make
```

It took about two hours to be installed. The virtual environment needs to be activate so OpenCV is compiled against the virtual environment Python and NumPy. Otherwise, OpenCV will be compiled against the system Python and NumPy which can lead to problems down the line. In final step, I was able to install OpenCV.

```
$ sudo make install
$ sudo ldconfig
```

OpenCV should now be installed in /usr/local/lib/python2.7/site-packages. But in order to utilize OpenCV within a cv virtual environment, first I needed to sym-link OpenCV into the site-packages directory:

```
$ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/
$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
$ ln -s /usr/local/lib/python2.7/site-packages/cv.py cv.py
```

Finally, I was able to test the OpenCV and Python installation.

```
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__
'2.4.10'
```

OpenCV and Python is now successfully installed on the Raspberry Pi and I'm able to go through the smart glasses features.

4.5 American Sign Language

Sign language is a visual language used by people with speech and hearing disabilities for communication in their daily conversation activities. It is completely an optical communication language through its native grammar, be unlike fundamentally from that of oral languages. The sign language (SL) is made by specifications of hand and facial idioms to express their views and thoughts of speech and hearing disabled persons with normal speech and hearing people. Most of the normal persons may not clearly understand the sign language. For that reason, there is a massive communication gap between the deaf communities with the general public. There is an inevitability of technology support for speech impairment to patronage speech impaired persons in their daily activities all the time. By the advancement in science and technology, the smart glasses came with a feature that can interpret gesture signs into humanoid or machine decipherable text. This smoothens the conversation between normal and impaired people (S, Shivashankara and S, Srinath 2018).

The gesture is a non-verbal communication technique generated by the movement of the limb or body as an expression of thought or feeling. Gestures can be static, where the user assumes a certain type of configuration or dynamic,

defined by movement. Hand Gesture can be subdivided into two types, firstly global motion where the entire hand moves whereas in the second one local motion only the fingers move. A sign language is a language where people use gestures instead of speech to communicate with others. The main advantage of using a visual input is the possibility of communication without the need for any physical contact with the device to be controlled. This can be achieved by eliminating input devices like joysticks, mice, keyboards and allowing the body to give signals to the computer through gestures (Hussain, Sarma, Talukdar 2014).

Compared to voice commands, hand gestures are advantageous in some situations like noisy environments, sound prohitbited areas and transferring quantitative information. A generic block diagram of the gesture recognition process is as shown in figure 16. Image acquisition is the first stage of a gesture recognition system where a set of image frames are captured by using a web-camera. Segmentation procedure partitions an image into its constituent parts or objects (Hussain, Sarma, Talukdar 2014).

Figure 16. Generic block diagram of gesture recognition system

Tracking is another important aspect of the gesture recognition system which is used to know the consecutive position of the fingers/hands of the user to represent the object in 3D. The feature extraction is the process of transformations, zones and geometric features. The hand portion has more convex and concave contours rather than the arm part and it has the information that required.

The system is implemented using the OpenCV library based on Python programming language. The experiment is running on the Raspberry Pi and the video image is captured by the Pi camera with good resolution. Since the hand

gesture can be in various different types, I defined some certain types of postures used in American Sign Language to show the experiment results. In the project, the arm can be straight or tilt and hand also can be straight or tilt. The input frame is a video captured using a camera. The video is converted to HSV color space from RGB color space to get rid of the illumination problem. Then by applying the OpenCV face detection algorithm to detect and delete face region. A track bar is also implemented which allows the user to play with different thresholds.

After proper segmentation, the image is converted into binary image over which some morphological operations like erosion and dilation are applied to make it smooth and noise-free as shown in figure 17.

Figure 17. I/P is converted to binary and Morphological operation carried out

The feature was configured by using Python, OpenCV and TensorFlow for training InceptionV3 model, a convolutional neural network model for classification. The requirements for this project were OpenCV which already installed in the Raspberry Pi, installing TensorFlow, Matplotlib and Numpy. For a Raspberry Pi, I used pip command to install the script (Loicmarie 2018).

