الآية

بسم الله الرحمن الرحيم

قال تعالى:

(لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا ۚ لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ ۗ رَبَّنَا لَا تُؤَاخِذْنَا إِنْ نَسِينَا أَوْ أَخْطَأْنَا ۚ رَبَّنَا وَلَا تَحْمِلْ عَلَيْنَا إِصْرًا كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِنْ قَبْلِنَا ۚ رَبَّنَا وَلَا تُحَمِّلْنَا مَا لَا طَاقَةَ لَنَا بِهِ ۖ وَاعْفُ عَنَّا وَاغْفِرْ لَنَا وَارْحَمْنَا ۚ أَنْتَ مَوْلَانَا فَانْصُرْنَا عَلَى الْقَوْمِ الْكَافِرِينَ)

صَدَقَ اللهُ العَظِيمُ

سورة البقره الآية (28)

DEDICATION

This Project is dedicated to our parents for their emotional and financial support, and for all our brothers, sisters, and friends for encouraging us, and without their love and support this project would not have been made possible. And finally for all who have visually impaired and especially our uncle Al-Imam Muhammad Abdel-Wahab.

ACKNOWLEDGEMENT

First and always, we thank Allah for his guidance and blessings, the reason behind our success. We would like to give our sincere thanks to our supervisor, **Ust. Jalal Abdurrahman Mohamed** for his valuable guidance continues encouragement, and suggestions.

ABSTRACT

The blind people and those who have visually impaired are going through difficulties in recognizing things in the surrounding environment and the embarrassment that the blind person receives from asking people.

We aim to create an intelligent system, imitating the human eye, which transfers different scenes and images to the brain. The brain in turn analyzes the images or scenes and based on previously stored information, the surrounding objects are identified. For this purpose, we use a small device that performs similar to the human brain, called "Raspberry Pi"; it is a small device that analyzes the images and scenes with the help of the camera, which moves the images to the small device. Then, the process of analysis begins through long complex algorithms known as the neural network algorithms. This network analyzes the images into parts to compare them with the most important characteristics of the objects in the images related to the database, through which the images are compared. When ensuring that the characteristics match the mathematical equations programmed in the language of Python, the objects in the image are detected. Finally, the sound of each tool in the database is called, and a message is sent to tell the blind about the tools in front of him.

The project aims to help blind people identify the surrounding things around them, which they see through a small camera, the camera is fixed on the glasses, This technique helps to provide job opportunities for the blind, and to identify things easily and conveniently through a voice message sent to an earphone placed on the blind ear to help him find various items easily and independently.so that it helps in searching things and tools easily without help, which saves time and efforts.

المستخلص

يواجه المكفوفون و ضعاف البصر صعوبات في التعرف على الأشياء في البيئة المحيطة والإحراج الذي يتلقاه الشخص الكفيف من سؤال الناس.

المبدأ الاساسي الذي استندنا عليه في بناء نظام ذكي يشبه العين البشرية والتي بدورها تنقل المشاهد اوالصور المختلفة الى الدماغ. والذي بدوره يقوم بتحليل الصور او المشاهد التي امامه. وبناء على معلومات مخزنة سابقة يتم التعرف على الاشياء المحيطة حوله ، وفي مشروعنا هذا استخدمنا مايشبه عمل الدماغ البشري بشكل جزئي بسيط ، وليس كأداء عمل الدماغ كونها ليست من صنع البشر لكن يتشابه في المبدأ الذي يساعدنا في بناء مشروع النظارة الذكية ويعرف بإسم "Raspberry Pi" وهو جهاز صغير الحجم يقوم بتحليل الصور او المشاهد التي امامه بمساعدة الكاميرا والتي بدورها تنقل المشاهد الى المتحكم ، ومن ثم تبدأ عملية تحليل هذه الصور من خلال خوارزميات معقدة طويلة والمعروفة باسم خوارزمية الشبكة العصبية وهي بدورها تحلل الصور الى اجزاء لتتم مقارنتها بأهم ميزات وخصائص الاشياء الموجود في الصور والمتصلة بقاعدة البيانات ، والتي يتم مقارنة الصور من خلالها ، وعند التأكد من تطابق الخصائص باستخدام معادلات رياضية مبرمجة بلغة البايثون يتم الكشف والتعرف على الاشياء الموجود في الصورة او المشهد ، ومن ثم استدعاء الصوت الخاص بكل أداة موجودة بقاعدة البيانات وارسال رسالة صوتية لتخبر المكفوف عن الأدوات الموجوده أمامه.

يهدف المشروع لمساعدة الاشخاص المكفوفين في التعرف على الأشياء المحيطة حوله، والتي يشاهدها امامه من خلال كاميرا صغيرة ، يتم تثبيتها على النظارة والتي بدورها توفر فرصة عمل للمكفوفين ، والتعرف على الأشياء بكل سهولة ويسر وذلك من خلال رسالة صوتية يتم نقلها باستخدام سماعة أذن توضع على اذن المكفوف بحيث تساعدة في البحث على الاشياء والأدوات بكل سهولة ويسر دون الحاجة لمساعدة الاشخاص الاخرين مما يوفر الوقت والجهد للمكفوف.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

WHO	World Health Organization
LED	Light Emitting Diode
USB	Universal Serial Port
SoC	System on Chip
BLE	Bluetooth Low Energy
GPU	Graphics Process Unit
OpenGL	Open Graphics Library
OpenVG	Open Vector Graphics
ARM	Advanced RISC Machine
RISC	Reduce Instruction Set Computing
LPDDR	Low-power Double Data Rate
SD	Secure Digital
GPIO	General-Purpose Input / Output
HDMI	High-Definition Multimedia Interface
MIPI	Mobile Industry Processor Interface
DSI	Display Serial Interface
SDIO	Secure Digital Input / output
CSI	Camera Serial Interface
VGA	Video Graphics Array
PWM	Pulse Width Modulation
OLED	Organic Light Emitting Diode
CNN	Convolution Neural Network
WIFI	Wireless Fidelity

CHAPTER ONE

INTRODUCTION

1.1 Overview

According to the WHO (World Health Organization) report as of 2012, there were 285 million people estimated to be visually impaired worldwide, of which 246 million had low vision and 39 million were blind. The majority of people with poor vision are in the developing world and over the age of 50.Individuals with a visual disability are difficult to communicate effectively not only with ordinary people, but also with their environment, If the visually impaired people can “see” the world with the help of other devices, they will gain increased independence and freedom in city life, and this is precisely why we built such a smart glass system.Nowadays, engineering solutions are improving our daily life in many ways some of which we do not even notice. Of the most important improvements are those aiming to help people with disabilities to overcome their challenges, and to cope with the changing environment around them.

In our lives, many people are suffering from different diseases or handicaps. But now we will talk about those who have a visual impairment and blind people. These people need some help to make their life easier and better. The main goal of “Smart Glasses” is to help blind people and people who have vision difficulties by introducing a new technology that makes them able to live their life normally. These glasses are provided with technology to detect any object in front of you and convert it into audio by using computer vision and tensor flow. The goal of “Smart Glasses” is to help those people in different life aspects. For example, these glasses effectively helpful in their lifestyle. Blind people and people with vision difficulties can be able to live without any help. “Smart Glasses” encourage blind people or people with vision difficulties to open their self into the world.

1.2 Problem Statement

The blind are going through difficulties of recognizing things in the surrounding environment and the embarrassment that the blind person receives from asking people.

1.3 Objectives

- To design a control system to assist blind people in recognizing objects.
- To learn about intelligent control systems and artificial intelligence.
- To implement computer image processing.

1.4 Methodology

The project is the creation of an image processing and recognition system. The system design includes:

- Study the intelligent control methods.
- Study the microcontroller types.
- Study the mechanism of the digital camera.
- Design image processing subsystem.
- Design text-to-speech subsystem.
- Programming each of the subsystems in Python language and TensorFlow Library.
- Implement the software program into the raspberry pi microcontroller chip.

- Assemble the final design containing the microcontroller chip, camera, headphones, control button, and indicator LED.

1.5 Project Layout

This project consists of five chapters: Chapter One gives an introduction to the project. Chapter Two discusses the literature view and related works of the project. Chapter Three describes system modeling and components. Chapter four handles the simulation and implementation of the project and testing results. Finally, Chapter Five provides conclusions and recommendations.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The human eye is the organ that gives us a sense of sight, allowing us to observe and learn more about the surrounding world than we do with any of the other from sense. We use our eye in almost every activity we perform, whether reading, working, watching television, writing a letter, driving a car, and in countless other ways. Most people probably would agree that sight is the sense they value mare than all the rest.

The World Health Organization (WHO) defines Visual impairment decrease or severe reduction in vision that cannot be corrected with standard glasses or contact lenses and reduce an individual's ability to function at a specific or all tasks. Blindness a severe sight loss, where a person is unable to see clearly how many fingers are being held up at a distance of 3m (9.8 feet) or less, even when they are wearing glasses or contact lenses. However, someone who is blind may still have some degree of vision.

Developing a tool for visually impaired people and developing a computer-aided tool is a developing area. All these systems aim to help the user live his life without the help of a second person. There are several works using computer vision techniques. But there is no existing method that helps to solve the all basic needs of a blind person. This system is designed only for a specific purpose. In this project, we propose a new method that combines the key aspects of some use full methods for assisting the visually impaired person. This new system may solve some of the major problems of blind persons that are still existing.

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do, [1]. Computer vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding. One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects, and providing a semantic description of the scene, [1, 2].

