新加坡国立大学苏州研究院
NUS (Suzhou) Research Institute

Assistive Navigation Device for the Low Vision Users

Submitted by

Ai Zekai

U2202048

In partial satisfaction of the requirements for the

Degree of Bachelor of Engineering in Mechanical Engineering

Supervisor: Prof. ONG Soh Khim

Examiner: Prof. Andrew Nee Yeh Ching

Session 2022/2023

30 April 2023

Abstract

Visual impairment is a significant issue for many people around the world. Despite the progress of urbanization, there is still a lack of convenient facilities for blind people, making most low-vision individuals reluctant to travel alone.

The intelligent cane and object recognition device I have designed aims to improve low-vision users' mobility and independence significantly. These devices can assist users in navigating their surroundings, avoiding obstacles, and recognizing the environment by utilizing Raspberry Pi, Arduino, ultrasonic sensors, and Raspberry Pi Camera. Real-world experiments have been conducted to ensure the reliability of the devices and enhance the user experience.

Acknowledgements

How time flies! I never thought that this year would pass by so quickly. This year, many people helped me a lot. I want to express my thanks to all those who have helped me.

First, I want to express my heartfelt thanks to Prof. Ong Soh Khim and Prof. Andrew Nee Yeh Ching. They are always very patient and provide valuable feedback on my design, guiding me towards the direction of my research. At the same time, I would like to thank the NUSRI for providing me with such an opportunity to communicate with the professors. Lastly, I want to thank my parents for their support. They have always given me their selfless dedication without expecting anything in return.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background Information

According to the World Health Organization[1], the number of people worldwide with visual impairment has surpassed 220 million, steadily increasing. Visual problems become more common as people age, with about one-third of those aged 65 and above having some form of visual impairment. Due to declining physical abilities with age, they often encounter more difficulties in their daily travel. Currently, relatively few assistive travel devices are designed for the visually impaired, and many cities only have tactile paving. In contrast, other navigation or positioning devices for people with low vision have yet to be developed. This underscores the urgent need to create assistive navigation devices for low-vision users.

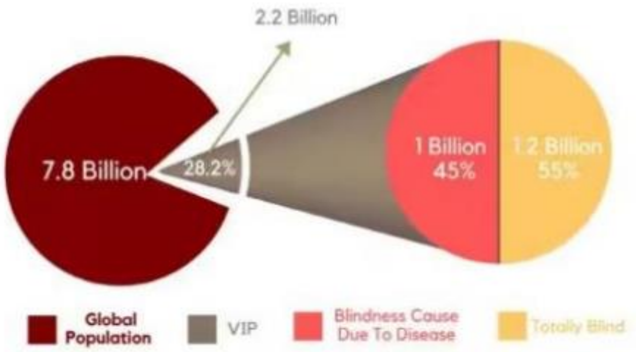


Figure 1: The ratio of visually impaired people [1]

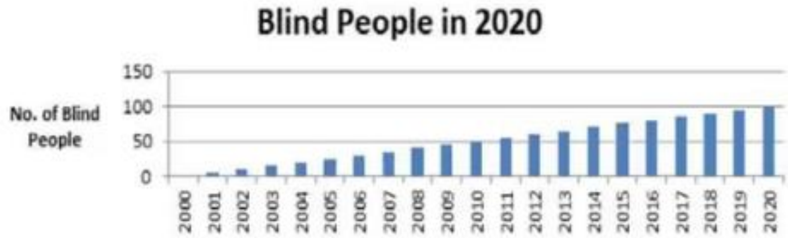


Figure 2: The prediction of the number of low-vision people [1]

1.2 Existing assistive navigation methods

1.2.1 Guide dog

The guide dog [2] is a type of working dog whose main job is to act as the eyes for visually impaired individuals, leading and helping them. They are often called "the second pair of eyes" for the blind. While guiding, the dogs provide the person with prompts for obstacles and directions to reach the intended destination rather than directly taking them there. The blind person will use verbal commands to instruct the guide dog to go straight, turn right, or turn left. During the journey, the guide dog is responsible for preventing the owner from colliding with obstacles, avoiding sudden changes in terrain, and leading the person around obstacles. Therefore, guide dog users must possess a stable and good sense of direction.