```
pip3 install -r requirements.txt
```

The model needed to be trained by using the following command. In case the provided dataset is used, it may take up to three hours.

```
python3 train.py \
  --bottleneck_dir=logs/bottlenecks \
  --how_many_training_steps=2000 \
```

```
  --model_dir=inception \
  --summaries_dir=logs/training_summaries/basic \
  --output_graph=logs/trained_graph.pb \
  --output_labels=logs/trained_labels.txt \
  --image_dir=./dataset
```

Then, when installing is ready to test, I used the command below for testing.

```
python3 classify.py path/to/image.jpg
```

There is a command which is required to use webcam.

```
python3 classify_webcam.py
```

I have proposed a method for bare hand posture recognition followed by accurate palm and fingertip position estimation based on hand contour. Fore-arm can also be included in the contour and the system has good toleration against rotation and tiltation of hand.

4.6 TensorFlow Object Detection

Creating accurate Machine Learning Models that are capable of identifying and localizing multiple objects in a single image remained a core challenge in computer vision. But with recent advancements in Deep Learning, Object Detection applications are easier to develop than ever before. TensorFlow's Object Detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. Object Detection is the process of finding real-world object instances like car, bike, TV and humans in still images or videos. It allows for the recognition, localization and detection of multiple objects within an image which provides users with a much better understanding of an image as a whole. It is commonly used in applications such as image retrieval, security, surveillance and advanced driver assistance system (Kurt 2019).

There are different ways to create object detection that could be done by Feature-based Object Detection, Viola Joes Object Detection, SVM Classification with HOG Features and Deep Learning Object Detection. In this project of smart glasses, I used the Deep Learning Object Detection as Tensorflow uses Deep Learning for computation. Object Detection can be also used for people counting. It is used for analyzing store performance or crowd statistics during festivals. These tend to be more difficult as people move out of the frame quickly.

Tensorflow is Google's Open Source Machine Learning Framework for dataflow programming across a range of tasks. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. Tensors are just multidimensional arrays, an extension of 2-dimensional tables to data with a higher dimension. There are many features of Tensorflow which makes it appropriate for Deep Learning (Kurt 2019).

I did not need the graphical system, but it required to have the latest Raspbian Stretch operating system image I installed in the beginning of the project already. Then I needed to check that Pi camera is enabled by opening a terminal and run a command below.

```
sudo raspi-config
```

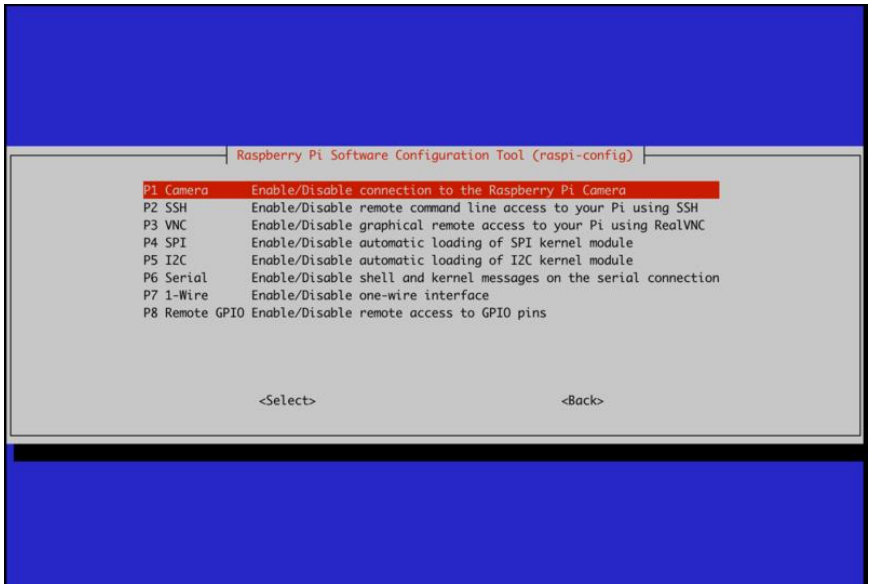

Figure 18. Camera configuration

After checking that cmera was enabled, it required to update and upgrade the system, libraries and installing python3- pip if it’s not available.

