Modern University for Technology and Information Faculty of Computers & Artificial intelligence Department of Computer science

Design and Implementation of Smart Glasses for Blind People

Computer Science Graduation Project

Submitted By

Dina Ahmed El Sayed Fathy

Mohamed Gamal Abdallah

Ismail Hany Ismail

Ahmed Sayed Mohamed

Mohamed Omar Abdallah

Ziad Waleed Sayed

Supervised By

Prof. Dr. Gouda Ismail

T.A. Riham Faris

2023

Acknowledgment

All praise and thanks to God, who provided us the ability to complete this work despite the circumstances of this pandemic. We hope that you accept this work from us. Our greatest appreciation to Modern University for Technology and Information (M.T.I) for the huge role it played in forming our character and personalities, providing us with the necessities for our educational development.

Most of all, we would love to express our deep and sincere gratitude to our supervisor Prof. Dr. Gouda Ismail for his inspiration, patience, and encouragement. His guidance helped us throughout this project. We would also like to acknowledge the contributions of Prof. Dr. Mohamed El Mayah, Prof. Dr. Mohamed El Gazzar, Prof. Dr. Alaa Abd El Rahim, and Prof. Dr. Hanafy Ismail for their insightful comments and support. Special thanks to T.A. Riham Faris and T.A. Omnia Ahmed for their valuable assistance throughout the project.

Additionally, we extend our thanks to our families for their unwavering support during our academic journey. We are grateful for their constant encouragement and belief in our abilities.

Abstract

These smart glasses are designed to assist visually impaired individuals in independently performing their daily activities without relying on the assistance of others. To achieve this objective the smart glasses were designed to carry eleven tasks which are reading text, translating text, scene description, determining prayer times and direction for praying (qibla), age gender reveal, color identification, Q R Code scanner, country/city detection, recording notes and obstacle detection using ultrasonic sensor.

The glasses used many technologies to perform its tasks which are OCR, Google translate API. OpenCV, Optical Character Recognition technology (OCR), MobileNet SSD model, CNN deployed by Caffe Framework, Hue, Saturation and value model, ALAdhan API, ZBar Library, IP Geolocation API.

The smart glasses device incorporates a camera and ultrasonic sensors to capture and analyze the user's surroundings. The camera captures images, which are then processed using computer vision algorithms to detect and identify objects, obstacles, and landmarks in real-time. The ultrasonic sensors emit sound waves and measure their reflection to detect the presence and distance of obstacles in the environment.

By analyzing the data from the camera and ultrasonic sensors, the smart glasses provide real-time audio feedback to the user, describing the objects and their spatial location. This enables blind and visually impaired individuals to navigate their surroundings more effectively, avoid obstacles, and identify important landmarks or points of interest.

The integration of these features in the smart glasses’ device offers a transformative solution for blind and visually impaired individuals, enhancing their independence, safety, and overall quality of life. By leveraging advanced sensor technology and computer vision algorithms, smart glasses empower users

to interact with their environment and navigate with greater confidence and freedom.

This report is structured into several chapters, starting with the introduction that provides an overview of the project's motivation, related work objectives and challenges. The system planning chapter covers the project team tasks and distribution, along with Gantt and network diagrams. The system analysis and design chapter focus on hardware design and module analysis. The System Implementation & Experimental Results outlines the software's main functions, algorithms, flowcharts, and provides snapshots of the output.

The conclusion based on the experimental results and future work includes a summary of the implemented application and possible future improvements. Finally, the references are included.

Table of Contents

Table of Figures

Table of Tables

Chapter One

Introduction

Chapter One

Introduction

1.1 Overview

Imagine living in a world where simple tasks we take for granted, like walking independently or recognizing objects, become daunting challenges for individuals who are blind or visually impaired. These individuals rely heavily on others for assistance, limiting their independence and access to education. Recognizing the pressing need to empower and enhance the lives of the visually impaired, we embarked on a mission to develop a groundbreaking solution – a smart glasses project that would revolutionize their independence, interaction with the world, and educational opportunities.

Vision for this project was to create smart glasses equipped with advanced technology that could assist blind individuals in navigating unfamiliar environments and perceiving their surroundings with newfound clarity. Leveraging the power of artificial intelligence, we integrated Optical Character Recognition (OCR) technology to enable text recognition and utilized Google Translate for real-time translation. This transformative feature empowers blind individuals to access and understand written information, opening a world of educational resources previously inaccessible to them.

Furthermore, we incorporated ultrasonic sensors into the smart glasses, enabling obstacle detection and avoidance. By detecting objects and obstacles in the user's path, the smart glasses provide real-time audio feedback, alerting the wearer and ensuring safe navigation through their environment. This obstacle avoidance capability further enhances their independence, allowing them to move freely without constant reliance on others.

The aim of the project is to address the limitations of existing solutions by providing a comprehensive tool that combines navigation assistance, OCR technology, and obstacle avoidance in a single, user-friendly package. We recognize the complexity of developing a computer-aided tool for the visually impaired, but our unwavering commitment to making a positive impact drives us forward.

Through this innovative endeavor, we aspire to empower blind individuals in their pursuit of education by enabling access to printed materials through OCR and translation features. Additionally, we aim to enhance their mobility and independence by offering seamless obstacle detection and avoidance. By bridging the gaps in existing technologies and offering a holistic solution, we strive to transform the lives of visually impaired individuals, granting them newfound freedom, educational opportunities, and the ability to navigate the world with confidence and independence.

1.2 Project Motivation

Our project is driven by strong motivations to make a meaningful impact and deliver the best possible solution to address the challenges faced by visually impaired individuals. However, like any ambitious endeavor, we encounter various obstacles and challenges along the way that test our dedication and perseverance. Despite these hurdles, we remain committed to overcoming them and completing the project with excellence.

The population of visually impaired individuals is steadily increasing, yet the existing solutions available for them fail to meet their comprehensive needs and lack significant advancements.

Considering this, we have embarked on the development of smart glasses that aim to fulfill all their requirements, enabling them to gain independence and improve their overall quality of life.

By addressing the challenges faced by blind individuals, we aim to provide an innovative alternative to conventional aids such as white canes or guide dogs, empowering them to engage in daily activities with greater ease and confidence.

Statistics of Blind People in North Africa and Middle

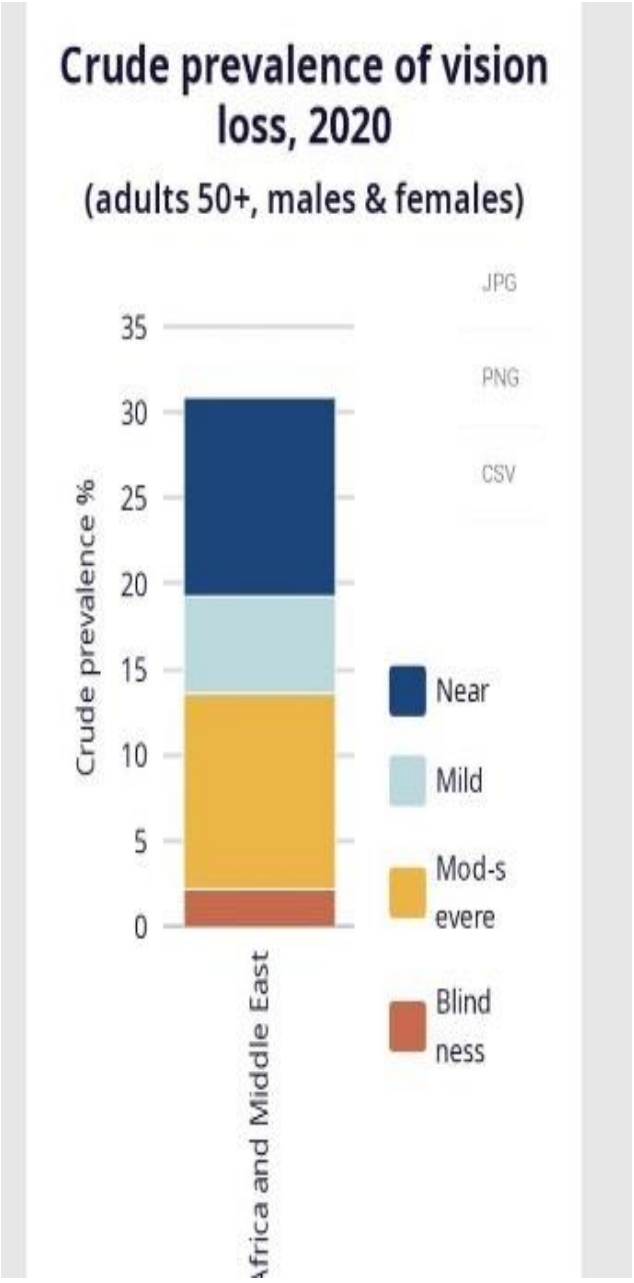


Figure 1-Statistics of Blind People in North Africa and Middle [1]

The number of visually impaired individuals is steadily increasing each year, particularly in North Africa and the Middle East. These statistics emphasize the pressing need to develop effective solutions that cater to the needs of this growing population.

Statistics of Blind People in Egypt

The year 2020 witnessed an estimated 8.3 million individuals in Egypt experiencing vision loss, with 370,000 among them being classified as blind. Unfortunately, individuals with physical challenges in Egypt have long faced discrimination and limited accessibility, and those with partial or complete loss of sight are no exception. There is a common misconception that visual disability solely refers to total blindness, which is untrue. Visual impairment encompasses any visual condition that hinders an individual's ability to carry out daily activities effectively.

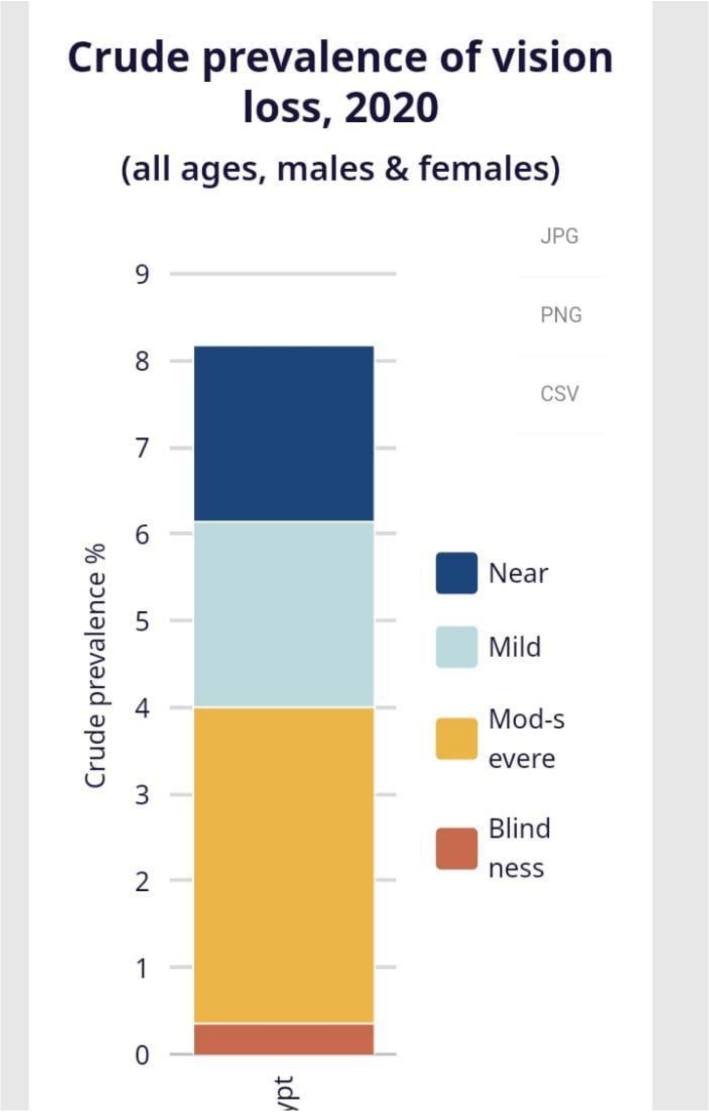


Figure 2-Statistics of Blind People in Egypt [1]

It is possible to say that 60% of the blindness in the world can be cured, and 20% can be prevented.

Statistics of Blind People All Over the World

The year 2020 witnessed an estimated 54 million individuals in North Africa and the Middle East experiencing vision loss, with 3.1 million among them classified as blind. Blinding eye diseases continue to pose a significant health issue in many developing countries, with an estimated 38 million people at risk of becoming blind. The prevalence of blindness in developing countries is substantially higher compared to developed nations, and the majority of blind individuals reside in Africa, Asia, and Latin America. Various studies conducted on blindness prevalence indicate rates ranging from 0.6% in Lebanon to 1.5% in Saudi Arabia. Cataracts, trachoma, and glaucoma are identified as leading causes of blindness, with cataract being the major contributor. However, it is important to note that a significant portion of blindness in the region is preventable or treatable. Efforts should be directed towards addressing universally prevalent eye problems through cost-effective remedies such as cataract and refractive issues, as well as focusing on focal problems that can be prevented through primary healthcare measures, such as trachoma, onchocerciasis, and vitamin A deficiency. Achieving the goals of Vision 2020 will require substantial development in staffing levels, infrastructure, and community programs.

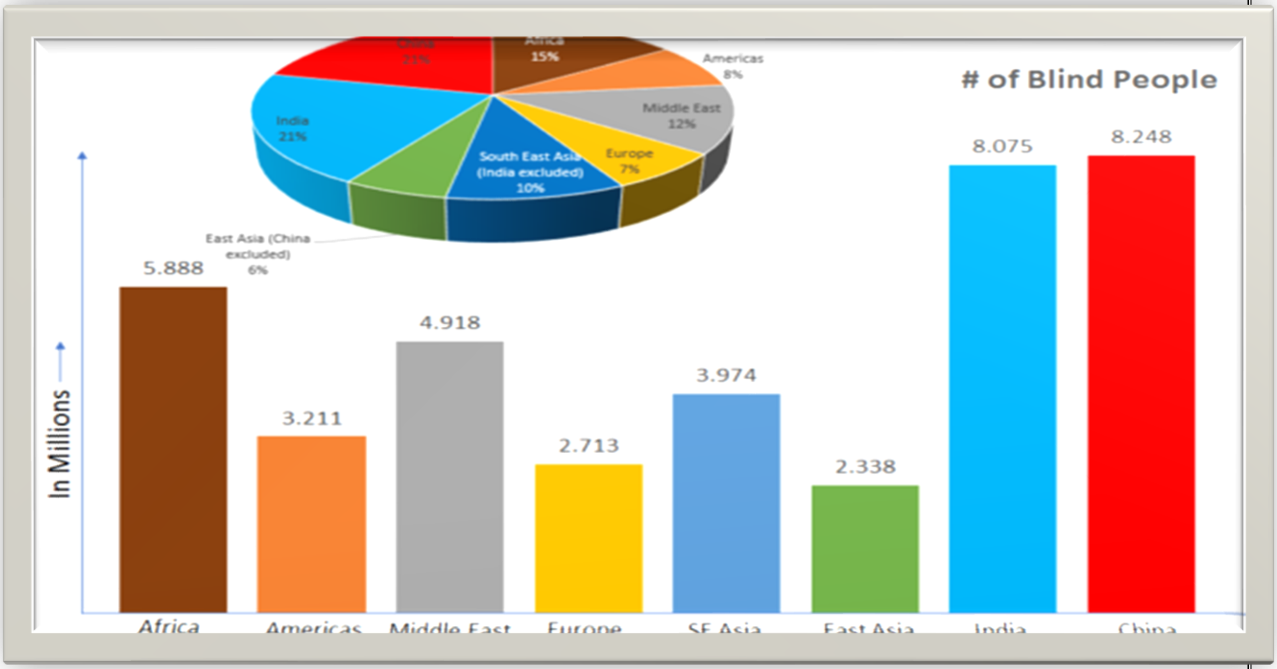


Figure 3-Statistics of Blind People all Over the world [1]

1.3 Related Work

We are not the first to invent these glasses, as there have been numerous commercials and researchers who have developed them.

1.3.1. Commercial Product

<u>Al-Amal (2010-2021)</u>

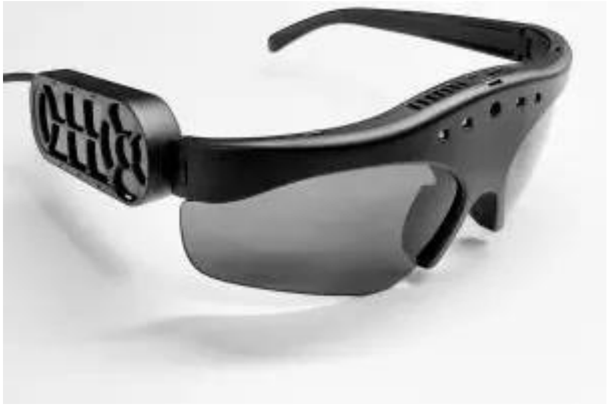


*Figure 4- **Amal Glass***

Amal Glass launches Version 2 of Smart glasses for blind and visually impaired people [2]

Dubai, UAE: Amal Glass, a Dubai-based company has launched Version 2 of the unique features Smart glasses for the blind and visually impaired people. It utilizes Artificial Intelligence (AI) to enable the users to become self-dependent and eliminates the need for multiple devices or guide dogs and human assistance. Amal Glass Version 2 is completely new and has unique features. It supports Arabic and English languages with plans for adding Spanish, French and Hindi. The Mobile Application running on the Android system makes it affordable and updatable.

It reads the texts of books, magazines, computer screens, cell phone screens, signboards and ATMs. Visually impaired people can go shopping alone at any time and confidently pay the cash and get back the correct amount, it can recognize over 100 banknotes from across the world.

The company has started Amal Glass V2 production on a pilot basis. Mohammed Islam, the company's Head of Development said: "The response it has generated has been tremendous in the UAE, Saudi Arabia, Egypt, Sultanate of Oman, Kuwait, Turkey, India and the UK. It got amazing customer feedback and received inquiries from the US, Korea, Argentina and Brazil."

It also allows the users to identify the surrounding objects like cars, street signs and furniture. The users can seek help from family members or friends by sending SMS with their current location to pre-registered three contacts. It offers to download over 15 additional applications from Amal Glass Application Store. The first Specified Application store for blind and visually impaired people enables the user to download new applications or install future updates. Amal Glass can read more than 100 languages and translates them to Arabic or English. The glass allows recording notes and playing voice notes like shopping lists and things to do. It eliminated the need for braille watches or talking watches – it can tell the exact time, date and day along with the stopwatch feature. It has a flashlight and guides about Qibla direction.

The start-up has won several awards beginning in 2014 the First Place At AINC Abu Dhabi Competition. In 2019, it was selected By the World Economic Forum (WEF) as one of the top 100 start-ups that will lead the Fourth Industrial Revolution in the Middle East. In the same year won the first place in UAE at entrepreneurship world cup and Nominated among top 40 over 100,000 Startups from all over the world.

Amal Glass Research and Development started in 2010 by The Egyptian Inventor: Mohammed Islam. After five years of hard work and perseverance Amal Glass came to life.

In 2016, Saudi investor, Al Bidnah Family, funded the project for commercial production and marketing. Two years later, another Saudi investor, Al Hozimi Family, joined Amal Glass for funding the production and continuing the research and development.

Envision Glasses (2020-2023) [3]

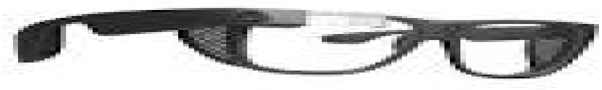


Figure 5 Envision Glasses

Version 2.0.0 January 26, 2023

The latest software update for Envision Glasses, version 2.1.0! This update introduces three new features designed to make life easier: Ask Envision, Wi-Fi Wizard, and Scan Wi-Fi QR Code.

Ask Envision

The new 'Ask Envision' feature allows users to ask questions about a piece of text or document they have just scanned, such as a menu card, electricity bill, or cooking instructions.

For example, after scanning a menu, user could ask Envision, '*What are the vegetarian options available?*' and receive a helpful response. This feature is powered by GPT-4 by OpenAI, one of the most advanced language models in the world, to understand and answer natural language questions.

To use 'Ask Envision', access the text 'Reader' after scanning document with the Scan Text feature. Press and hold the 'Hinge Button' on Envision Glasses until hear a chime, then ask the question and wait for the virtual assistant to answer.

Currently, voice prompts are only supported in English, however, the 'Ask Envision' output can be in any of the languages supported by Envision Glasses.

Since this feature uses ChatGPT, it may occasionally produce inaccurate information about people, places, or facts.

Wi-Fi Wizard

The Wi-Fi Wizard provides step-by-step guidance to help user connect Envision Glasses to the internet, whether it’s an iPhone hotspot, an Android hotspot, or a regular router.

This feature aims to make it easier for new and existing users to stay connected anywhere, anytime. To access the Wi-Fi Wizard, navigate to the sub-menu Wi-Fi, in the Settings menu or the sub-menu Trainings, in the Help menu, and perform a one-finger double tap to start the guide. Simply follow the instructions to get connected.

Scan Wi-Fi QR Code

Based on user feedback, Envision Glasses integrated Wi-Fi QR Codes into the existing Scan QR Code feature. Now, if the router has a QR Code it can easily connect the Envision Glasses by opening the Scan QR Code feature and pointing the camera at the QR Code.

Envision Glasses will automatically and immediately connect to the router, making the process seamless and hassle-free.

In addition to these new features, numerous under-the-hood changes and performance improvements to provide a better experience were made.

eSight Go 2023 smart glasses. [4]

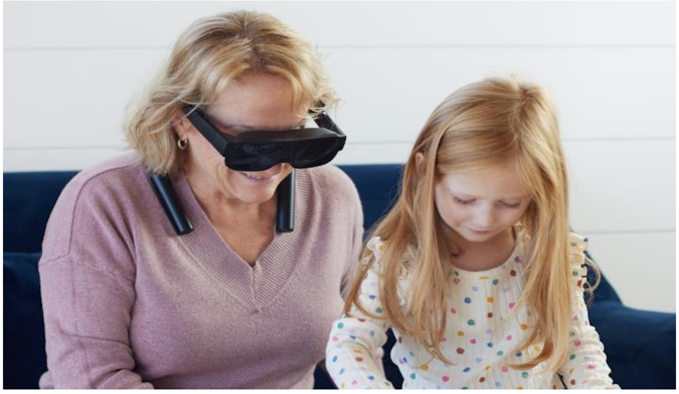


Figure 6 eSight's smart glasses

eSight's smart glasses are designed to help people with central vision loss regain their vision, and they do so using a combination of cameras, OLED screens, and clever software algorithms. The 2023 eSight Go model shows just how much technology has moved on and how much more difference it can potentially make to people's lives.

Incredible changes happen in people's lives when they use eSight smart glasses, which help partially sighted people see more clearly. It's life-changing technology, and the aim of its latest model is to help more people than ever.

The eSight isn't subtle, so some people are shy to wear it in everyday life because it puts a bit of a spotlight on them," the current smart glasses provide the same technological capability in a smaller format."

The eSight Go smart glasses are far more subtle than before, looking like a large pair of sunglasses.

"It's lighter, smaller, has a better camera, a better chip, and easier to navigate software. All the things are one level better. the battery has been taken out of the device, which was, in part, what has made it so small and light. The batteries are worn around the neck and connected with a short wire [to the headset]. On previous generations, it had a long wire to a suitcase on user hip, and it could get caught on things as user walked around. Now it’s not obstructive at all.”

1.3.2. Similar Projects for Smart Glasses

Authors	Title /Year	Features	Cons
Dr Ahmed El-Abbasi Eng. Hadeer Alim [5]	Smart Glasseses for Blind People 2017	-Can recognize images and determine each object in the images -Compute the distance between the person and each object	The glasses didn’t have the following: -GPS Locator -Wireless charging to recharge battery -Panic button -Sim card slot for data connectivity -Roller device at top to generate powers
Maghfirah Ali Tong Boon Tang [6]	Smart Glasses for the Visually Impaired People 2016	-New design of smart glasses that can provide assistance in multiple tasks at a low cost. -The design use Raspberry Pi 2 single board computer, a camera, and an earpiece to convey information to the user.	The paper only demonstrates reading task only.
Nallapaneni Manoj Kumar, Neeraj Kumar Singh, V. K. Peddiny [7]	Wearable Smart Glasses: Features, Applications, Current Progress and Challenges 2018	- voice recording - text preparation -video recording -location services, -data exchange	According to the paper conclusion more improvements are needed for the smart glasses to make itself as the best fit in many potential areas.

S.H. Vardhan, Jalluri Vardhan, Garapati Rakesh [8]	• SMART GLASSES FOR BLIND PEOPLE • 2021	1. It has an in-built sensor in it which spreads ultrasonic waves in the direction the person is going by scanning at most 5-6 meters of 30o range. As soon as the obstacle is detected, the sensor detects it and sends it to the device which generates an automated buzzer voice near the ear. 2. An IR sensor detects the position of the glasses when they fell off to the ground. As soon as it fell down it starts a buzzer that intimate the position to pick it again by the user.	1. Supporting only English language 2. The maximum distance of capturing the images is between150-300cm. 3. Design need to be more smaller and comfortable to wear.

Table 1-Similar Projects for Smart Glasses

1.4 . Project Objectives

Our objective is to develop smart glasses that can do the following features:

1) OCR (Optical Character Recognition)

OCR is a technology that enables the extraction and recognition of text from images or documents. By utilizing image processing techniques and machine learning algorithms, OCR systems convert printed or handwritten text into machine-readable format, facilitating tasks such as text extraction, document digitization, and accessibility for visually impaired individuals.

2) Text Translation

Text translation involves the automated conversion of text from one language to another. With advancements in natural language processing and machine translation algorithms, translation systems can accurately and efficiently translate text in real-time. This technology bridges language barriers and facilitates communication, making it useful in various scenarios, such as international business, travel, and multilingual content creation.

3) Scene Description

Scene description aims to provide a textual representation of the visual content within an image or video. Using computer vision techniques and deep learning algorithms, scene description systems analyze visual elements, objects, and their relationships to generate meaningful descriptions. This technology is beneficial for visually impaired individuals, as it allows them to comprehend the visual world through descriptive text.

4) Age/Gender Reveal

Age and gender reveal systems utilize facial analysis techniques to estimate a person's age and gender based on their facial features. By analyzing facial landmarks, wrinkles, skin texture, and other visual cues, these systems can provide approximate age and gender predictions. This technology finds applications in various fields, such as targeted advertising, demographic analysis, and personalized user experiences.

5) Color Identification

Color identification systems analyze the visual input from images or objects to identify and classify colors. By extracting color features using image processing algorithms, these systems can determine the dominant color or color range present in the input. Color identification has applications in areas such as fashion, design, and assistive technologies for color-blind individuals.

6) Scan QR code

QR code scanning involves the detection and decoding of QR (Quick Response) codes, which are two-dimensional barcodes containing encoded information. By utilizing computer vision techniques and QR code decoding algorithms, scanning systems can extract the embedded data from QR codes, enabling actions such as accessing websites, retrieving product information, or making payments.

7) Prayer-Time

Prayer-time systems provide accurate and up-to-date information regarding the timing of Islamic prayers. By integrating with relevant APIs or databases, these systems can retrieve prayer times based on geographical location and display them for users. Prayer-time systems are beneficial for individuals adhering to Islamic practices and help them in managing their prayer routines.

8) El Qibla Detection

El Qibla detection systems assist users in accurately determining the direction of the Kaaba (the holiest shrine in Islam) from their current location. By leveraging GPS coordinates or compass readings, these systems calculate the precise Qibla direction, allowing users to align themselves correctly during prayers.

9) Target Country/City Detection

Target country/city detection systems identify the country or city based on visual cues, such as landmarks, flags, or street signs. These systems employ computer vision techniques and pattern recognition algorithms to match the visual input with pre-defined reference images or databases. Target country/city detection has applications in tourism, navigation, and cultural exploration.

10) Recording Notes

Recording notes systems enable users to capture and store audio or text-based notes for later reference. These systems often provide features such as voice recognition for transcribing spoken notes, organization tools for categorizing notes, and synchronization across multiple devices. Recording notes systems are useful for individuals in academic, professional, or personal settings.

11) Distance Measurement

Distance measurement systems utilize various sensors, such as ultrasonic to estimate the distance between objects or surfaces. By emitting and measuring the time it takes for the signal to reflect, these systems can calculate distances accurately.

1.5 The Main Challenges [9]

During the design and implementation of these glasses, we encountered several challenges, including the following:

1. **Hardware and software requirements**

Developing fast, small, lightweight, and highly power-efficient processors that can fit into seamless and comfortable frames while considering hardware cost is a key challenge. Additionally, addressing concerns such as cameras that can account for tilting and head movement, optimizing battery life, and incorporating eye-tracking capabilities further add to the complexity of the project. Striking a balance between performance, cost, and user experience remains a critical aspect of the development process.

2. **Network security and bandwidth**

Providing seamless AR experiences based on cloud-hosted services and content requires uninterrupted Wi-Fi availability and high-security standards which is a challenge.

3. **Safety, privacy, and regulations**

Manufacturers are still trying to develop a standard on how to best interact with displayed content naturally. This has led to the continued iteration of controls and frame design.

4. **Education**

Eyeglasses must continue to serve their purpose, which is to correct and protect vision. In this regard, the public will need to learn how to operate these new tools and adapt to the influx of wireless data.

5. **Fashion**

Appearance was a significant hurdle for geeky-looking Google Glasses. The smart glasses of the future need to be sleek and stylish.

1.6 Report Organization

This report is organized as follows:

Chapter One introduces the project, discussing the motivation, challenges, and organization of the report.

Chapter Two presents the background and literature survey, including related work.

Chapter Three focuses on hardware design and module analysis.

Chapter Four outlines the software's main functions, algorithms, flowcharts, and provides snapshots of the output.

Finally.

Chapter Five presents the conclusion based on the experimental results and outlines future work in the field.

Chapter Two
Project Planning

Chapter Two

Project Planning

Introduction

Project planning is a discipline for stating how to complete a project within a certain timeframe, usually with defined stages, and with designated resources. One view of project planning divide into sections. Section 2.1 describes the project management and its description. Section 2.2 describes the Software tool that’s used to build up the planning which consists of 2.2.1 Project Subsystem, 2.2.2 Gantt chart for system tasks, and 2.2.3 The network diagram of the project.

2.1 Project Management

The project includes the following tasks, duration times for these tasks and dependency of each task.

Number	Task Name	Duration	Predecessors	Assigned Students
1	Problem Definition & Survey on Smart Glasses	7 days	-----	All
2	Identify Required Hardware & Features	14 days	1	All
3	Specify Functional Block Diagram	7 days	2	Ziad Waleed
4	Design of Glasses	60 days	2,3	Ziad Waleed
5	Identify required technique for each feature	30 days	2	Dina Mohamed Gamal Mohamed Omar Ismail Hany Ahmed Sayed
6	System Implementation	42.25 days	5	Dina Mohamed Gamal Mohamed Omar Ismail Hany Ahmed Sayed
7	Testing	60 days	6	Dina Mohamed Gamal Mohamed Omar
8	System documentation task	32 days	7	All

Network diagram for the project showing estimated times for each activity and the earliest and latest expected completion time for each activity.