Guide dogs, while invaluable, have their limitations. Their ability to navigate obstacles is impressive, but their vision is limited to shades of gray, making it challenging to distinguish traffic lights. Additionally, some guide dogs may struggle with street signs. These limitations and the fact that guide dogs cannot provide information about the user's surroundings highlight the pressing need for more advanced and reliable navigation devices for the visually impaired.

Figure 3: The guide dog [2]

1.2.2 WeWALK

WeWALK [3] is a white cane with the remarkable feature of sensing objects within a 45-degree range to the left and right of the cane, enabling visually impaired people to be aware of their surroundings while walking and maintaining social distance. The white exterior is designed with a threaded grip to prevent slipping and can be used with one hand. The sensor is in front of the touchpad and provides feedback through vibration to alert the user of obstacles ahead. WeWALK can connect to smart devices via Bluetooth, and the built-in microphone can provide voice navigation instructions to ensure users find the right direction. The touchpad on the cane's surface can control a mobile phone, allowing users to use it without taking it out of their pocket. WeWALK can be charged via USB and has a battery life of about 5 hours. The device uses sensors and vibration feedback systems to inform users about direction and obstacles through vibration. It also includes voice control and gesture recognition functions for convenient operation of mobile phones, navigation software, and other applications. Additionally, the device can be paired with other smart devices for additional functionality, such as tracking health data.

While the WeWALK can offer numerous advantages, it's essential to consider its drawbacks. First, the price of this intelligent cane is relatively high, typically around $300, which may pose a financial challenge for some users. Second, the weight of the WeWALK smart cane may be a physical challenge for some users. Additionally, the device requires regular charging, and if the user fails to charge it in time, the intelligent system may not function properly. Finally, there is a learning curve for using the voice control system, which may require user practice.

Figure 4: WeWALK [3]

1.2.3 OrCam MyEye

OrCam MyEye [4] 2.0 is a lightweight and convenient wireless AI visual aid device that weighs only 22.5 grams and features a built-in speaker. It has five main functions: text recognition, facial recognition, product recognition, color recognition, and time recognition. OrCam uses advanced machine learning systems and optical character recognition technology to distinguish text structure and capture printed or digital text elements on any surface. The device's high-resolution camera is magnetically attached to the glass frame. It analyzes the captured text or objects through intelligent algorithms, converting the image to text in 1-2 seconds and reading it aloud through the built-in speaker, enabling users to "read" newspapers, menus, or books. The device's database can also store portraits, and users can take a picture of a person by pressing a button and dictating their identity information. When the person appears in the camera's view again, the device will inform the user of their identity. OrCam MyEye 2.0 is a wearable visual aid device. Users with partial vision can use manual indicators. They can take pictures by pressing a button when unable to determine their surroundings or objects, and the device will read from far to near.

The device's two built-in LED lights will automatically supplement the light in low-light conditions. OrCam MyEye 2.0 can be used continuously for 90 minutes and comes with a charging dock that is the size of a lighter and can be carried with you. A fully charged dock can provide four charges to the device, allowing for six hours of use. It is worth noting that OrCam MyEye 2.0 does not require internet connectivity while in use. Personal privacy items such as ID cards, social security cards, and insurance policies should be protected. The design of no internet connectivity is intended to protect patient privacy and avoid signal strength's impact on the continuity and completeness of user information reception.

Figure 5: OrCam MyEye [4]

1.2.4 Apple

With the help of advanced software, hardware, and machine learning capabilities, people with visual or mobility impairments can use door detection to guide themselves through the final steps toward their destination using an iPhone or iPad. Users with physical or mobility disabilities can use voice to control and switch functions through Apple Watch Mirroring on their iPhones. Deaf or hard-of-hearing individuals can view live captions on their iPhones, iPads, and Macs. Apple is also expanding its support for the industry-leading screen reader tool, VoiceOver, adding more than 20 language and region options. These features will be available through software updates across Apple's platforms later this year.