```
sudo apt update
sudo apt-get update
sudo apt-get upgrade

sudo apt-get install -y python3-pip

pip3 install --upgrade setuptools
```

Finally, I rebooted the system and did the initial test with the camera which should display the camera sees on the display.

```
raspistill -t 0
```

For TensorFlow, there are a few depemdency requirements to installed in the python environment, it was done by the following command.

```
pip3 install virtualenv Pillow numpy pygame
```

The program originally was written by Leigh Johnson that uses the MobileNet v2 model to detect objects, so it required to clone it by using git clone.

```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ git clone --depth 1 https://github.com/adafruit/rpi-vision.git
Cloning into 'rpi-vision'...

remote: Enumerating objects: 99, done.
remote: Counting objects: 100% (99/99), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 99 (delta 19), reused 64 (delta 18), pack-reused 0
Unpacking objects: 100% (99/99), done.
Checking out files: 100% (80/80), done.
pi@raspberrypi:~ $ cd rpi-vision
pi@raspberrypi:~/rpi-vision $ python3 -m virtualenv -p $(which python3) .venv
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/pi/rpi-vision/.venv/bin/python3
Also creating executable in /home/pi/rpi-vision/.venv/bin/python
Installing setuptools, pip, wheel...
done.
pi@raspberrypi:~/rpi-vision $ source .venv/bin/activate
(.venv) pi@raspberrypi:~/rpi-vision $
```

Figure 19. Install rpi-vision

Inside the environment, I downloaded and installed Tensorflow 2.0 RC0.

```
wget https://github.com/PINTO0309/Tensorflow-bin/raw/master/tensorflow-2.0.0rc0-cp37-cp37m-linux_armv7l.whl

pip3 install --upgrade setuptools
```

Figure 20a. Install TensorFlow 2.0

By typing the following command, it took a couple minutes to finish installing TensorFlow 2.0.

```
pip3 install tensorflow-2.0.0rc0-cp37-cp37m-linux_armv7l.whl

pip3 install -e .
```

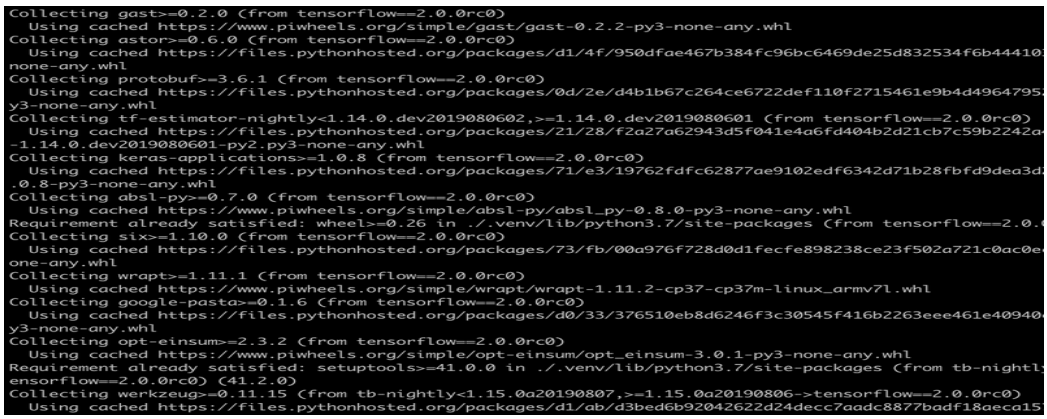

Figure 20b. Install TensorFlow 2.0

When the installation was ready, I had to reboot the system and run the detection software. First I wanted to run as root so that Python can access the Frame Buffer. Then activate the virtual environment again.

```
sudo bash
cd rpi-vision && . .venv/bin/activate
```

When program was run which displayed the object that sees on screen type by the following command.