2.2 Control System

A control system is a system, which provides the desired response by controlling the output. The figure 2.1 shows the simple block diagram of a control system. In the following figure, the control system is represented by a single block. Since the output is controlled by varying input, the control system got this name.

Figure 2.1: Control system

2.2.1 Continuous-time and discrete-time control systems

Control Systems can be classified as continuous-time control systems and discrete-time control systems based on the type of signal used. In continuous-time control systems, all the signals are continuous in time. But, in discrete time control systems, there exist one or more discrete-time signals.

2.2.2 SISO and MIMO control systems

Control Systems can be classified as SISO control systems and MIMO control systems based on the number of inputs and outputs present.SISO (Single Input and Single Output) control systems have one input and one output. Whereas, MIMO (Multiple Inputs and Multiple Outputs) control systems have more than one input and more than one output.

2.2.3 Open loop and closed loop control systems

Control Systems can be classified as open-loop control systems and closed-loop control systems based on the feedback path. In open-loop control systems, the output is not fed-back to the input. So, the control action is independent of the desired output. The figure 2.2 shows the block diagram of the open-loop control system.

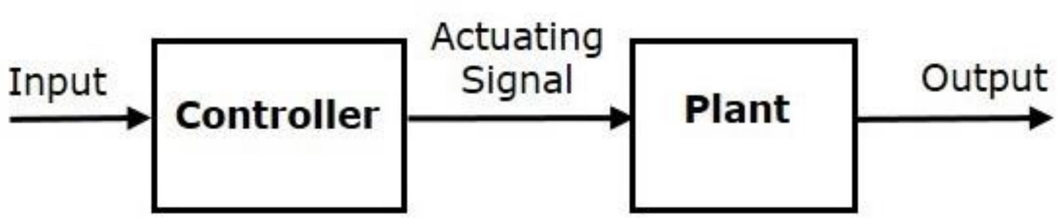

Figure 2.2: Open-loop control system

Here, the input is applied to a controller and it produces an actuating signal or controlling signal. This signal is given as an input to a plant or process which is to be controlled. So, the plant produces an output, which is controlled. In closed-loop control systems, the output is fed back to the input. So, the control action is dependent on the desired output. The figure 2.3 shows the block diagram of the negative feedback closed-loop control system.

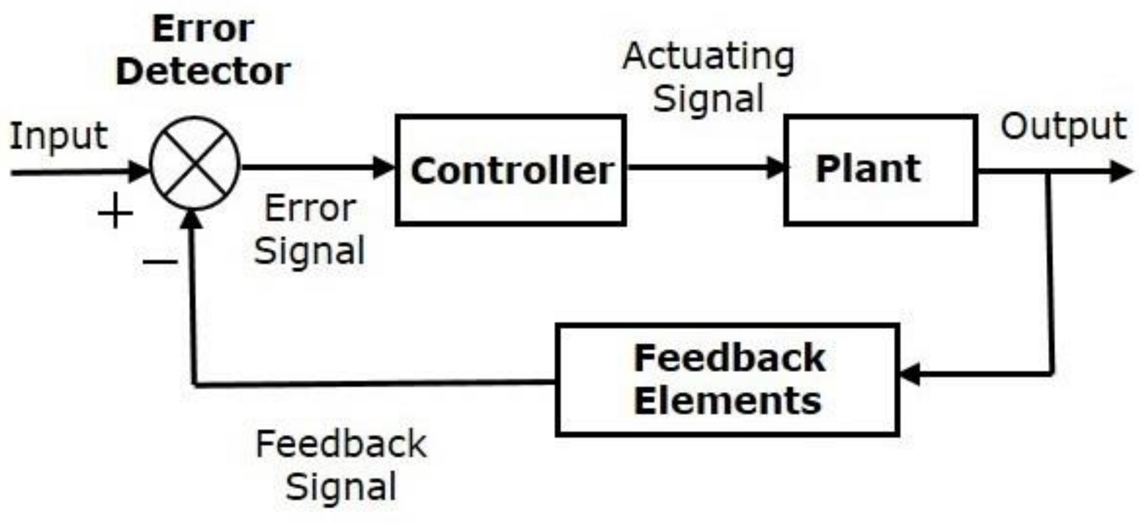

Figure 2.3: Close loop control system

The error detector produces an error signal, which is the difference between the input and the feedback signal. This feedback signal is obtained from the block (feedback elements) by considering the output of the overall system as an input to this block. Instead of the direct input, the error signal is applied as an input to a controller. So, the controller produces an actuating signal which controls the plant. In this combination, the output of the control system is adjusted automatically till we get the desired response. Hence, closed-loop control systems are also called automatic control systems. Traffic lights control system having a sensor at the input is an example of a closed-loop control system.

2.3 Microcontroller

A microcontroller is a small and low-cost microcomputer, which is designed to perform the specific tasks of embedded systems like displaying microwave's information, receiving remote signals, etc. The general microcontroller consists of the processor, the memory (RAM, ROM, and EPROM), Serial ports, peripherals (timers, counters), etc.

2.3.1 Types of microcontrollers

Microcontrollers are divided into various categories based on memory, architecture, bits, and instruction sets. Based on bit configuration, the microcontroller is further divided into three categories.

- 8-bit microcontroller: This type of microcontroller is used to execute arithmetic and logical operations like addition, subtraction, multiplication division, etc. For example, Intel 8031 and 8051 are 8 bits microcontrollers.
- 16-bit microcontroller: This type of microcontroller is used to perform arithmetic and logical operations where higher accuracy and performance are required. For example, Intel 8096 is a 16-bit microcontroller.
- 32-bit microcontroller: This type of microcontroller is generally used in automatically controlled appliances like automatic operational machines, medical appliances, etc.

Based on the memory configuration, the microcontroller is further divided into two categories.

- External memory microcontroller: This type of microcontroller is designed in such a way that they do not have a program memory on the chip. Hence, it is named as external memory microcontroller. For example Intel 8031 microcontroller.

- Embedded memory microcontroller: This type of microcontroller is designed in such a way that the microcontroller has all programs and data memory, counters, and timers, interrupts, I/O ports are embedded on the chip. For example Intel 8051 microcontroller.

Based on the instruction set configuration, the microcontroller is further divided into two categories.

- CISC − CISC stands for complex instruction set computer. It allows the user to insert a single instruction as an alternative to many simple instructions.
- RISC − RISC stands for Reduced Instruction Set Computers. It reduces the operational time by shortening the clock cycle per instruction.

2.3.2 Applications of microcontrollers

Microcontrollers are widely used in different devices such as:

- Light sensing and controlling devices like LEDs.
- Temperature sensing and controlling devices like microwave oven, chimneys.
- Fire detection and safety devices like a Fire alarm.
- Measuring devices like Volt Meter.

2.4 Camera Mechanism

A camera is an optical instrument used to record images. At their most basic, cameras are sealed boxes (the camera body) with a small hole (the aperture) that allow light in to capture an image on a light-sensitive surface (usually photographic film or a digital sensor). Cameras have various mechanisms to control how the light falls onto the light-sensitive surface. Lenses focus the light entering the camera, the size of the aperture can be widened or narrowed to let more or less light into the camera, and a shutter mechanism determines the amount of time the photosensitive surface is exposed to the light.

The still image camera is the main instrument in the art of photography and captured images may be reproduced later as a part of the process of photography, digital imaging, and photographic printing. The similar artistic fields in the moving image camera domain are film, videography, and cinematography.

2.4.1 Aperture

Aperture is a small opening that allows the light to travel inside into the camera. These blades create an octagonal shape that can be opened closed. And thus it makes sense that, the more blades will open, the hole from which the light would have to pass would be bigger. The bigger the hole, the more light is allowed to enter. The effect of the aperture directly corresponds to the brightness and darkness of an image. If the aperture opening is wide, it would allow more light to pass into the camera. More light would result in more photons, which ultimately result in a brighter image.

2.4.2 Shutter

After the aperture, there comes the shutter. The light when allowed to pass from the aperture, falls directly onto the shutter. The shutter is a cover, a closed window, or can be thought of as a curtain. Remember when we talk about the CCD array sensor on which the image is formed. Well behind the shutter is the sensor. So shutter is the only thing that is between the image formation and the light when it is passed from the aperture. As soon as the shutter is open, light falls on the image sensor, and the image is formed on the array. If the shutter allows light to pass a bit longer, the image would be brighter. Similarly, a darker picture is produced, when a shutter is allowed to move very quickly, and hence, the light that is allowed to pass has very few photons, and the image that is formed on the CCD array sensor is very dark.

2.4.3 ISO

ISO factor is measured in numbers. It denotes the sensitivity of light to the camera. If the ISO number is lowered, it means our camera is less sensitive to light and if the ISO number is high, it means it is more sensitive. The higher is the ISO, the brighter the picture would be. If ISO is set to 1600, the picture would be very brighter and vice versa. If the ISO increases, the noise in the image also increases. Today most of the camera manufacturing companies are working on removing the noise from the image when ISO is set to a higher speed.

2.5 Image Processing

There has been much groundbreaking research conducted in the field of computer vision, primarily in the areas of object detection, recognition, and segmentation, from the 1960s to 2000s. This work was partly divided into two categories: one focused on techniques to perform image recognition, which started in the 1960s, and the other, which started after 2000, focused on collecting images data as a benchmark to evaluate the techniques.