Apple is about to launch door detection, an advanced navigation feature explicitly designed for people who are blind or have low vision. Door detection can help users determine the location of a door as they approach their destination, telling them how far away the door is and describing its features, including whether the door is open or closed. If the door is closed, it can also tell the user whether they can open it by pushing, pulling, or rotating the handle. Door detection can also read signs and symbols on and around the door, such as office room numbers or accessibility entrance signs. This new feature combines laser scanner, camera, and on-device machine learning capabilities and will be available on iPhone and iPad models equipped with a laser scanner.

Door detection will become a new detection mode within the Magnifier app, an integrated app designed by Apple for blind and low-vision users. Door detection can be used separately or in combination with people detection and image description features in the detection mode, providing customizable tools for visually impaired users to navigate and obtain rich environmental information. In addition to the navigation tools within the Magnifier app, the Apple Maps app also provides sound and haptic feedback for users who use the VoiceOver feature to identify the starting point of walking navigation.

Figure 6:Apple [5]

1.2.5 HumanWare Trekker Breeze

The HumanWare Trekker Breeze [6] provides visually impaired travelers with the most accurate, intuitive, and effective navigation information. Based on popular user feedback, the Breeze+ uses the same practical design as the original Trekker Breeze for use on the go. With clear tactile buttons, all device functions can be operated with one hand, leaving the other free for your white cane or guide dog. This simple all-in-one handheld GPS is ideal for blind or low-vision individuals.

The HumanWare Trekker Breeze comes equipped with GPS, which prompts you with the names of streets, intersections, and landmarks as you walk; with the press of a button, the HumanWare Trekker Breeze will tell you about your location on the spot. When you are in a vehicle, the Breeze will announce all intersections, guiding you in the right direction. Place a landmark in front of your house, and the Breeze will guide you back to your exact location at any time; use landmark locations to get from point A to point B in open areas such as campuses or parks.

Figure 7: HumanWare Trekker Breeze [7]

1.2.6 BlindSquare

BlindSquare[8][8] is a navigation application specifically designed for users with low vision or blindness. It combines GPS technology and location data to provide direction and information about the surrounding environment, making it easier for users to navigate to their desired destination. BlindSquare uses GPS technology to locate the user's position and provide direction and navigation prompts. Users can input the address they want to go to and receive detailed navigation instructions. BlindSquare can identify landmarks around the user, such as shops, restaurants, and public facilities. When the user approaches these landmarks, the application provides them with relevant information such as name, address, and business hours. BlindSquare can provide navigation prompts and landmark identification using voice prompts. It can also be customized based on user preferences, such as volume and speech rate. Users can use Bluetooth devices to work with BlindSquare, making receiving navigation prompts and landmark information more convenient. BlindSquare can be integrated with social media platforms such as Foursquare and Facebook to provide more information about surrounding landmarks.

Figure 8: BlindSquare [8]

1.2.7 Seeing AI

Seeing AI [9] is an artificial intelligence assistive tool developed by Microsoft that aims to help people with visual impairments better understand their surroundings. It uses computer vision and natural language processing technologies to identify text, objects, and facial expressions and provides relevant information to the user through voice or vibration prompts.

Seeing AI can recognize text in photos and convert it into readable text, which is very useful for visually impaired users. It supports multiple languages and font types and can help users read documents, labels, menus, and more. Seeing AI can also recognize objects and provide information about their appearance, color, distance, and other aspects to the user. It can help users identify items in unfamiliar environments, such as finding a specific product in a supermarket. Seeing AI can recognize facial expressions and provide information about emotions, gender, age, and more to the user. This can help users better understand social interactions and emotional expression.