```
python3 tests/pitft_labeled_output.py –tflite
```

The result I got

By starting holding up various items in front of the camera and it should display what it thinks it sees, which is not actually what the item may be. Some items that it’s pretty good about identifying are coffee mugs and animals. Also, it’s possible to hook up a pair of headphones and speakers to the Raspberry Pi and it will

actually tell what it is detecting. There is one similar feature that was also installed before on the Raspberry Pi but there was no voice assistant that could tell what kind of object is in front of the user, it's better than the one that I explained how to install because it can detect any kind of objects.

5 CONCLUSION

The original problem in this thesis was the assistant for handicap persons who face problems with their daily life. Therefore, The final goal of this research was a smart glasses product with different integration features and implemented software. The product was delivered in the requested timeframe with all the requirements satisfied. The solution required by the commissioner of this thesis presented smart glasses based on a voice assistant for handicap persons which contained few features that can provide quick help for them. The main features were provided in this project such as image detection which can help the visually impaired track and recognize the objects in front based on a voice assistant, but the problem was with the operating system that didn't support the application. So it required to install the feature on different OS environments without voice assistants. The other feature was installed for deaf persons which provide help for normal persons to understand the person in front if he/she was deaf, the smart glasses will use the camera to translate the sign language and show on screen the set of letters were detected from the movement of hands. The application was based on American Sign Languages.

During the development process, a certain skill set was gained by the author of this work and understanding of the Internet of Things that contains two parts the hardware and software processes in the production environment knowledge was gained. In this field, the informal comparison of different options for development was made by the author of this work's implementation. All the work including the research, learning and implementation took about five months.

The solution provided has certain areas that could be improved in the future and were not achieved so due to the lack of time resources and lack of knowledge of the project's author. In the future, there will be additional improvements to the features such as find solution for the assistant in the image detection application, develop the ASL application to translate full sentence instead of letters, also there is one feature will be added to the smart glasses used to control the glasses using hand gesture recognition, that would be handy to use hands instead of mouse controller.

The design implementation during this thesis implementation was the easiest part. But it took time to get the final design. During the research time, there were few similar projects but with different ideas. The design, in general, includes the Raspberry Pi installed on a normal eyeglasses and 3D printed case. But because the 3D printed cases require a good knowledge in designing and how to use the device and that would take a couple weeks to study and create one case. Therefore I used a simple case of simple items just to prove the concept.

REFERENCES

Due, Brian L. 2014a. The future of smart glasses: An essay about challenges and possibilities with smart glasses.pdf [Accessed March,2019]

Due, Brian. 2014b. The future of Smart Glasses.pdf [Accessed August, 2019]

Ronald T. Azuma. A Survey of Augemented Reality. Pdf [Accessed April,2019]