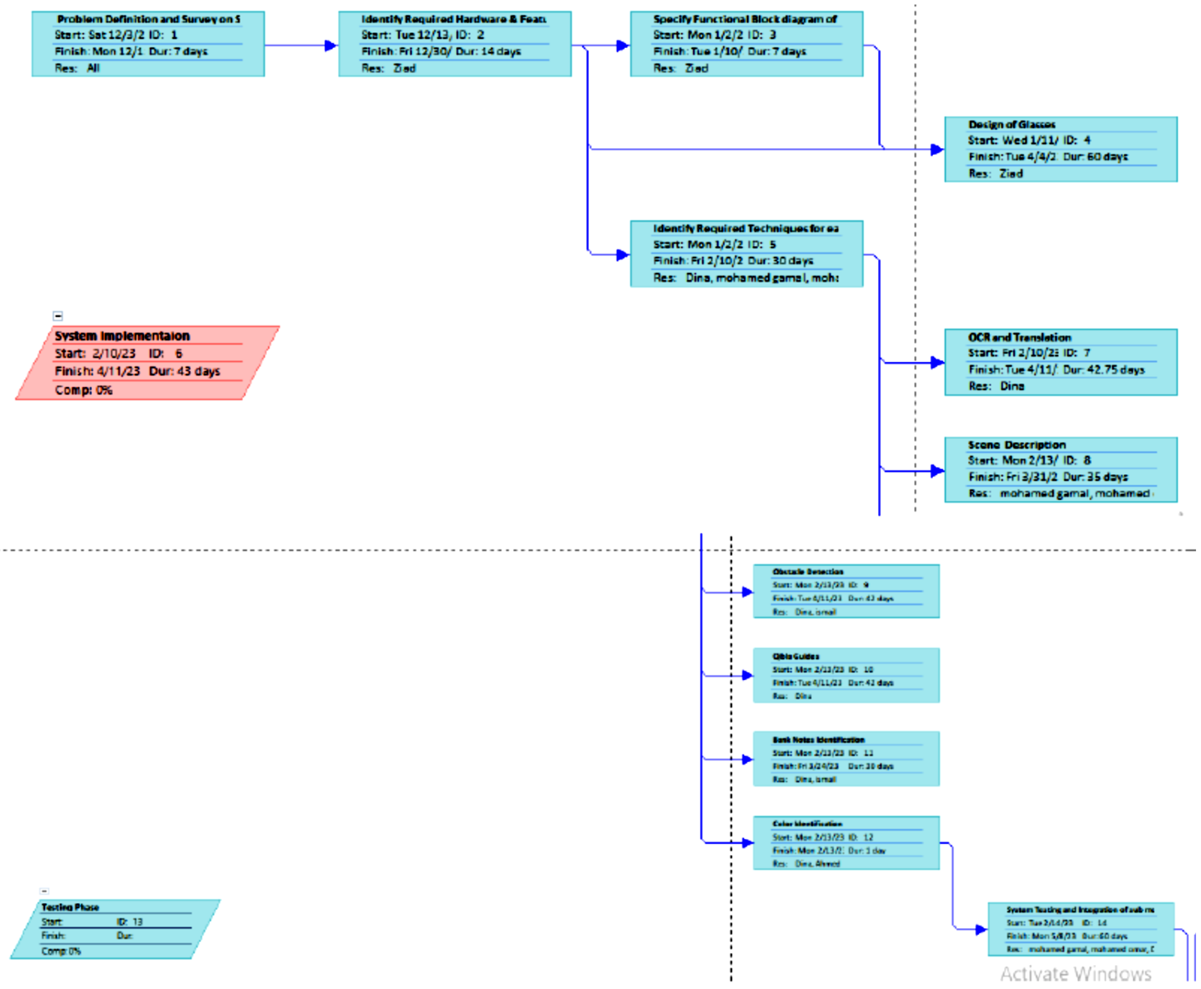

Problem Definition and Survey on S
Start: Sat 12/3/2 ID: 1
Finish: Mon 12/1 Dur: 7 days
Res: All
Identify Required Hardware & Featu
Start: Tue 12/13/ ID: 2
Finish: Fri 12/30/ Dur: 14 days
Res: Ziad
Specify Functional Block diagram of
Start: Mon 1/2/2 ID: 3
Finish: Tue 1/10/ Dur: 7 days
Res: Ziad
Design of Glasses
Start: Wed 1/11/ ID: 4
Finish: Tue 4/4/2 Dur: 60 days
Res: Ziad
Identify Required Techniques for ea
Start: Mon 1/2/2 ID: 5
Finish: Fri 2/10/2 Dur: 30 days
Res: Dina, mohamed gamal, moh
System Implementaion
Start: 2/10/23 ID: 6
Finish: 4/11/23 Dur: 43 days
Comp: 0%
OCR and Translation
Start: Fri 2/10/23 ID: 7
Finish: Tue 4/11/ Dur: 42.75 days
Res: Dina
Scene Description
Start: Mon 2/13/ ID: 8
Finish: Fri 3/31/2 Dur: 35 days
Res: mohamed gamal, mohamed
Obstacle Detection
Start: Mon 2/13/23 ID: 9
Finish: Tue 4/11/23 Dur: 42 days
Res: Dina, ismail
Qibla Guides
Start: Mon 2/13/23 ID: 10
Finish: Tue 4/11/23 Dur: 42 days
Res: Dina
Bank Notes Identification
Start: Mon 2/13/23 ID: 11
Finish: Fri 3/24/23 Dur: 30 days
Res: Dina, ismail
Color Identification
Start: Mon 2/13/23 ID: 12
Finish: Mon 2/13/2 Dur: 1 day
Res: Dina, Ahmed
Testing Phase
Start: ID: 13
Finish: Dur:
Comp: 0%
System Testing and Integration of sub m
Start: Tue 2/14/23 ID: 14
Finish: Mon 5/8/23 Dur: 60 days
Res: mohamed gamal, mohamed omar, [illegible]

Figure 7-Network Diagram

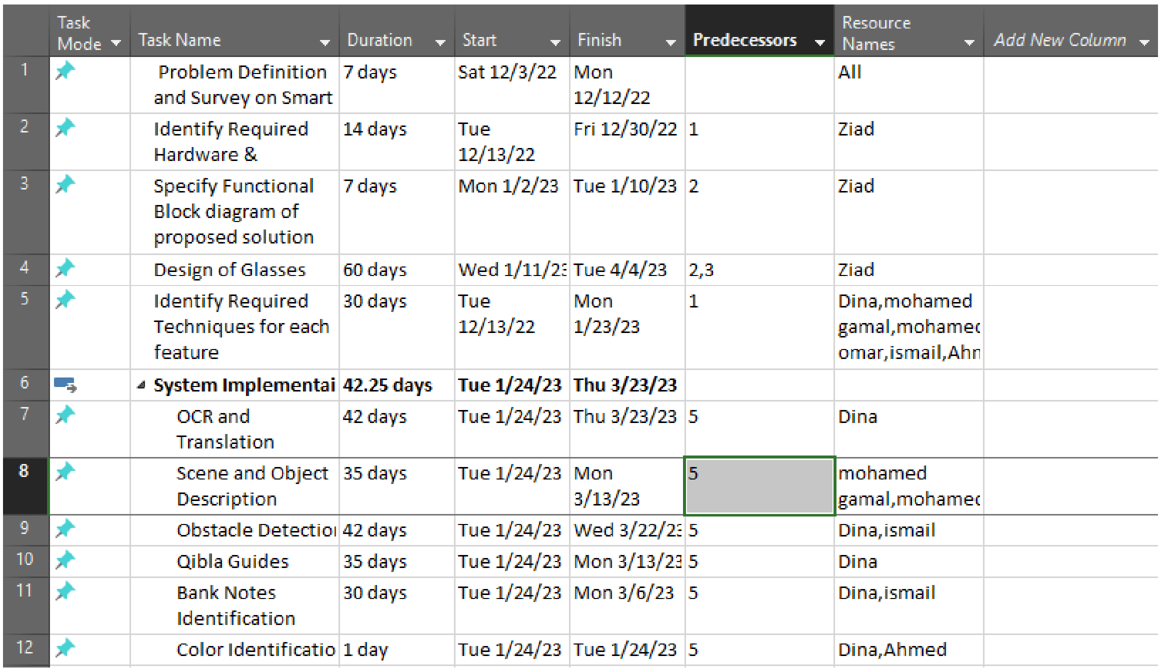
2.2 Software Tools

Software Tool for planning task made by (Microsoft project planning) Microsoft Project can do:

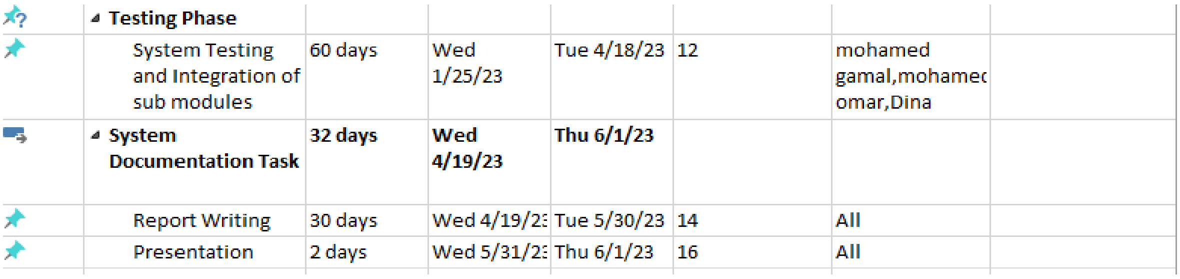
- Draw the action plan and represent it on both the Network Diagram and the Gantt chart.
- Allocate and organize resources for each activity.
- Follow-up progress of the project.
- Project budget management, and workload analysis.

- Group of interconnected and interactive parts that perform an important job or task as a component of a larger system.

2.2.1Gantt Chart



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
1	📌	Problem Definition and Survey on Smart	7 days	Sat 12/3/22	Mon 12/12/22		All	
2	📌	Identify Required Hardware &	14 days	Tue 12/13/22	Fri 12/30/22	1	Ziad	
3	📌	Specify Functional Block diagram of proposed solution	7 days	Mon 1/2/23	Tue 1/10/23	2	Ziad	
4	📌	Design of Glasses	60 days	Wed 1/11/2:	Tue 4/4/23	2,3	Ziad	
5	📌	Identify Required Techniques for each feature	30 days	Tue 12/13/22	Mon 1/23/23	1	Dina,mohamed gamal,mohamed omar,ismail,Ahn	
6	▬	**◢ System Implementai**	**42.25 days**	**Tue 1/24/23**	**Thu 3/23/23**			
7	📌	OCR and Translation	42 days	Tue 1/24/23	Thu 3/23/23	5	Dina	
8	📌	Scene and Object Description	35 days	Tue 1/24/23	Mon 3/13/23	5	mohamed gamal,mohamed	
9	📌	Obstacle Detectioi	42 days	Tue 1/24/23	Wed 3/22/2:	5	Dina,ismail	
10	📌	Qibla Guides	35 days	Tue 1/24/23	Mon 3/13/2:	5	Dina	
11	📌	Bank Notes Identification	30 days	Tue 1/24/23	Mon 3/6/23	5	Dina,ismail	
12	📌	Color Identificatio	1 day	Tue 1/24/23	Tue 1/24/23	5	Dina,Ahmed	



📌?	**◢ Testing Phase**						
📌	System Testing and Integration of sub modules	60 days	Wed 1/25/23	Tue 4/18/23	12	mohamed gamal,mohamed omar,Dina	
▬	**◢ System Documentation Task**	**32 days**	**Wed 4/19/23**	**Thu 6/1/23**			
📌	Report Writing	30 days	Wed 4/19/2:	Tue 5/30/23	14	All	
📌	Presentation	2 days	Wed 5/31/2:	Thu 6/1/23	16	All	

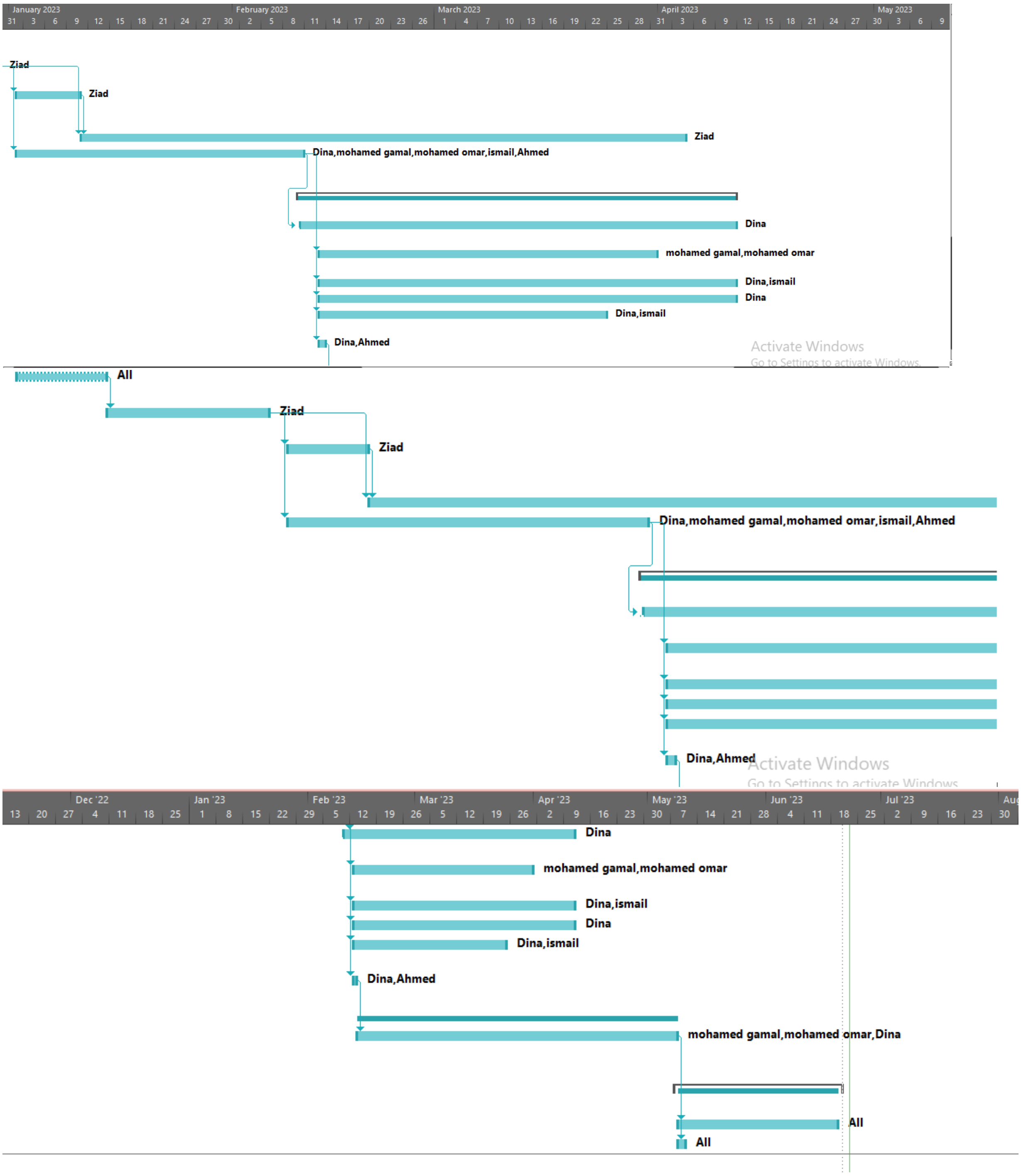


Figure 8-Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity,

Chapter 3

Smart Glasses System Analysis & Design

Chapter 3

Smart Glasses System Analysis & Design

Introduction

This chapter delves into the back end and analysis of the smart glasses, exploring their impressive features. **Optical Character Recognition (OCR)** is introduced, enabling real-time conversion of printed text into a digital format that can be easily accessed by screen readers or text-to-speech software. This feature allows visually impaired users to effortlessly access information from books, documents, and other printed materials.

The Text-to-Speech functionality further enhances the user experience by converting the digital text generated by OCR into audible speech. This intuitive and natural way of accessing information eliminates the need for braille or tactile methods, providing a more seamless and inclusive experience.

Scene Description takes advantage of computer vision algorithms to analyze the environment and verbally describe the scene. It provides information about object and people locations, spatial layout, and potential hazards, offering visually impaired users a comprehensive understanding of their surroundings and enhancing their independence and safety.

QR Code Scanning enables users to scan QR codes using the built-in camera of the smart glasses. This functionality grants instant access to additional information about products, locations, or exhibits. For example, in a museum, users can scan QR codes to receive detailed audio descriptions of artwork, enhancing their engagement and enjoyment of the exhibits.

Color Identification utilizes the camera to analyze colors in the environment and provide verbal descriptions. This feature assists visually impaired users in matching clothing or differentiating between items, allowing them to make informed decisions and navigate their daily lives more effectively.

Additionally, smart glasses incorporate **Age/Gender reveal** technology, which leverages advanced algorithms to estimate the age and gender of individuals in the environment. This contextual information can be valuable in various scenarios, such as social interactions or personalized services, enhancing the overall user experience.

By combining these powerful features, smart glasses offer visually impaired individuals' real-time access to information, improved independence, and a more inclusive and engaging interaction with the world. This chapter provides a comprehensive overview of the functionality and operation of each feature, showcasing the practical application and potential impact of our smart glasses project in various fields, including education, accessibility, and everyday life.

3.1 Functional Block Diagram

The functional block diagram of the smart glasses system, which consists of key components such as the Raspberry Pi, ultrasonic sensor, camera module, power bank, and headphones. The Raspberry Pi acts as the main processing unit, running the necessary software to control the other system components. The ultrasonic sensor measures distances using sound waves and provides distance information to the Raspberry Pi. The camera module captures images or video of the surroundings and is connected to the Raspberry Pi. The power bank serves as a portable battery, supplying power to the Raspberry Pi and other system components. Finally, the EarPods are worn to listen to audio output from the Raspberry Pi.

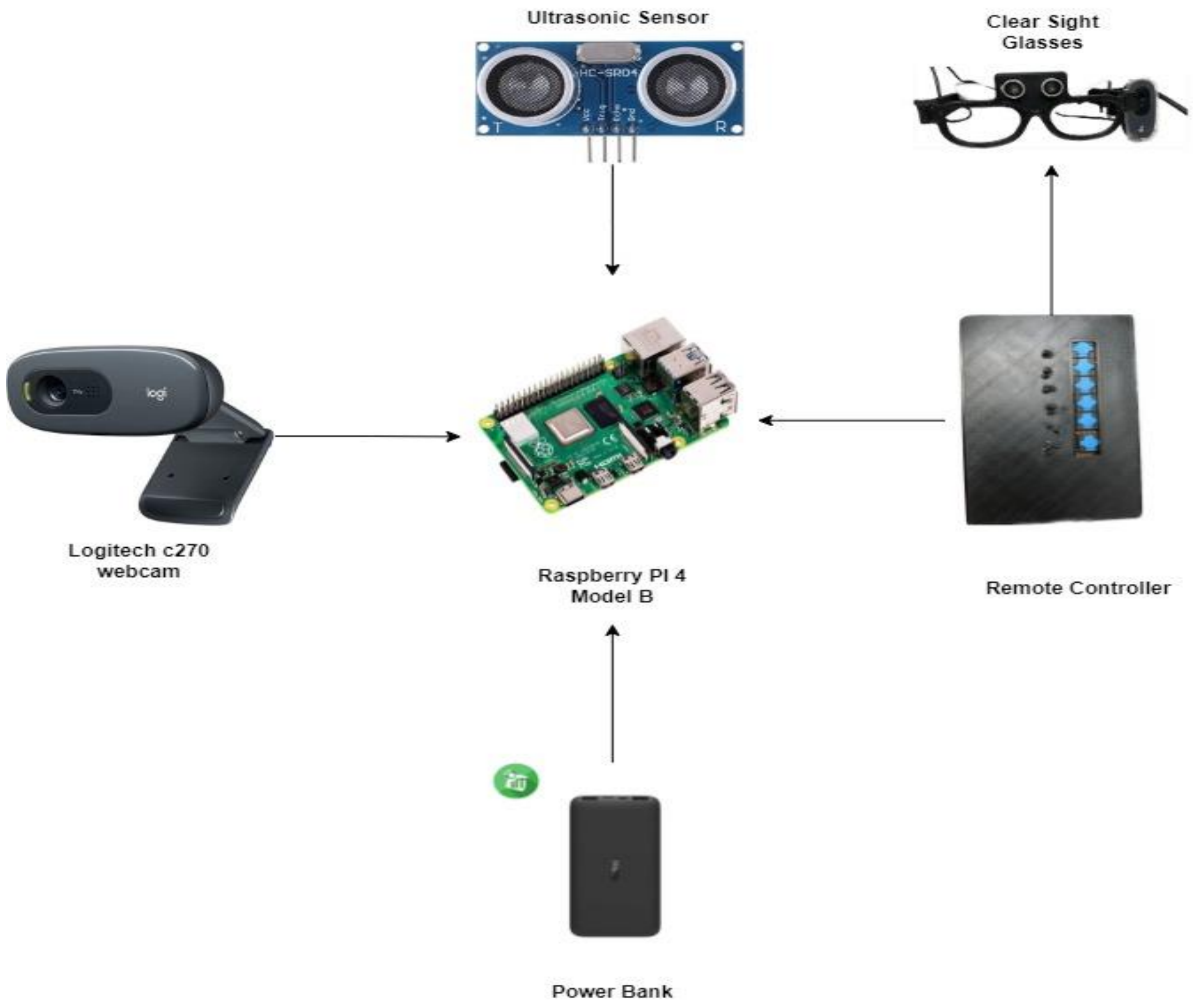


Figure 9-Functional Block Diagram

3.2 Hardware Components

Smart glasses for blind individuals are designed to enhance their daily lives and promote independence. Our goal is to develop a discreet and inconspicuous design that maintains the wearer's privacy, allowing them to navigate public spaces confidently without drawing unnecessary attention to their visual impairment.

The hardware components work together seamlessly to provide essential functionalities, including capturing images, analyzing the environment, and delivering audio feedback, empowering visually impaired individuals to navigate their surroundings effectively.

Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B is a powerful and versatile single-board computer that serves as the brain of a smart glasses system designed to assist visually impaired individuals. It features a microprocessor rather than a microcontroller, offering enhanced processing power and computational capabilities. With its 4GB of RAM, it can handle complex tasks and run multiple applications simultaneously. The Raspberry Pi 4 Model B also includes dual-band wireless LAN and Bluetooth 5 connectivity for seamless communication with other devices. Its micro-SD card slot provides storage for the operating system and data, allowing for efficient performance and data management. This makes the Raspberry Pi 4 Model B an ideal choice for building advanced smart glasses systems that can provide comprehensive assistance to visually impaired individuals.

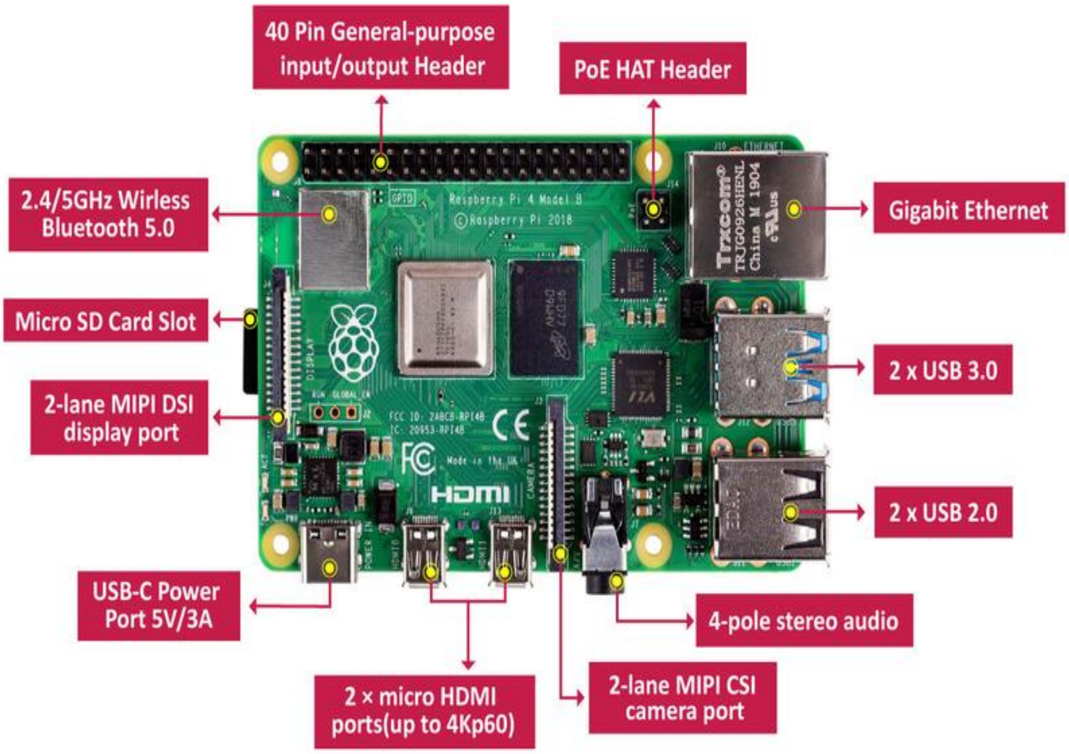

40 Pin General-purpose input/output Header
PoE HAT Header
2.4/5GHz Wirless Bluetooth 5.0
Gigabit Ethernet
Micro SD Card Slot
2 x USB 3.0
2-lane MIPI DSI display port
2 x USB 2.0
USB-C Power Port 5V/3A
4-pole stereo audio
2 × micro HDMI ports(up to 4Kp60)
2-lane MIPI CSI camera port

Figure 10-Raspberry PI 4

Ultrasonic sensors

Ultrasonic sensors are electronic devices that utilize sound waves to detect the presence, distance, or movement of objects. They work based on the principle of sending out ultrasonic waves and measuring the time it takes for the waves to bounce back after hitting an object. This is achieved using two main components: a trigger and an echo.

The trigger component emits a short burst of ultrasonic waves, usually in the frequency range of 20 kHz to 200 kHz, which is beyond the range of human hearing. These waves propagate through the environment until they encounter an object. Upon hitting the object, the waves are reflected as an echo.

The echo component of the ultrasonic sensor receives the reflected waves and calculates the time it takes for the waves to return. By knowing the speed of sound in the medium, which is approximately 343 meters per second in dry air at room temperature, the sensor can determine the distance to the object using the equation distance = speed × time.

Ultrasonic sensors offer several advantageous properties. They can detect objects at varying distances, ranging from a few centimeters to several meters, with a high degree of accuracy and resolution. Additionally, their directionality can be controlled, allowing for targeted detection within a specific area. They are robust and capable of operating in challenging environments, such as those with dust or moisture. Furthermore, ultrasonic sensors consume relatively low power, making them suitable for applications that rely on battery power.

Overall, ultrasonic sensors are valuable components in many systems, including smart glasses for the visually impaired, as they enable accurate object detection and provide essential information for navigation and obstacle avoidance.

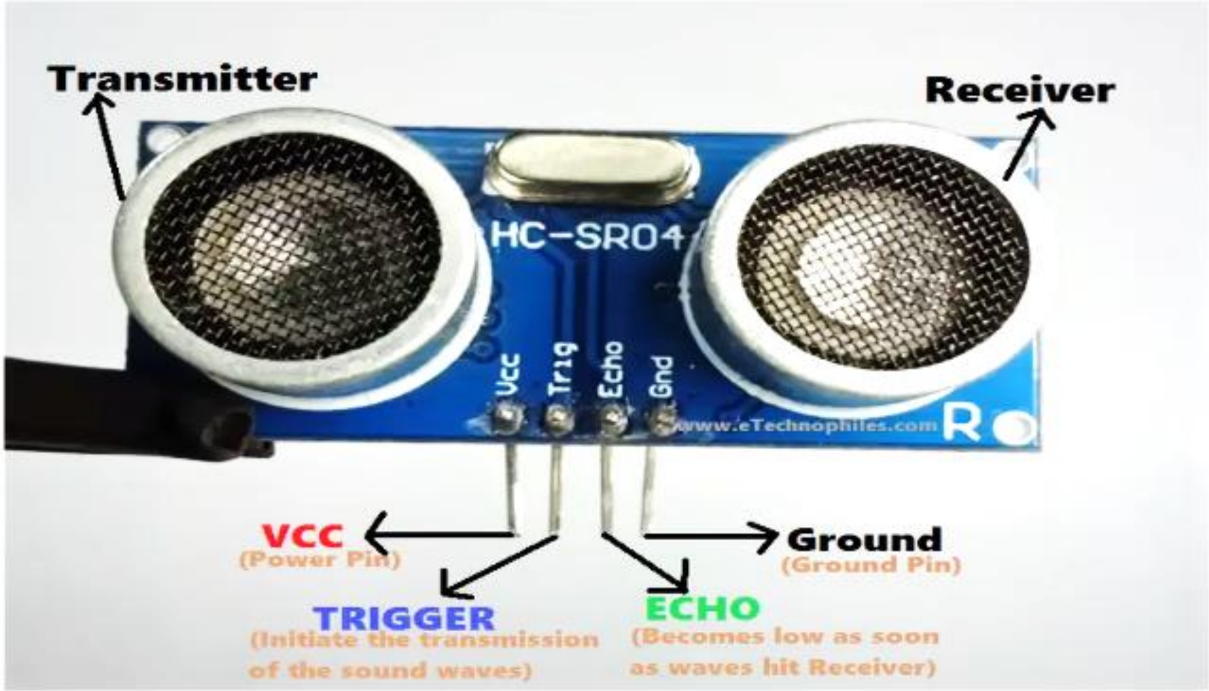


Figure 11-Ultrasonic sensor

Logitech C270 HD Webcam

The Logitech C270 HD Webcam is a popular webcam model known for its high-definition video quality and versatile features. When paired with the Raspberry

Pi 4, it can be used for various applications, such as video conferencing, live streaming, surveillance, and computer vision projects. Here are some details about the Logitech C270 HD Webcam and its compatibility with the Raspberry Pi 4:

1. Video Quality: The Logitech C270 HD Webcam supports video resolutions up to 720p, which allows for clear and sharp video output. It also features a built-in microphone for capturing audio during video recordings or online calls.

2. Compatibility: The Logitech C270 HD Webcam is compatible with a variety of operating systems, including Linux, which is the operating system commonly used on the Raspberry Pi. This ensures seamless integration and compatibility with the Raspberry Pi 4.

3. Software Support: The webcam is supported by various software applications and libraries on the Raspberry Pi 4. For example, OpenCV, a popular computer vision library, can be used to capture video frames from the webcam and perform image processing tasks.

4. Cost-Effective Solution: The Logitech C270 HD Webcam is an affordable option compared to higher-end webcams, making it an ideal choice for budget-conscious projects or individuals looking for a reliable webcam solution without compromising on video quality.

When combined with the Raspberry Pi 4, the Logitech C270 HD Webcam offers a powerful and accessible tool for capturing high-definition video.

Figure 12-Logitech C270 HD Webcam

The Xiaomi Redmi PB100LZM Power Bank

The Xiaomi Redmi PB100LZM Power Bank is a portable charging device with a capacity of 10000mAh, designed to provide a convenient power source for various electronic devices. While it doesn't have a direct connection to the Raspberry Pi 4, it can be used alongside the Raspberry Pi 4 as a power supply. Here are some details about the Xiaomi Redmi PB100LZM Power Bank and its compatibility with the Raspberry Pi 4:

1. Capacity: With a capacity of 10000mAh, the power bank can provide multiple charges for the Raspberry Pi 4. The Raspberry Pi 4 is a low-power device, and depending on the usage, it can run for several hours on a fully charged power bank.

2. Portability: The compact and lightweight design of the power bank makes it highly portable, making it convenient for powering the Raspberry Pi 4 on the go.

3. Safety Features: The Xiaomi Redmi PB100LZM Power Bank incorporates multiple safety features, such as overcharge protection, short circuit protection, and temperature control, ensuring the safe operation of the connected devices, including the Raspberry Pi 4.

Figure 13-Xiaomi Redmi PB100LZM Power Bank

Samsung EVO Plus 128GB Micro SD Card

The Samsung EVO Plus 128GB microSDXC UHS-I U3 memory card is a high-performance storage solution that offers ample capacity and fast data transfer speeds. When used with the Raspberry Pi 4 Model B, it can provide expanded storage for the device and enhance its capabilities. Here's how you can set up a 64-bit operating system on the Raspberry Pi 4 using the Samsung EVO Plus memory card:

1. Compatibility: The Samsung EVO Plus memory card is compatible with the Raspberry Pi 4 Model B, as it supports the required microSDXC format.

2. Capacity: With its 128GB capacity, the Samsung EVO Plus memory card offers plenty of space to store your operating system, applications, files, and media.

3. Speed: The UHS-I U3 speed rating of the Samsung EVO Plus memory card ensures fast data transfer rates of up to 130MB/s. This speed is beneficial when reading or writing data, as it reduces loading times and enhances overall system performance.

The Samsung EVO Plus 128GB microSDXC UHS-I U3 memory card is an excellent choice for installing a 64-bit operating system on the Raspberry Pi 4

Model B. With its high storage capacity and fast data transfer speeds, it provides ample space for the operating system, applications, and files.

Figure 14-Samsung Evo Plus 128GB Micro SD Card

Push Buttons

Push buttons are physical switches that can be integrated into the Raspberry Pi 4 Model B to enable user interaction and activate various features. These buttons serve as input devices, allowing users to trigger specific actions or functions within the smart glasses system. By pressing a button, users can initiate processes such as capturing images, playing audio, or activating voice commands.

In the context of smart glasses for blind people, push buttons play a crucial role in providing tactile feedback and navigation assistance. The buttons are enhanced with braille stickers, allowing visually impaired users to locate and differentiate between different buttons by touch. This tactile feedback helps users navigate through menus, select options, and control the smart glasses functionalities with ease.

To enable the functionality of these push buttons, a Python library can be utilized. This library provides an interface to interact with the Raspberry Pi's GPIO (General Purpose Input Output) pins. Through programming, developers can assign specific functions or commands to the push buttons, linking them to desired actions within the smart glasses system. This allows users to conveniently access and control various features and capabilities by simply pressing the corresponding buttons.

The integration of push buttons with braille stickers, combined with the Python library, enhances the usability and accessibility of smart glasses for blind people. It empowers visually impaired individuals to navigate through different functionalities and interact with the system more effectively, contributing to their overall independence and improved user experience.

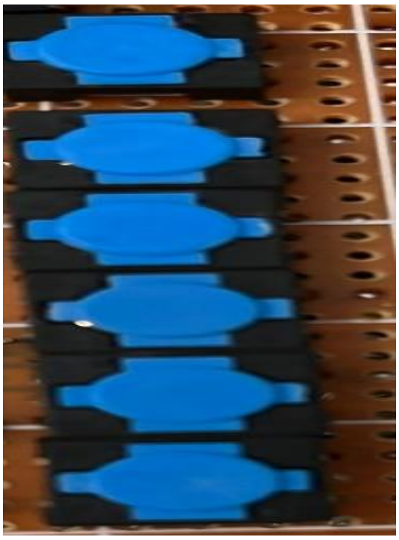


Figure 15-Push Buttons

Jumper Wires

Jumper wires are essential components used in electronics and can be employed in conjunction with the Raspberry Pi 4 Model B to establish connections between different electronic components. These wires typically consist of insulated conductive material, such as copper, with pins or connectors on each end that can be easily inserted into the GPIO (General Purpose Input Output) pins of the Raspberry Pi.

Jumper wires play a crucial role in creating electrical pathways between various components, such as sensors, buttons, and modules, and the Raspberry Pi. They facilitate the transmission of signals, power, and data between these components, enabling them to communicate and interact effectively.

With the Raspberry Pi 4 Model B, jumper wires allow for flexible and customizable circuit connections, as they can be easily plugged in and removed without the need for soldering. This feature is particularly advantageous when prototyping or experimenting with different configurations and setups.

Additionally, jumper wires enable the Raspberry Pi to interface with external devices and modules, expanding its functionality and capabilities. By connecting sensors, actuators, and other components using jumper wires, users can create complex systems and projects that can interact with the physical world and respond to various inputs.

Overall, jumper wires are versatile and indispensable tools in electronics and Raspberry Pi projects. They provide the necessary connectivity and flexibility to establish connections between components, allowing for seamless integration and efficient communication within the system.