Seeing AI can recognize the surrounding environment and provide information about landmarks, buildings, roads, sidewalks, and more to the user. This is very useful for visually impaired users to navigate. Seeing AI uses voice prompts to provide relevant information to the user, which can be heard through headphones or speakers. It can also provide vibration prompts when the user is near an object.

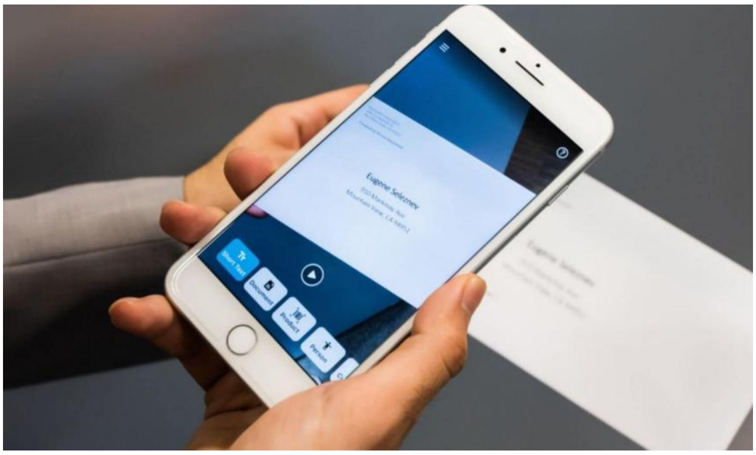


Figure 9: Seeing AI [9]

Chapter 2: Solution Selection

2.1 Design Ideas

I considered several options when designing the model and ultimately decided on an intelligent walking stick and a head-mounted target recognition device. Here, I will elaborate on the process of making the final decision on this chosen approach.

2.1.1 Smart Shoes

At first, I planned to install a vibration motor in the shoe and use GPS for positioning. The walking route could be scheduled in advance, and the direction of travel could be conveyed through the vibration of the left and right foot motors at specific intersections. Additionally, an ultrasonic sensor could be installed at the front of the shoe to detect obstacles ahead. If an obstacle is detected, the user can be alerted through a vibration motor at a specific location, allowing for timely avoidance.

Using shoes as an auxiliary visual aid has several advantages. First, the distance between the shoe and the ground is very close, ensuring that the obstacle information collected by the ultrasonic sensor is from the user's path. Secondly, putting on and taking off shoes is relatively convenient, making it more user-friendly. Furthermore, transmitting directional information through vibration is relatively simple, and users do not need to invest much in training costs.

Despite the many benefits of using this approach as a navigation method for low-vision users, there are still many shortcomings. First, this technology requires specific shoe components, such as vibration motors, which may make the shoes bulky and uncomfortable, affecting the user experience. Secondly, the contact between the shoes and the ground may cause the position of the sensor to change, affecting the accuracy and reliability of the sensor and other modules. For example, the sensor may not correctly recognize the user's direction or position, causing the user to deviate from the expected path and not reach the destination. Additionally, if a GPS module is required, the hardware may be large, and it may not be easy to integrate fully into the shoes.

2.1.2 Smart Glove

Later, I considered designing an intelligent glove that integrates sensors and can also carry a GPS module. The glove can provide navigation guidance through vibration or voice prompts, such as vibrating the left-hand glove to indicate the need to turn left and vibrating the right-hand glove to indicate the need to turn right. At the same time, the tactile sensors on the glove can detect the shape and texture of objects and simulate this information through vibration.

However, this design still has some drawbacks. Firstly, it is relatively costly to achieve the ability to detect the shape and texture of objects and simulate them through vibration, and not all low-vision users can afford it. Secondly, the operation of the intelligent glove is relatively complex, and users need to spend some time adapting to using the intelligent glove and learning how to interpret the information provided by the glove correctly. In addition, because there are multiple vibration motors and tactile sensors on the glove, there may be mutual interference between signals from different sensors. Finally, since the intelligent glove needs to use batteries to drive the vibration motors and tactile sensors, there is a high demand for batteries. It needs to be recharged or have its batteries replaced regularly.