Stevens, Laura. 2018. What are smart glasses? The much-desired wearable technology explained. www, at: https://home.bt.com/tech-gadgets/future-tech/what-are-smart-glasses-11364214160834 [Accessed October, 2019]

Augemented Reality in real estate,at: https://thinkmobiles.com/blog/what-is-augmented-reality/. [Accessed April, 2019]

Mehdi Mekni, Andr´e Lemieux. Augmented Reality: Applications, Challenges and Future Trends.pdf [Accessed April, 2019]

LIK-HANG, LEE and PAN, HUI. Interaction Methods for Smart Glasses.pdf [Accessed April, 2019]

Raspberry Pi 3, at https://pythonprogramming.net/introduction-raspberry-pi-tutorials/ [Accessed April, 2019]

Raspberry Pi 3 Network : Specs, benchmarks & testing.2016. www, at: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/ [Accessed April, 2019]

Python with Raspberry. 2018. In this article, learn how to use the programming language Python to create projects in Raspbian for your

Raspberry Pi.www, at: https://maker.pro/raspberry-pi/tutorial/how-to-use-python-with-raspberry-pi [Accessed April, 2019]

Python language on raspberry pi. 2018. Getting started with Python programming and the Raspberry Pi.www , at: https://raspberrypihq.com/getting-started-with-python-programming-and-the-raspberry-pi/ [Accessed April, 2019]

Circuit Basics. 2015. How to Write and Run a Python Program on the Raspberry Pi.www, at: http://www.circuitbasics.com/how-to-write-and-run-a-python-program-on-the-raspberry-pi/ [Accessed October , 2019]

Google assistant voice controller. 2019. Google Assistant: Qwik Start - Dialogflow.www, at: https://www.qwiklabs.com/focuses/2196?parent=catalog/ [Accessed August, 2019]

Google Actions. 2019. Build Actions for the Google Assistant.www, at: https://codelabs.developers.google.com/codelabs/actions-1/#1 [Accessed October, 2019]

Draper, Sam. 2018. These Smartglasses Aren't Just Smart They're Also Very Fashionable.www , at: https://www.wearable-technologies.com/2018/12/these-smartglasses-arent-just-smart-theyre-also-very-fashionable/ [Accessed August, 2019]

Nobel, Steve. 2019. The 10 best augmented reality smartglasses in 2019.www , at: https://www.aniwaa.com/best-of/vr-ar/best-augmented-reality-smartglasses/ [Accessed April, 2019]

Aniwaa, Epson. 2019. Epson-Moverio-BT-300-3 smart glasse.www, at: https://www.aniwaa.com/product/vr-ar/epson-moverio-bt-300/ [Updated August, 2019]

Lee, Clifford. 2018. Everysight Raptor.www, at: https://www.cxmagazine.com/review-everysight-raptor-augmented-reality-glasses-gravel-cyclocross [Accessed April, 2019]

Puzak, Tom. 2018. Everysight Raptor review.www, at: https://gearjunkie.com/everysight-raptor-ar-cycling-glasses-review [Accessed August, 2019]

Kerton, Stu. 2018. Everysight Raptor glasses.www, at: https://road.cc/content/review/251030-everysight-raptor-glasses [Accessed August, 2019]

D'Angelo, Matt. 2019. Google Glass Enterprise Edition 2: Is It Good for Business?.www, at: https://www.businessnewsdaily.com/10313-google-glass-enterprise-business.html [Updated August, 2019]

Wang, Jules. 2019. Google Glass refuses to die Enterprise Edition 2 gives it a new lease on life.www, at: https://www.androidpolice.com/2019/05/20/google-glass-refuses-to-die-enterprise-edition-2-gives-it-a-new-lease-on-life/ [Accessed August, 2019]

Aniwaa, Solos. 2019. Kopin SOLOS.www, at: https://www.aniwaa.com/product/vr-ar/kopin-solos/ [Accessed April, 2019]

Essential. 2019. Kopin SOLOS glass.www, at: http://essentialdesign.com/case-study/kopin-solos/ [Accessed August, 2019]

Rubin, Peter. 2017. Meta 2.www, at: https://www.aniwaa.com/product/vr-ar/meta-2/ [Accessed April, 2019]

CES.2016. ODG R-7 Delivers the Most Advanced Platform for Augmented Reality Head-worn Computing Experiences.www, at:

https://www.businesswire.com/news/home/20160104005518/en/ODG-R-7-Delivers-Advanced-Platform-Augmented-Reality [Updated August, 2019]