Work in the area of collecting images data for benchmarking began in the year 2006, starting with the PASCAL Visual Object Classes challenge, which had a data set of 20 object categories and was composed of several thousand to about 10,000 labeled objects per category. Many groups started using this data set to test their techniques, and, thus, a new paradigm shift occurred. At about the same time, a group of scholars at Princeton and Stanford Universities began to consider whether we were ready to recognize most objects or all objects, which led to a project called ImageNet. This is a collection of 14M+ images spread over 22K categories. It was the biggest artificial intelligence (AI)-related data set at the time. The ImageNet Large Scale Visual Recognition Challenge resulted in many groundbreaking algorithms, one of which was AlexNet (CNN), which beat all other algorithms, to win the ImageNet challenge in 2012, [8].

2.6 Neural Networks

Neural networks are a type of machine learning algorithm that tries to mimic the human brain. Computers always have been better at performing complex computations, compared to humans. They can do the calculations in no time, whereas for humans, it takes a while to perform even the simplest of operations manually. The human body consists of neurons, which are the basic building blocks of the nervous system. A neuron consists of a cell body, or soma, a single axon, and dendrites (Figure 2.4). Neurons are connected by the dendrites and axon terminals. A signal from one neuron is passed to the axon terminal and dendrites of another connected neuron, which receives it and passes it through the soma, axon, and terminal, and so on, Neurons are interconnected in such a way that they have different functions, such as sensory neurons, which respond to such stimuli as sound, touch, or light; motor neurons, which control the muscle movements in the body; and interneurons, which are connected neurons within the same region of the brain or spinal cord, [3].

The figure 2.4 is showing the structure of a neuron.

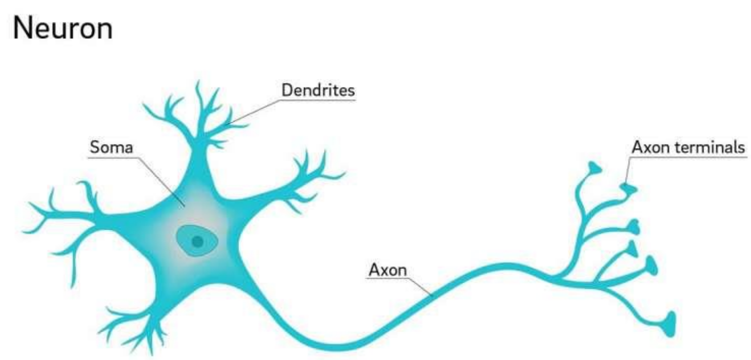

Figure 2.4: Structure of a neuron

Neural Networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on. (Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification, and regression).

The Convolutional Neural Networks (CNNs), an important and powerful kind of learning architecture widely diffused especially for Computer Vision applications. They currently represent the state of the art algorithm for image classification tasks and constitute the main architecture used in Deep Learning, [4].

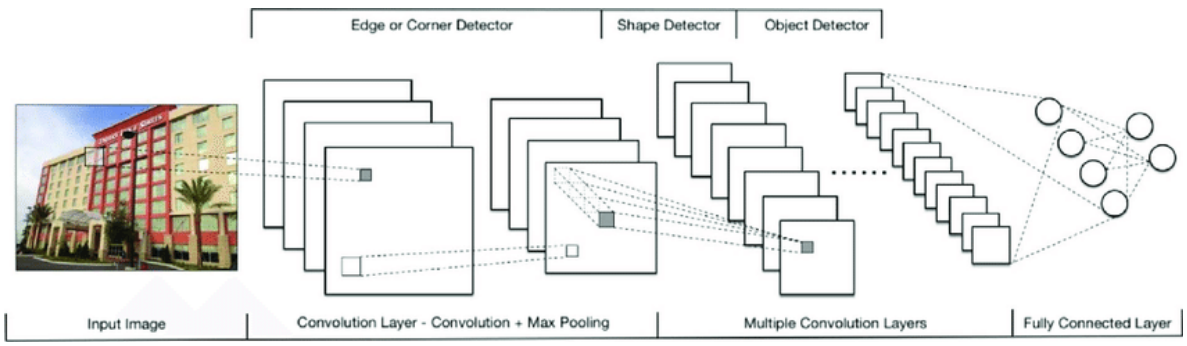

Figure 2.5: Convolutional neural networks

2.7 Python Coding Language

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Two major versions of Python are currently in active use:

- Python 3.x is the current version and is under active development.
- Python 2.x is the legacy version and will receive only security updates until 2020. No new features will be implemented.

2.8 Tensor

A tensor is a mathematical entity with which to represent different properties, similar to a scalar, vector, or matrix. A tensor is indeed a generalization of a scalar or vector. In short, tensors are multidimensional arrays that have some dynamic properties. A vector is a one-dimensional tensor, whereas two-dimensional tensors are matrices, [5].

The figure 2.6 is showing the Tensor flow library.

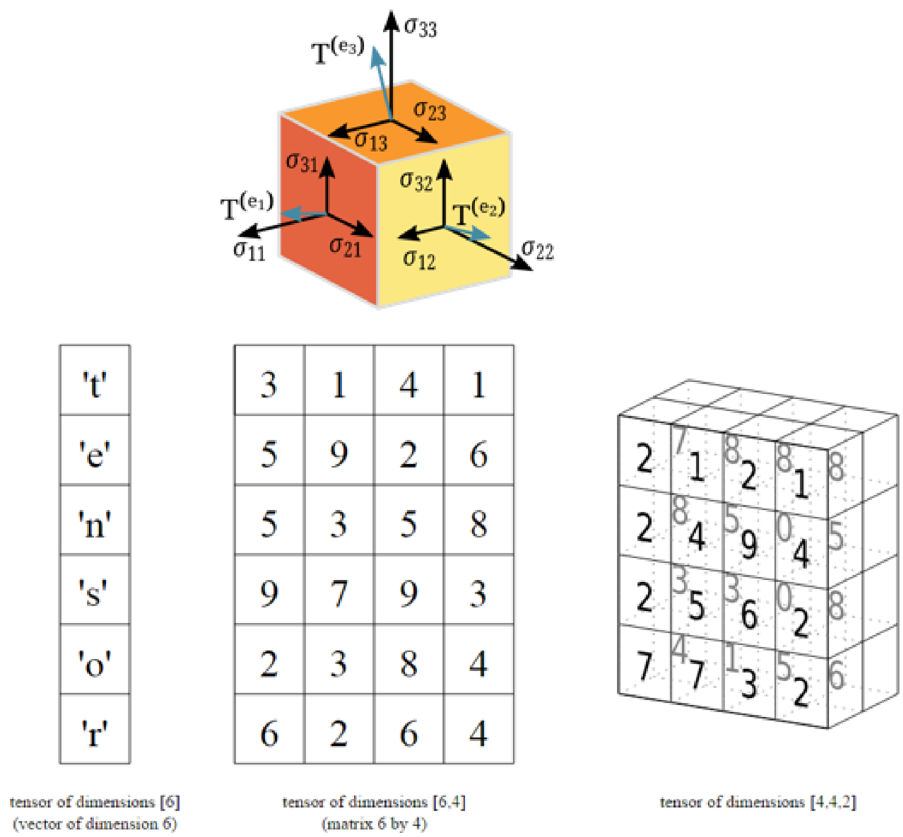

Figure 2.6: Tensor

2.9 Tensor Flow

Tensor flow is Google's open-source machine learning library, with the motto "An Open Source Machine Learning Framework for Everyone." It was developed internally at Google and first released to the public in 2015. Since then a large external community has grown up around the software, with more contributors outside Google than inside. It's aimed at Linux, Windows, and macOS desktop and server platforms, [6].

2.9.1 Tensor flow lite

This library is aimed at running neural network models efficiently and easily on mobile devices and microcontrollers to reduce the size and complexity of the framework, it doesn't support training, just running inference on models that were previously trained. It also doesn't support the full range of data types (such as double).Tensor flow Lite can fit within just a few hundred kilobytes, it also has highly optimized libraries for Arm Cortex-A-series CPUs and GPUs through OpenGL. Another key advantage is that it has good support for the 8-bit quantization of networks, [6].

2.10 Related Works

This subsection is a global review of some of the common smart glasses products available in the market. Increasing reality glasses are a rare sight in daily life, but they hold a large place in the imagination. AR headset has not made the jump to consumers but they are still a full-fledged industry. 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).

2.10.1 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 invariant 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 2.7: 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.

2.10.2 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 1459 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 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 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 2.8: 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.

2.10.3 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 on top of the QR code itself. It can also be useful for other technological activities such as flying a compatible drone in a 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 gaming purposes and browsing the web, the controller will make for an easier controlling experience.

Figure 2.9: 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.

2.10.4 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 feature three beamforming microphones and a multi-touch gesture touchpad, it also includes 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 projects. 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 2.10: 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 it's worth it for business.

2.10.5 Toshiba dynaEdge AR100 viewer smart glasses

Toshiba dynaEdge AR smart glasses are wearable, hands-free augmented reality solutions 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 a 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 2.11: Toshiba dynaedge ar100 viewer

DynaEdge can connect to the corporate network, send and receive data, stream live video, and track assets. It can record and live stream video, take photos, and scan barcodes. Also, it's configurable for use with the 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.

CHAPTER THREE

SYSTEM MODELING

3.1 Introduction

In this chapter, the study was tackled the system components and discus about each component to give the specification of each component. Raspberry pi, camera model, power supply, and also the secondary components.

3.2 The Raspberry Pi 4 Model B

Computers are being developed in a short time with increased speed, hardware, software, lower cost, and the increased availability of access technology. Hence, several works on assistive technologies are created to enable localization, navigation, and object recognition. The best interface can be customized based on a user request whether that is vibrations, sounds, or the spoken word.

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice, and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles, [7].

The figure 3.1 shows the raspberry pi 4 model B.