Figure 16-Jumper Wires

PCB HS-03

A printed circuit board (PCB) is a fundamental component in electronics that serves as a platform for connecting and supporting electronic components. The main function of a PCB is to provide a mechanical structure and electrical connections for various electronic elements, such as integrated circuits, resistors, capacitors, and other components.

The primary purpose of a PCB is to provide a reliable and efficient means of routing electrical signals between different components on the board. Copper traces, typically etched onto the surface of the PCB, form the pathways that carry electrical current between the components. These traces are carefully designed and laid out to ensure proper signal flow, minimize interference, and maintain signal integrity.

PCBs also serve as a means of mechanical support for the electronic components. They provide a sturdy platform for mounting and securing the components, ensuring proper alignment and preventing damage during operation or handling.

Furthermore, PCBs help to organize and optimize the circuit layout. By carefully arranging the components and their interconnections on the board, PCB designers can minimize signal interference, reduce noise, and optimize the overall performance of the circuit.

PCBs offer several advantages over traditional point-to-point wiring or breadboard circuits. They provide a compact and space-efficient solution, allowing for complex circuits to be implemented in a small form factor. They also improve the reliability and durability of electronic systems by eliminating loose connections and reducing the risk of short circuits or accidental component damage.

In summary, the primary function of a PCB is to provide a robust and reliable platform for connecting and supporting electronic components. By facilitating efficient electrical connections, mechanical support, and optimized circuit layout, PCBs play a crucial role in the design and manufacturing of electronic devices and systems.

Figure 17-PCB HS-03

Resistors

Resistors are passive electronic components that are widely used in electronic circuits to control the flow of electric current. A 1k ohm resistor has a resistance value of 1,000 ohms, while a 2k ohm resistor has a resistance value of 2,000 ohms.

The primary function of resistors is to limit the amount of current flowing through a circuit. They are used to create voltage drops, divide voltage, and control the flow of current in various parts of a circuit. Resistors can be connected in series or parallel to achieve specific resistance values and desired circuit characteristics. In addition to controlling current flow, resistors also play a role in voltage division. By using resistors in voltage divider circuits, it is possible to obtain a desired voltage level at a specific point in the circuit. This is commonly used in amplifier circuits, sensor interfaces, and voltage level shifting.

Resistors are available in various power ratings, tolerance levels, and physical sizes. The power rating of a resistor determines its ability to dissipate heat generated when current flows through it. Higher power ratings are required for resistors that handle larger currents or are subjected to higher voltages.

The tolerance of a resistor indicates the range within which its actual resistance value may deviate from the stated nominal value. For example, a resistor with a 5% tolerance means that its resistance can vary by up to 5% from the specified value. Tolerance levels are important in ensuring the accuracy and reliability of circuit designs.

Resistors are typically color-coded to indicate their resistance value, tolerance, and other characteristics. The color bands on the resistor body provide a quick visual reference to determine its resistance value based on a standardized color code.

In summary, resistors are essential components in electronic circuits that help control current flow and voltage levels. The 1k ohm and 2k ohm resistors, with their specific resistance values, are commonly used in various applications, including voltage dividers, current limiting circuits, and signal conditioning circuits. Their precise resistance values and tolerance levels contribute to the accurate and reliable operation of electronic system.

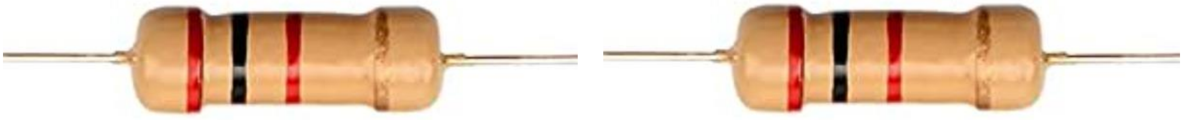


Figure 18-Resistors 1k ohm and 2k ohm

Breadboard

A breadboard is a fundamental tool used in electronics prototyping and experimentation. It is a rectangular board with numerous interconnected holes or sockets that allow electronic components to be easily connected and tested without the need for soldering.

The primary function of a breadboard is to provide a platform for creating temporary circuits. It has rows and columns of interconnected metal clips or spring connectors beneath the surface, which allow components to be inserted and connected by simply pushing their leads or pins into the holes. The interconnected clips establish electrical connections between the components and facilitate the flow of current through the circuit.

Breadboards are particularly useful for quickly testing and prototyping electronic circuits. They enable rapid experimentation, allowing components to be easily rearranged and connected in different configurations without the need for permanent soldering. This flexibility allows for iterative design and troubleshooting, making it an essential tool for electronics enthusiasts, students, and professionals.

When it comes to testing glasses, it is essential to clarify the context or specific purpose of testing. If you are referring to testing the functionality of smart glasses for visually impaired individuals, the process would involve verifying the proper operation of features such as object detection, text-to-speech conversion, scene description, or any other relevant functionality. This testing could include verifying the accuracy of object recognition, assessing the clarity and intelligibility of speech output, evaluating the effectiveness of scene descriptions, and ensuring the overall usability and performance of the glasses in assisting visually impaired users.

Testing glasses typically involves a combination of functional testing, where each feature is evaluated individually for its intended purpose, and user testing, where visually impaired individuals provide feedback and insights on the effectiveness and usability of the glasses in real-world scenarios. This iterative testing process helps refine and improve the design and functionality of the glasses to better meet the needs of visually impaired users.

In conclusion, a breadboard is a versatile tool for prototyping electronic circuits, allowing easy connection and testing of components without soldering. Testing glasses involves assessing the functionality and usability of smart glasses designed for visually impaired individuals, ensuring that the features perform as intended and meet the needs of the users.

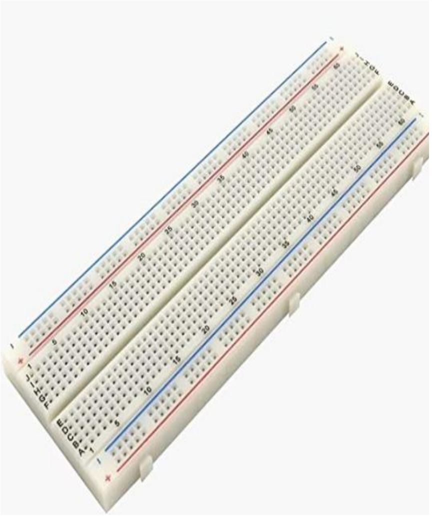


Figure 19- Breadboard

3.3 Design of Smart Glasses

As Shown in figure 20 the user interacts with the smart glasses through a remote control that consists of Braille words, allowing them to access the desired features.

Additionally, the glasses incorporate sensors for obstacle detection, and the built-in Logitech C270 camera provides visuals with a resolution of 1280x720 pixels.

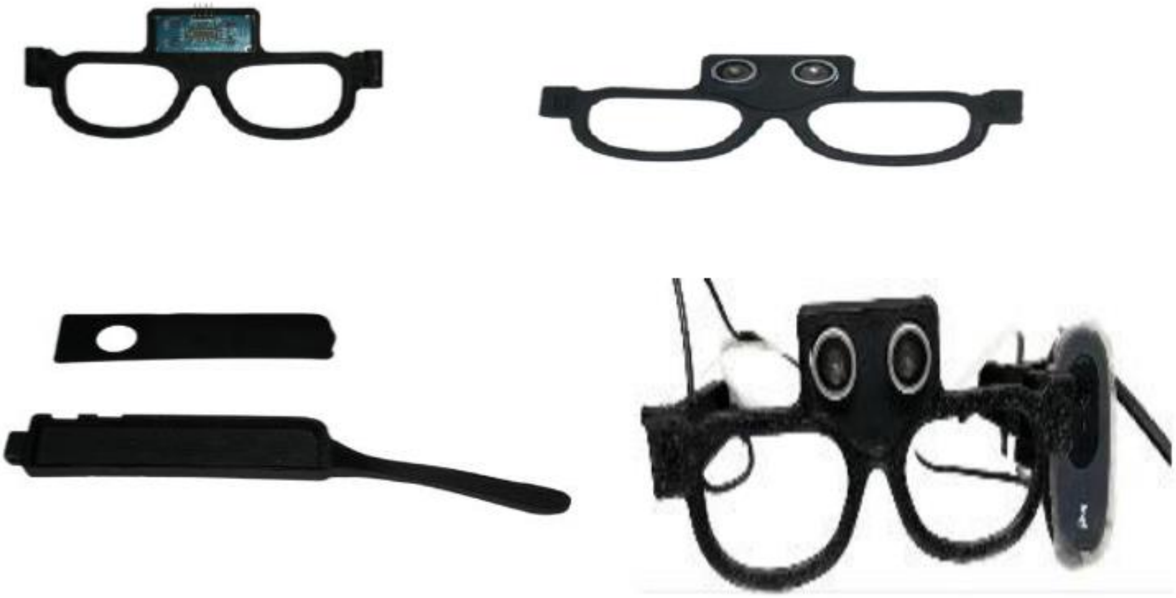


Figure 20-Smart Glasses Design

The Braille Alphabet

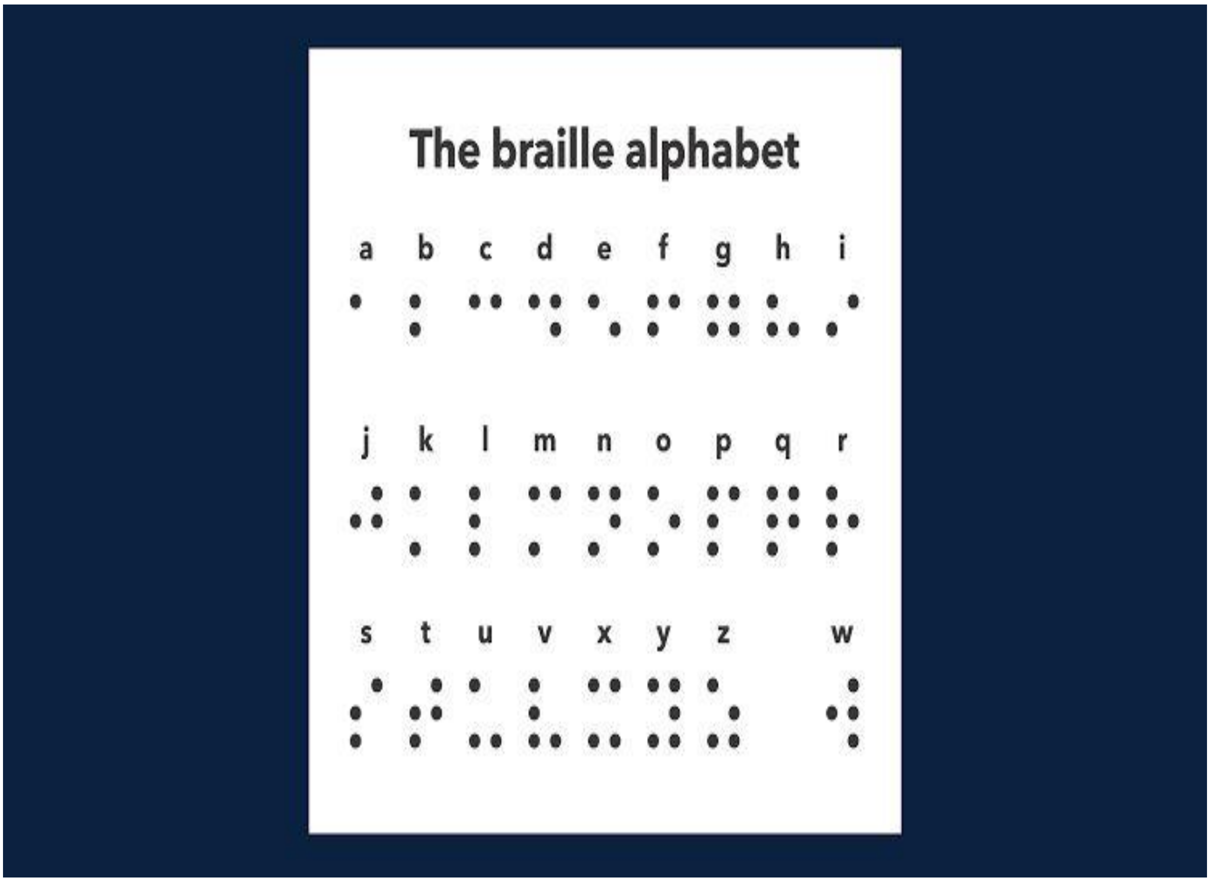


Figure 21-Braille Alphabet

As depicted in Figure 21, Braille is a writing system designed for individuals with visual impairments. It utilizes raised dots arranged in specific patterns to represent letters, numbers, and symbols. Each Braille cell consists of six dots, and letters are formed by modifying dot combinations. Braille enables blind individuals to read through touch and incorporates contractions to enhance reading efficiency, facilitating independent communication, and promoting literacy among people with visual impairments.

3.4 Optical Character Recognition Module

Optical Character Recognition (OCR) is a technology that enables the conversion of printed or handwritten text into machine-readable text.

It involves the use of algorithms and pattern recognition techniques to identify and extract characters from images or scanned documents, such as photographs, PDF files, or paper documents.

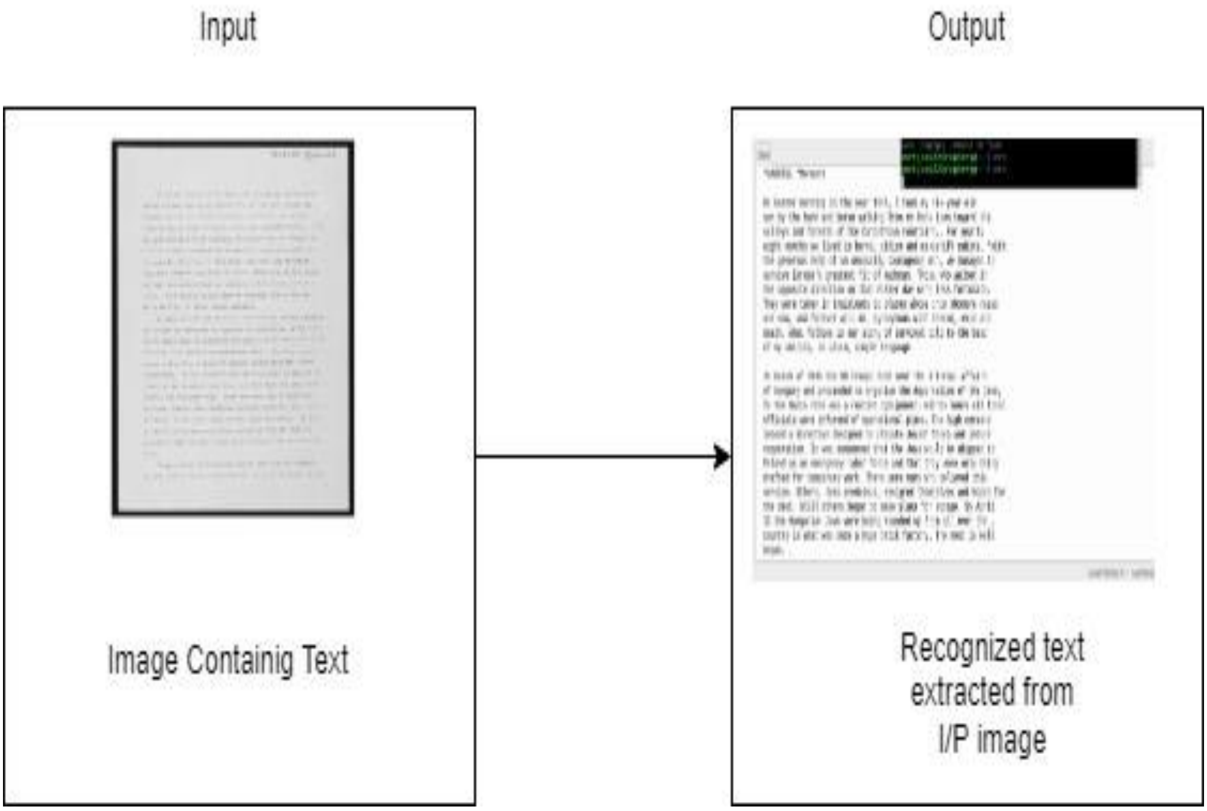


Figure 22-OCR

How OCR Works

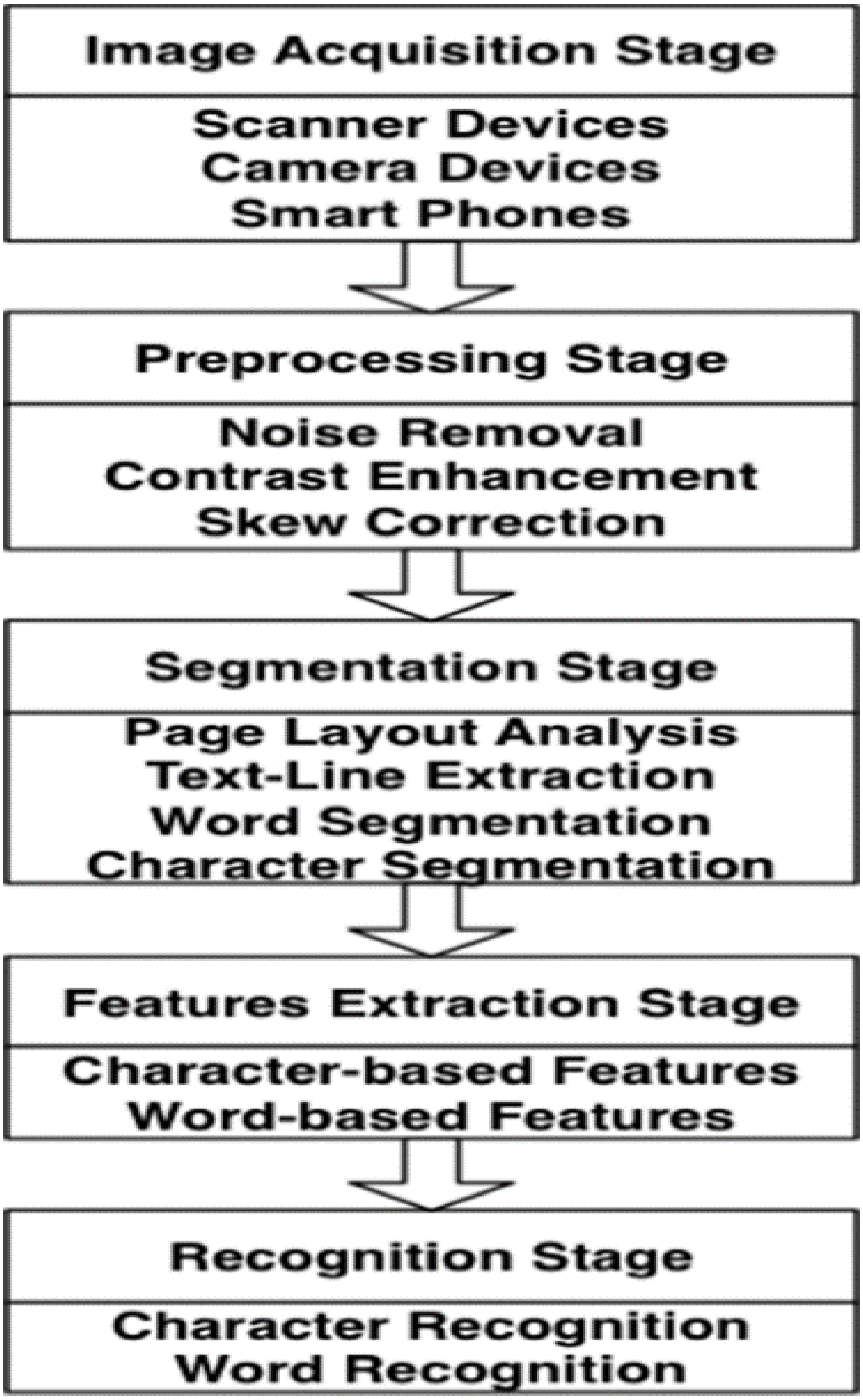


Figure 23-How OCR Works

1. Image Acquisition: In the first step, a Logitech webcam is used to capture an image of the text. The webcam is connected to a Raspberry Pi, which serves as the processing unit for the smart glasses. The captured image is then saved on the Raspberry Pi for further processing.

2. Preprocessing Stage: Once the image is acquired, preprocessing techniques are applied to enhance the quality and prepare the image for subsequent stages. Common preprocessing steps include resizing the image to a standard size, converting it to grayscale to simplify processing, and applying filters to reduce noise or improve contrast.

3. Segmentation Stage: In the segmentation stage, the preprocessed image is analyzed to identify individual characters or regions of text. This step is crucial to separate the text from the background and other visual elements. Techniques such as edge detection, thresholding, or connected component analysis can be used to segment the text regions.

4. Feature Extraction: After the segmentation stage, the segmented text regions need to be converted into a suitable format for recognition. Feature extraction involves extracting relevant characteristics or features from the segmented text regions. This process can include techniques such as extracting pixel intensity patterns, shape descriptors, or texture features to represent each character or text region.

5. Recognition Stage: Once the features are extracted, the recognition stage involves matching the extracted features with known patterns or characters. This step can be performed using various algorithms, such as machine learning-based classifiers or template matching techniques. The recognized characters are then converted into text, and the final result is provided as the output to the user, typically via an audio output or display on the smart glasses.

Overall, these stages work together to enable the smart glasses to capture an image, preprocess it, segment the text regions, extract relevant features, and recognize the text, providing valuable information to visually impaired individuals.

Image Preprocessing Phase

Image preprocessing plays a crucial role in optimizing Optical Character Recognition (OCR) results by improving the quality of input images and enhancing the visibility of text. Several techniques are commonly employed during the preprocessing stage to prepare the image for accurate OCR analysis. Let's delve into four key techniques: Image Inversion, Gray-Scale Conversion, Binarization, and Noise Removal.

1. Image Inversion: In this technique, the colors of the image are inverted, resulting in a negative image. This is done to enhance the text present in the image, particularly when the text is lighter than the background. By inverting the colors, the text becomes more prominent, aiding the subsequent OCR process.

2. Gray-Scale Conversion: Gray-scale conversion involves transforming the image into shades of gray, removing color information. This process reduces the color space, simplifying the image and making it easier to extract text features. By converting the image to gray-scale, it eliminates the potential distractions caused by color variations, allowing OCR algorithms to focus solely on analyzing the text.

3. Binarization: Binarization is the process of converting the gray-scale image into a binary image, where only black and white pixels are present. The objective of binarization is to separate the foreground, which includes the text, from the background of the image. By thresholding the gray-scale image, pixels are assigned either black or white based on their intensity values, making the text more distinguishable for subsequent OCR analysis.

4. Noise Removal: Noise removal is a vital step to eliminate any unwanted details that may interfere with the OCR process. Small artifacts, scratches, or speckles in the image can degrade the accuracy of character recognition. Noise removal techniques, such as smoothing filters or morphological operations, are applied to clean up the image and suppress unwanted elements while preserving the essential text information.

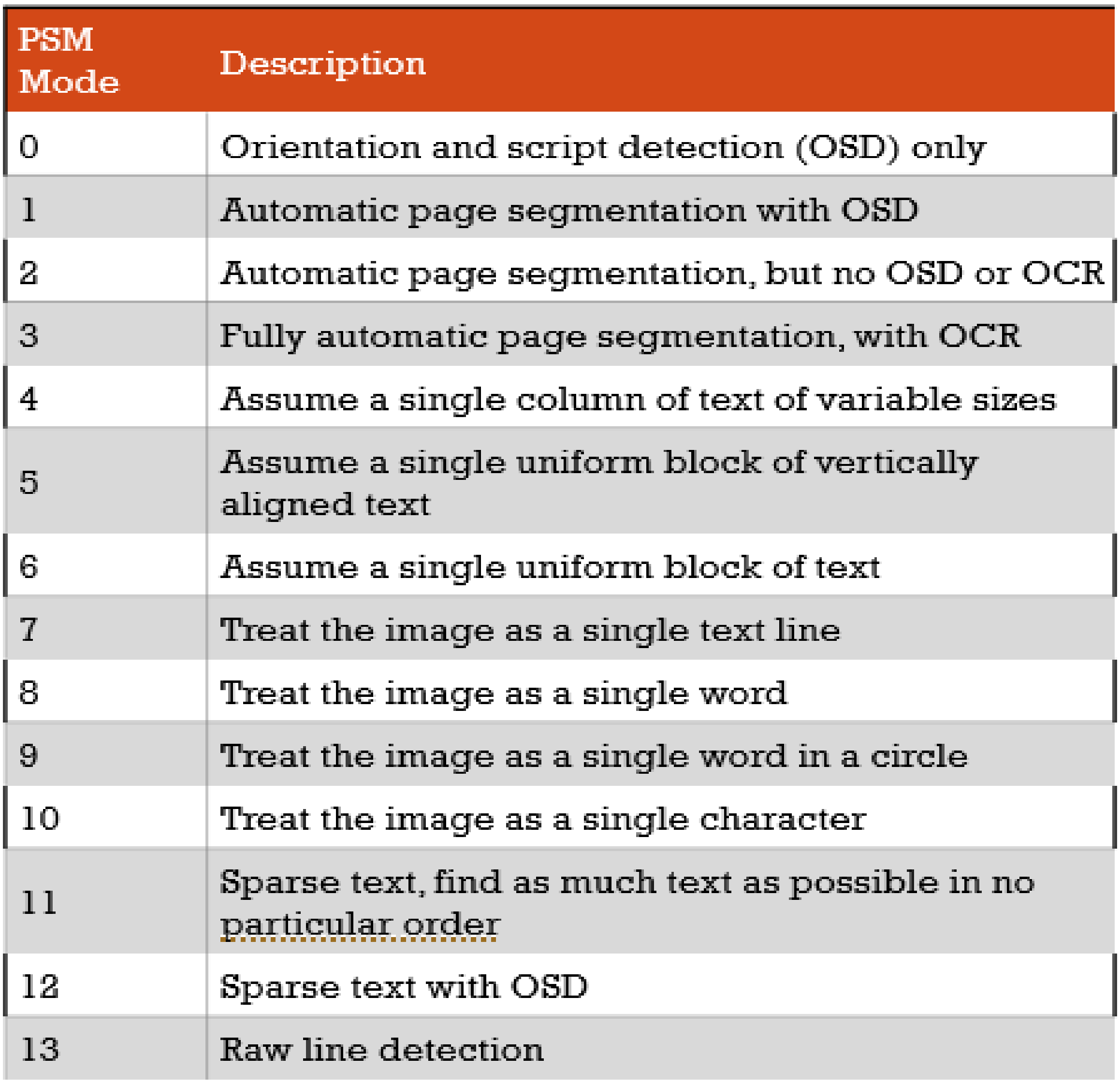
By incorporating these image preprocessing techniques into the OCR workflow, the quality and accuracy of text extraction are significantly enhanced. These steps help to optimize the image for better recognition, improve the contrast between text and background, and reduce potential distractions caused by noise or color variations. The goal is to provide clear and well-defined text regions that enable the OCR algorithms to perform accurate character recognition and produce reliable results.

Page Segmentation Modes (PSM)

Page Segmentation Mode (PSM) is a crucial parameter in OCR that determines how an input image is segmented into text regions.

It significantly impacts the accuracy and quality of OCR output. PSM defines the level of granularity in text segmentation, ranging from treating the entire image as a single text block to segmenting the text line by line, word by word, or even character by character.

The choice of PSM depends on the layout and structure of the input document. Selecting the appropriate PSM is essential as it directly affects the OCR's ability to correctly identify and interpret text regions, impacting the accuracy of the recognized output.



PSM Mode	Description
0	Orientation and script detection (OSD) only
1	Automatic page segmentation with OSD
2	Automatic page segmentation, but no OSD or OCR
3	Fully automatic page segmentation, with OCR
4	Assume a single column of text of variable sizes
5	Assume a single uniform block of vertically aligned text
6	Assume a single uniform block of text
7	Treat the image as a single text line
8	Treat the image as a single word
9	Treat the image as a single word in a circle
10	Treat the image as a single character
11	Sparse text, find as much text as possible in no particular order
12	Sparse text with OSD
13	Raw line detection

Figure 24-PSM Modes [10]

Tesseract OCR Library

The Tesseract library is an open-source Optical Character Recognition (OCR) engine developed by Google. It is widely used for extracting text from images or scanned documents. Tesseract supports multiple languages and has gained popularity for its accuracy and reliability in recognizing text in various contexts. The library utilizes advanced image processing techniques, pattern recognition algorithms, and machine learning to analyze and interpret the visual patterns of characters in an image. It can handle various types of documents, including printed text, handwriting, and typewritten text.

Tesseract is available as a command-line tool, but it also provides language bindings for different programming languages, such as Python, Java, and C++. These language bindings allow developers to integrate the Tesseract OCR engine into their applications and leverage its capabilities for text extraction and recognition tasks.

With the Tesseract library, developers can build applications that automate data entry, enable text search in images, perform document analysis, and enable accessibility features for visually impaired individuals. It has become a popular choice for OCR-related tasks due to its accuracy, extensibility, and wide language support.

Install Tesseract Library on Raspberry Pi

To install the Tesseract library on Raspberry Pi, you can follow these steps. First, ensure that your Raspberry Pi is connected to the internet and accessible via SSH or directly. Open a terminal and execute the following commands to update the system's package lists and upgrade the existing packages:

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

Next, install Tesseract and its required dependencies by running the following command in the terminal:

```
sudo apt-get install tesseract-ocr
```

By default, Tesseract only installs the English language data. If you need support for additional languages, you can install them using the command:

```
Sudo apt-get install tesseract-ocr-[language code]
```

Replace `[language code]` with the appropriate language code, such as `spa` for Spanish.

To verify the installation, use the following command to view the installed version of Tesseract:

```
tesseract --version
```

If the installation was successful, you should see the Tesseract version and other relevant information.

After installing Tesseract on your Raspberry Pi, you can use it in your Python projects by installing its Python wrapper called `pytesseract`. Install it by running the following command:

```
pip install pytesseract
```

With `pytesseract`, you can leverage the Tesseract OCR capabilities within your Python code to extract text from images or scanned documents.

3.5 Text to Speech Module

When it comes to text-to-speech (TTS) functionality, there are various modules and libraries available that can convert written text into spoken words.

These modules are valuable tools for applications that require auditory output, such as voice assistants, accessibility tools, and audio content generation.

Two popular modules for implementing TTS are pyttsx3 and gTTS.

Pyttsx3 [11]

Pyttsx3 is a feature-rich Python library that facilitates text-to-speech conversion. It provides developers with a comprehensive set of tools and functionalities for generating high-quality speech output from written text. One of the key advantages of pyttsx3 is its support for multiple speech synthesis engines. It offers compatibility with popular engines like the Microsoft Speech Platform and e Speak, allowing developers to choose the engine that best suits their needs in terms of voice quality, language support, and other preferences.

The library is designed to be cross-platform, meaning it can seamlessly run on various operating systems including Windows, macOS, Linux, and Raspberry Pi OS. This cross-platform compatibility ensures that developers can create TTS applications that can be deployed on different platforms without significant modifications. Whether you are developing on a Raspberry Pi device running Raspberry Pi OS or another platform, pyttsx3 can be leveraged to implement text-to-speech functionality.

Pyttsx3 provides a straightforward and user-friendly interface for controlling speech parameters. Developers can adjust settings such as volume, rate (speed of speech), and voice selection to tailor the generated speech according to specific requirements.

This level of control allows for the creation of more personalized and natural-sounding audio output.

Implementing TTS with pyttsx3 is relatively simple. Developers can initialize the pyttsx3 engine, set the desired properties, and then use the `speak ()` method to convert written text into speech. Furthermore, pyttsx3 supports event-driven programming, enabling the handling of events such as completion of speech or any errors that may occur during the synthesis process.

Overall, pyttsx3 stands as a reliable and flexible text-to-speech library for Python. Its extensive features, compatibility across platforms including Raspberry Pi OS, and customizable speech parameters make it a valuable asset for developers seeking to incorporate speech synthesis capabilities into their applications running on Raspberry Pi devices.

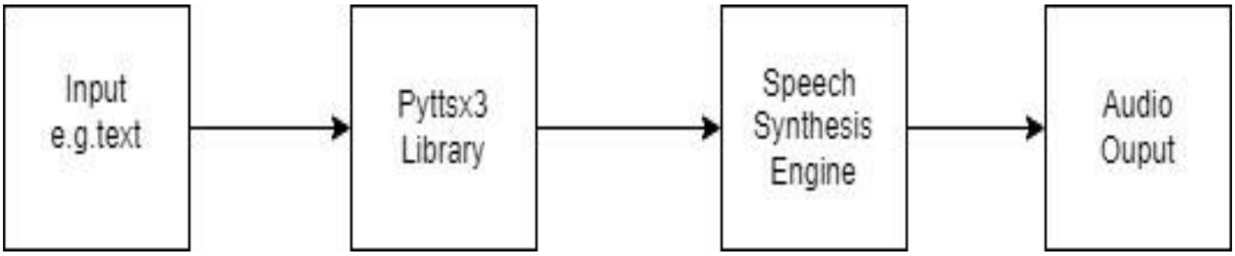


Figure 25-pyttsx3

<u>**Google Text-To-Speech**</u> [12]

Google Text-to-Speech (gTTS) is a Python library that utilizes the Google Text-to-Speech API to convert written text into spoken words. It provides developers with a straightforward and efficient way to generate high-quality speech output in multiple languages.

One of the key features of gTTS is its integration with the Google Text-to-Speech API. By leveraging this API, gTTS taps into Google's advanced speech synthesis technology, which produces natural-sounding and human-like speech.

Using gTTS is simple and convenient. Developers can provide the desired text as input to the library, and gTTS takes care of the conversion process by sending the text to the Google Text-to-Speech API. The API processes the text and returns an audio file containing the synthesized speech.