Dynabook. 2019. Toshiba dynaEdge AR100 Viewer.www, at: https://us.dynabook.com/smartglasses/products.html [Updated August, 2019]

Harding, Scharon. 2019. Vuzix Blade Smart Glasses Review: AR Fun Over Fashion.www , at: https://www.tomshardware.com/reviews/vuzix-blade-ar-smart-glasses-consumer,5667.html [Accessed April, 2019]

Prospero, Mike. 2019. Vuzix Blade Review: These $1,000 AR Glasses Are Fun But Frustrating.www, at: https://www.tomsguide.com/us/vuzix-blade,review-6065.html [Accessed April, 2019]

Aniwaa, X1. 2019. ThirdEye Gen X1.www, at: https://www.aniwaa.com/product/vr-ar/thirdeye-x1/ [Accessed April, 2019]

ThirdEye Gen, Inc. 2018. ThirdEye Gen showcases X1 Augmented Reality Smart Glasses™ with Enterprise AR Software.www, at: https://www.prnewswire.com/news-releases/thirdeye-gen-showcases-x1-augmented-reality-smart-glasses-with-enterprise-ar-software-300578567.html [Accessed August, 2019]

Flynn, Conner. 2019. Vuzix M300 Smart Glasses: Hands Free Mobile Computing, at: http://www.gadgetreview.com/vuzix-m300-smart-glasses [Accessed April, 2019]

Emteria. 2019. [emteria.OS] Installation Step-by-Step.www, at: https://help.emteria.com/kb/emteria-os-installation/ [Accessed October, 2019]

Yolkhovyy. 2018. Raspberry PI: How to install Android emteria.OS, Google Play and Synthesia.www, at: http://geomodule.com/sw-eng-

notes/2018/12/23/raspberry-pi-how-to-install-emteria-os-android-google-play-and-synthesia/ [Accessed August, 2019]

Raspbian. 2019. Welcome to Raspbian.www, at: https://www.raspbian.org/ [Accessed October, 2019]

Zach. 2019. Download Raspbian Stretch.www, at: https://howchoo.com/g/nzc0yjzjy2u/raspbian-stretch-download [Accessed October, 2019]

Techcoil. 2018. How to setup Raspbian Stretch on Raspberry Pi 3 for developing Python 3 applications.www, at: https://www.techcoil.com/blog/how-to-setup-raspbian-stretch-on-raspberry-pi-3-for-developing-python-3-applications/ [Accessed October, 2019]

Mischie, Matiu-lovan, Gasparesc. 2018. Implementation of Google Assistant on Rasberry Pi.pdf, at: https://www.researchgate.net/publication/329817651_Implementation_of_Google_Assistant_on_Rasberry_Pi [Accessed October, 2019]

Gus. 2018. Build your own Raspberry Pi Google Assistant.www, at: https://pimylifeup.com/raspberry-pi-google-assistant/ [Accessed October, 2019]

Mordvintsev. Alexander, K. Abid. 2017. OpenCV-Python Tutorials Documentation.pdf, at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjl_dPvgJTlAhVD2aYKHWsWCUcQFjAAegQIABAC&url=https%3A%2F%2Freadthedocs.org%2Fprojects%2Fopencv-python-tutroals%2Fdownloads%2Fpdf%2Flatest%2F&usg=AOvVaw3zyOCJCjMjJ7F7syrUISIu [Accessed October, 2019]

Rosebrock, Adrian. 2015. Install OpenCV and Python on your Raspberry Pi 2 and B+.www, at: https://www.pyimagesearch.com/2015/02/23/install-opencv-and-python-on-your-raspberry-pi-2-and-b/ [Accessed September, 2019]

S, Shivashankara and S, Srinath. 2018. American Sign Language Recognition System: An Optimal Approach.pdf, at: https://www.researchgate.net/publication/326972551_American_Sign_Language_Recognition_System_An_Optimal_Approach [Accessed September, 2019]

Hussain. Imran, Sarma. Kandarpa Kumar, Talukdar. Anjan Kumar. 2014. Hand Gesture Recognition System with Real-Time Palm Tracking.pdf, at: https://www.researchgate.net/publication/272182855_Hand_Gesture_Recognition_System_with_Real-Time_Palm_Tracking [Accessed September, 2019]

Loicmarie. 2018. sign-language-alphabet-recognizer.www, at: https://github.com/loicmarie/sign-language-alphabet-recognizer [Accessed September, 2019]

Kurt. 2019. Object Detection Tutorial in TensorFlow: Real-Time Object Detection.www, at: https://www.edureka.co/blog/tensorflow-object-detection-tutorial/#application [Accessed October, 2019]

M. LeBlanc-Williams. 2019. Running TensorFlow Lite Object Recognition on the Raspberry Pi 4.www, at: https://learn.adafruit.com/running-tensorflow-lite-on-the-raspberry-pi-4?view=all [Accessed September, 2019]