Figure 3.1: raspberry pi 4 model B 2 GB

Table 3.1: Specification of Raspberry Pi 4 Model B

Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE.
GPU	Dual-Core Video core IV® Multimedia Co-Processor. Provides Open GL ES 3.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s, or 24GFLOPs with texture filtering and DMA infrastructure.
Memory	2GB LPDDR4-3200 SDRAM.
Operating System	Boots from Micro SD card, running a version of the Linux operating system.
Dimensions	85 x 56 x 17mm.
Power	5V DC via USB-C connector. 5V DC via GPIO header.[15]

Table 3.2: Connectors in Raspberry Pi 4 Model B

Ethernet	10/100 Base t Ethernet socket.
Video Output	2 × micro-HDMI ports (up to 4kp60 supported).
Audio Output	Audio Output 3.5mm jack.
GPIO	Connector 40-pin 2.54 mm (100 mils) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V, and GND supply lines.
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2).
Display Connector	Display Serial Interface (DSI) 15-way flat flex cable connector with two data lanes and a clock lane.
Memory Card	Slot Push/pull Micro SDIO.

3.3 Raspberry Pi Camera Module v2

A camera module is an image sensor integrated with a lens, control electronics, and an interface like CSI, Ethernet, or plain raw low-voltage differential signaling. The raspberry pi camera module v2 replaced the original camera module in April 2016. The v2 camera module has a Sony IMX219 8-megapixel sensor and it supports 1080p30, 720p60, and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. In our system, the camera module used to take a live image with Specific dimensions and quality to detect and recognize the object from the image, the figure 3.2 shows the raspberry pi camera module v2.

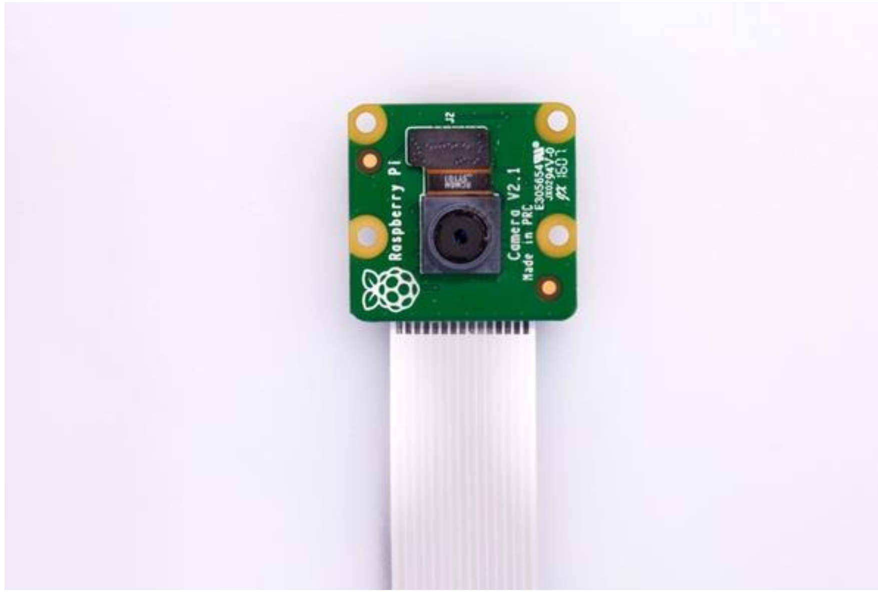

Figure 3-2: Raspberry pi camera v2

3.4 Headphones

The wired headphone will be used in the project since Raspberry pi 4 model B comes with an audio jack, it is better to take advantage of this feature rather than occupying one of the four USB ports that can be useful for other peripherals in the project.

The headphones will be used to help the user listen to the object that is been converted to the audio after it is been captured by the camera. The headphones are going to be small, light, and connected to the glasses, so the user will not worry about losing the headphones or bothered by wearing them, the figure 3.3 shows the headphones.

Figure 3.3: Headphones

3.5 Battery

The Raspberry pi 4 model B uses a normal power supply 5V and 2A. But the user can’t move freely with it. So we decided to use a power bank 5V and 2.5 A, the figure 3.4 shows the power bank charger.

Figure 3.4: Power bank charger

3.6 Light-Emitting Diode Module

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

The 3mm two-color LED module KY-011, emits red and green light. We can adjust the amount of each color using PWM. This module consists of a common cathode 3mm red/green LED and a 0Ω resistor, since the operating voltage, is 2.0v ~2.5v we will need to use limiting resistors to prevent burnout when connecting to the raspberry pi. In our system its use as an indicator light if the headphone gets an input signal from the recognized object, the table 3.3 shows the Specification of led module KY-011,and the figure 3.5 shows the led module.

Table 3.3: Specification of led module KY-011

Operating Voltage	2.0v ~ 2.5v
Working Current	10mA
Diameter 3mm	3mm
Package Type	Diffusion
Color	Red + Green
Beam Angle	150
Wavelength	571nm + 644nm
Luminosity Intensity (MCD)	20-40; 40-80

Figure 3.5: LED

3.7 Secure Digital Cards

Secure Digital cards or SD cards are an ever-important commodity of the tech world. Since the dawn of electronic devices, the need to have a storage medium for saving data became imperative. Several gadgets like audio/video consoles, gaming consoles, cameras, camcorders, handheld computers, mobile phones, and then a few years later, smartphones, were introduced to the world. This, in turn, increased the amount of information being generated by those devices. SD cards have minimal risk of data loss and come in a wide array of storage capacities that you can choose from. There are three main types of SD cards based on their physical size, SD, micro SD, and Compact flash.

In our system micro SD card (size 32GB) contains all software components and operating system files. The figure 3.6 shows the SD card chip.

Figure 3.6: SD cards 32GB

3.8 Push Button Module

The push-button module allows detection in states of high or low from the onboard momentary push button. In our system, the push-button module runs the camera and the main program. The figure 3.7 shows the push button module.

Figure 3.7: Push-button

3.9 System Block Diagram

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams. The main idea of a system block diagram is to work as a similar part eye in the human of the person who is not blind and help blind people for understanding the object around his life by detection and recognition using camera and tell the result on earphone what’s the camera can see. This block can descript it. The figure 3.8 shows the block diagram of the project.

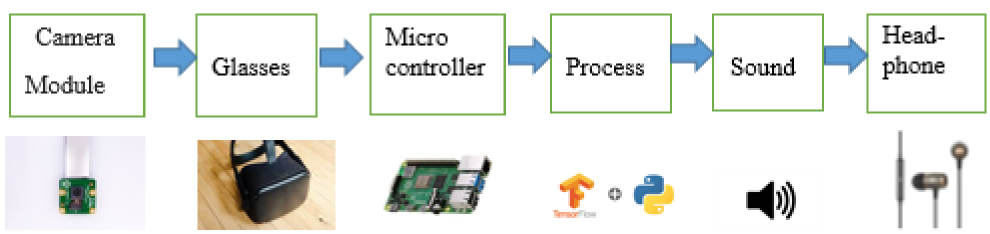

Figure 3.8: Block diagram of the project.

3.10 System Flowchart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.

Image classification approach is presented by applying several steps:

- Import software libraries.
- Load the labels text file into an array.
- Load the TensorFlow model into memory.
- Run the interpreter allocates method.
- Get image height and width from interpreter details.
- Instantiate the camera and start the stream input select.
- Set image format to JPEG.
- Capture image converted to RGB and resize the image.
- Call classification image function and set the image as input into the interpreter.
- Return classification result into an array.
- Get the highest object probability and set it as input text to speech function.
- Return an audio output.

The figure 3.9 shows the system flowchart.