One notable advantage of gTTS is its support for multilingual text-to-speech conversion. The library offers a wide range of language options, enabling developers to generate speech output in different languages to cater to diverse user needs.

The gTTS library provides additional features to enhance the speech synthesis process. It allows developers to control aspects such as speech speed, audio volume, and language selection, enabling customization of the generated speech output.

The generated audio output can be saved as an audio file for playback or integrated into various applications that require speech output, such as voice assistants, audiobook creation, and accessibility tools.

Overall, gTTS serves as a powerful and accessible tool for incorporating text-to-speech capabilities into Python applications. By leveraging the Google Text-to-Speech API, gTTS empowers developers to create natural and dynamic speech output in multiple languages, enriching user experiences and enabling a wide range of applications.

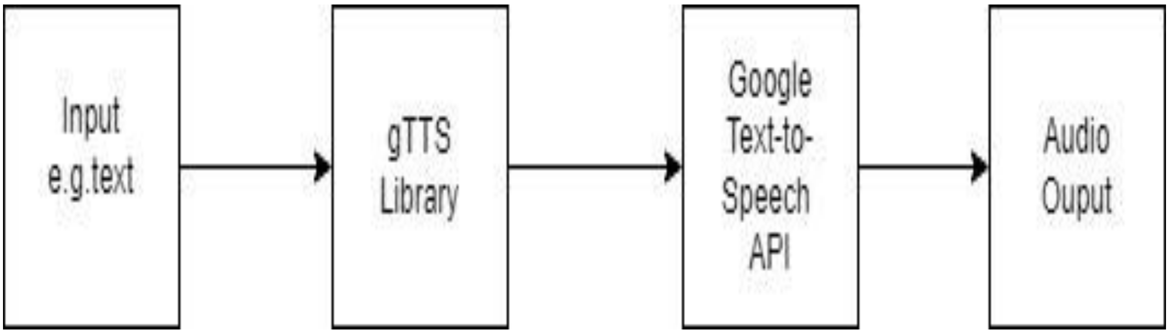


Figure 26-gTTS Library

gTTS vs pyttsx3

	gTTS	pyttsx3
Offline	No	Yes
Speech Quality	Google’s advanced speech synthesis	Highly customizable and natural
Language Support	Wide range of Languages Support	Multiple speech synthesis engines
Speed Control	No	Yes
Volume Control	No	Yes
Volume Selection	No	Yes

Table 2-gTTS vs pyttsx3

Offline Capability: pyttsx3 has the advantage of being able to work offline as it comes with built-in speech synthesis engines. This makes it suitable for applications that require speech synthesis without an internet connection, such as smart glasses that may not always have reliable internet access.

Online Requirement: gTTS relies on the Google Text-to-Speech API, which requires an internet connection. It sends the text input to the API and retrieves the synthesized speech output, making it dependent on online connectivity.

Speech Quality: pyttsx3 provides highly customizable and natural-sounding speech output. With support for multiple speech synthesis engines, developers have greater control over speech parameters, resulting in more personalized and engaging audio output. gTTS leverages Google's advanced speech synthesis technology, providing high-quality speech, but with less flexibility in customization.

Language Support: gTTS offers a wide range of language options, allowing developers to generate speech output in multiple languages. pyttsx3, with its support for multiple speech synthesis engines, provides language options based on the specific engine being used.

In the context of smart glasses, pyttsx3 is preferred due to its offline capability and the flexibility to customize speech parameters. This allows the smart glasses to provide real-time, customizable speech output without relying on an internet connection, making them more independent and user-friendly.

Overall, the choice between pyttsx3 and gTTS depends on specific requirements, such as offline capabilities, customization needs, and language support.

While gTTS may be suitable for online applications with a wide language range, pyttsx3's offline capability and customization options make it a preferred choice for smart glasses and similar devices.

3.6 Text Translation Module

Text translation implementation plays a crucial role in the development of smart glasses systems, enabling real-time language translation capabilities. By utilizing technologies such as the Google Translate API and integrating them into smart glasses, users can receive instant translations of text from one language to another directly through their glasses. This empowers users to overcome language barriers and facilitates seamless communication in various scenarios, including travel, business, and daily interactions.

Google Translate API

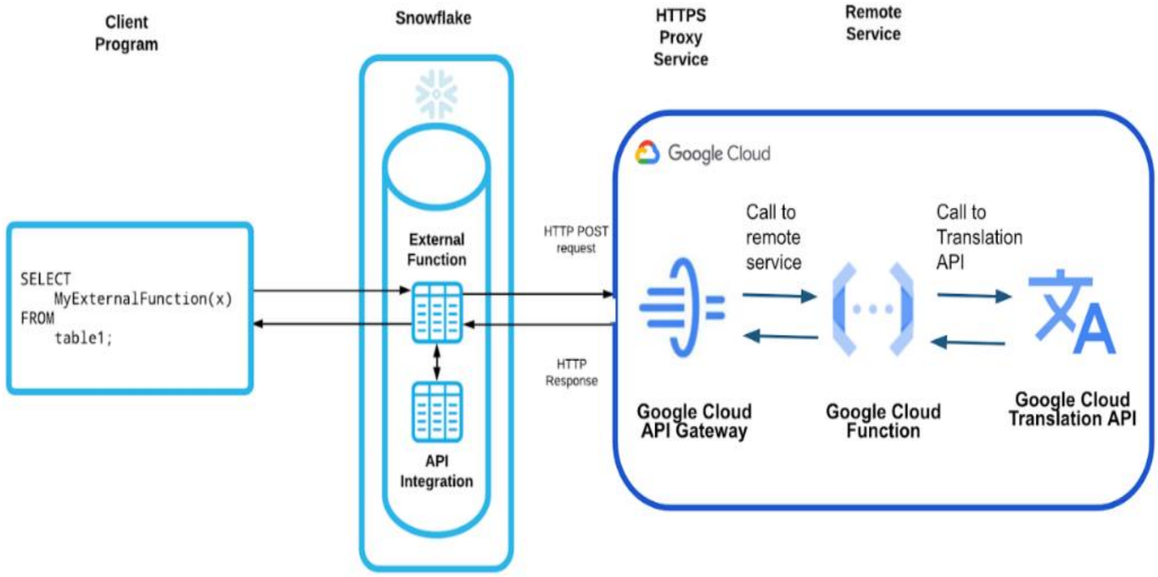
Google Translate API is a service provided by Google that allows developers to integrate language translation capabilities into their applications. It provides an interface for automatically translating text from one language to another. The Google Translate API Python library is a wrapper around the Google Translate API that simplifies the process of making API requests and handling the responses.

To use the Google Translate API, developers need to obtain an API key from the Google Cloud Platform. This API key acts as an authentication mechanism, allowing access to the translation service.

Once the API key is obtained and the library is installed, developers can use the library to make translation requests. They can specify the source language and target language, along with the text to be translated. The library then sends a request to the Google Translate API server, which processes the request and returns the translated text.

The Google Translate API uses advanced machine learning techniques to provide accurate and high-quality translations. It leverages a vast amount of multilingual data to train its models and improve translation accuracy over time. The API supports a wide range of languages, allowing developers to translate text between different language pairs.

The Python library provides a convenient interface for developers to interact with the Google Translate API. It abstracts away the complexity of making HTTP requests and handling the API response. Developers can focus on the translation logic within their applications without worrying about the underlying implementation details.



```
SELECT
    MyExternalFunction(x)
FROM
    table1;
```

Figure 27-Google Translate API

Overall, the Google Translate API and its Python library make it easy for developers to integrate language translation capabilities into their applications. It provides a reliable and efficient way to automate language translation tasks, enabling seamless communication and understanding across different languages.

3.7 Scene Description Module

Scene description and object detection are powerful computer vision techniques that use advanced models such as RCNN, YOLO, MobileNet SSD, and SSD to analyze images or video and provide concise information about the scene and identify specific objects. These models leverage deep learning algorithms to accurately detect and classify objects in real-time, enabling machines to understand visual content.

Scene description involves providing a verbal description of the scene, including the layout, object locations, and potential hazards. Object detection focuses on identifying and localizing objects within the scene, ranging from everyday items to complex entities like landmarks or animals.

By combining these techniques, machines can provide valuable insights, enhance accessibility, and assist visually impaired individuals. This technology has the potential to revolutionize various domains, from navigation systems to automation, by enabling machines to interpret visual information and interact with the world more effectively.

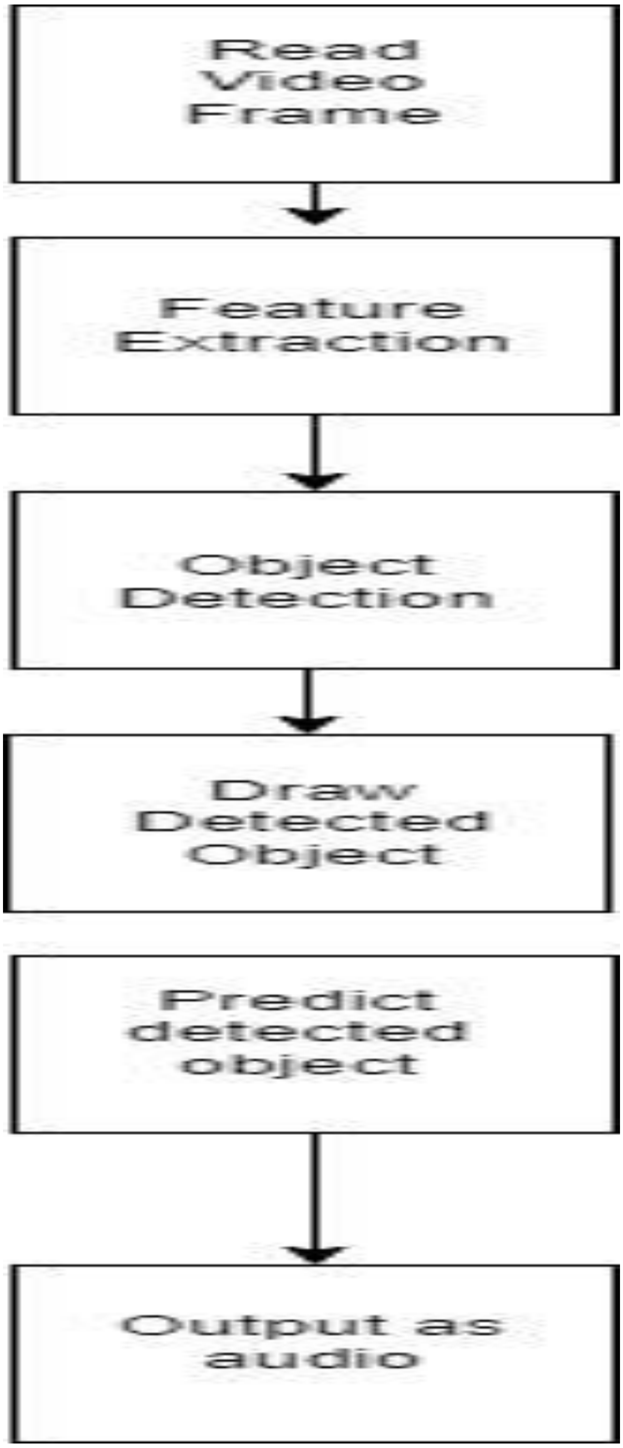


Figure 28-Scene Description Block Diagram

Yolo V7 [13]

YOLOv7 is a state-of-the-art object detection model that offers significant advancements in real-time accuracy and efficiency. Let's explore how it works step by step:

1. Network Architecture

YOLOv7 utilizes a carefully designed network architecture that combines convolutional layers, pooling layers, and other specialized layers to extract features from input images. This architecture is optimized for real-time object detection tasks.

2. Feature Integration

YOLOv7 employs an effective feature integration method that combines features from different layers of the network. This integration allows the model to capture both low-level and high-level visual cues, enabling accurate detection of objects at various scales and complexities.

3. Object Detection

Once the features are extracted and integrated, YOLOv7 applies bounding box regression and class prediction to identify and locate objects in the input image. The model predicts the bounding box coordinates and assigns class probabilities to each detected object.

4. Loss Function

YOLOv7 employs a robust loss function that measures the discrepancy between the predicted bounding boxes and the ground truth annotations. This loss function guides the model to refine its predictions and improve the accuracy of object detection during the training process.

5. Label Assignment and Training Efficiency

YOLOv7 improves label assignment and model training efficiency. It optimizes the process of assigning ground truth labels to anchor boxes, which are predefined bounding box priors. This optimization helps the model better understand the spatial relationship between objects and reduces false positives.

6. Inference Speed and Accuracy

YOLOv7 achieves faster inference speed and higher detection accuracy compared to other object detection models. By reducing the number of parameters and computation while maintaining strong performance, YOLOv7 enables real-time object detection on resource-constrained devices.

7. Cost-Effectiveness

One of the advantages of YOLOv7 is its ability to run on cheaper computing hardware compared to other deep learning models. This makes it more accessible and cost-effective for deployment in various applications.

Overall, YOLOv7 combines an optimized network architecture, effective feature integration, robust loss function, improved label assignment, and training efficiency to provide accurate and efficient real-time object detection. Its advancements in speed, accuracy, and cost-effectiveness make it a preferred choice for applications that require fast and reliable object detection capabilities.

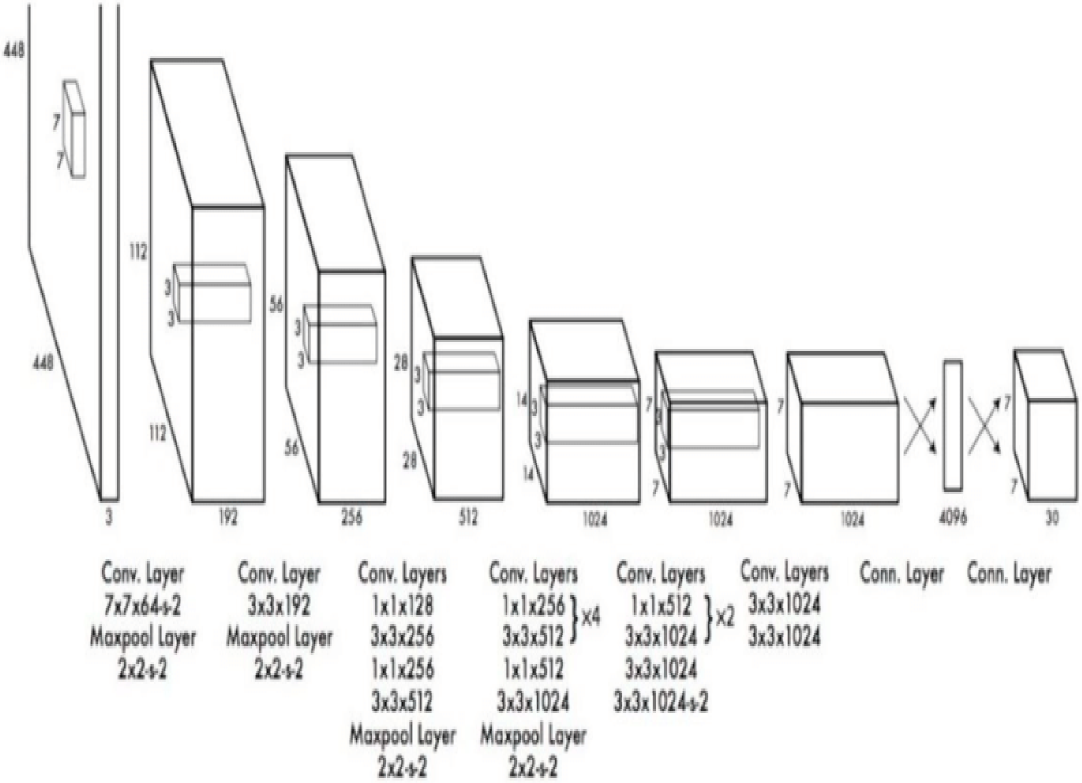


Figure 29-YoloV7

R-CNN

R-CNN (Region-based Convolutional Neural Network) is a widely used object detection model that operates in a two-step process: region proposal and classification. Let's explore how R-CNN works step by step:

1. Region Proposal

R-CNN begins by generating a set of region proposals in the input image. These proposals are candidate bounding boxes that potentially contain objects. Various methods can be used for region proposal, such as selective search, which generates a diverse set of possible regions.

2. CNN Feature Extraction

For each proposed region, R-CNN applies a convolutional neural network (CNN) to extract features. CNN processes the region and generates a fixed-length feature vector that captures the visual information within the region.

3. Object Classification

The extracted features are fed into a set of class-specific linear support vector machines (SVMs). Each SVM is trained to classify the presence or absence of a specific object class within the proposed region. The SVMs analyze the features and assign a class label to each region.

4. Bounding Box Refinement

In addition to classifying the presence of an object, R-CNN also refines the bounding box coordinates of the proposed regions. A regression model is trained to adjust the coordinates of the region's bounding box to more accurately align with the object within the region.

5. Non-Maximum Suppression:

Since the region proposal step may generate overlapping bounding boxes for the same object, R-CNN applies non-maximum suppression to remove redundant detections. This step selects the most confident detection for each object by suppressing overlapping bounding boxes based on their confidence scores.

6. Training Process:

R-CNN is trained in a supervised manner using a large labeled dataset. The CNN backbone is pre-trained on a large-scale image classification task, such as ImageNet, and then fine-tuned on the object detection dataset. The SVMs and regression models are trained using the extracted features and ground truth annotations.

By following these steps, R-CNN can accurately detect and classify objects in an input image. However, R-CNN is computationally expensive due to its two-stage approach and requires significant memory and computational resources.

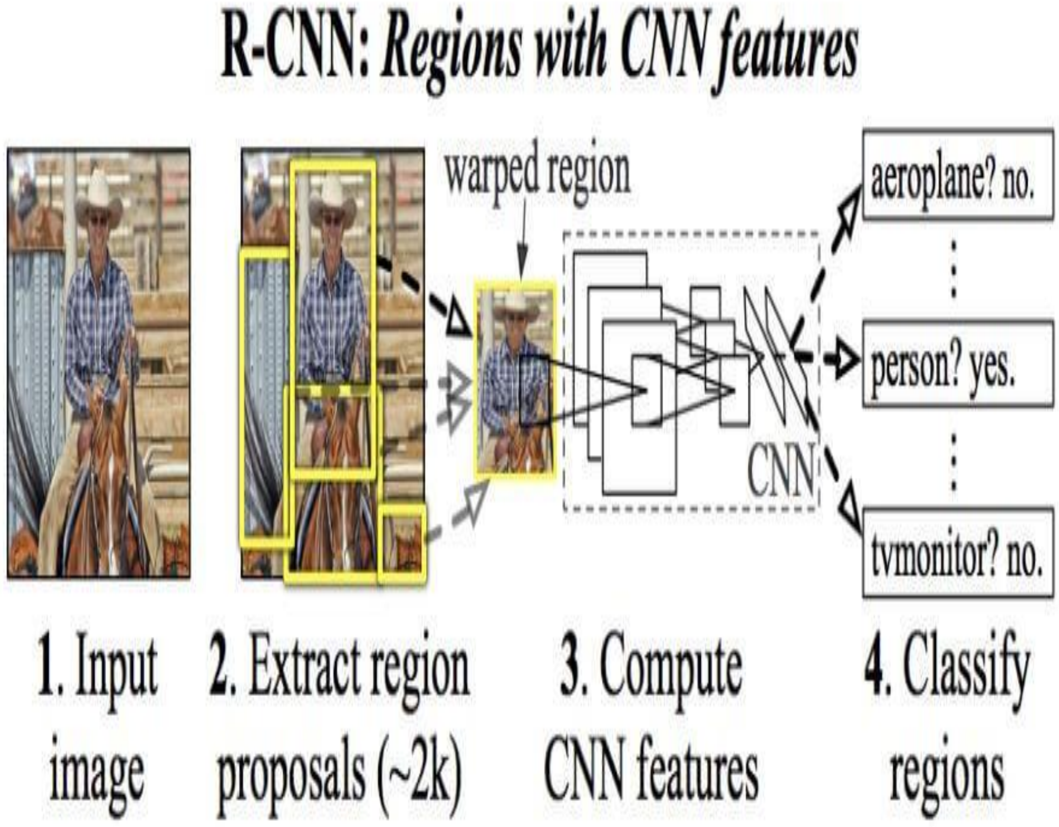


Figure 30-R-CNN

-

SSD

SSD (Single Shot MultiBox Detector) is a popular object detection model known for its real-time inference capabilities. Unlike R-CNN, which operates in a two-stage process, SSD performs object detection in a single shot. Let's explore how SSD works step by step:

1. Base Network:

SSD begins by utilizing a base network, typically a pre-trained convolutional neural network (CNN) such as VGG or ResNet. The base network processes the input image and extracts a set of feature maps with different spatial resolutions.

2. Multi-scale Feature Maps:

SSD applies a set of convolutional layers on top of the base network to generate feature maps at multiple scales. These feature maps capture information at different levels of detail, allowing the model to detect objects of various sizes.

3. Anchor Boxes:

SSD defines a set of anchor boxes or default boxes on each feature map cell. These anchor boxes act as reference bounding boxes of different aspect ratios and scales. The anchor boxes are used to predict the location and size of objects in the image.

4. Object Class Prediction:

SSD applies a series of convolutional layers on each feature map to predict the presence of objects and their corresponding class labels. For each anchor box, the model outputs the confidence scores for different object classes, indicating the likelihood of an object belonging to each class.

5. Bounding Box Regression:

In addition to class prediction, SSD also performs bounding box regression. For each anchor box, the model predicts the offsets or deltas to adjust the anchor box coordinates to better fit the shape of the detected object.

6. Non-Maximum Suppression:

Similar to other object detection models, SSD employs non-maximum suppression to filter out redundant detections. This process removes overlapping bounding boxes and selects the most confident detection for each object class.

7. Training Process:

SSD is trained in a supervised manner using annotated training data. The model is optimized using a combination of classification loss, which measures the accuracy of class predictions, and regression loss, which measures the accuracy of bounding box predictions.

By following these steps, SSD can efficiently detect objects in an input image in a single pass. It benefits from the use of multi-scale feature maps and anchor boxes, allowing it to handle objects of different sizes and aspect ratios. SSD achieves real-time inference speeds while maintaining high detection accuracy, making it suitable for applications that require fast and accurate object detection.

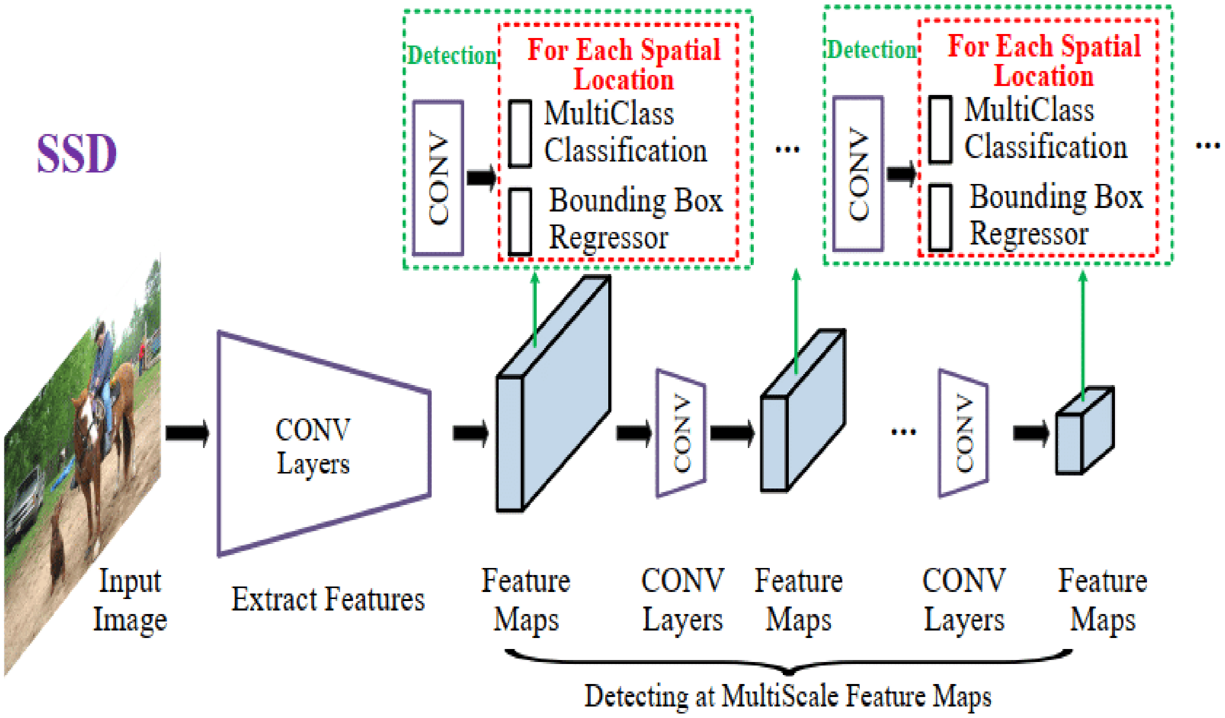


Figure 31-SSD

MobileNet SSD [14]

MobileNet SSD is an artificial intelligence model that combines the MobileNet architecture with the Single Shot MultiBox Detector (SSD) framework. It is designed for real-time object detection, allowing computers and devices to identify and locate objects within images or video streams.

The MobileNet architecture, which forms the base of MobileNet SSD, is specifically designed for resource-constrained devices such as mobile phones or embedded systems. It utilizes depth wise separable convolutions, a technique that reduces computational complexity while maintaining accuracy. This makes MobileNet SSD suitable for running on devices with limited processing power, such as the Raspberry Pi 4 Model B.

In MobileNet SSD, the architecture is divided into two main components: the base network and the detection network. **The base network**, based on the MobileNet architecture, is responsible for extracting features from the input image. These features capture important visual patterns and information.

The detection network, built on top of the base network, performs object detection. It consists of several convolutional layers and prediction layers. The **convolutional layers** analyze the extracted features from the base network and generate a feature map that represents different aspects of the image. The **prediction layers** utilize these feature maps to predict the presence of objects and their bounding box coordinates.

MobileNet SSD incorporates multiple detection layers at different scales to handle objects of various sizes. These layers are responsible for detecting objects at different resolutions, allowing the model to capture objects both near and far from the camera.

During inference, MobileNet SSD processes an input image through the base network, followed by the detection network. The detection network analyzes the extracted features and generates a set of bounding box predictions and class probabilities. These predictions indicate the location and class of objects present in the image.

By combining the efficiency of the MobileNet architecture with the object detection capabilities of SSD, MobileNet SSD achieves real-time object detection while maintaining a small model size and low computational requirements. This makes it an ideal choice for running on devices like the Raspberry Pi 4 Model B, enabling real-time object detection applications with limited resources.

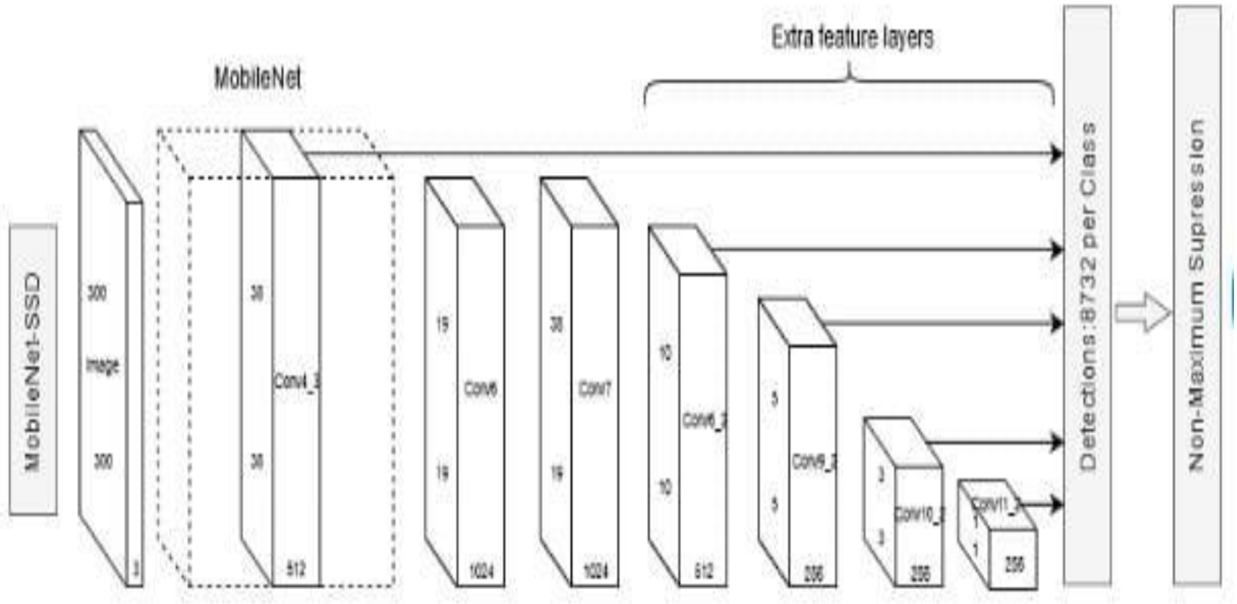


Figure 32-5 -MobileNET SSD Architecture

In the MobileNet SSD architecture, there are several key layers that play a crucial role in the model's object detection capabilities. Here is an explanation of some of these layers:

1. Convolutional Layers: The convolutional layers are the building blocks of the neural network. They perform convolution operations on the input image or feature maps to extract visual patterns and features. These layers consist of filters that scan across the input data, capturing relevant information at different spatial locations.

2. Depthwise Separable Convolution: MobileNet SSD utilizes depthwise separable convolutions, which are a more efficient variant of standard convolutions. Depthwise separable convolutions split the convolution operation into two separate layers: depthwise convolution and pointwise convolution. The depthwise convolution applies a separate filter to each input channel, while the pointwise convolution performs a 1x1 convolution to combine the outputs of the depthwise convolution. This approach reduces computational complexity while preserving accuracy.

3. Feature Maps: Feature maps are intermediate representations of the input image obtained after passing through the convolutional layers. These maps capture different levels of visual information, highlighting important features such as edges, textures, and shapes. Feature maps encode higher-level representations of the image, which are then used for object detection.

4. Prediction Layers: The prediction layers in MobileNet SSD are responsible for generating the final object detection results. These layers take the feature maps from different scales and apply additional convolutional operations to predict the presence of objects and their corresponding class labels and bounding box coordinates. Each prediction layer is associated with a specific scale of the input image, allowing the model to detect objects at different resolutions.

5. Non-Maximum Suppression (NMS): After the prediction layers produce a set of bounding box predictions, a non-maximum suppression algorithm is applied. This algorithm removes redundant or overlapping bounding boxes, selecting only the most confident and accurate detections. NMS helps in eliminating duplicate detections and improving the final object detection results.

The combination of these layers in MobileNet SSD enables the model to effectively detect objects in real-time. The convolutional layers extract meaningful features, the prediction layers generate bounding box predictions, and the NMS algorithm refines the results to provide accurate and reliable object detection. This architecture is designed to balance efficiency and accuracy, making it well-suited for deployment on resource-constrained devices like the Raspberry Pi 4 Model B.

Yolo vs R-CNN vs SSD vs MobileNet SSD

Model	R-CNN	SSD	Yolo	MobileNet SSD
Architecture	Two-Stage	Single-Shot	Single-Shot	Single-Shot
Inference Speed	Slower	Faster	Faster	Faster
Accuracy	High	Moderate	Moderate	Moderate
Computation	High	Moderate	Hight	Moderate
Embedded	Not Suitable	Suitable	Suitable	Suitable

Table 3-Yolo vs R-CNN vs SSD vs MobileNet SSD

Coco Dataset

The COCO (Common Objects in Context) dataset is a large-scale image recognition dataset used for object detection, segmentation, and captioning tasks. It consists of over 330,000 images, each annotated with 80 object categories and 5 captions describing the scene. It is widely utilized in computer vision research and has been instrumental in training and evaluating state-of-the-art object detection and segmentation models.

The COCO dataset offers different types of annotations, including object detection with bounding box coordinates and full segmentation masks for 80 object categories. Additionally, it provides stuff image segmentation with pixel maps depicting 91 amorphous background areas. The dataset includes a wide range of classes, such as people, bicycles, cars, trees, and roads.

The COCO dataset serves as a foundational resource for various computer vision tasks. For object detection, it provides bounding box coordinates for different objects, enabling the training of models to detect and classify objects in images. Key point detection, another application, involves identifying specific points of interest in an image, such as object corners or human body joints. The COCO dataset includes key point annotations for over 250,000 people, facilitating tasks like object tracking and motion analysis. Additionally, the dataset supports stuff image segmentation, which helps in understanding the scene, identifying objects, and their context based on semantic classes.

Researchers and practitioners can leverage the COCO dataset to train deep learning models like Multi-Person Pose Estimation (MPPE) or OpenPose, which take images as input and generate key points as output. Furthermore, the dataset's stuff image segmentation aids in scene understanding and provides insights into the presence and location of objects based on the context, materials, and geometric properties of the scene.

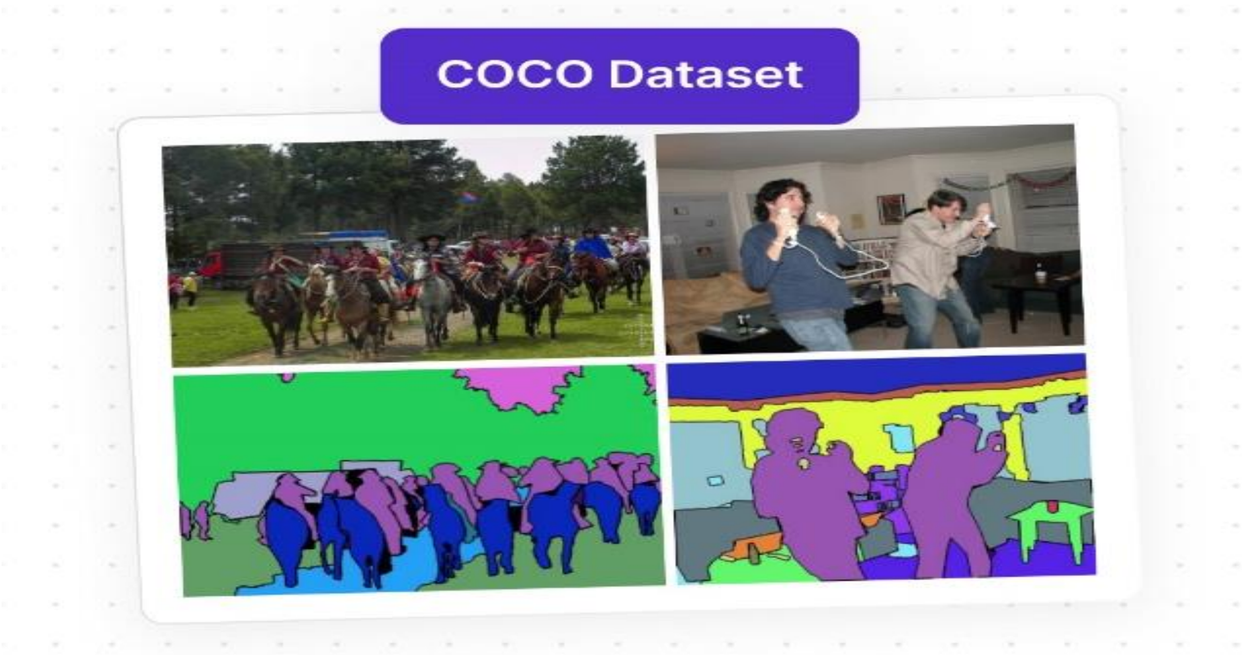


Figure 33-Coco Dataset

3.8 QR Code Module

QR code scanning in smart glasses revolutionizes hands-free information retrieval by seamlessly integrating camera capture, real-time processing, and augmented reality capabilities.

With a built-in camera and wearable display, users can simply glance at QR codes in their field of view to instantly extract and display encoded information.

This technology finds applications in logistics, maintenance, healthcare, and more, enabling quick access to shipment details, equipment manuals, troubleshooting guides, and other relevant content.

By eliminating the need for handheld devices, QR code scanning in smart glasses enhances efficiency, accuracy, and user experience.

This seamless integration of QR code scanning and smart glasses paves the way for a new era of immersive, hands-free information interaction.

ZBAR Library

ZBar is an open-source software suite that provides tools and libraries for barcode scanning and decoding. It is widely used for recognizing and extracting barcode information from images or video frames. The library supports various barcode formats such as EAN-13, UPC-A, Code 128, QR Code, and more.

ZBar is written in C but offers interfaces for multiple programming languages, including Python. The Python interface is provided through a wrapper library called pyzbar. Pyzbar simplifies the process of integrating barcode and QR code scanning functionality into Python applications by providing a high-level API for decoding barcodes and QR codes.

By utilizing ZBar and pyzbar, developers can easily incorporate barcode scanning capabilities into their Python projects. This can be particularly useful for applications that involve inventory management, product tracking, document processing, or any scenario where barcode data needs to be extracted and processed.

With the ZBar library and the convenience of pyzbar, developers have access to a powerful and flexible solution for barcode and QR code scanning in Python, enabling efficient and accurate data retrieval from a wide range of barcode formats.

How ZBAR Library Works?

The ZBar library, along with its Python wrapper pyzbar, provides functionality to scan and decode QR codes. Here's an overview of how the library performs QR code scanning:

1. Image Input: To scan a QR code, you need to provide an image or a frame containing the QR code. This image can be a file on disk or obtained from a video stream.

2. Image Processing: The library uses image processing techniques to analyze and extract the QR code from the provided image. This involves steps such as converting the image to grayscale, enhancing contrast, and applying filters to improve QR code detection.

3. QR Code Detection: ZBar implements algorithms to detect and locate QR codes within the processed image. It looks for the distinctive square patterns and alignment patterns that constitute a QR code.

4. Decoding QR Code: Once the QR code is successfully detected, the library proceeds to decode the encoded data within the QR code. It analyzes the positioning patterns, timing patterns, and other components to extract the information embedded in the QR code.

5. Data Retrieval: The decoded data, such as text, URLs, or other types of information, is then made available for further processing in your application. You can access this data programmatically to perform actions based on the scanned QR code's content.

By utilizing the ZBar library and pyzbar, developers can easily incorporate QR code scanning capabilities into their Python applications.

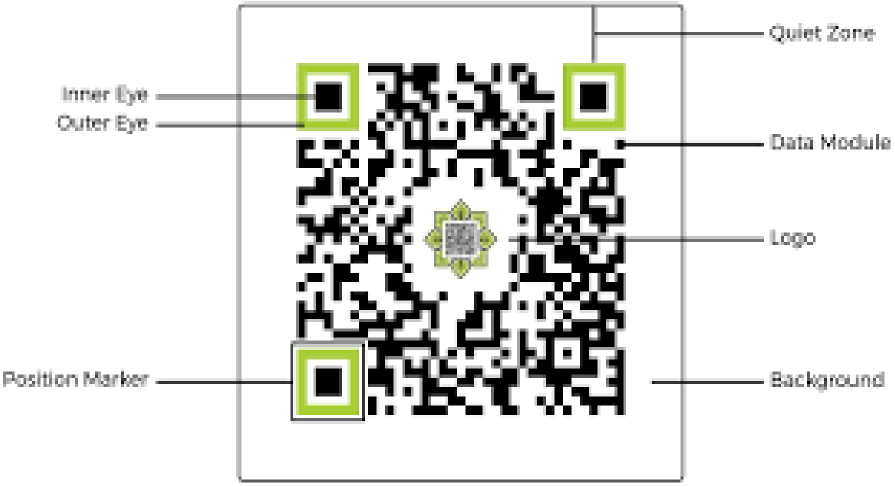


Figure 34-QR Code

Color Identification Module

Color modules are essential tools in computer vision and image processing applications that involve color identification and analysis. These modules provide functionalities to work with different color models, such as RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value), enabling accurate color representation and manipulation.

The **RGB color model** is widely used in digital imaging systems and displays. It represents colors as combinations of red, green, and blue intensities. In this model, each pixel's color is specified by three values, ranging from 0 to 255, indicating the intensity of each primary color component. By combining different intensities of red, green, and blue, a wide range of colors can be produced.

On the other hand, **the HSV color model** focuses on the perceptual properties of color. It describes colors in terms of hue, saturation, and value. Hue represents the dominant color tone, saturation indicates the intensity or purity of the color, and value represents the brightness or lightness of the color. The HSV model provides a more intuitive representation of colors, making it useful in tasks such as color identification, image segmentation, and color-based object detection.

In addition to RGB and HSV, there are other color models used for specific purposes. For example, the **CMYK** (Cyan, Magenta, Yellow, Key/Black) color model is commonly used in printing and graphic design. It represents colors using a combination of subtractive primary colors (cyan, magenta, yellow) and a key color (black) to achieve accurate color reproduction on printed materials.

Color modules in programming languages or libraries provide functionalities to convert between different color models, perform color space transformations, extract color information from images or videos, and identify specific colors based on defined criteria. These modules enable developers to implement color identification algorithms, color-based image processing, and computer vision applications that rely on color analysis and recognition.

By leveraging color modules and the diverse color models they support, developers can accurately represent, manipulate, and identify colors in various computer vision and image processing tasks, leading to enhanced visual perception and analysis capabilities.

Red, Green, Blue (RGB) Model

The RGB color model is indeed defined by specifying the amount of red, green, and blue that is present in a color. Each primary color component is assigned a value ranging from 0 to 255, indicating the intensity or brightness of that color component. By combining different intensities of red, green, and blue, a wide range of colors can be achieved.

In the RGB model, the combination of full intensity for all three-color components (red, green, and blue) results in the color white, while the absence of all three components yields the color black. This additive method of color mixing means that as more of each primary color is added, the resulting color becomes brighter.

When observing a specific color, it can be challenging to precisely determine the exact composition of red, green, and blue that forms it. The human eye perceives colors as a combination of various wavelengths of light, and our perception of color can be influenced by factors such as lighting conditions and individual differences in color perception.

Additionally, different devices or systems may have variations in how they represent colors within the RGB color model. Factors like color profiles, display settings, and color calibration can affect the accuracy and consistency of color reproduction.

To address these challenges, color analysis and colorimetry techniques are employed to measure and quantify color properties objectively. Color spaces, such as CIE XYZ or CIE LAB, provide standardized representations of colors based on human perception and can be used for more accurate color analysis and comparison.

In practical applications, color identification and analysis often involve comparing color values or ranges within a specific tolerance. By defining color thresholds or using color distance metrics, it is possible to determine whether a given color falls within a certain range or matches a target color, even with slight variations in the RGB values.

While the RGB color model provides a versatile and widely used representation of colors in digital systems, accurately defining the exact composition of a color based solely on RGB values can be challenging due to the subjective nature of color perception and the limitations of human vision. Therefore, color analysis often involves additional techniques and considerations to ensure accurate color identification and analysis in various applications.

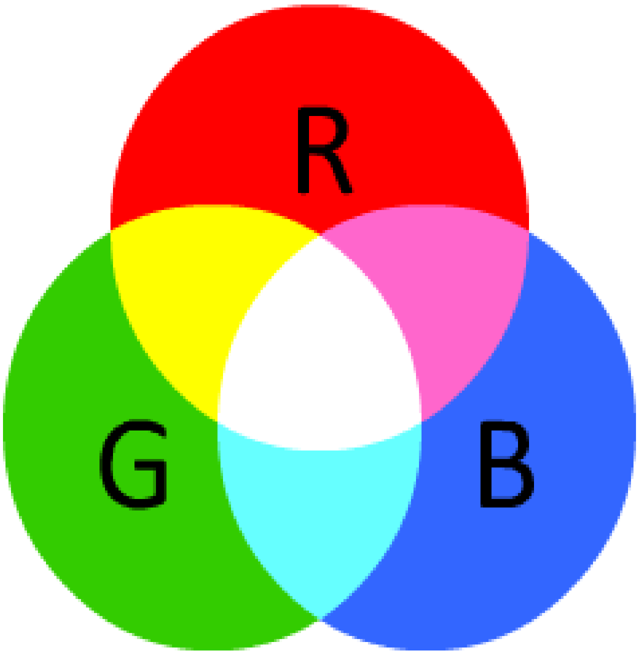


Figure 35-RGB Model

Hue, Saturation, Value (HSV) Model [15]

HSV (Hue, Saturation, Value) color model is used to represent colors in a way that is more intuitive and perceptually meaningful. The HSV color model defines the following ranges for each component:

1. Hue (H): The hue component **represents the color itself** and is measured in degrees on a color wheel. It ranges from 0 to 360, where 0 corresponds to red, 120 to green, and 240 to blue. The hues in between represent intermediate colors, allowing for a continuous representation of the color spectrum. For example, yellow is represented at approximately 60 degrees, while purple is represented at around 300 degrees.

2. Saturation (S): The saturation component describes the **intensity or purity of a color.** It represents how much gray is mixed with the hue. Saturation is measured as a percentage, ranging from 0% (completely desaturated or grayscale) to 100% (fully saturated or pure hue). Higher saturation values indicate vibrant and vivid colors, while lower values indicate more subdued or pastel colors.

3. Value (V): The value component **represents the brightness or lightness of a color.** It determines how much light is reflected from the surface. The value ranges from 0% (black or no light) to 100% (maximum brightness or full lightness). Increasing the value results in brighter colors, while decreasing it leads to darker shades.

By utilizing the HSV color model, color identification algorithms can analyze the hue, saturation, and value values of pixels in an image to identify and categorize different colors. This allows for the recognition of specific colors or color ranges based on their unique hue values. For example, a color identification system can be programmed to detect colors within a certain range of hues, such as identifying red colors with hues ranging from 0 to 30 degrees or blue colors with hues ranging from 210 to 240 degrees.

The range of colors in the hue component is important for distinguishing and categorizing different color families. By defining specific hue ranges, color identification algorithms can accurately recognize and differentiate between various colors in an image or scene. This information can be utilized in applications such as color matching, object recognition, or even assisting visually impaired individuals in distinguishing between different colors.

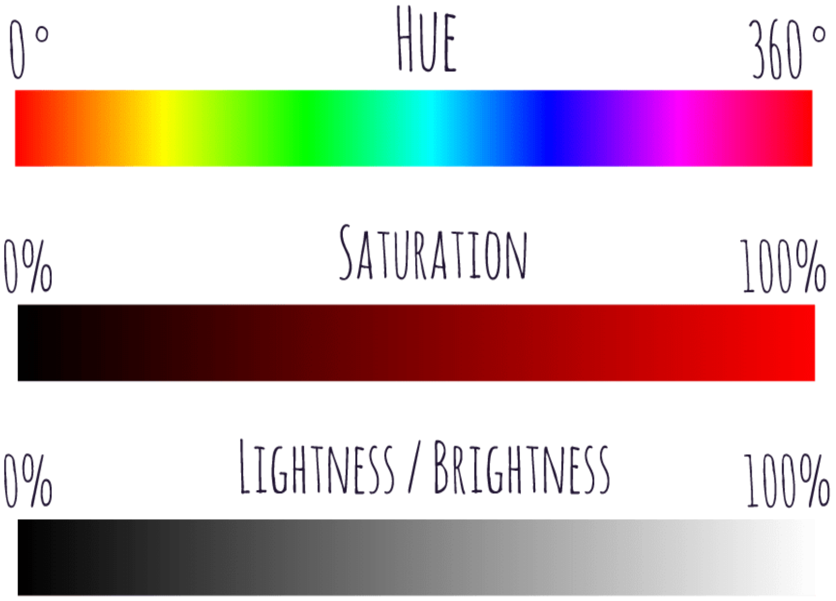


Figure 36-HSV Model

HSV VS RGB

	HSV	**RGB**
Representation	Separates Color info into hue, saturation, value	Combines red, green and blue channels.
Color Components	Hue (0° to 360°), Saturation (0% to 100%), Value (0% to 100%)	Red (0 to 255), Green (0 to 255), Blue (0 to 255)
Human Perception	Intuitive representation, relates to specific colors	Not directly intuitive, represents color combinations.
Lighting Robustness	Less sensitive to lighting variations	More sensitive to lighting variations.

Table 4-HSV vs RGB [16]

HSV is preferred over RGB in color identification on Raspberry Pi due to its intuitive representation, robustness to lighting variations, and simplified color range specification, which aligns well with human perception and natural color relationships.

Block Diagram

The first step is to open the camera and initialize it to capture frames. This can be done using libraries such as OpenCV in Python.

Once the camera is open, each frame is captured and processed. The captured frame is then converted from the RGB color space to the HSV color space, which is more suitable for color detection.

Next, color detection algorithms are applied to the HSV image to identify specific colors of interest. These algorithms use thresholding techniques or color range definitions to isolate the desired colors.

The detected colors are then displayed or visualized on the screen by overlaying them on the original captured frame. This helps the user visually identify and differentiate the detected colors.

This process of opening the camera, capturing frames, converting to HSV, detecting colors, and displaying the results is repeated continuously to provide real-time color detection feedback.

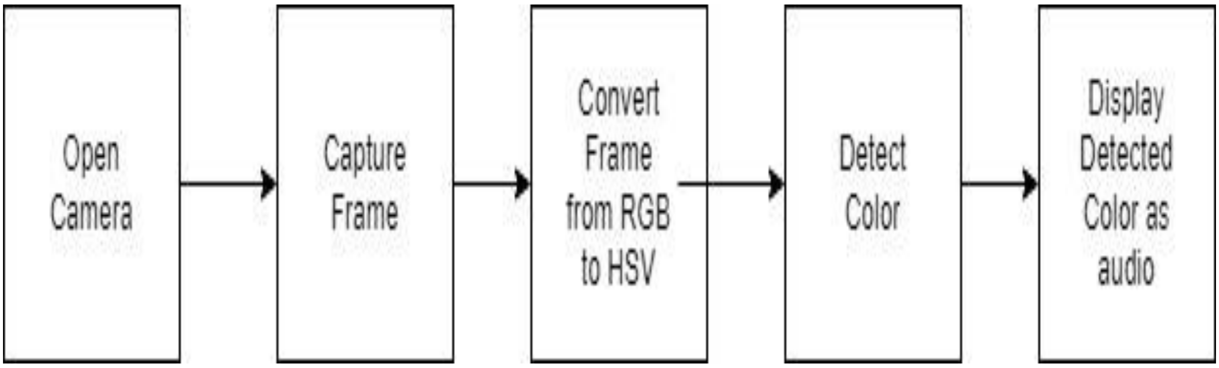

Open Camera
Capture Frame
Convert Frame from RGB to HSV
Detect Color
Display Detected Color as audio

Figure 37-Color Identification Block Diagram

3.9 Age/Gender Reveal Module

Age and gender reveal technology integrated into smart glasses offers a novel way to gather demographic information in real-time. By leveraging the power of

Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs), coupled with the Caffe framework, smart glasses can provide immediate and accurate age and gender estimation of individuals within the wearer's field of view.

These deep learning models are trained on extensive datasets to recognize and classify facial features related to age and gender.

The Caffe framework, a popular deep learning framework, provides the necessary tools and libraries to deploy CNN and DNN models efficiently on smart glasses. Its optimized implementation allows for real-time inference on resource-constrained devices, making it suitable for on-device age and gender estimation in smart glasses.

The process typically involves the following steps: the captured video frames are fed into the CNN or DNN models, which analyze facial features and extract relevant information such as age and gender. The models use their learned knowledge to make predictions based on the input data, providing near-instantaneous results.

By integrating age and gender reveal capabilities into smart glasses, users can gain valuable demographic insights without the need for additional devices or manual data entry. This technology finds applications in various fields, including retail, marketing, security, and customer service, enabling personalized experiences, targeted advertising, and improved situational awareness.

The combination of smart glasses, CNNs, DNNs, and the Caffe framework empowers users with real-time age and gender estimation capabilities, enhancing their interactions and decision-making processes. However, it is important to ensure that privacy and ethical considerations are considered when implementing such technologies to respect individuals' rights and maintain data privacy.

Convolution Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a class of deep learning models primarily designed for processing and analyzing visual data.

Let's dive into the details of CNN architecture and layers:

1. Input Layer: The input layer receives the input data, typically an image, and passes it forward for further processing.

2. Convolutional Layers: Convolutional layers are the core building blocks of CNNs. They consist of multiple filters or kernels that convolve over the input image. Each filter performs element-wise multiplication and summation operations on a small region of the input, known as the receptive field. The filters learn to extract different features from the input, such as edges, textures, and patterns, by sliding across the image and applying convolution operations. The output of each filter is referred to as a feature map or a convolutional feature.

3. Activation Function: After the convolution operation, an activation function (e.g., ReLU - Rectified Linear Unit) is applied element-wise to introduce non-linearity. The activation function helps in learning complex relationships between the extracted features.

4. Pooling Layers: Pooling layers reduce the spatial dimensions of the input, decreasing the computational complexity while retaining important features. Common pooling techniques include max pooling, which selects the maximum value within a region, and average pooling, which calculates the average value within a region.

5. Fully Connected Layers: Fully connected layers, also known as dense layers, are responsible for making predictions based on the learned features. The outputs from the previous layers are flattened into a 1D vector and connected to a series of fully connected layers. These layers perform operations involving weights and biases, gradually transforming the input into the desired output shape.

6. Output Layer: The output layer produces the final predictions based on the task at hand, such as image classification or object detection. The number of nodes in the output layer corresponds to the number of classes or categories in the problem domain.

CNNs are typically trained using large datasets through a process called backpropagation, where the weights of the network are adjusted to minimize the difference between predicted outputs and ground truth labels. This training process allows the network to learn meaningful representations of the input data. The architecture and specific configuration of CNNs can vary depending on the task and network complexity. More advanced architectures, such as VGGNet, ResNet, or InceptionNet, introduce additional architectural components like skip connections, residual blocks, and parallel branches to improve performance and address specific challenges.

Overall, CNNs excel at capturing hierarchical representations of visual data, automatically learning and extracting meaningful features from images, making them highly effective for a wide range of computer vision tasks.

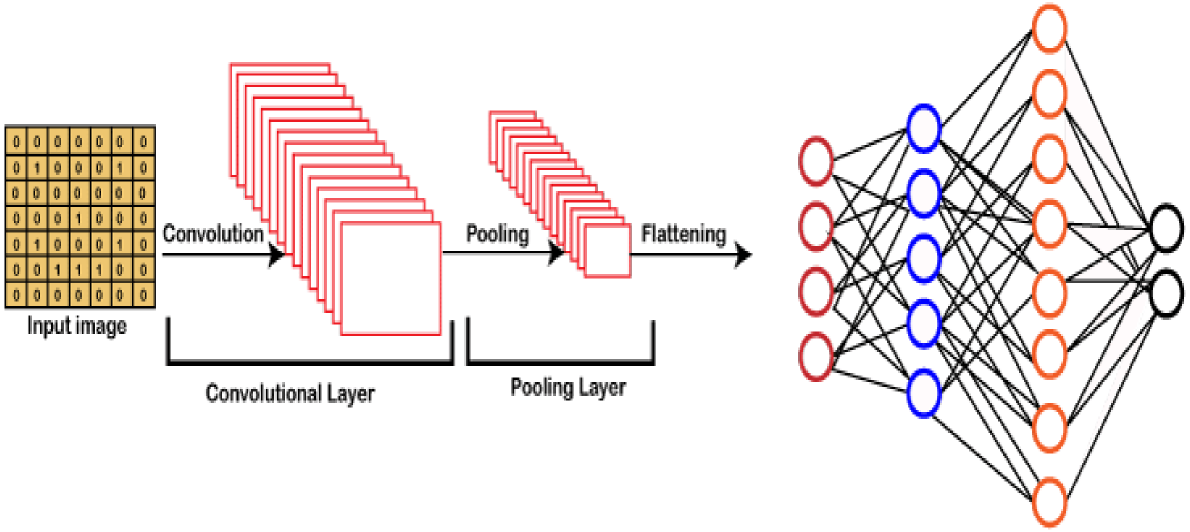


Figure 38-CNN

Deep Neural Network (DNN)

DNN stands for Deep Neural Network, which is a type of artificial neural network (ANN) with multiple hidden layers between the input and output layers. DNNs are designed to learn and represent complex patterns and relationships in data by employing a hierarchical structure of interconnected nodes or neurons.

Unlike traditional neural networks with only one or two hidden layers, DNNs are characterized by their depth, referring to the presence of multiple hidden layers. The depth of the network allows for the extraction of increasingly abstract and high-level features as the data passes through the layers, leading to more expressive and powerful representations.

DNNs consist of several key components:

1. Input Layer: Receives the initial input data, such as images, text, or numerical features, and passes it forward for further processing.

2. Hidden Layers: DNNs contain multiple hidden layers between the input and output layers. Each hidden layer consists of a set of interconnected neurons. The hidden layers progressively transform the input data, learning and extracting increasingly complex representations and features.

3. Neurons: Are computational units within the hidden layers of a DNN. Each neuron receives inputs from the previous layer, applies a linear transformation (weighted sum), and passes the result through an activation function. The activation function introduces non-linearity, allowing the DNN to model complex relationships and capture non-linear patterns in the data.

4. **Weights and Biases:** Like other neural networks, DNNs employ weights and biases associated with each connection between neurons. The weights determine the strength of the connections and influence the output of each neuron during the computation. Biases are additional parameters added to each neuron, providing an offset or threshold for the activation function.

5. Output Layer: The output layer of a DNN produces the final predictions or outputs based on the learned representations from the hidden layers. The number of neurons in the output layer depends on the specific task, such as classification, regression, or sequence generation.

Training a DNN involves optimizing the weights and biases using algorithms such as backpropagation and stochastic gradient descent. Through iterative training on labeled data, DNNs learn to make accurate predictions and generalize to unseen examples.

DNNs have achieved remarkable success in various domains, including computer vision, natural language processing, speech recognition, and more. Their ability to learn hierarchical representations from complex data has made them instrumental in advancing the field of deep learning and pushing the boundaries of artificial intelligence.

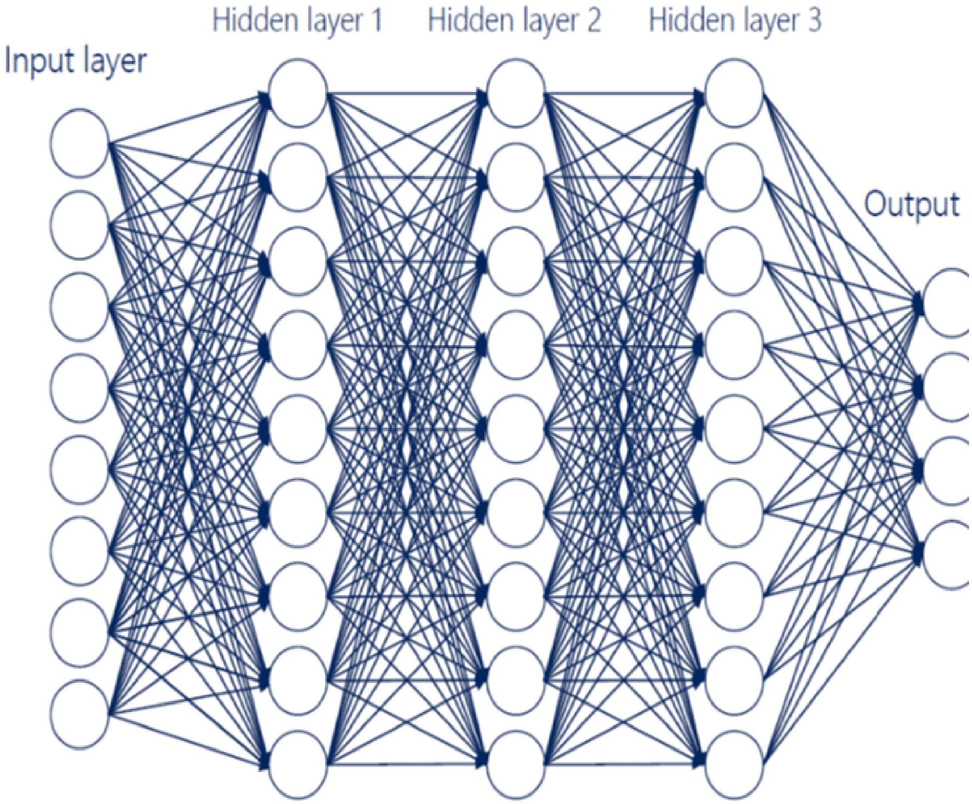


Figure 39-DNN

Convolutional Architecture for Fast Feature Embedding

Convolutional Architecture for Fast Feature Embedding (CAFFE) is a popular deep learning framework that specializes in the deployment and training of convolutional neural networks (CNNs).

It provides a flexible and efficient platform for various computer vision tasks, including image classification, object detection, and segmentation.

CAFFE excels in the domain of CNNs due to its architecture and optimization techniques. CNNs in CAFFE typically consist of convolutional layers, pooling layers, activation functions, and fully connected layers. Convolutional layers apply learnable filters to extract local patterns and spatial hierarchies from input data. Pooling layers reduce spatial dimensions while preserving essential features. Activation functions introduce non-linearity to capture complex relationships.

Fully connected layers in CAFFE make high-level decisions based on the learned features, enabling classification or regression tasks. The framework also provides support for deploying other types of deep neural networks (DNNs) that incorporate fully connected layers, although its primary focus remains on CNNs. CAFFE offers several key features and optimizations that contribute to its effectiveness and efficiency. These include GPU acceleration, which leverages compatible hardware to speed up computations; memory optimization, which minimizes memory overhead by reusing intermediate results; and model serialization, allowing trained models to be stored, shared, and deployed easily.

Additionally, CAFFE provides a collection of pretrained models, including popular architectures like AlexNet and VGGNet. These pretrained models can serve as starting points for transfer learning or fine-tuning on specific tasks, saving time and computational resources.

Overall, CAFFE's architecture, optimizations, and pretrained models make it a powerful framework for deploying and training CNNs, particularly for computer vision applications. Its focus on CNNs, along with its efficient implementation and rich set of features, has made it widely used in academia and industry for developing cutting-edge computer vision models.

Models Used

1. age_deploy.prototxt: This file is the configuration file for the age discrimination network. It defines the architecture and parameters of the network. The prototxt file specifies the layers, their types (e.g., convolutional, pooling), sizes, and connectivity, enabling the construction of the network. It contains information about the input and output layers, activation functions, and other configuration details required to define the network's structure.

2. age_net.caffemodel: This file contains the trained model parameters for the age discrimination network. During the training process, the network learns to assign appropriate weights to the connections between neurons to make accurate age predictions. The caffemodel file stores these learned parameters, allowing the network to make age predictions on new input data without going through the training process again.

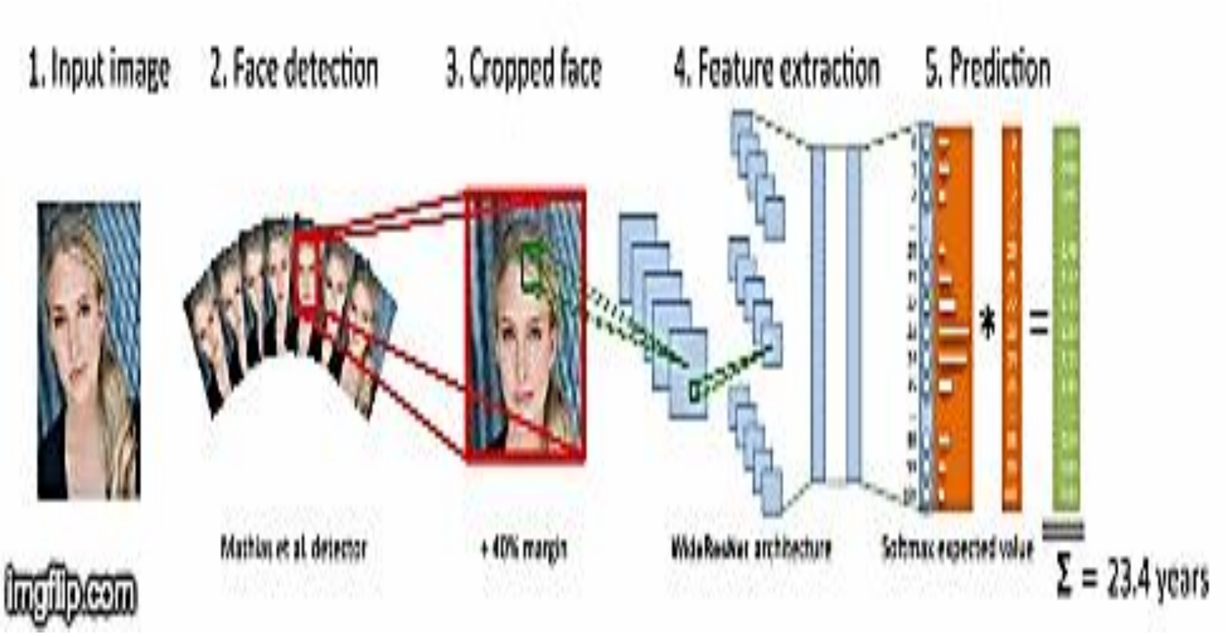
3. gender_deploy.prototxt: Is the model file for the gender discrimination network.Similar to age_deploy.prototxt, it defines the architecture and configuration of the gender discrimination network.It specifies the layers, connectivity, and other necessary information to construct the network for predicting the gender of input images.

4. opencv_face_detector.pbtxt: This file is the profile or configuration file for the face recognition network provided by OpenCV.It describes the structure and settings of the face detection model.The pbtxt file contains information about the layers, their parameters, and the necessary preprocessing steps required for accurate face detection.

5. opencv_face_detector_uint8.pb: The opencv_face_detector_uint8.pb file is a model file for the face recognition network. It contains the trained weights and biases for the face detection model. The model uses a deep learning architecture, such as a CNN, to identify and locate faces in images or videos.

These files collectively provide the necessary components for age and gender discrimination and face detection tasks. The configuration files define the network architecture and parameters, while the model files store the trained weights for making accurate predictions or detections.

Block Diagram [17]



1. Input in Real Time: The system receives a continuous stream of real-time input, such as video frames from a camera. Each frame is processed sequentially to perform age and gender prediction.

2. Face Detection: The first step is to detect and locate faces within the input frames. Face detection algorithms or models are applied to identify and localize faces accurately. This step ensures that only the regions of interest (faces) are further processed for age and gender prediction.

3. Cropped Face: Once a face is detected, it is cropped or extracted from the original frame. The cropped face region becomes the input for subsequent processing steps. By isolating the face, the system focuses specifically on the relevant features for age and gender prediction.

4. Feature Extraction: The cropped face is passed through a feature extraction module. This module analyzes the facial features and extracts relevant information. Common approaches involve using pre-trained deep learning models or handcrafted feature descriptors to capture discriminative facial characteristics.

5. Prediction: The extracted facial features are used as input to a prediction model. For age prediction, the model could be a trained deep neural network or regression model that estimates the person's age based on the input features. Similarly, for gender prediction, a trained model classifies the input features into male or female categories.

6. Age and Gender Reveal: The final step is to present the age and gender predictions based on the processed input frame. The system can display the predicted age and gender labels on the output video or convey the information through audio or visual cues. The entire process described above is performed in a continuous loop, allowing the system to provide real-time age and gender predictions for each input frame.

3.10 Prayer-Times Module

Prayer-times implementation is a crucial component of applications or services that cater to the needs of individuals seeking accurate and timely prayer timings. It involves the integration of various data sources, algorithms, and calculations to determine the specific prayer times for a given location and date. By leveraging geographic location information, time zone data, and calculation methods, prayer-times implementation ensures the provision of reliable and precise prayer timings. This implementation often employs APIs or libraries that offer convenient access to pre-computed data or real-time calculations, allowing developers to seamlessly integrate prayer times functionality into their applications. With prayer-times implementation, users can effortlessly stay connected to their religious obligations and practice their faith with punctuality and ease.