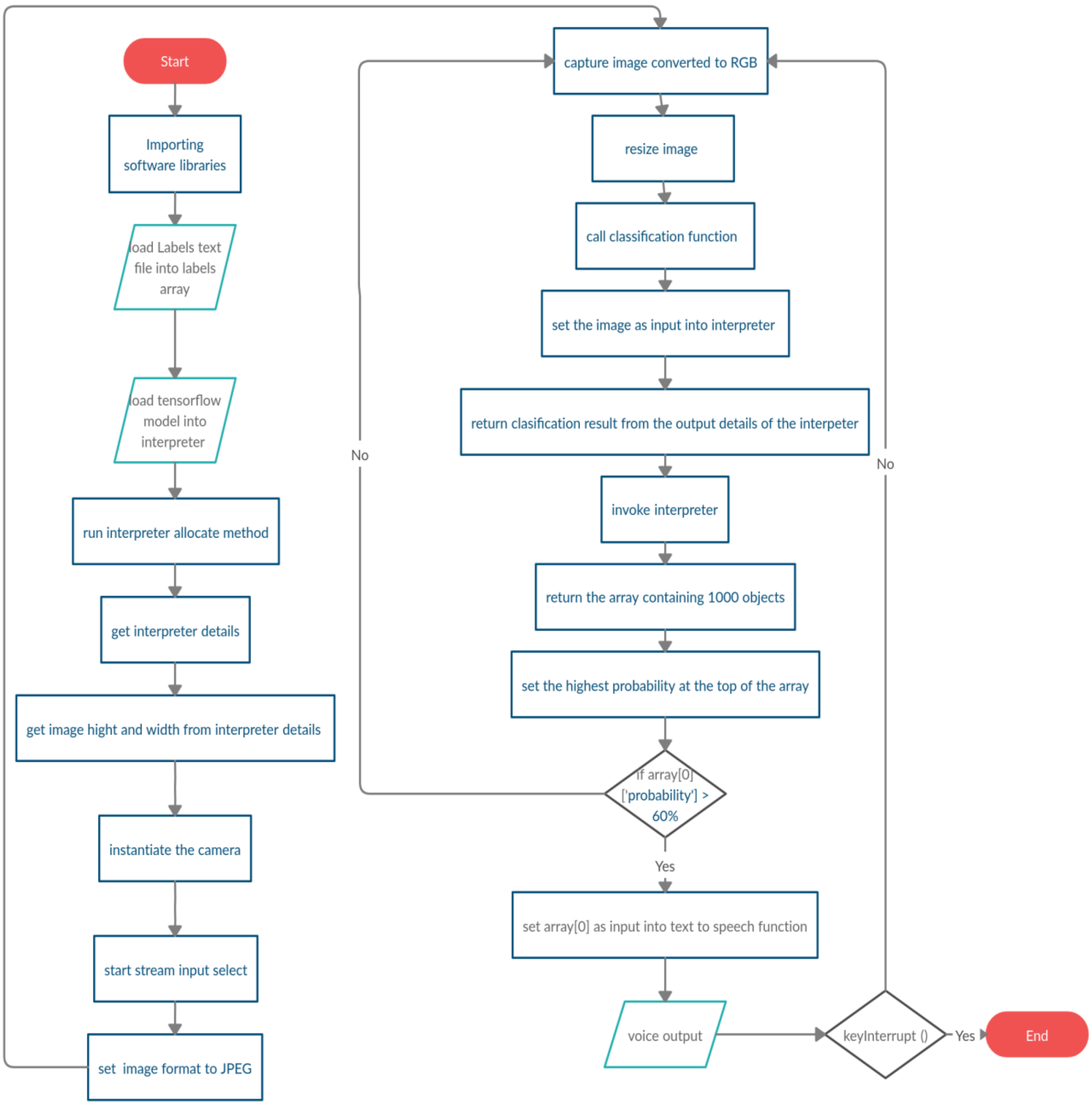

Figure 3.9: System flowchart

CHAPTER FOUR

SIMULATION AND IMPLEMENTATION

4.1 Overview

In this chapter, we covered the simulation part of the project and the program we used in the simulation, and we talk about system software implementation. Finally, we test our project and conclude the result.

4.2 Simulation

A simulation is an approximate imitation of the operation of a process or system; that represents its operation over time. Simulation is used in many contexts, such as simulation of technology for performance tuning or optimizing, safety engineering, testing, training, education, Andrew video games. Often, computer experiments are used to study simulation models. Simulation is also used with scientific modeling of natural systems or human systems to gain insight into their functioning, as in economics. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. Simulation is also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not yet built, or it may simply not exist, [8-11], Key issues in simulation include the acquisition of valid sources of information about the relevant selection of key characteristics and behaviors, the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the simulation outcomes. Procedures and protocols for model verification and validation are an ongoing field of academic study, refinement, research, and development in simulations technology or practice, particularly in the work of computer simulation.

4.2.1 Computer simulation

A computer simulation is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables in the simulation, predictions may be made about the behavior of the system. It is a tool to virtually investigate the behavior of the system under study [8] ,Computer simulation has become a useful part of modeling many natural systems in physics, chemistry, and biology, and human systems in economics and social science (e.g., computational sociology) as well as in engineering to gain insight into the operation of those systems. A good example of the usefulness of using computers to simulate can be found in the field of network traffic simulation. In such simulations, the model behavior will change each simulation according to the set of initial parameters assumed for the environment.

Traditionally, the formal modeling of systems has been via a mathematical model, which attempts to find analytical solutions enabling the prediction of the behavior of the system from a set of parameters and initial conditions. Computer simulation is often used as an adjunct to, or substitution for, modeling systems for which simple closed-form analytic solutions are not possible. There are many different types of computer simulation, the common feature they all share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states would be prohibitive or impossible.

4.3 Proteus Design Suite

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards. It was developed in Yorkshire, England by Labcenter Electronics Ltd and is available in English, French, Spanish and Chinese languages. The figure 4.1 shows the Proteus logo.

Figure 4.1: Proteus

4.3.1 Visual designer for raspberry pi

Visual Designer for Raspberry Pi is a breakthrough product allowing you to design, simulate, and debug complete Raspberry Pi systems. For the first time, users can create a Raspberry PI schematic and a controlling program and then simulate and debug the entire system in software. The creation of the schematic is vastly simplified by a large number of ready-made Raspberry Pi hats and breakout boards. Meanwhile, the design of the firmware program can take place in an easy to use flowchart, complete with high-level methods for controlling the hardware.

4.3.2 Project Workflow

Creating and building new embedded gadgets with Visual Designer for Raspberry Pi is much easier than its might think. Visual Designer is unique in integrating the hardware design and software design into a single software package which vastly simplifies the project design workflow. The first stage is to select your 'virtual hardware' from the Peripheral Gallery. Choose from a library of popular Raspberry Pi hats from Adafruit and Pimoroni as well as dozens of loose sensors, buttons, LEDs, and other widgets. When added these to your project they will automatically be placed on the schematic and connected to the Raspberry Pi. High-level control methods for the peripherals will also be added to Visual Designer. Then design our software as a flowchart so you can easily drag and drop these methods along with decisions, delays, and assignments to drive the connected hardware from the Raspberry Pi. Compile and simulate at the press of the button, making use of our renowned simulation and debugging technology to watch your design come to life on screen. Finally, transfer to the physical Raspberry Pi hardware with a single mouse click and see it working the first time in the real world.

4.4 Circuit of system simulation

Connect the microcontroller (raspberry pi model 4 B) with his camera (raspberry pi camera model v2), the led module in GPIO 20, and the push bottom module in GPIO 21.The figure 4.2 shows the circuit of system simulation.

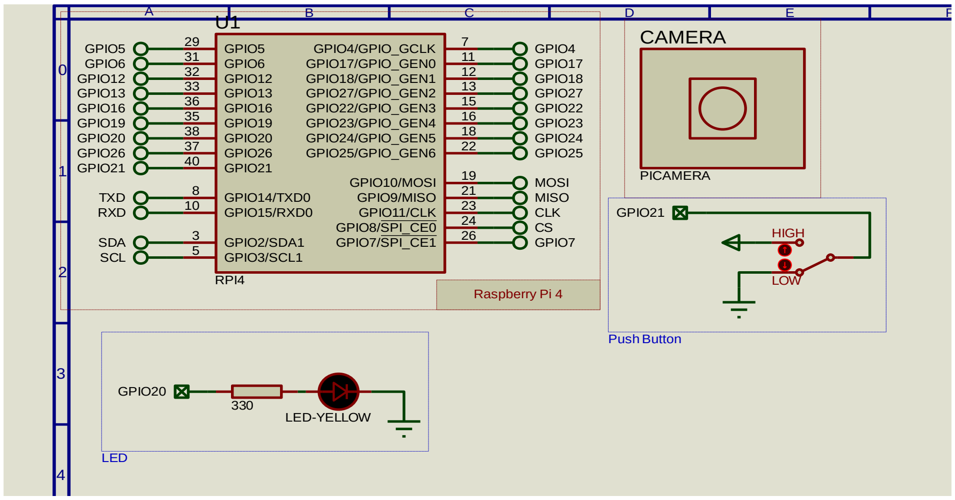

Figure 4.2: Circuit of system simulation

4.5 System Software Implementation

4.5.1 Installing raspbian stretch operating system

Before downloading the operating system on the SD card, the SD card should be formatted, it was used SD Card Formatter for this purpose. Then the Raspbian stretch files were moved to the SD card by using Win32 Disk Imager. After that, the SD card was inserted into the Raspberry Pi. Finally, the keyboard, the mouse, HDMI cable, and lastly the power were inserted into the correct raspberry ports. The figure 4.3 shows SD card formatter, figure 4.4 shows the win disk imager and figure 4.5 shows the installing raspbian stretch operating system.