The IslamicFinder API

The IslamicFinder API is a comprehensive service that provides a wide range of Islamic data, including accurate prayer timings, Islamic calendar information, Qibla direction, and more. It offers developers a convenient and reliable way to integrate Islamic features into their applications or services. Let's explore the IslamicFinder API in more detail:

1. Prayer Timings: The IslamicFinder API allows users to retrieve accurate prayer timings based on specific parameters such as location, date, and calculation method. Users can obtain prayer times for various prayers, including Fajr (pre-dawn), Dhuhr (noon), Asr (afternoon), Maghrib (sunset), and Isha (night). The API considers geographic location, time zone, and calculation methods to provide precise prayer timings.

2. Islamic Calendar:With the IslamicFinder API, developers can access Islamic calendar information, including the dates of important Islamic events and holidays. Users can retrieve the dates of Islamic months, as well as details about significant occasions such as Ramadan, Eid al-Fitr, and Eid al-Adha. This information is crucial for users to stay connected with Islamic observances.

3. Qibla Direction: The IslamicFinder API offers Qibla direction functionality, which helps users determine the correct direction (towards the Kaaba in Mecca) for their prayers. Users can obtain the Qibla angle or the compass bearing based on their current location. This feature is particularly useful for individuals who are traveling or in areas where determining the Qibla direction may be challenging.

4. Calculation Methods and Adjustments: The IslamicFinder API supports different calculation methods for prayer timings, including popular methods such as Islamic Society of North America (ISNA) and Muslim World League (MWL). Users can specify their preferred calculation method to ensure prayer timings align with their local practices. Additionally, the API takes into account adjustments such as daylight saving time (DST) to ensure accurate prayer timings throughout the year.

5. API Integration: Developers can easily integrate the IslamicFinder API into their applications or services by making HTTP requests to the API endpoints. The API provides comprehensive documentation and guidelines to assist developers in implementing the desired features effectively. The response from the API is typically in a structured format such as JSON, making it easy to parse and utilize the retrieved Islamic data.

By leveraging the IslamicFinder API, developers can enhance their applications with valuable Islamic features, empowering users to access accurate prayer timings, Islamic calendar information, and Qibla direction. Whether it's for building Islamic prayer apps, Islamic event trackers, or Islamic educational platforms, the IslamicFinder API simplifies the integration process and ensures the availability of essential Islamic data for users.

Muslim Salat API

The Muslim Salat API is a service that provides prayer timing data and Islamic calendar information for Muslims worldwide. It offers developers a convenient way to integrate prayer timings and Islamic calendar functionality into their applications or services. Let's delve into the details of the Muslim Salat API:

1. Prayer Timings: The Muslim Salat API provides accurate prayer timings based on specific parameters such as location, date, and calculation method. Users can obtain prayer times for various prayers, including Fajr (pre-dawn), Dhuhr (noon), Asr (afternoon), Maghrib (sunset), and Isha (night). The API takes into account geographic location, time zone, and calculation methods to ensure precise prayer timings.

2. Islamic Calendar: With the Muslim Salat API, developers can access Islamic calendar information, including the dates of Islamic months and important Islamic events. Users can retrieve details about significant occasions such as

Ramadan, Eid al-Fitr, and Eid al-Adha. This feature enables users to stay connected with Islamic observances and plan their activities accordingly.

3. Calculation Methods and Adjustments: The Muslim Salat API supports various calculation methods for prayer timings, including widely recognized methods such as Islamic Society of North America (ISNA), Muslim World League (MWL), and Egyptian General Authority of Survey (EGAS). Users can select their preferred calculation method to align with their local practices. Additionally, the API accounts for adjustments such as daylight-saving time (DST) to provide accurate and consistent prayer timings.

4. API Integration: Integrating the Muslim Salat API into applications or services involves making HTTP requests to the API endpoints. Developers can specify the desired parameters, such as the location and date, to retrieve the relevant prayer timings and Islamic calendar information. The API provides clear documentation and guidelines to facilitate seamless integration, making it easy for developers to incorporate prayer timings and Islamic calendar features into their applications.

By utilizing the Muslim Salat API, developers can enhance their applications with prayer timings and Islamic calendar functionality, empowering users to access accurate and up-to-date Islamic data. Whether it's for creating prayer reminder apps, Islamic event trackers, or Islamic lifestyle platforms, the Muslim Salat API simplifies the process of integrating essential Islamic features and ensures that users have access to reliable prayer timings and Islamic calendar information.

Aladhan API

The AlAdhan API is a comprehensive prayer timing service API that provides accurate prayer timings based on geographic location and time zone information. It offers developers a convenient and reliable way to integrate prayer times functionality into their applications or services.

Here's how the AlAdhan API works:

1. Requesting Prayer Times: Users make HTTP requests to the AlAdhan API, specifying parameters such as the desired location, date, and calculation method for determining prayer times. The location can be provided as geographic coordinates (latitude and longitude) or as a city name. The API also supports additional parameters such as time zone adjustments and calculation methods.

2. Processing the Request: Upon receiving the request, the AlAdhan API processes the provided parameters to determine the specific prayer times required. It utilizes geographic location information to accurately calculate the sunrise, sunset, and other relevant astronomical data necessary for calculating prayer times. The API also considers additional factors such as altitude, daylight saving time adjustments, and specific calculation methods requested by the user.

3. Data Retrieval or Calculation: The AlAdhan API retrieves the relevant prayer times data from its database or performs calculations based on established algorithms. The database may contain pre-computed prayer times for various locations, enabling faster retrieval and reducing computational overhead. Alternatively, the API employs mathematical formulas and algorithms to calculate the prayer times on the fly based on the provided parameters.

4. Returning the Response: The AlAdhan API formats the retrieved or calculated prayer times data into a structured format, commonly JSON (JavaScript Object Notation). The response includes prayer times such as Fajr (pre-dawn), Dhuhr (noon), Asr (afternoon), Maghrib (sunset), and Isha (night), along with their corresponding timings for the specified location and date. Additional information such as the time zone, calculation method used, and any adjustments made are often included in the response for clarity and accuracy.

By integrating the AlAdhan API into their applications or services, developers can provide users with a reliable and convenient way to access accurate and up-to-date prayer timings. The API handles the complex calculations, considerations, and data retrieval, allowing users to obtain precise prayer times for their specific locations and dates effortlessly. Whether it's for displaying prayer times on a website or mobile app, setting up prayer reminders, or incorporating prayer timings into personal calendars, the AlAdhan API simplifies the process and ensures accurate prayer time information for users.

ALADHAN API VS ISLAMICFINDER API VS MUSLIM SALAT API [18]

	AlAdhan API	**IslamicFinder API**	**Muslim Salat API**
Prayer Timings	Accurate & Reliable	Accurate & Reliable	Accurate & Reliable
Request Flexibility	Customizable Parameters.	Customizable Parameters.	Customizable Parameters.
Adjustments	Handles DST adjustments.	Handles DST adjustments.	Handles DST adjustments.
Location Input	Latitude/Longitude or City	Latitude/Longitude or City	Latitude/Longitude or City
Data Format	JSON	JSON	JSON
Ease of Use	User-Friendly	User-Friendly	User-Friendly
Pricing	Freemium Model Available	Freemium Model Available	Freemium Model Available

Table 5-Aladhan vs IslmaicFinder vs Muslim Salat API

3.11Target Country/City Detection

Global Positioning System (GPS) [19]

GPS is a technology that utilizes a network of satellites to determine the precise geographic location of a device. The GPS system works by receiving signals from multiple satellites, calculating the device's distance from each satellite, and then triangulating the device's exact position. **Here's a detailed explanation of how GPS works:**

1. Satellite Signal Reception: The GPS device or receiver receives signals transmitted by multiple GPS satellites orbiting the Earth. Each satellite broadcasts a unique signal that contains information about the satellite's location and time of transmission.

2. Distance Calculation: The GPS device measures the time it takes for the signals to reach the device from each satellite. Since the signals travel at the speed of light, the GPS receiver can calculate the distance between itself and each satellite based on the time delay.

3. Trilateration: Trilateration is the mathematical process used by the GPS device to determine its precise location. By comparing the distance measurements from at least four satellites, the GPS receiver can calculate the device's position in three-dimensional space. The intersection of the distance spheres from each satellite determines the device's location.

4. Position Calculation: Using the trilateration calculations and the known positions of the satellites, the GPS device calculates its latitude, longitude, and sometimes altitude. This information provides the device's precise geographic coordinates.

5. Geolocation Display: Once the GPS device has determined its location, it can display the coordinates on a map or provide other location-based information to the user. This allows users to navigate, track their movements, or utilize location-based services.

It's important to note that GPS accuracy can be affected by factors such as signal obstructions (buildings, trees, tunnels), atmospheric conditions, and the quality of the GPS receiver itself. To improve accuracy, GPS devices often utilize additional technologies like Assisted GPS (A-GPS), which combines GPS signals with data from cellular networks or Wi-Fi to enhance positioning accuracy, especially in urban areas or indoors.

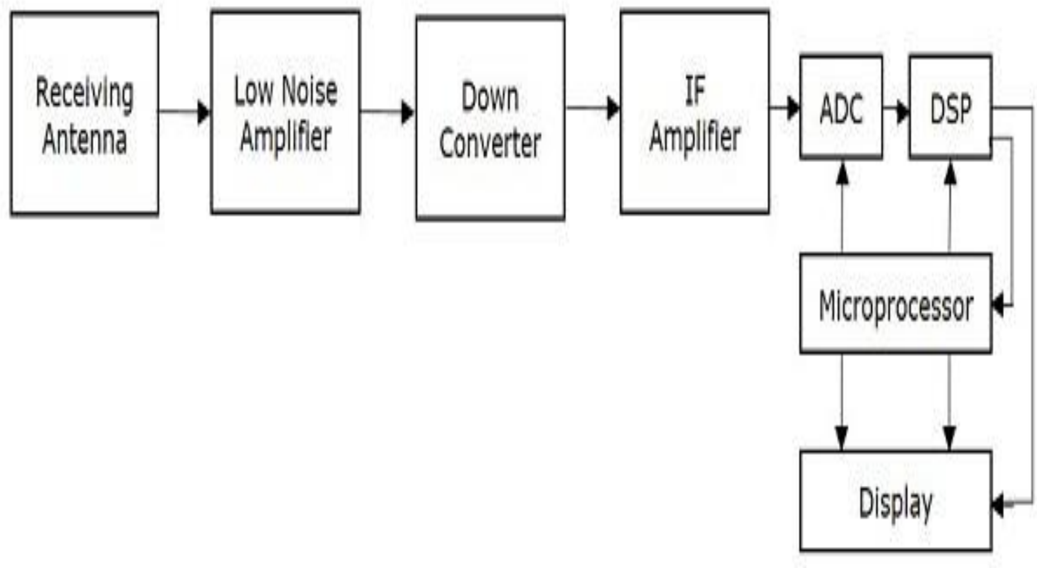


Figure 40-GPS

<u>IP Geolocation API</u> [20]

IP geolocation is a technique used to determine the physical location of a device, such as a computer or mobile device, based on its IP address. The IP Geolocation API simplifies this process by automatically extracting the IP address from the request headers without requiring explicit user input.

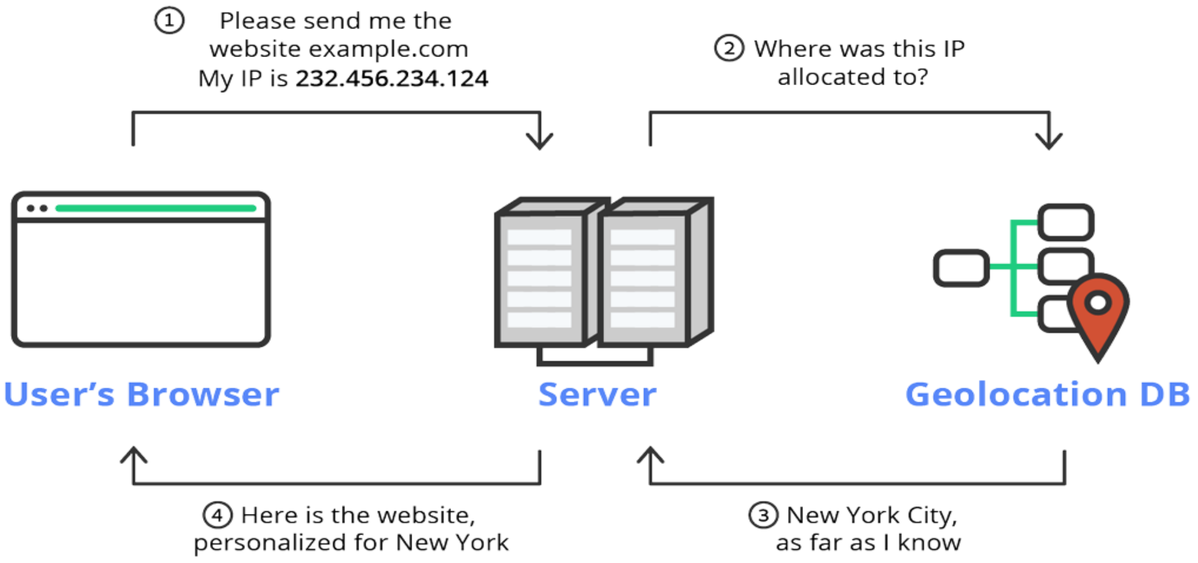

① Please send me the website example.com My IP is 232.456.234.124
② Where was this IP allocated to?
User's Browser
Server
Geolocation DB
④ Here is the website, personalized for New York
③ New York City, as far as I know

Figure 41-IP Geolocation API

Let's explore how it works:

1. Receiving the HTTP Request: When a device sends an HTTP request to a server, the IP Geolocation API intercepts and processes the request. It analyzes the request headers to extract the IP address of the device that initiated the request.

2. Extracting the IP Address: The IP Geolocation API retrieves the IP address from the request headers. This IP address serves as the key piece of information that is used to determine the geolocation of the device.

3. Determining Geolocation Information: Once the IP address is obtained, the IP Geolocation API utilizes a database or geolocation algorithms to map the IP address to a physical location. This process involves matching the IP address against a vast database of IP ranges and associated geographic information.

4. Returning the Geolocation Response: After determining the geolocation information, the IP Geolocation API formats the data into a structured response, commonly in JSON format. The response typically includes details such as the country, region, city, latitude, longitude, and even additional information like time zone and ISP.

By following these steps, the IP Geolocation API enables developers to retrieve accurate geolocation information based on the IP address of the device that made the request.

It's worth noting that the accuracy of IP geolocation may vary. While it can provide reasonably accurate results at the country or city level, the precision may decrease when attempting to pinpoint a specific physical address. Factors such as IP address allocation, network infrastructure, and the use of VPNs or proxies can affect the accuracy of geolocation data.

Overall, the IP Geolocation API streamlines the process of determining the physical location of a device based on its IP address. It eliminates the need for users to manually input their location and provides developers with an efficient solution to retrieve geolocation information for various applications and services.

IP Geolocation API Vs GPS API [21]

	IP Geolocation	**GPS**
Method	Based on mapping IP address to geographic	Based on satellite-based positioning
Accuracy	Moderate to low	High
Precision	Rough estimate of location	Precise coordinates
Dependency	Relies on IP address and associated data.	Requires GPS-enabled device and satellite signal.
Limitations	Affected by network configurations, proxies, and network address allocations.	Limited accuracy in obstructed areas (indoors, dense urban environments)

Table 6-IP Geolocation VS GPS

Chapter 4

System Implementation

&

Experimental Results

Chapter 4

System Implementation &Experimental Results

Introduction

This chapter provides a comprehensive overview of the implementation and experimental results of a versatile smart glasses system. The system incorporates a wide range of modules, including Optical Character Recognition (OCR) for text recognition, text-to-speech conversion, scene description for visual understanding, color identification for object recognition, age and gender identification, Prayer-Times for daily Islamic prayer schedules, Target City & Country for location identification, Distance Measurement using ultrasonic sensors, and QR Code scanning for information retrieval. Each module plays a vital role in enhancing the functionality of the smart glasses system, enabling it to provide a rich and immersive user experience.

Throughout this chapter, we will delve into the details of each module, discussing their algorithms, techniques, and practical implementations. Additionally, we will present the experimental results obtained from testing the system in various scenarios, assessing the accuracy, efficiency, and overall performance of each module. By the end of this chapter, you will gain a thorough understanding of the capabilities and potential applications of the smart glasses system, along with valuable insights into the experimental findings.

4.1.OCR (Optical Character Recognition) Module

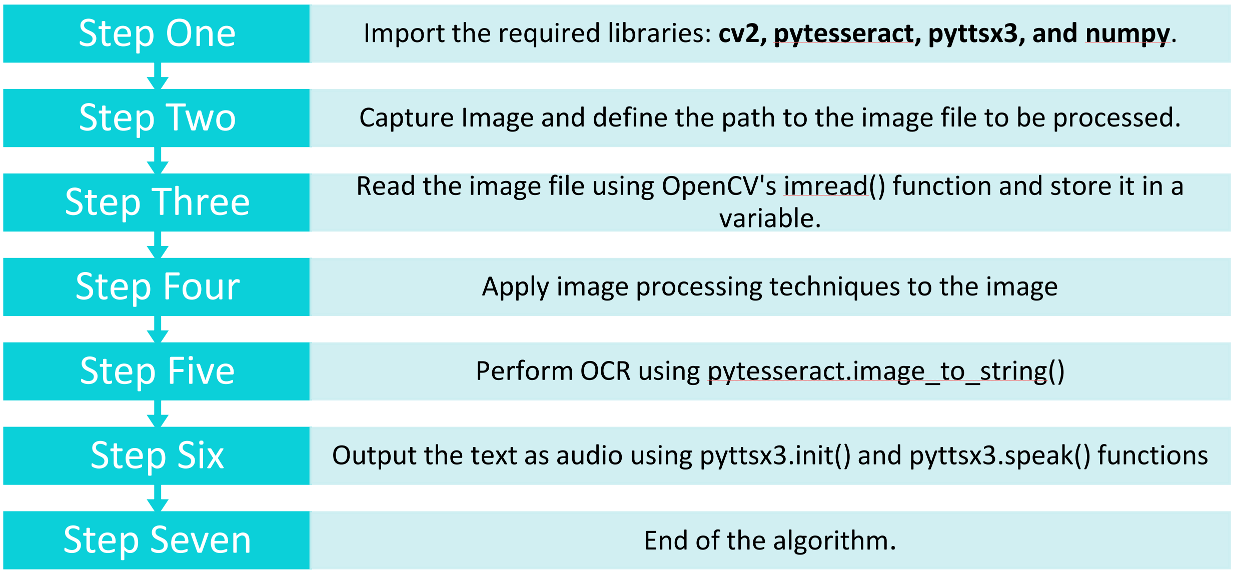
OCR algorithms employ advanced image processing techniques, such as image enhancement, noise reduction, and edge detection, to improve the accuracy of character recognition. Additionally, machine learning models are often employed to train the OCR algorithms, enabling them to learn and adapt to different fonts, languages, and styles of handwriting.

We support two languages, English and Arabic, in our "smart glasses".

4.1.1English OCR

Algorithm

In this algorithm, we will walk through the steps involved in performing Optical Character Recognition (OCR) on an image using Python and several libraries such as cv2, pytesseract, pyttsx3, and numpy. By importing the necessary libraries, capturing and processing the image, and utilizing OCR techniques, we can extract the text from the image and convert it into audio output.



Step One

Import the required libraries: **cv2, pytesseract, pyttsx3, and numpy**.

Step Two

Capture Image and define the path to the image file to be processed.

Step Three

Read the image file using OpenCV's imread() function and store it in a variable.

Step Four

Apply image processing techniques to the image

Step Five

Perform OCR using pytesseract.image_to_string()

Step Six

Output the text as audio using pyttsx3.init() and pyttsx3.speak() functions

Step Seven

End of the algorithm.

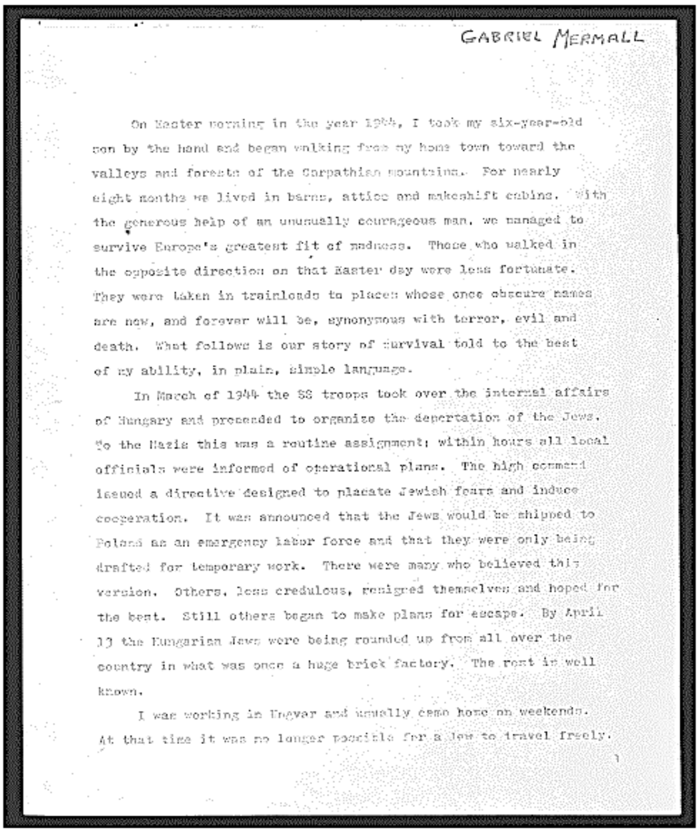
Figure 42-Englsih OCR Algorithm

Experimental Results

To validate the effectiveness of the algorithm, we conducted experiments where we tested its performance using various input images and observed the corresponding output of the OCR process.

Input

The user will capture a picture using a webcam, which will be saved and then processed to remove noise. Subsequently, the algorithm will convert the text within the image to a string and read it aloud using the pyttsx3 library. The processing time will vary depending on the size of the picture. The picture captured by the user may contain noise.



GABRIEL MERMALL

On Easter morning in the year 1944, I took my six-year-old son by the hand and began walking from my home town toward the valleys and forests of the Carpathian mountains. For nearly eight months we lived in barns, attics and makeshift cabins. With the generous help of an unusually courageous man, we managed to survive Europe's greatest fit of madness. Those who walked in the opposite direction on that Easter day were less fortunate. They were taken in trainloads to places whose once obscure names are now, and forever will be, synonymous with terror, evil and death. What follows is our story of survival told to the best of my ability, in plain, simple language.

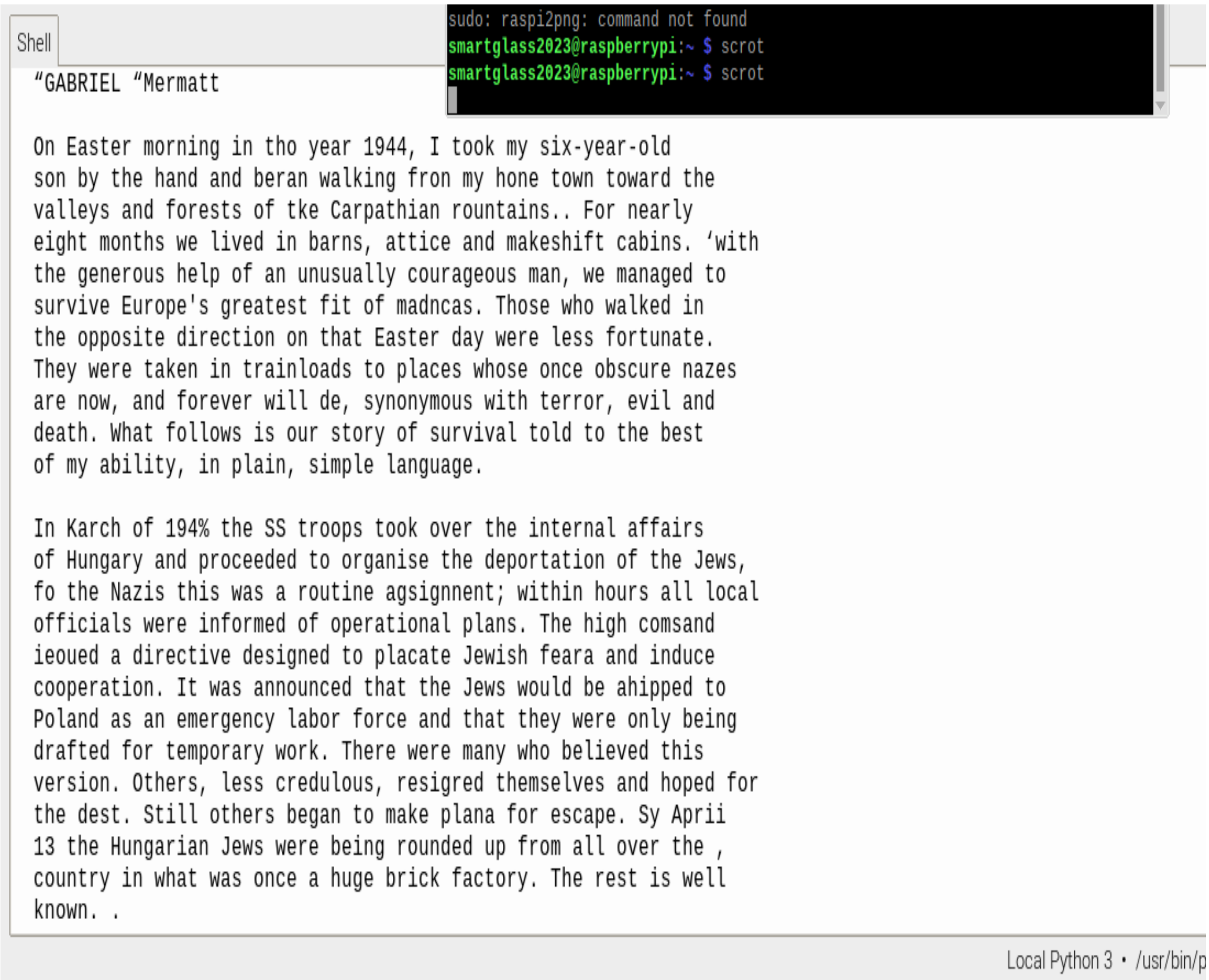
In March of 1944 the SS troops took over the internal affairs of Hungary and proceeded to organize the deportation of the Jews. To the Nazis this was a routine assignment; within hours all local officials were informed of operational plans. The high command issued a directive designed to placate Jewish fears and induce cooperation. It was announced that the Jews would be shipped to Poland as an emergency labor force and that they were only being drafted for temporary work. There were many who believed this version. Others, less credulous, resigned themselves and hoped for the best. Still others began to make plans for escape. By April 13 the Hungarian Jews were being rounded up from all over the country in what was once a huge brick factory. The rest is well known.

I was working in Ungvar and usually came home on weekends. At that time it was no longer possible for a Jew to travel freely.

1

Figure 43--Input Image English OCR

Output



```
sudo: raspi2png: command not found
smartglass2023@raspberrypi:~ $ scrot
smartglass2023@raspberrypi:~ $ scrot
```

```
“GABRIEL “Mermatt

On Easter morning in tho year 1944, I took my six-year-old
son by the hand and beran walking fron my hone town toward the
valleys and forests of tke Carpathian rountains.. For nearly
eight months we lived in barns, attice and makeshift cabins. ‘with
the generous help of an unusually courageous man, we managed to
survive Europe's greatest fit of madncas. Those who walked in
the opposite direction on that Easter day were less fortunate.
They were taken in trainloads to places whose once obscure nazes
are now, and forever will de, synonymous with terror, evil and
death. What follows is our story of survival told to the best
of my ability, in plain, simple language.

In Karch of 194% the SS troops took over the internal affairs
of Hungary and proceeded to organise the deportation of the Jews,
fo the Nazis this was a routine agsignment; within hours all local
officials were informed of operational plans. The high comsand
ieoued a directive designed to placate Jewish feara and induce
cooperation. It was announced that the Jews would be ahipped to
Poland as an emergency labor force and that they were only being
drafted for temporary work. There were many who believed this
version. Others, less credulous, resigred themselves and hoped for
the dest. Still others began to make plana for escape. Sy Aprii
13 the Hungarian Jews were being rounded up from all over the ,
country in what was once a huge brick factory. The rest is well
known. .
```

Figure 44-Output English OCR

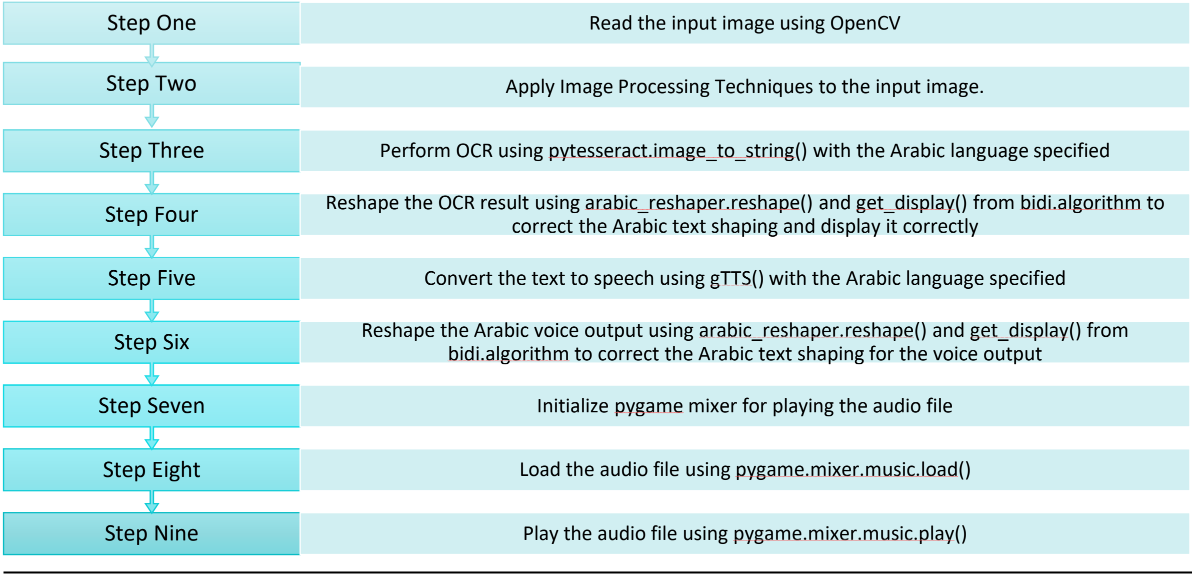
The output is produced with good accuracy, as it successfully extracts all characters from the previous picture and converts them into text.

4.1.2 Arabic OCR

Arabic OCR algorithms are specifically tailored to handle the unique characteristics of the Arabic script, including right-to-left reading direction, complex ligatures, and varying letter forms based on their position within a word. These algorithms leverage advanced image processing techniques and language-specific models to accurately identify and convert Arabic text from scanned documents, images, or even handwritten text into machine-readable format. By enabling automated data extraction and analysis, Arabic OCR plays a crucial role in enhancing productivity, accessibility, and information retrieval in various domains, such as document digitization, archival preservation, and multilingual text analysis.

Algorithm.

The Arabic OCR algorithm utilizes libraries such as Arabic reshape and the bidi algorithm to improve accuracy by standardizing letter shapes and correctly ordering Arabic script for display, enhancing recognition and interpretation of the text.



Step One	Read the input image using OpenCV
Step Two	Apply Image Processing Techniques to the input image.
Step Three	Perform OCR using pytesseract.image_to_string() with the Arabic language specified
Step Four	Reshape the OCR result using arabic_reshaper.reshape() and get_display() from bidi.algorithm to correct the Arabic text shaping and display it correctly
Step Five	Convert the text to speech using gTTS() with the Arabic language specified
Step Six	Reshape the Arabic voice output using arabic_reshaper.reshape() and get_display() from bidi.algorithm to correct the Arabic text shaping for the voice output
Step Seven	Initialize pygame mixer for playing the audio file
Step Eight	Load the audio file using pygame.mixer.music.load()
Step Nine	Play the audio file using pygame.mixer.music.play()

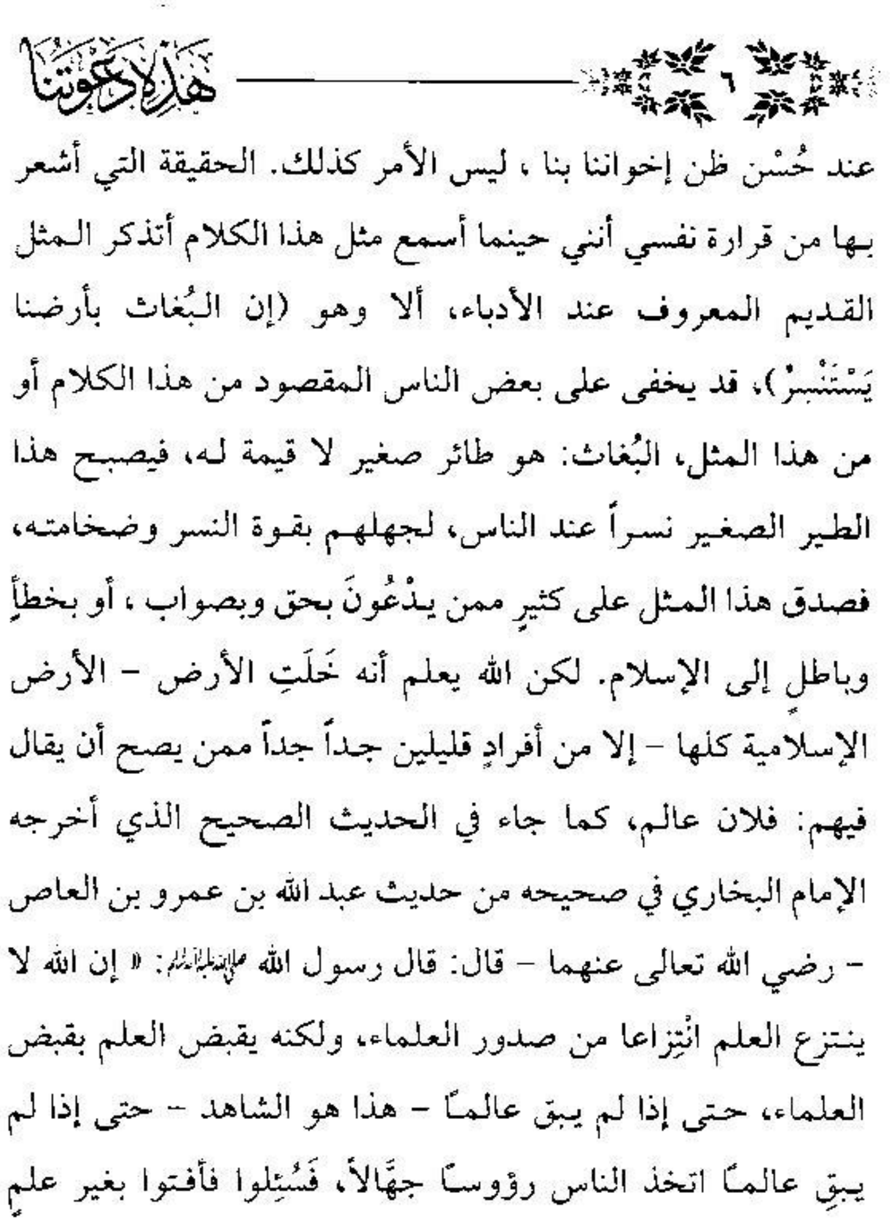
Figure 45-Arabic OCR Algorithm

Experimental Results

To validate the effectiveness of the algorithm, we conducted experiments where we tested its performance using various input images and observed the corresponding output of the OCR process.

Input

The user will capture a picture using a webcam, which will be saved and then processed to remove noise. Subsequently, the algorithm will convert the text within the image to a string and read it aloud using the pyttsx3 library. The processing time will vary depending on the size of the picture. The picture captured by the user may contain noise.

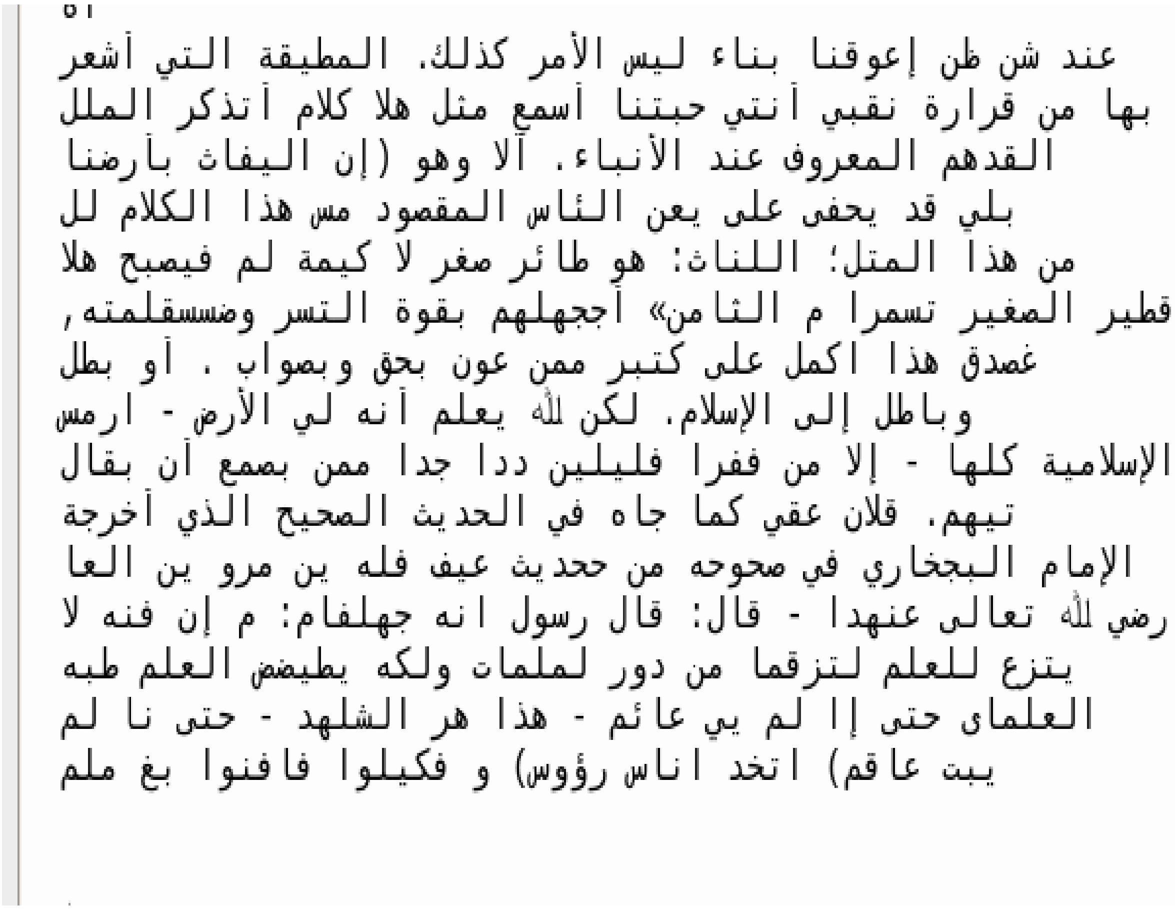


٦ هذه دعوتنا

عند حُسْن ظن إخواننا بنا ، ليس الأمر كذلك. الحقيقة التي أشعر بـها من قرارة نفسي أنني حينما أسمع مثل هذا الكلام أتذكر الـمثل القـديم المعروف عند الأدباء، ألا وهو (إن الـبُغاث بأرضنا يَسْتَنْسِرْ)، قد يخفى على بعض الناس المقصود من هذا الكلام أو من هذا المثل، البُغاث: هو طائر صغير لا قيمة لـه، فيصبح هذا الطـير الصغـير نسراً عند الناس، لجهلهـم بقـوة النسر وضخامتـه، فصدق هذا المـثل على كثيرٍ ممن يـدْعُونَ بحق وبصواب ، أو بخطأٍ وباطلٍ إلى الإسلام. لكن الله يعلم أنه خَلَتِ الأرض – الأرض الإسلامية كلها – إلا من أفرادٍ قليلين جداً جداً ممن يصح أن يقال فيهم: فلان عالم، كما جاء في الحديث الصحيح الذي أخرجه الإمام البخاري في صحيحه من حديث عبد الله بن عمرو بن العاص – رضي الله تعالى عنهما – قال: قال رسول الله ﷺ: « إن الله لا ينتزع العلم انْتِزاعا من صدور العلماء، ولكنه يقبض العلم بقبض العلماء، حتى إذا لم يبق عالمـاً – هذا هو الشاهد – حتى إذا لم يبقِ عالمـاً اتخذ الناس رؤوسـاً جهّالاً، فَسُئِلوا فأفتوا بغير علمٍ

Figure 46-Arabic OCR

Output



عند شن ظن إعوقنا بناء ليس الأمر كذلك. المطيقة التي أشعر
بها من قرارة نقبي أنتي حبتنا أسمع مثل هلا كلام أتذكر الملل
القدهم المعروف عند الأنباء. آلا وهو (إن اليفاث بأرضنا
بلي قد يحفى على يعن الئاس المقصود مس هذا الكلام لل
من هذا المتل؛ اللناث: هو طائر صغر لا كيمة لم فيصبح هلا
قطير الصغير تسمرا م الثامن» أججهلهم بقوة التسر وضسقلمته،
غصدق هذا اكمل على كتبر ممن عون بحق وبصواب . آو بطل
وباطل إلى الإسلام. لكن لله يعلم آنه لي الأرض - ارمس
الإسلامية كلها - إلا من ففرا فليلين ددا جدا ممن بصمع آن بقال
تيهم. قلان عقي كما جاه في الحديث المحيح الذي آخرجة
الإمام البجخاري في صحوحه من حديث عيف فله ين مرو ين العا
رضي لله تعالى عنهدا - قال: قال رسول انه جهلفام: م إن فنه لا
يتزع للعلم لتزقما من دور لملمات ولكه يطيمض العلم طبه
العلماى حتى إا لم يي عائم - هذا هر الشلهد - حتى نا لم
يبت عاقم) اتخد اناس رؤوس) و فكيلوا فافنوا بغ ملم

Figure 47-Arabic OCR Output

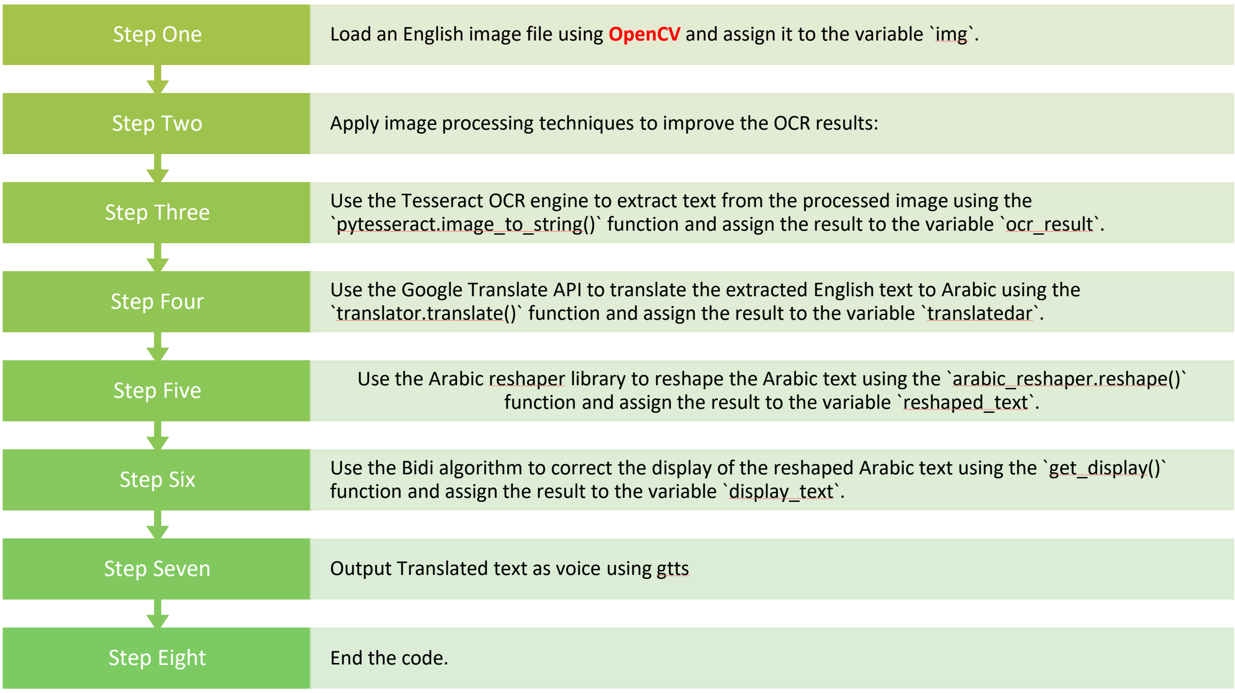
The output is produced with good accuracy, as it successfully extracts all characters from the previous picture and converts them into text.

4.2. Text Translation Module

Our "smart glasses" provide the capability to translate any text from English to Arabic using the Google Translate API. To utilize this feature, the user needs to have internet access.

Algorithm

In the following algorithm, we outline the steps required to process an English image, extract text using OCR, translate it to Arabic using the Google Translate API, reshape the Arabic text, correct its display, and finally output the translated text as voice using the gtts library.



Step One	Load an English image file using OpenCV and assign it to the variable `img`.
Step Two	Apply image processing techniques to improve the OCR results:
Step Three	Use the Tesseract OCR engine to extract text from the processed image using the `pytesseract.image_to_string()` function and assign the result to the variable `ocr_result`.
Step Four	Use the Google Translate API to translate the extracted English text to Arabic using the `translator.translate()` function and assign the result to the variable `translatedar`.
Step Five	Use the Arabic reshaper library to reshape the Arabic text using the `arabic_reshaper.reshape()` function and assign the result to the variable `reshaped_text`.
Step Six	Use the Bidi algorithm to correct the display of the reshaped Arabic text using the `get_display()` function and assign the result to the variable `display_text`.
Step Seven	Output Translated text as voice using gtts
Step Eight	End the code.

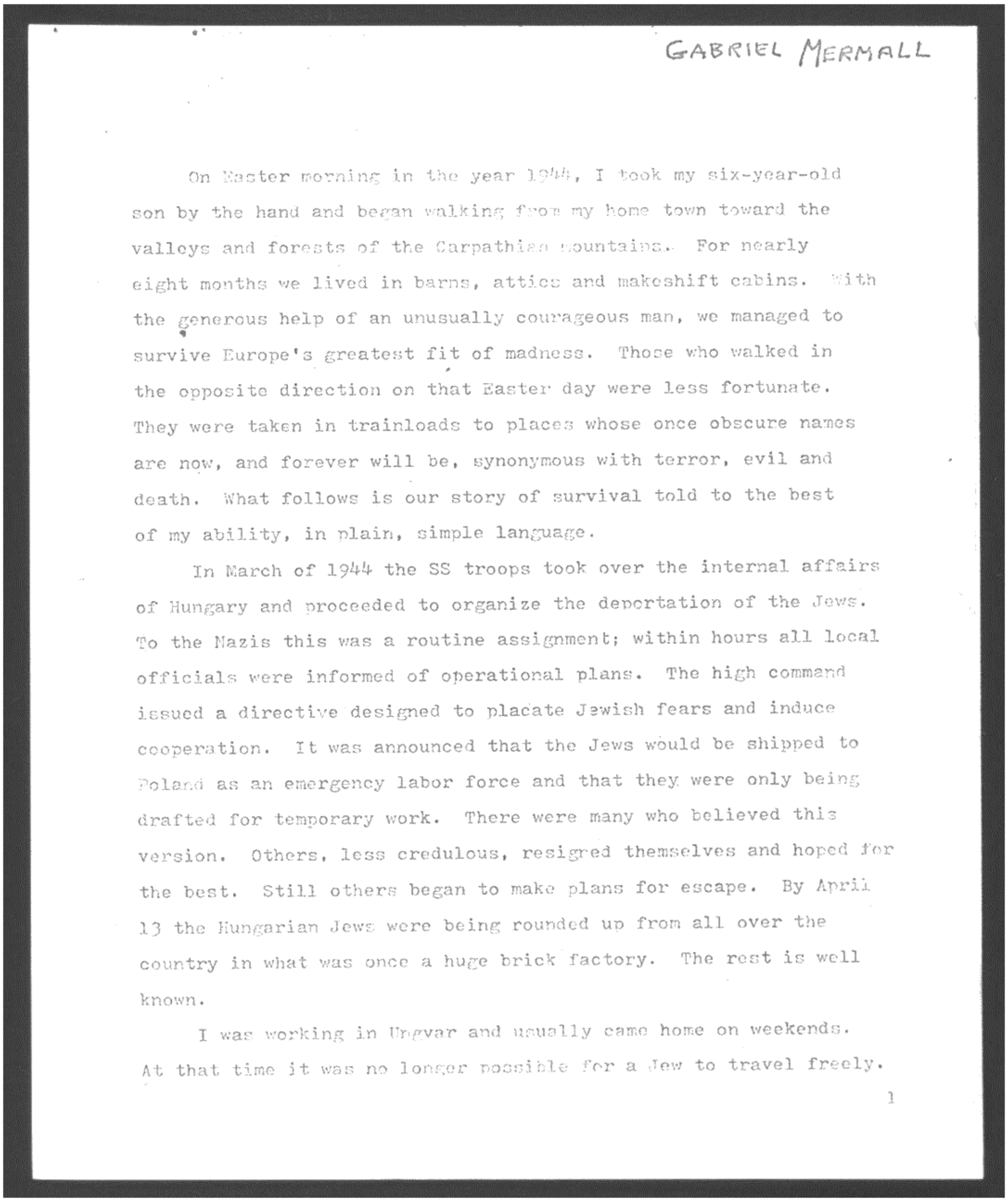
Figure 48-Text Translation Algorithm

Experimental Results

To validate the algorithm's performance, we conducted tests using an image containing noise which was successfully removed through pre-processing techniques, followed by OCR and translation performed on the extracted text.

Input

The user will capture a picture using a webcam, which will be saved and then processed to remove noise.



GABRIEL MERMALL

On Easter morning in the year 1944, I took my six-year-old son by the hand and began walking from my home town toward the valleys and forests of the Carpathian mountains. For nearly eight months we lived in barns, attics and makeshift cabins. With the generous help of an unusually courageous man, we managed to survive Europe's greatest fit of madness. Those who walked in the opposite direction on that Easter day were less fortunate. They were taken in trainloads to places whose once obscure names are now, and forever will be, synonymous with terror, evil and death. What follows is our story of survival told to the best of my ability, in plain, simple language.

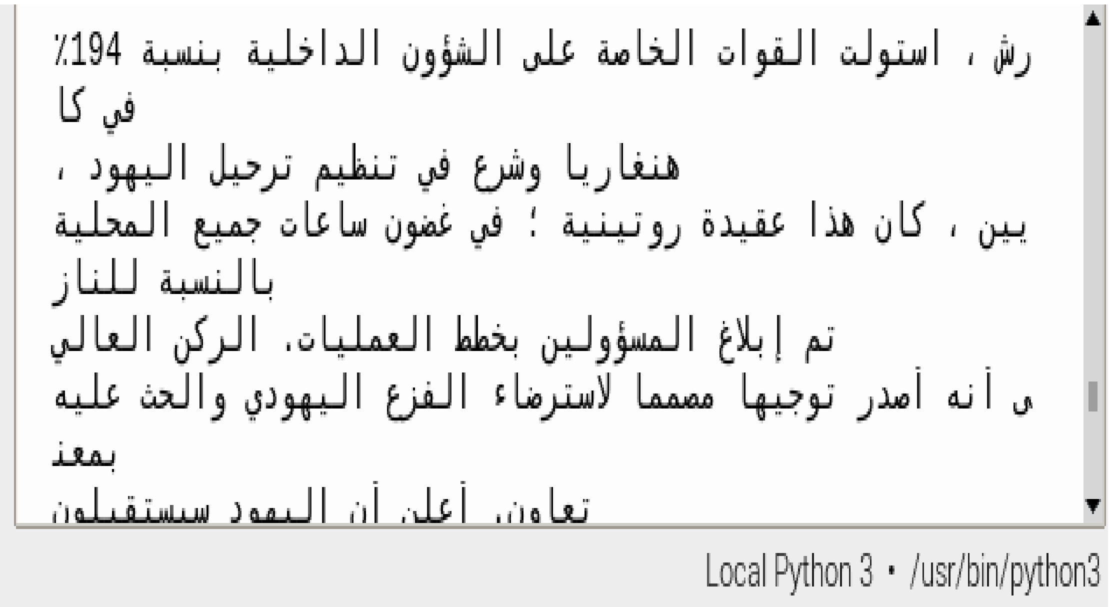
In March of 1944 the SS troops took over the internal affairs of Hungary and proceeded to organize the deportation of the Jews. To the Nazis this was a routine assignment; within hours all local officials were informed of operational plans. The high command issued a directive designed to placate Jewish fears and induce cooperation. It was announced that the Jews would be shipped to Poland as an emergency labor force and that they were only being drafted for temporary work. There were many who believed this version. Others, less credulous, resigred themselves and hoped for the best. Still others began to make plans for escape. By April 13 the Hungarian Jews were being rounded up from all over the country in what was once a huge brick factory. The rest is well known.

I was working in Ungvar and usually came home on weekends. At that time it was no longer possible for a Jew to travel freely.

1

Figure 49-Text Translation Input Image

Output



```
رش ، استولت القوات الخاصة على الشؤون الداخلية بنسبة 194٪
في كا
هنغاريا وشرع في تنظيم ترحيل اليهود ،
يين ، كان هذا عقيدة روتينية ؛ في غضون ساعات جميع المحلية
بالنسبة للناز
تم إبلاغ المسؤولين بخطط العمليات. الركن العالي
ى أنه أصدر توجيها مصمما لاسترضاء الفزع اليهودي والحث عليه
بمعذ
تعاون. أعلن أن اليهود سيستقبلون
```

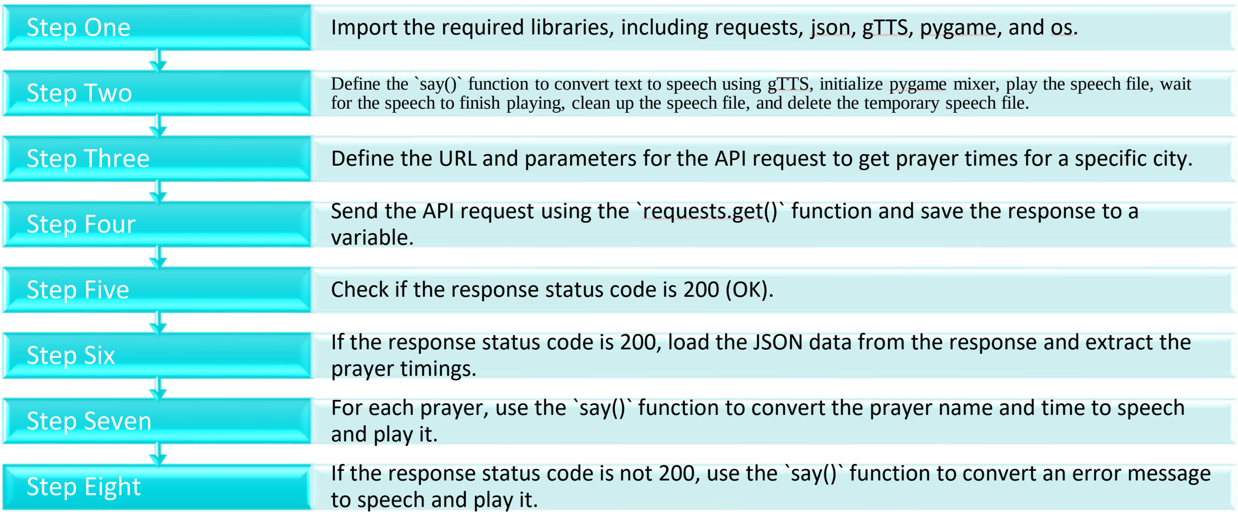
Figure 50--Text Translation Output

The code demonstrates good accuracy by successfully reading all the English text from the input picture using OCR techniques and accurately translating it to Arabic.

4.3. Prayer Times Module.

Aldhan API is a powerful financial data API that provides comprehensive and up-to-date information on financial markets, including stocks, currencies, and commodities. We chose to utilize the Aldhan API due to its extensive coverage, reliable data sources, and user-friendly interface. With the Aldhan API, we gain access to real-time market data, historical price information, company fundamentals, and technical indicators, enabling us to develop robust financial applications and conduct in-depth market analysis. The API's versatility allows us to retrieve data in various formats, such as JSON or CSV, and provides flexible options for filtering and customization. By leveraging the capabilities of the Aldhan API, we can deliver accurate and timely financial data to our users, empowering them with valuable insights for informed decision-making.

Algorithm



Step One	Import the required libraries, including requests, json, gTTS, pygame, and os.
Step Two	Define the `say()` function to convert text to speech using gTTS, initialize pygame mixer, play the speech file, wait for the speech to finish playing, clean up the speech file, and delete the temporary speech file.
Step Three	Define the URL and parameters for the API request to get prayer times for a specific city.
Step Four	Send the API request using the `requests.get()` function and save the response to a variable.
Step Five	Check if the response status code is 200 (OK).
Step Six	If the response status code is 200, load the JSON data from the response and extract the prayer timings.
Step Seven	For each prayer, use the `say()` function to convert the prayer name and time to speech and play it.
Step Eight	If the response status code is not 200, use the `say()` function to convert an error message to speech and play it.

Figure 51-Prayer Times Algorithm

Experimental Results

The output of the algorithm will be the sound audio playing the prayer times (Azan) for each prayer, allowing users to audibly hear and be reminded of the respective prayer timings

4.4. Scene Description Module

For the scene description algorithm on Raspberry Pi 4 Model B, we chose the MobileNet SSD (Single Shot MultiBox Detector) model due to its lightweight architecture, which enables real-time object detection and scene analysis on resource-constrained devices.

Algorithm

Step One	Import necessary libraries: **cv2**, **numpy**, and **pyttsx3**
Step Two	Load the class names from the coco.names file and create a detection model(mobile net ssd) using the pre-trained weights and configuration file
Step Three	Define a function called getObjects that takes an image value as input and detects the objects in the image using the detection model.
Step Four	The function returns the image with the detected objects drawn on it and a list of object information, such as the object's bounding box and class name.
Step Five	Use pyttsx3 to provide an audio output of the detected object information.
Step Six	End Algorithm

Figure 52-Scene Description Algorithm

Experimental Results

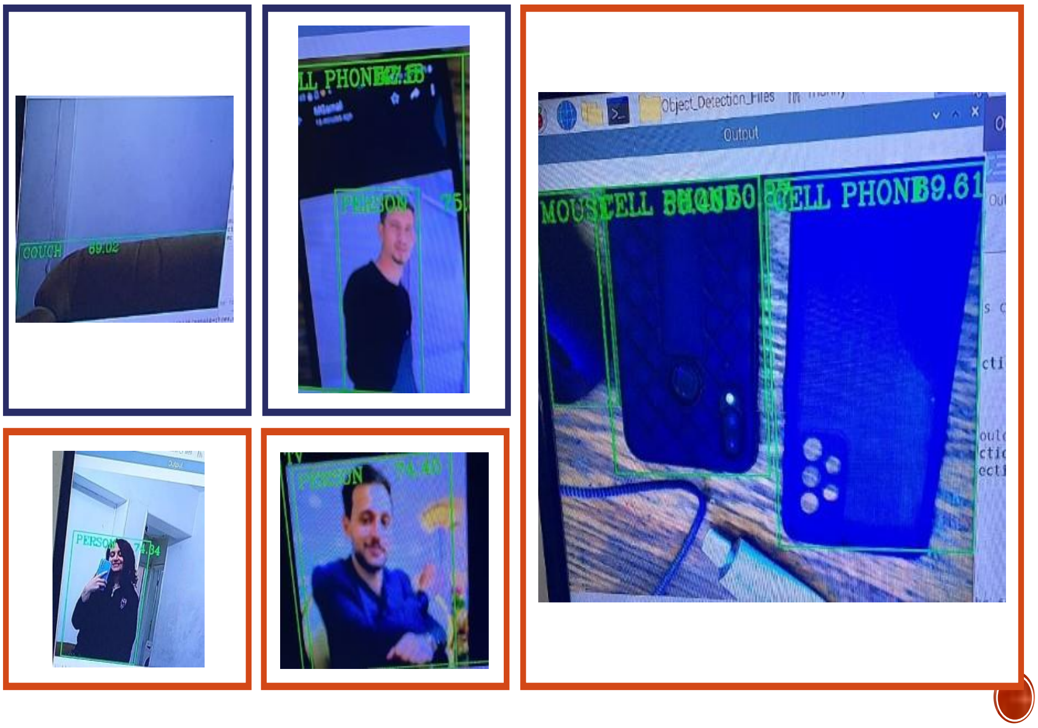


Figure 53-Scene Description Experimental Results

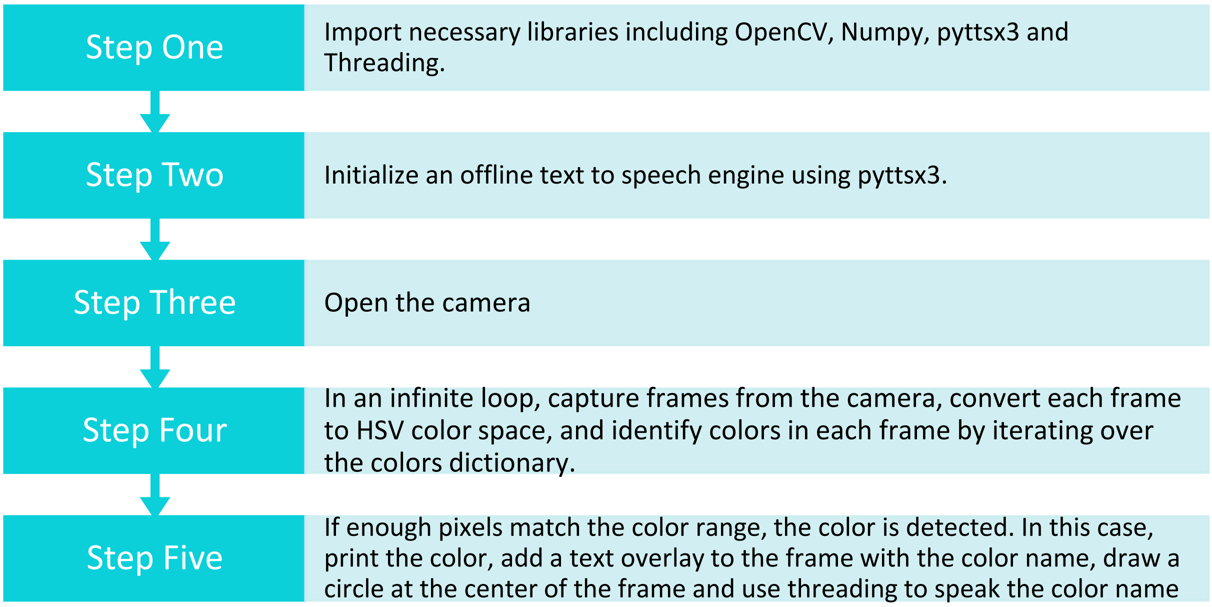
The experimental results demonstrate the algorithm's accuracy in successfully detecting objects in images and providing reliable object information, achieving a high level of accuracy in object detection.

4.5. Color Identification Module

The color identification module algorithm utilizes the HSV (Hue, Saturation, Value) color space for color detection, as it provides robustness to lighting variations and allows for more accurate color identification across different environments.

We chose the HSV color space for its ability to separate color components effectively and provide reliable color detection in various lighting conditions, enhancing the algorithm's accuracy and performance.

Algorithm



Step One

Import necessary libraries including OpenCV, Numpy, pyttsx3 and Threading.

Step Two

Initialize an offline text to speech engine using pyttsx3.

Step Three

Open the camera

Step Four

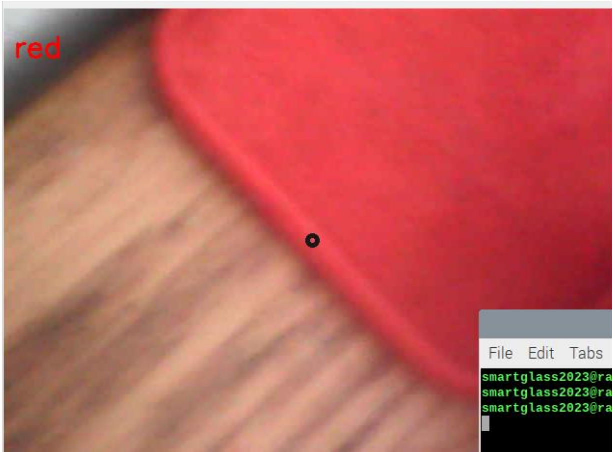
In an infinite loop, capture frames from the camera, convert each frame to HSV color space, and identify colors in each frame by iterating over the colors dictionary.

Step Five

If enough pixels match the color range, the color is detected. In this case, print the color, add a text overlay to the frame with the color name, draw a circle at the center of the frame and use threading to speak the color name

Figure 54-Color Identification Algorithm

Experimental Results



```
smartglass2023@ra
smartglass2023@ra
smartglass2023@ra
```

Figure 55-Red Color

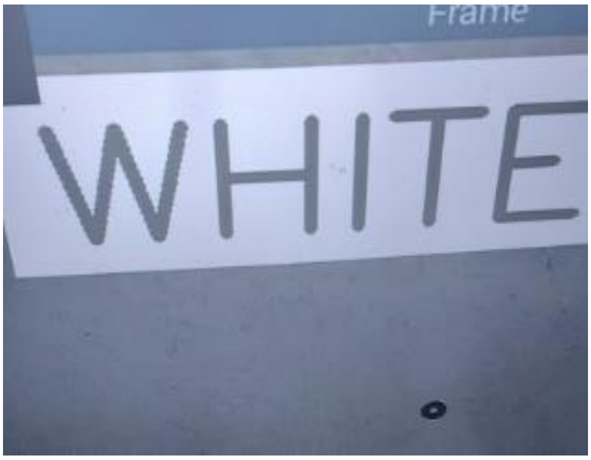


Figure 56-White Color

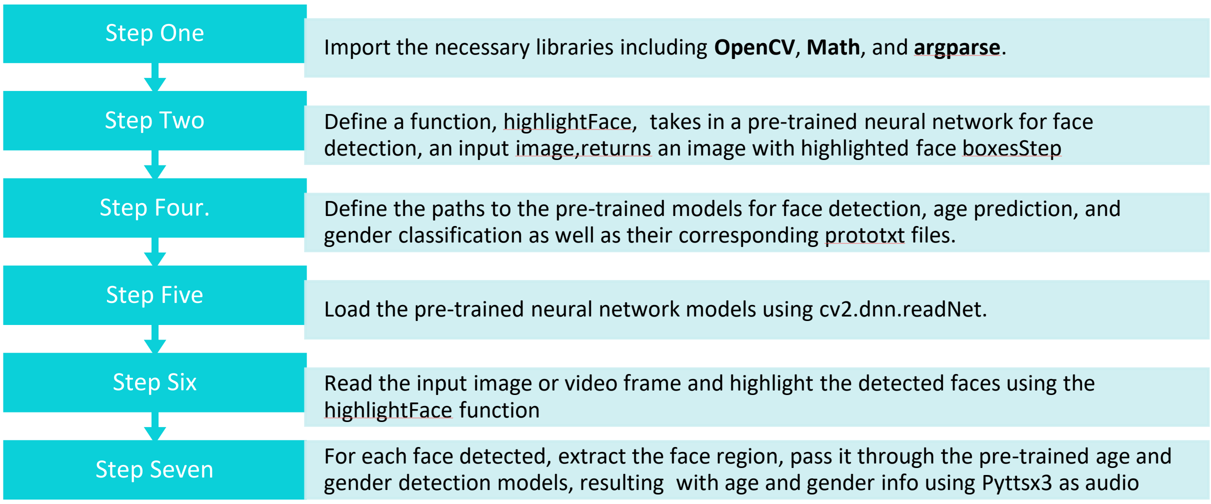
The experimental results demonstrate the algorithm's effectiveness in accurately identifying colors using the HSV color space, achieving a high level of accuracy across different lighting conditions and improving the reliability of color detection.