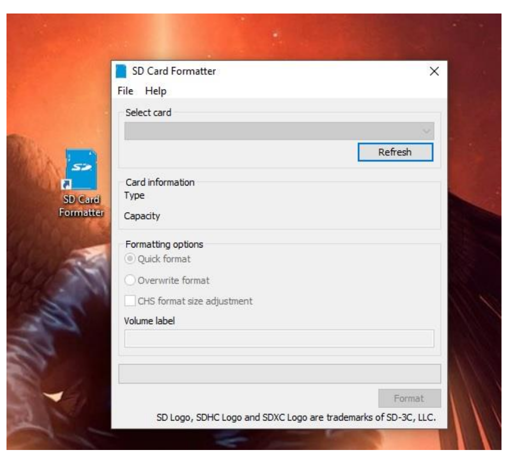

Figure 4.3: SD card formatter

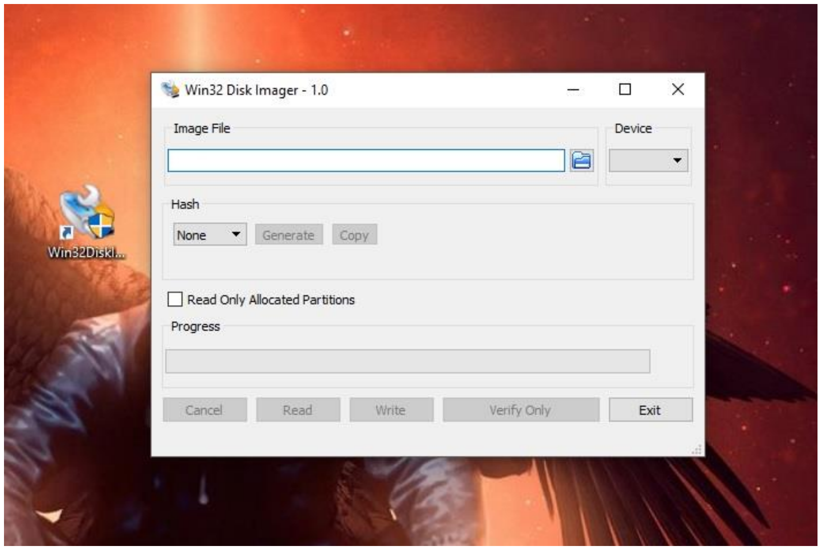

Figure 4.4: Win32 disk imager -1.0

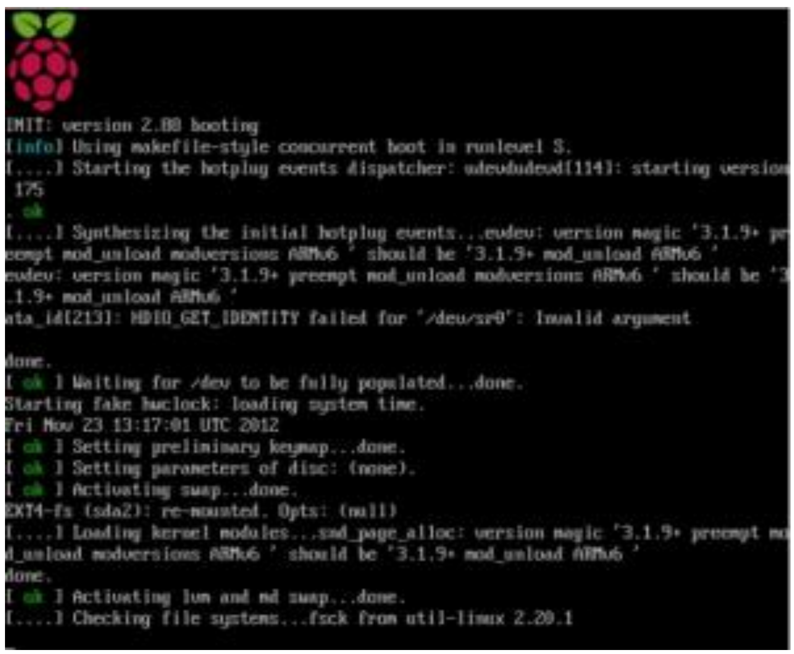

Figure 4.5: Installing raspbian stretch operating system

After raspberry pi boot up connect the raspberry pi to the internet and opened the terminal and update the Raspbian operating system and upgrade it, by the following commands:

$ `sudo apt-get update`

$ `sudo apt-get upgrade`

The figure 4.6 shows updating and upgrading rasbian system.

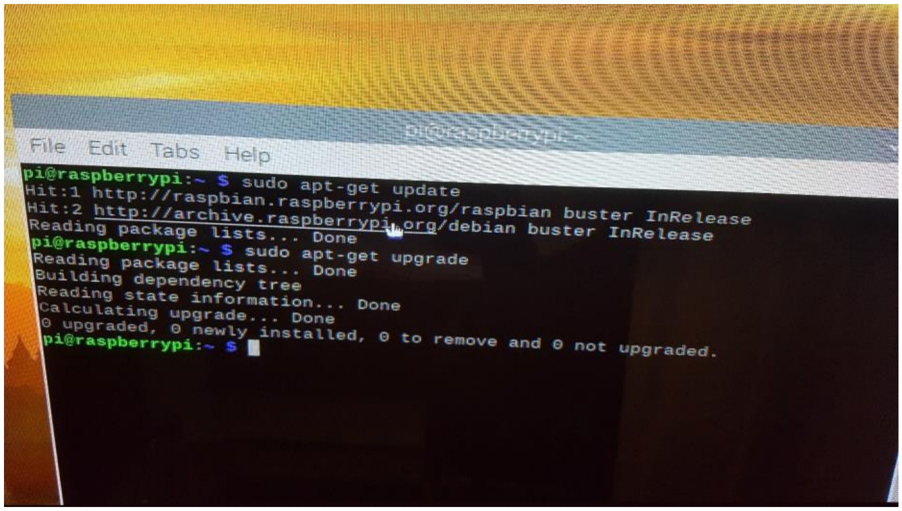

Figure 4.6: Update and upgrade rasbian

4.5.2 Enabling Raspberry Pi Camera

To enable the Raspberry Pi Camera opened the terminal and opening the Raspberry Pi configuration by the following commands.

$ `sudo raspi-config`

From the configuration list, chose interfacing options and enable the camera. The figure 4.7 shows the configuration list.

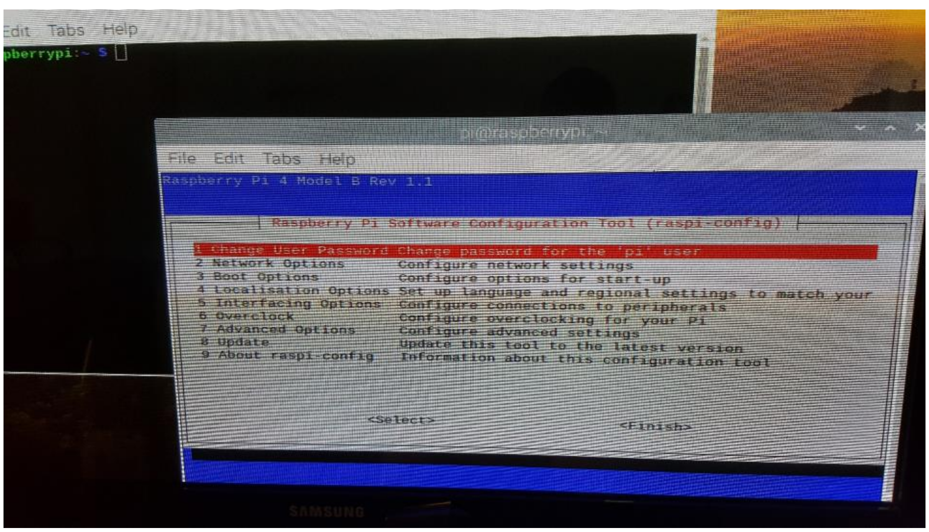

Figure 4.7: Configuration list

4.5.3 Installing tensor flow lite interpreter

To installing the tensor flow lite interpreter typed the following command in the terminal.

```
$ pip3 install
https://dl.google.com/coral/python/tflite_runtime-
2.1.0.post1-cp38-cp38-linux_armv7l.whl
```

4.5.4 Tensor flow lite image processing steps

Tensor flow lite interface typically follows the following steps:

- ➢ Loading the model into memory, which contains the model's execution graph.
- ➢ Raw input data for the model generally does not match the input data format expected by the model. We need to resize an image or change the image format to be compatible with the model.
- ➢ Using the tensor flow lite API to execute the model. It involves a few steps such as building the interpreter, and allocating tensors, as described in the following steps:

- ❖ Load the model into memory.
- ❖ Build an interpreter based on an existing model.
- ❖ Set input tensor values. (Optionally resize input tensors if the predefined sizes are not desired.)
- ❖ Invoke inference.
- ❖ Read output tensor values.

- Receive results from the model inference, we must interpret the tensors in a meaningful way that's useful in our application.

4.5.5 Image classification by tensor flow

The task of identifying what an image represents is called image classification. An image classification model is trained to recognize various classes of images. During training, an image classification model is fed images and their associated labels. Each label is the name of a distinct concept, or class, that the model will learn to recognize.

4.5.5.1 Mobile net quantization and labels model

It is sufficient training data (often hundreds or thousands of images per label), we download it from:

https://www.tensorflow.org/lite/models/image_classification

- The TensorFlow Lite quantized MobileNet model's accuracy range from 64.4 to 89.9%.
- The TensorFlow Lite quantized MobileNet model's sizes range from 0.5 to 3.4 MB.

4.5.6 The software libraries

4.5.6.1 Core tools for working with streams library

The io module provides Python's main facilities for dealing with various types of I/O. There are three main types of I/O: text I/O, binary I/O, and raw I/O. These are generic categories, and various backing stores can be used for each of them.

```
$ import io
```

4.5.6.2 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays, [12].

$ `import numpy as np`

4.5.6.3 Picamra library

This package provides a pure Python interface to the Raspberry Pi camera module for Python 2.7 (or above) or Python 3.2 (or above).

$ `import picamera`

4.5.6.4 Rpi.GPIO library

Raspberry-gpio-python or RPi.GPIO is a Python module to control the GPIO interface on the raspberry pi.

$ `import RPi.GPIO as GPIO`

4.5.6.5 Time access and conversions library

This library provides various time-related functions.

$ `import time`

4.5.6.6 Miscellaneous operating system interfaces library

This library provides a portable way of using operating system dependent functionality.

```
$ import os
```

4.5.6.7 Python imaging library

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

```
$ from PIL import Image
```

4.5.6.8 Google text-to-speech library

Google Text-to-Speech a Python library and CLI tool to interface with Google Translator's text-to-speech API. Writes spoken mp3 data to a file. We create audio files for all labels that our model can be recognized.

```
$ from gtts import gTTS
```

The figure 4.8 shows the hardware design of the project.

Figure 4.8: Hardware design

4.6 Testing and results

This paragraph talks about tested the performance of our project by putting some object in front of the camera and conclude the result, some of the items have high probability and easy to recognize it, and some of them were hard to recognize.

At first, a telephone placed in front of the camera at a suitable distance for the camera to recognize it and pronounce it through the handset. And found that the probability of detection on the phone is excellent and fast in recognition. The process was repeated 40 times and recorded in a table. The figure 4.9 and figure 4.10 shows the telephone testing and the result.