4.6Age/Gender Reveal Module

The algorithm employs specific pre-trained neural network models, namely "Caffe-based deep learning models" for face detection, "AgeNet" for age prediction, and "GenderNet" for gender classification.

We chose these models for their compatibility with resource-constrained devices like the Raspberry Pi 4 Model B, as they are lightweight and designed to achieve efficient inference on such platforms without compromising accuracy. These models strike a balance between computational efficiency and performance, making them well-suited for real-time applications on the Raspberry Pi 4 Model B.

Algorithm



Step One

Import the necessary libraries including **OpenCV**, **Math**, and **argparse**.

Step Two

Define a function, highlightFace, takes in a pre-trained neural network for face detection, an input image,returns an image with highlighted face boxesStep

Step Four.

Define the paths to the pre-trained models for face detection, age prediction, and gender classification as well as their corresponding prototxt files.

Step Five

Load the pre-trained neural network models using cv2.dnn.readNet.

Step Six

Read the input image or video frame and highlight the detected faces using the highlightFace function

Step Seven

For each face detected, extract the face region, pass it through the pre-trained age and gender detection models, resulting with age and gender info using Pyttsx3 as audio

Figure 57-Age/Gender Reveal Algorithm

Experimental Results

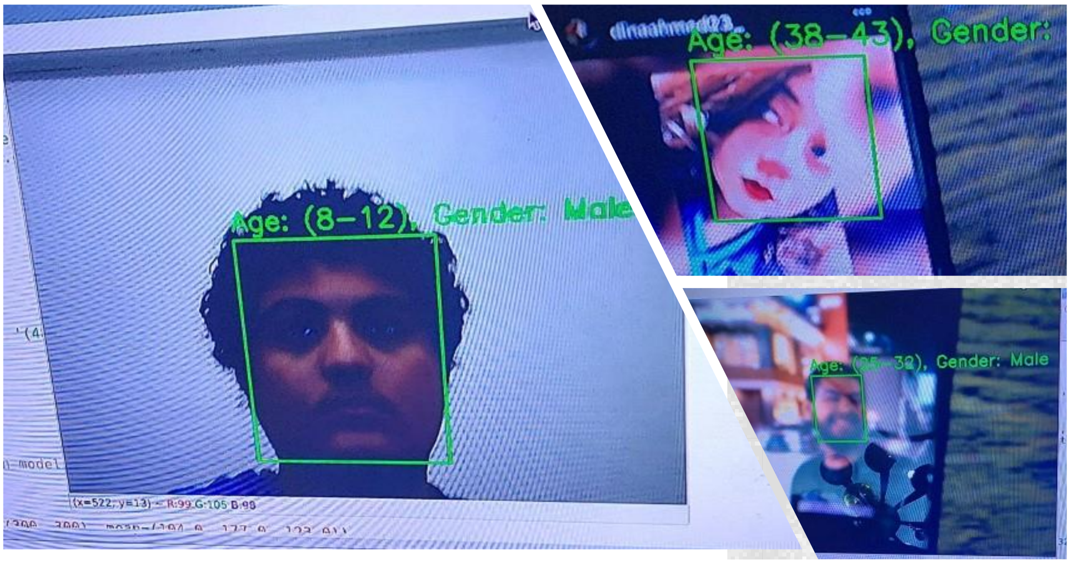


Figure 58-2-Age/Gender Reveal Experimental Result 1

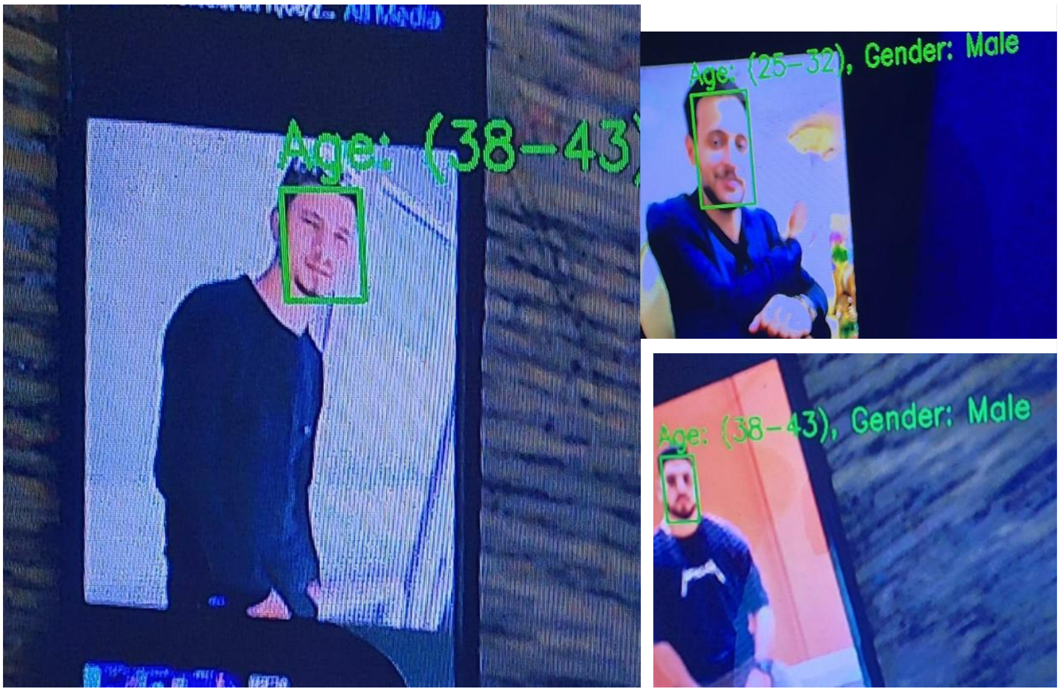


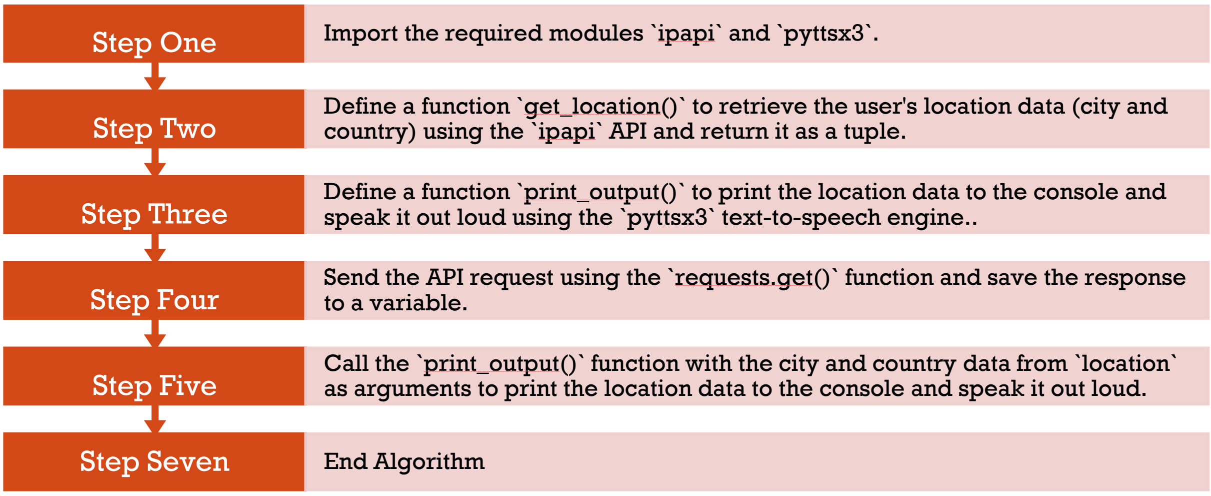
Figure 59--Age/Gender Reveal Experimental Results 2

The experimental results demonstrate the effectiveness of the chosen pre-trained neural network models in accurately detecting faces, predicting age, and classifying gender in real-time video streams, showcasing their compatibility and accuracy on the Raspberry Pi 4 Model B.

4.7 Target Country & City Module

When developing our smart glasses, we placed a strong emphasis on integrating a target city and country feature. To accomplish this, we opted to utilize an IP geolocation API that provides accurate location information based on the user's IP address. By leveraging this technology, our smart glasses can precisely determine the user's geographical details, enabling personalized functionalities and content tailored to their specific city and country. This enhances the user experience by offering location-specific information, such as local weather updates, points of interest, and language translations, making our smart glasses more useful and relevant to the user's surroundings.

<u>Algorithm</u>



Step One
Import the required modules `ipapi` and `pyttsx3`.

Step Two
Define a function `get_location()` to retrieve the user's location data (city and country) using the `ipapi` API and return it as a tuple.

Step Three
Define a function `print_output()` to print the location data to the console and speak it out loud using the `pyttsx3` text-to-speech engine..

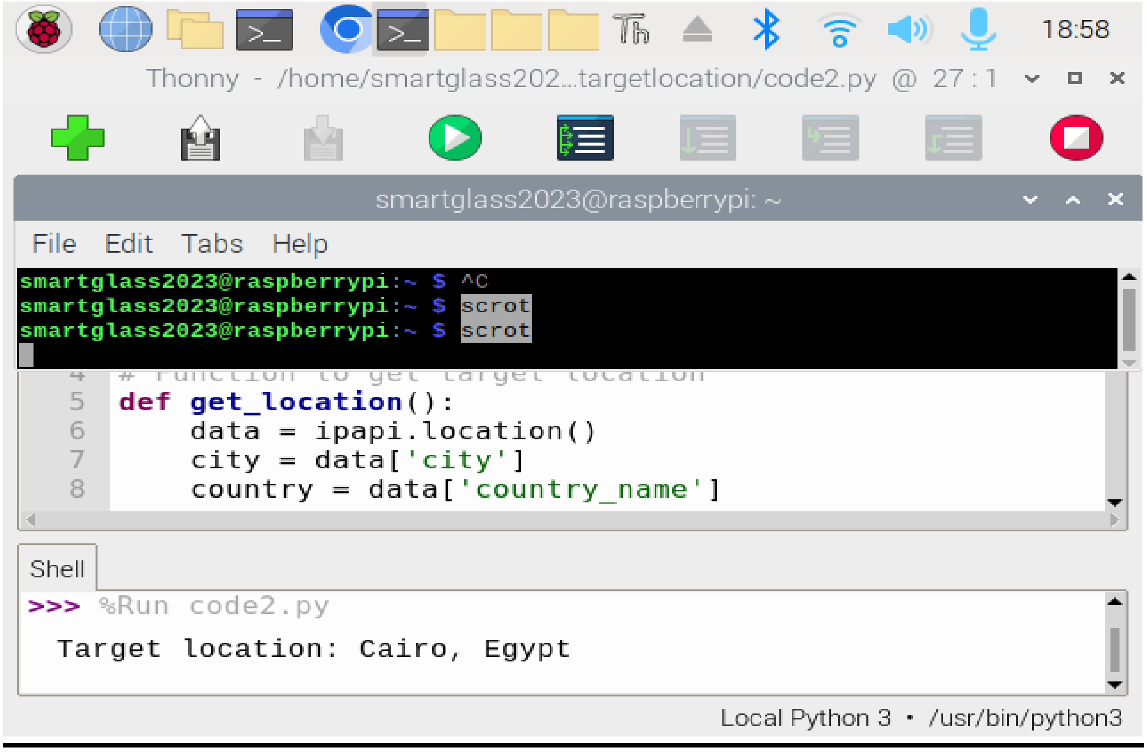
Step Four
Send the API request using the `requests.get()` function and save the response to a variable.

Step Five
Call the `print_output()` function with the city and country data from `location` as arguments to print the location data to the console and speak it out loud.

Step Seven
End Algorithm

Figure 60-Target Country & City Algorithm

Experimental Results



18:58
Thonny - /home/smartglass202...targetlocation/code2.py @ 27 : 1
smartglass2023@raspberrypi: ~
File Edit Tabs Help
Shell
Local Python 3 • /usr/bin/python3

```
smartglass2023@raspberrypi:~ $ ^C
smartglass2023@raspberrypi:~ $ scrot
smartglass2023@raspberrypi:~ $ scrot
```

```
# Function to get target location
def get_location():
    data = ipapi.location()
    city = data['city']
    country = data['country_name']
```

```
>>> %Run code2.py

  Target location: Cairo, Egypt
```

Figure 61-Target Country & City Experimental Result

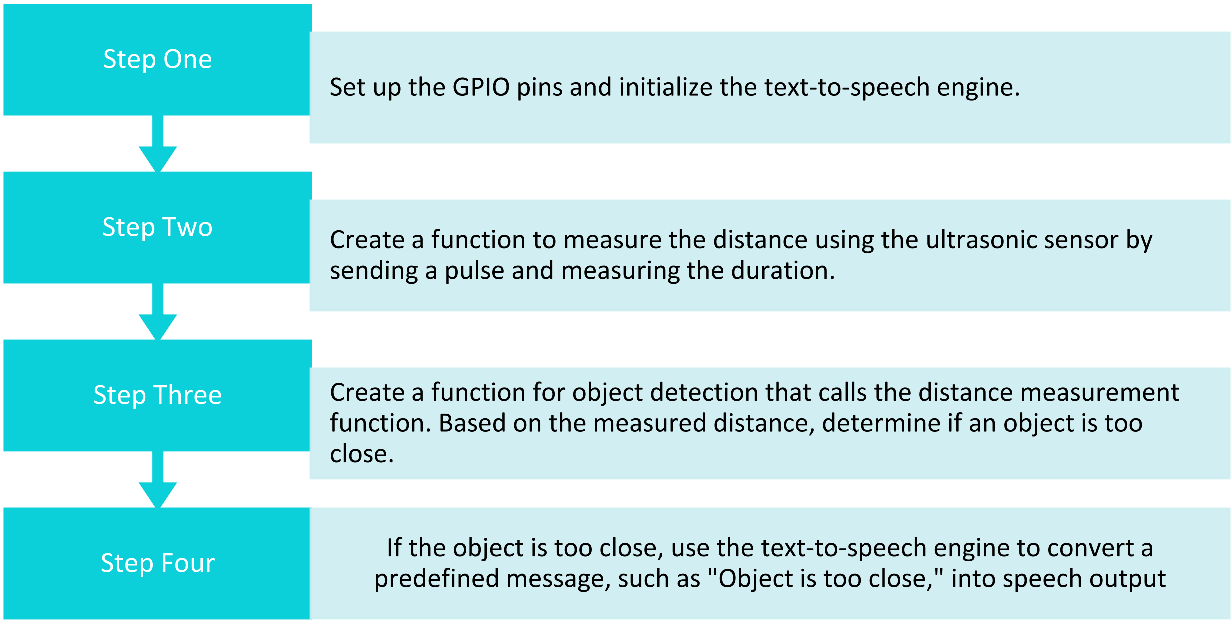
The experimental results confirmed the effectiveness of the IP geolocation API, as it accurately identified the user's location as Cairo, Egypt.

4.8. Distance Measurement Module

In our distance measurement feature, we incorporated an ultrasonic sensor to accurately determine the distance between objects and the smart glasses. The ultrasonic sensor utilizes sound waves to measure the time it takes for the waves to bounce back, enabling us to calculate distances and provide real-time feedback to the user. This functionality enhances the user experience by offering proximity awareness and assisting in tasks such as obstacle detection and navigation.

Algorithm

The algorithm utilizes an ultrasonic sensor, GPIO pins, and a text-to-speech engine to implement a distance measurement feature. By setting up the GPIO pins, initializing the text-to-speech engine, creating functions to measure the distance using the ultrasonic sensor and detect objects based on the measured distance, the algorithm enables real-time feedback through speech output when an object is detected to be too close.



Step One

Set up the GPIO pins and initialize the text-to-speech engine.

Step Two

Create a function to measure the distance using the ultrasonic sensor by sending a pulse and measuring the duration.

Step Three

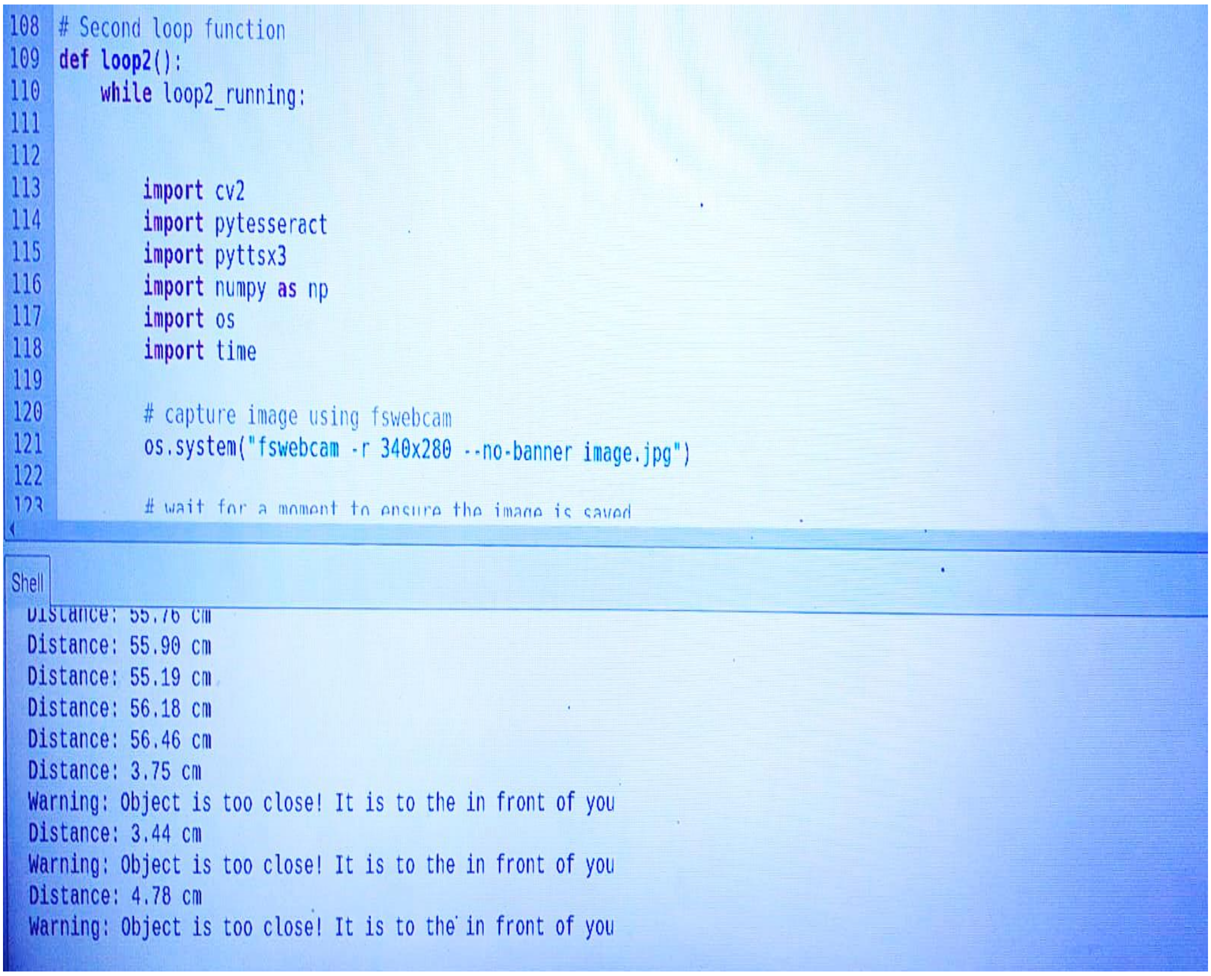
Create a function for object detection that calls the distance measurement function. Based on the measured distance, determine if an object is too close.

Step Four

If the object is too close, use the text-to-speech engine to convert a predefined message, such as "Object is too close," into speech output

Figure 62-Ultrasonic Algorithm

Experimental Results



```
# Second loop function
def loop2():
    while loop2_running:


        import cv2
        import pytesseract
        import pyttsx3
        import numpy as np
        import os
        import time

        # capture image using fswebcam
        os.system("fswebcam -r 340x280 --no-banner image.jpg")

        # wait for a moment to ensure the image is saved
```

```
Distance: 55.76 cm
Distance: 55.90 cm
Distance: 55.19 cm
Distance: 56.18 cm
Distance: 56.46 cm
Distance: 3.75 cm
Warning: Object is too close! It is to the in front of you
Distance: 3.44 cm
Warning: Object is too close! It is to the in front of you
Distance: 4.78 cm
Warning: Object is too close! It is to the in front of you
```

Figure 63-Distance Measurement Experimental Results

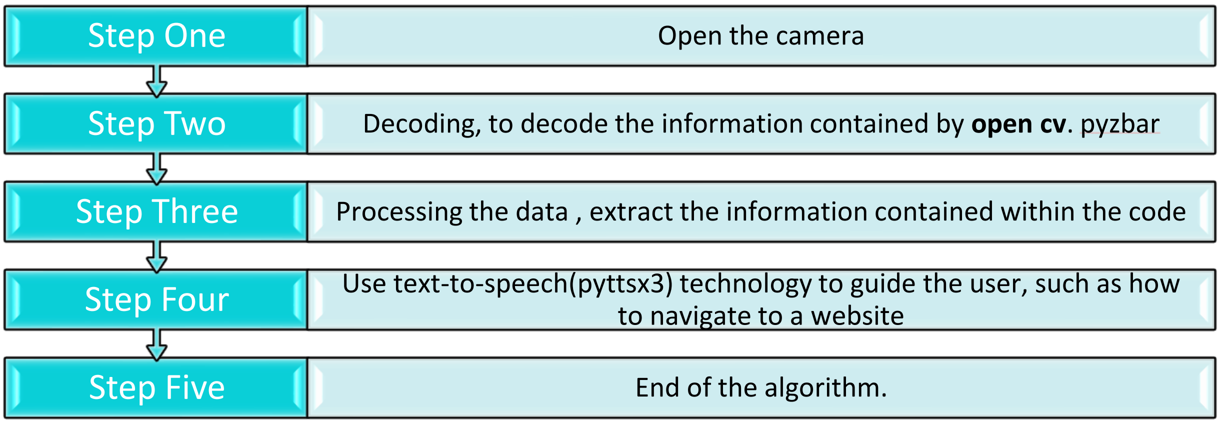
The experimental results demonstrated the effectiveness of the algorithm in detecting objects that are too close using the ultrasonic sensor, providing timely speech output to alert the user of potential obstacles.

4.9. QR Code Scan Module

In the context of computer vision and information retrieval, the process of scanning and decoding barcodes or QR codes has become increasingly prevalent. This technique enables the extraction of valuable information embedded within these codes, such as website URLs or product details.

By leveraging open-source libraries like OpenCV and pyzbar, we can implement a simple algorithm that utilizes the camera to scan and decode the codes, processes the extracted data, and provides guidance to the user through text-to-speech technology, ultimately enhancing their interactive experience with the information encoded in the codes.

Algorithm



Step One: Open the camera

Step Two: Decoding, to decode the information contained by **open cv**. pyzbar

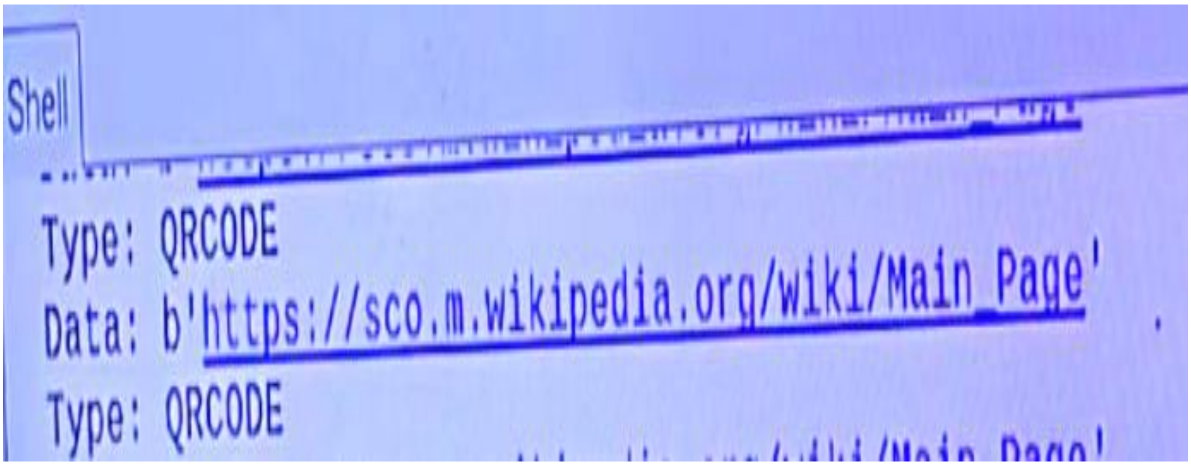
Step Three: Processing the data , extract the information contained within the code

Step Four: Use text-to-speech(pyttsx3) technology to guide the user, such as how to navigate to a website

Step Five: End of the algorithm.

Figure 64--QR Algorithm

Experimental Results



```
Type: QRCODE
Data: b'https://sco.m.wikipedia.org/wiki/Main_Page'
Type: QRCODE
```

Figure 65-QR Output

The experimental results demonstrate the effectiveness and accuracy of the proposed algorithm in successfully scanning and decoding barcodes or QR codes, extracting the embedded information, and providing user guidance through text-to-speech technology.

Chapter Five

Conclusion & Future Work

Chapter Five

Conclusion & Future Work

5.1. Conclusion

In conclusion, we have successfully developed smart glasses with a range of advanced features. These glasses incorporate OCR technology, allowing users to extract and recognize text from images or documents. Additionally, the glasses feature real-time text translation capabilities, bridging language barriers and facilitating communication across different languages.

Our smart glasses also include scene description functionality, enabling visually impaired individuals to comprehend the visual world through descriptive text. With facial analysis algorithms, the glasses can estimate the age and gender of individuals, finding applications in various fields such as targeted advertising and personalized user experiences.

Furthermore, the glasses incorporate color identification systems, allowing users to analyze and classify colors in their surroundings. They also feature QR code scanning capabilities, enabling users to extract embedded information from QR codes for various purposes such as website access or product information retrieval.

For individuals adhering to Islamic practices, our smart glasses provide accurate prayer-time information and assist in accurately determining the direction of the Qibla during prayers. Additionally, the glasses can detect and identify target countries or cities based on visual cues, supporting tourism, navigation, and cultural exploration.

To realize the full potential of these features, we have upgraded our smart glasses to the Jetson platform. Jetson provides high-performance GPU processing and deep learning capabilities, enabling faster and more accurate execution of image processing algorithms, OCR systems, facial analysis models, and scene description algorithms. With Jetson's AI capabilities, we can optimize and accelerate the execution of these algorithms, resulting in real-time and seamless user experiences.

5.2. Future work

In future iterations of the project, several aspects will be improved to enhance the functionality and usability of smart glasses for blind people. Firstly, an upgrade to **Jetson**, a powerful embedded AI computing platform, will be implemented. This upgrade aims to enhance overall performance, enabling faster image processing and analysis, thereby reducing processing time and providing more real-time assistance to the user.

Additionally, the **webcam** will be upgraded to a higher resolution one. This improvement will result in clearer and more detailed image capture, enabling better object recognition and scene understanding. The enhanced visual input will contribute to more accurate information retrieval and navigation assistance for the blind user.

To augment the navigation capabilities, **GPS** functionality will be integrated into the smart glasses. This addition will enable the device to determine the user's location and provide location-based information, such as nearby points of interest, directions, and navigation assistance.

In terms of additional modules, **Face Recognition technology** will be incorporated. This feature will allow the smart glasses to identify familiar faces and provide audio cues or descriptive information to the user, enhancing social interactions and personal recognition.

Furthermore, **Bank Note Identification** will be implemented to assist blind users in handling currency. The smart glasses will be able to recognize and provide spoken or tactile feedback regarding the denomination of banknotes, ensuring financial independence and security for visually impaired individuals.

Improving the design is a crucial aspect of the project. The smart glasses will be redesigned to resemble ordinary glasses, prioritizing the privacy of the blind person. This design enhancement will help the user feel more comfortable and confident while wearing the device in public. Additionally, efforts will be made to make the smart glasses lighter, reducing the burden on the user's face and improving long-term wearability.

Overall, these future improvements aim to create a more powerful and user-friendly smart glasses solution for the visually impaired, combining advanced technologies with a sleek and discreet design.

References

[1] I. V. Atlas, GBD Region Map & Estimates of Vision Loss, https://www.iapb.org/learn/vision-atlas/magnitude-and-projections/gbd-regions/north-africa-and-middle-east/.

[2] Zawya, https://www.zawya.com/en/press-release/amal-glass-launches-version-2-of-smart-glasses-for-blind-and-visually-impaired-people-ow9acukm, 2021.

[3] Envision, https://www.letsenvision.com/updates/glasses#:~:text=January%2026%2C%202023,based%20on%20their%20individual%20needs, 2023.

[4] A. Boxall, "CES 2023: How these smart glasses are giving people their vision back," https://www.digitaltrends.com/mobile/esight-go-smart-glasses-gives-people-their-vision-back-ces-2023/, 2023.

[5] Y. A. E.-H. A. I. I. F. E.-G. M. H. H. M. El-Shaarawy, "smart glasses for blind people," https://www.academia.edu/33796786/Smart_glasses_for_blind_people, 2017.

[6] M. B. T. Ali, "Smart Glasses for the Visually Impaired People," in *Conference: International Conference on Computers Helping People with Special Needs*, 2016.

[7] N. K. S. V. K. P. Nallapaneni Manoj Kumar, "Wearable Smart Glass: Features, Applications, Current Progress and Challenges," in *IEEE Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, Bangalore, India, 2018.

[8] . V. S. H. Vardhan, "Smart guiding glasses for visually impaired people in indoor environment".

[9] A. R. F. E. ALLIANCE, "5 Challenges that Providers of Smart Glasses Must Overcome," https://thearea.org/5-challenges-that-providers-of-smart-glasses-must-overcome-copy/, 2018.

[10] A. Rosebrock, "Tesseract Page segmentation modes (PSMs)Explained:How to Improve Your OCR Accuracy," Tesseract Page

Segmentation Modes (PSMs) Explained: How to Improve Your OCR Accuracy - PyImageSearch, 2021.

[11] p. Website, https://pyttsx3.readthedocs.io/en/latest/.

[12] Google, "gTTS," https://cloud.google.com/text-to-speech/docs/libraries.

[13] S. D. R. G. A. F. Joseph Redmon, "You Only Look Once: Unified, Real-Time Object Detection.You Only Look Once: Unified, Real-Time Object Detection (cv-foundation.org)," You Only Look Once: Unified, Real-Time Object Detection (cv-foundation.org).

[14] C. S. A. D. Shekofa GHOURY, "Real-Time Diseases Detection of Grape and Grape Leaves using Faster R-CNN and SSD MobileNet Architectures," International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2019), 2019.

[15] Understanding Hue, Saturation & Lightness (HSL) for Photo Retouching Giggster Guide.

[16] T. MAMDOUH, "Color spaces (RGB vs HSV) - Which one you should use?," Color spaces (RGB vs HSV) - Which one you should use? HubPages, 2020.

[17] S. M. K. P. M. K. S. Haseena, "Prediction of the Age and Gender Based on Human Face Images Based on Deep Learning Algorithm," in *https://www.hindawi.com/journals/cmmm/2022/1413597/*, 2022.

[18] I. Finder, https://www.islamicfinder.org/world/egypt/360630/cairo-prayer-times/.

[19] T. Point, Global Positioning System (tutorialspoint.com).

[20] B. Wu, "IP Geolocation: The Ultimate Guide," https://www.abstractapi.com/guides/ip-geolocation-the-ultimate-guide, 2023.

[21] F. Khatri, "GPS Geolocation vs IP Geolocation: How To Test Them On Real Devices?," https://www.lambdatest.com/blog/gps-geolocation-vs-ip-geolocation/, 2023.

[22] F. S. B. T. C. M. D. A. J. B. e. a. Bourne RRA, "Magnitude, temporal trends and projections of the global prevalence of blindness and distance and near

vision impairment: a systematic review and meta-analysis.," Lancet Glob Health, 2021.

[23] M. W. L. Rui Chen, "Analysis of the role and robustness of artificial intelligence in commodity image recognition under deep learning neural network," https://www.researchgate.net/publication/342760827_Analysis_of_the_role_and_robustness_of_artificial_intelligence_in_commodity_image_recognition_under_deep_learning_neural_network, 2020.

[24] A. M. a. F. D. Fiorella Fuentes, "The Usability of ICTs in People with Visual Disabilities: A Challenge in Spain," https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9517802/, Spain, 2022.

[25] "Smart guiding glasses for visually impaired people in indoor environment".