Figure 4.9: Telephone test

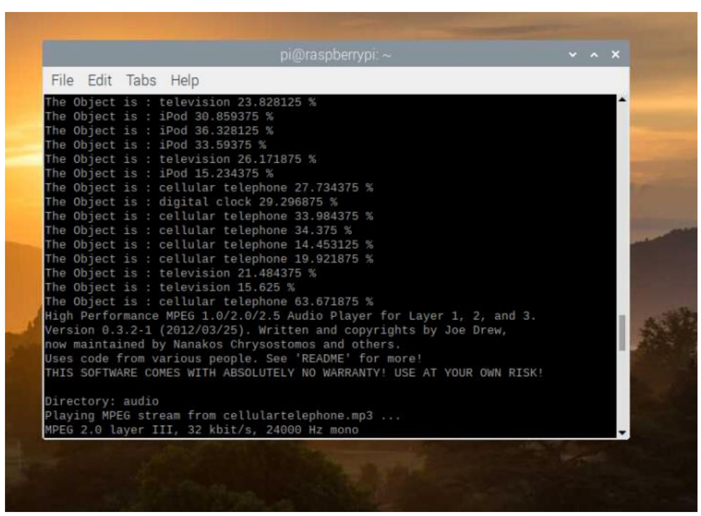

Figure 4.10: Telephone result

And then a notebook placed in front of the camera at a suitable distance for the camera to recognize it and pronounce it through the headset. And found that the probability of detection of the notebook is excellent and fast in recognition. The process was repeated 40 times and recorded in a table. The figure 4.11 and figure 4.12 shows the notebook testing and the result.

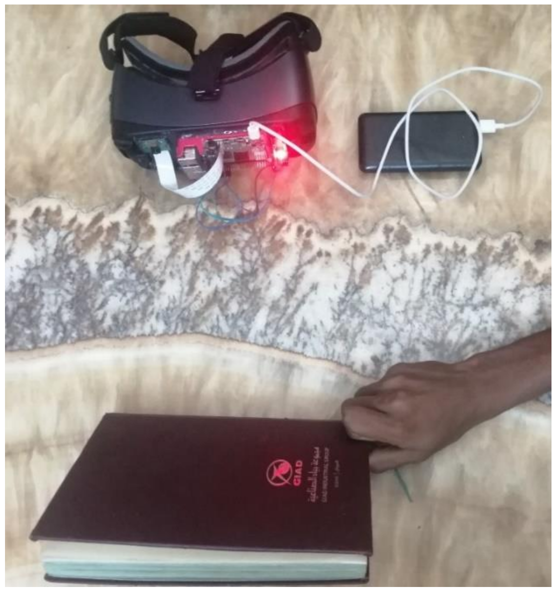

Figure 4.11: Notebook test

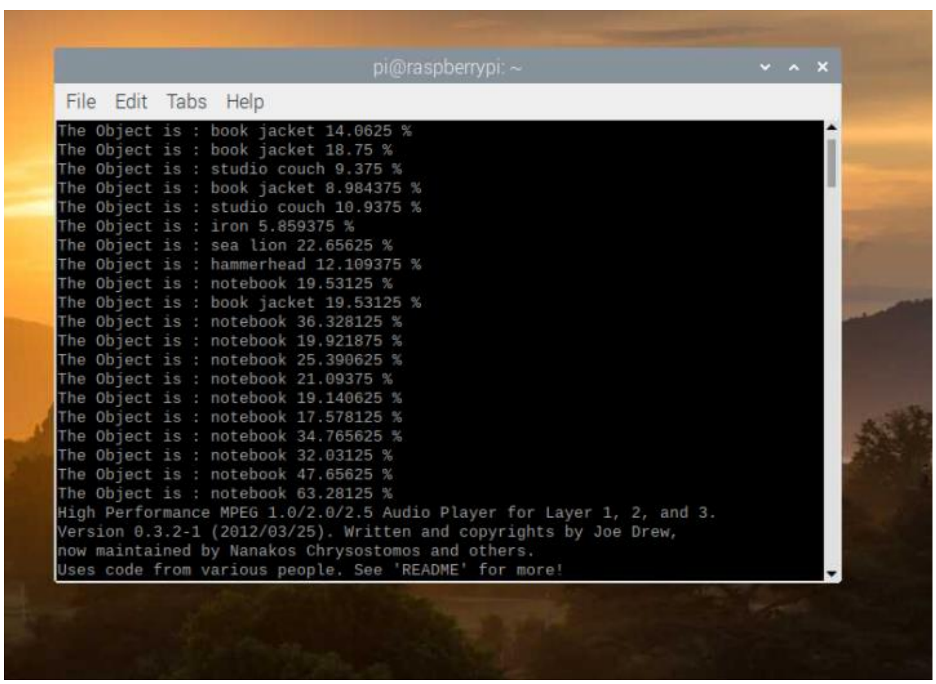

Figure 4.12: Notebook result

And also placed a laptop in front of the camera at a suitable distance for the camera to recognize it and pronounce it through the headset. And found that the probability of detection on the laptop was somewhat good and easy to recognize. The process was repeated 40 times and recorded in a table. The figure 4.13 and figure 4.14 shows the laptop testing and the result.

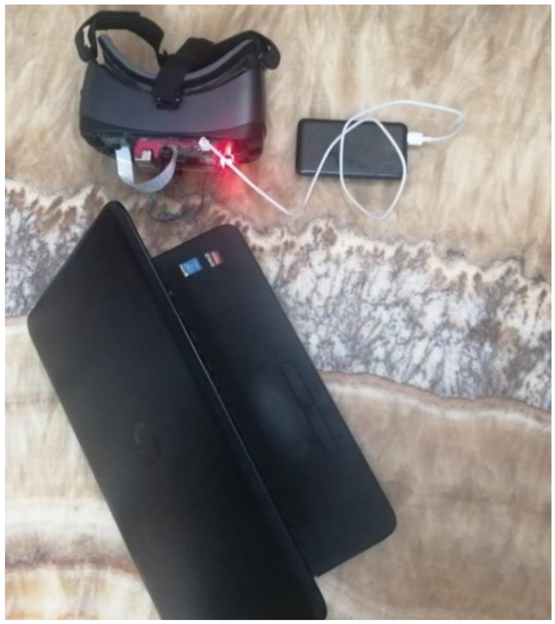

Figure 4.13: Laptop test

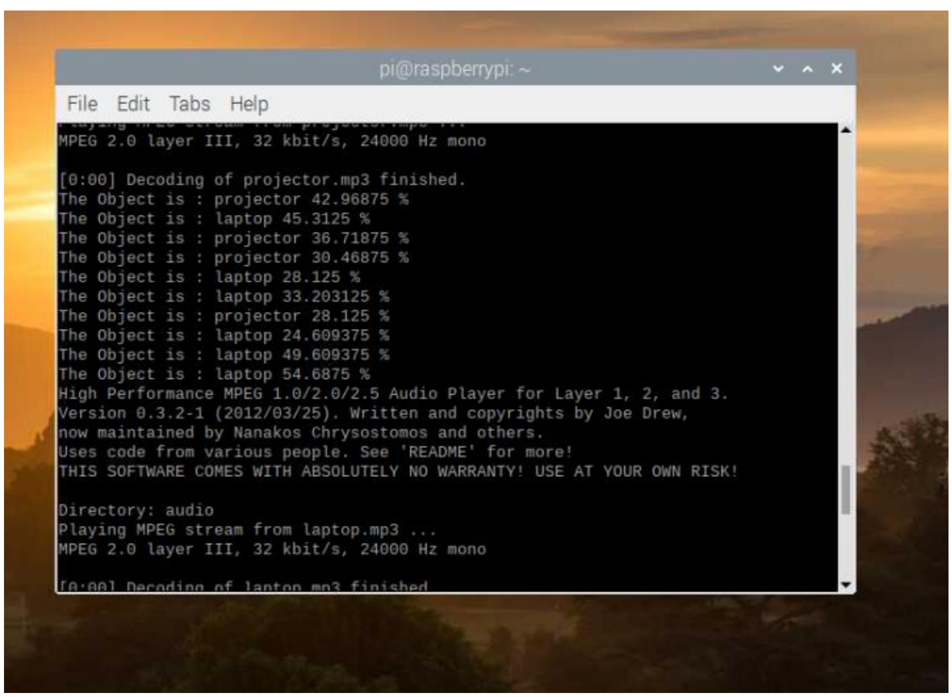

Figure 4.14: Laptop result

Then sunglasses placed in front of the camera at a suitable distance so that the camera could recognize it and speak it with the stethoscope. And found that the probability of detecting the glasses was somewhat weak and slow to recognize and the process was repeated 40 times and recorded in a table. The figure 4.15 and figure 4.16 shows the sunglasses testing and the result.

Figure 4.15: Sunglasses test

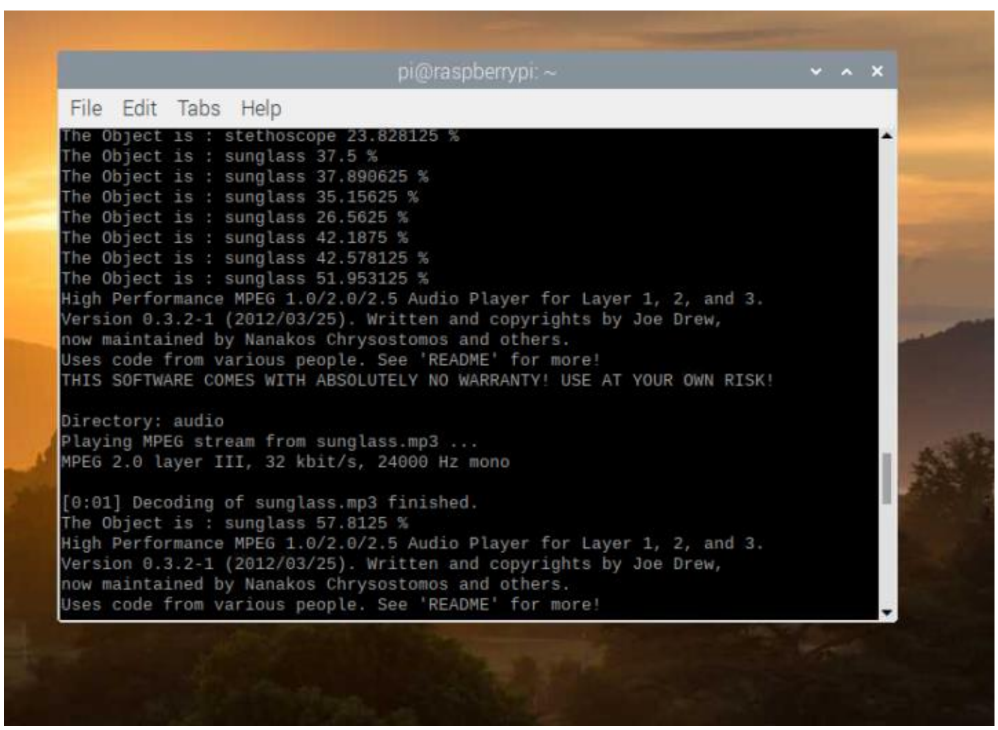

Figure 4.16: Sunglasses result

The table 4.1 shows the result of testing the objects.

Table 4.1: Test of objects

Object Name	**Number of Tries**	**Detection Ratio**	**Pass**	**Failure**	**Percentage Error**
Telephone	40	63.67	37	3	7.5%
Notebook	40	63.28	38	2	5%
Laptop	40	54.68	37	3	7.5%
Sunglasses	40	51.95	39	1	2.5%
Bottle	40	60.3	36	1	2.5%
Knife	40	54.1	39	4	10%

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The original problem in this project was to help the blind people 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 main features were provided in this project such as image detection which can help the visually impaired to recognize the objects in front based on a voice assistant. But the problem was with the dataset that was not enough and required to study image processing methods and programming language. During the development process, certain skills were developing and understanding the hardware and software design.

The project is implemented with emphasis on cost-effectiveness, which is kept as low as possible, this appears in the following cost analysis table, and the final prototype cost only $117.

Table 5-1: Total cost of the project

Item	**Raspberry pi 4 model B**	**Camera**	**Earpiece**	**Accessories**	**Total**
Cost	$55	$27	$10	$25	$117

5.2 Recommendations

We recommended:

- Improve the quality of the camera to get better performance.
- Add flash to improve image quality in dark places.

REFERENCES

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[2] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010.

[3] Pramod Singh, Avinash Manure - Learn TensorFlow 2.0_ Implement Machine Learning And Deep Learning Models With Python-Apress (2020)- Chapter 4 page 76

[4] Pramod Singh, Avinash Manure - Learn TensorFlow 2.0_ Implement Machine Learning And Deep Learning Models With Python-Apress (2020)- Chapter 3 page 54

[5] Pramod Singh, Avinash Manure - Learn TensorFlow 2.0_ Implement Machine Learning And Deep Learning Models With Python-Apress (2020)- Chapter 1 page 6

[6] Pete Warden, Daniel Situnayake - TinyML_ Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers-O'Reilly Media (2019) - CHAPTER 13 page 355

[7] **Jamal S. Zraqou, Wissam M. Alkhadour, and Mohammad Z. Siam, Isra University, Amman Jordan, Accepted 30 Jan 2017, Available online 31 Jan 2017, Vol.7, No.1 (Feb 2017).**

[8] J. Banks; J. Carson; B. Nelson; D. Nicol (2001). Discrete-Event System Simulation. Prentice-Hall. p. 3. ISBN 978-0-13-088702-3.

[9] Srinivasan, Bharath (2020-09-27). "Words of advice: teaching enzyme kinetics". The FEBS Journal. DOI:10.1111/febs.15537. ISSN 1742-464X

[10] In the words of the Simulation article in Encyclopedia of Computer Science, "designing a model of a real or imagined system and conducting experiments with that model".

[11] Sokolowski, J.A.; Banks, C.M. (2009). Principles of Modeling and Simulation. John Wiley & Son. p. 6. ISBN 978-0-470-28943-3.

[12] Charles R Harris; K. Jarrod Millman; Stéfan J. van der Walt; et al. (16 September 2020), "Array programming with NumPy" (PDF), Nature, 585

APPENDIX A

CREATE TEXT TO SPEECH FILES CODE

```
import os

from gtts import gTTS
file = open("labels_mobilenet_quant_v1_224.txt", 'r')
my_list = file.readlines()
def remove(string):
    return string.replace(" ", "")
def find(name, path):
    for root, dirs, files in os.walk(path):
        if name in files:
            return True
        else:
            return False
for i in range(len(my_list)):
    string = my_list[i][:-1]
    save_name = "{}.mp3".format(remove(string))
    find_ex = find(save_name,
'/home/achmed/Documents/glass/v0.2/label map for
TFlite')
    if find_ex != True:
        tts = gTTS(text=my_list[i], lang='en')
        tts.save(save_name)
           os.system("mpg321 {}".format(save_name))
```

APPENDIX B

RECOGNIZING AND PRONOUNCING THE OBJECTS CODE

```
import io
import numpy as np
import picamera
import RPi.GPIO as GPIO
import time
import os
from PIL import Image
from tflite_runtime.interpreter import Interpreter
GPIO.setmode(GPIO.BCM)
GPIO.setup(20, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(21, GPIO.OUT)
GPIO.setup(12, GPIO.OUT)
def load_labels(path):
    with open(path, 'r') as f:
        return {i: line.strip() for i, line in enumerate(f.readlines())}
def set_input_tensor(interpreter, image):
    tensor_index = interpreter.get_input_details()[0]['index']
    input_tensor = interpreter.tensor(tensor_index)()[0]
    input_tensor[:, :] = image
def classify_image(interpreter, image, top_k=1):
    """Returns a sorted array of classification results."""
    set_input_tensor(interpreter, image)
```

```python
        interpreter.invoke()
        output_details = interpreter.get_output_details()[0]
        output = np.squeeze(interpreter.get_tensor(output_details['index']))
        if output_details['dtype'] == np.uint8:
                scale, zero_point = output_details['quantization']
                output = scale * (output - zero_point)
        ordered = np.argpartition(-output, top_k)
        return [(i, output[i]) for i in ordered[:top_k]]
def add_temp(prev_label):
        add = open("temp.txt", "w+")
        added_lines = add.readlines()
        add.write(prev_label)
        add.close()
def check_temp(string_to_search):
        with open('temp.txt', 'r') as read_obj:
                for line in read_obj:
                        if string_to_search in line:
                                return True
        return False
def delete_temp():
        a_file = open("temp.txt", "r")
        lines = a_file.readlines()
        a_file.close()
        del lines[0:]
        new_file = open("temp.txt", "w+")
        for line in lines:
                new_file.write(line)
        new_file.close()
```

```
def remove(string):
        return string.replace(" ", "")
def main():
        labels = load_labels('labels_mobilenet_quant_v1_224.txt')
        interpreter = Interpreter('mobilenet_v1_1.0_224_quant.tflite')
        interpreter.allocate_tensors()
        _, height, width, _ = interpreter.get_input_details()[0]['shape']
        with picamera.PiCamera(resolution=(640, 480), framerate=30) as camera:
                camera.start_preview()
                try:
                        stream = io.BytesIO()
                        for _ in camera.capture_continuous(
                                        stream, format='jpeg', use_video_port=True):
                                stream.seek(0)
                                image                                                    =
Image.open(stream).convert('RGB').resize((width, height),

Image.ANTIALIAS)
                                start_time = time.time()
                                results = classify_image(interpreter, image)
                                elapsed_ms = (time.time() - start_time) * 1000
                                label_id, prob = results[0]
                                stream.seek(0)
                                stream.truncate()
                                if prob > 0.60:
                                        if check_temp(labels[label_id]) is False:
                                                camera.annotate_text = '%s %.2f\n%.1fms'
% (labels[label_id], prob, elapsed_ms)
```

```
                    GPIO.output(12, GPIO.LOW)
                    GPIO.output(21, GPIO.HIGH)
                    print("The Object is :", labels[label_id],
prob * 100, "%")
                    string = labels[label_id]
                    voice_output =
"audio/{}.mp3".format(remove(string))
                    os.system("mpg321
{}".format(voice_output))
                    add_temp(string)
            else:
                print("The Object is :", labels[label_id], prob *
100, "%")
                GPIO.output(21, GPIO.LOW)
                GPIO.output(12, GPIO.HIGH)
                if GPIO.input(20) == GPIO.LOW:
                    break
    finally:
        camera.stop_preview()
        delete_temp()
if __name__ == '__main__':
    os.system("mpg321 audio/pressthebutton.mp3")
    try:
        while True:
            input_state = GPIO.input(20)
            if GPIO.input(20) == GPIO.LOW:
                if True:
                    print("The Camera is Opened!")
```

```
                    os.system("mpg321
audio/theCameraisready.mp3")
                    main()
            else:
                GPIO.output(21, GPIO.LOW)
    except KeyboardInterrupt:
        GPIO.cleanup()
```