2.1.3 Smart Cap

I have also considered designing an intelligent cap that integrates devices such as a camera, Raspberry Pi, ultrasonic sensor, and infrared intensity sensor into the cap. The camera is mounted on the Raspberry Pi and can be used for visual recognition. For example, if the camera recognizes a zebra crossing, it will prompt the user that they are about to cross a zebra crossing. If the camera detects that the traffic light is red, it will broadcast the result to the user via Bluetooth earphones, and the user needs to wait for the red light. If the camera detects the green traffic light, it will also broadcast the result to the user, and the user can proceed.

In addition, the smart cap is also equipped with an ultrasonic sensor for obstacle detection. As the cap is located at the highest point of the user, it has a high reference value for obstacle detection. At the same time, the intelligent cap can obtain the user's real-time location information through GPS or other positioning technologies.

However, this device also has some drawbacks. Firstly, installing sensors and Raspberry Pi may make the cap heavy and unbalanced, causing significant discomfort to the user during wear and daily use. Secondly, the product requires the implementation of target recognition algorithms, which is also challenging and requires training models, making development difficult. Additionally, the weight of the Raspberry Pi and battery is relatively heavy, making it difficult to arrange the entire device.

2.1.4 Smart Cane

One of the devices I ultimately decided to use is the intelligent cane, on which I designed three main functions. Firstly, obstacle detection is achieved by installing ultrasonic sensors, along with a buzzer and vibration motor. When the ultrasonic sensor detects an obstacle in front, it will transmit a signal to the Arduino, which will then control the buzzer and vibration motor to alert visually impaired users of the obstacle. Secondly, a button alarm function triggers LED lights, a buzzer, and a vibration motor when the user encounters an emergency. This can manually alert bystanders to the user's urgent situation. Thirdly, there is fall detection. If the user falls, the cane will fall with them, and the tilt switch on the cane will sense the change in its state and transmit a signal to the Arduino, which will then activate the buzzer and vibration motor to alert people nearby to the user's condition.

The advantage of using a cane as a carrier for assistive navigation devices for the visually impaired is manifold. Firstly, visually impaired people commonly use white canes as walking aids, so adapting to an intelligent cane is relatively fast. In addition, the fall detection, one-button alarm, and obstacle detection functions added to the conventional cane are practical, providing more security for visually impaired users.

Moreover, the intelligent cane I designed is much lower in cost than those currently on the market, making it more convenient for widespread use.

2.1.5 Object Identification Device

The head-mounted recognition device is another device that I am planning to put into use. The head-mounted recognition device consists of three components, namely, a head-mounted fixture with a camera, ultrasonic sensor, and buzzer; a vest with a Raspberry Pi and power bank, which provides power to the Raspberry Pi; and a Bluetooth bone conduction headset. I have designed two main functions for the head-mounted recognition device: to capture images of the scene using the camera, recognize specific scenes, and notify the user through the Bluetooth headset. For example, when a user wants to cross the road, the Raspberry Pi camera can recognize the zebra crossing and prompt the user through the Bluetooth headset. It can also recognize traffic lights, stairs, and other objects. The second function is ultrasonic ranging. The ultrasonic sensor fixed on the head can detect obstacles in high places, such as billboards. The height information collected is relatively accurate since the sensor is placed on the head, which is the highest point of view. If the sensor detects an obstacle, it will send a signal to the Raspberry Pi, which controls the buzzer to warn the user of collision risks. I have also tried to apply object recognition in a home environment. By increasing the dataset of objects such as phones, computers, and cups, the device can recognize them, making it flexible for low-vision users to find items.

This device has many advantages. Its operation is relatively simple, and the device is easy to put on and take off, making it user-friendly. Secondly, the device's functions can help users avoid obstacles in high places, and the visual detection function can help users understand the scene and make judgments. The possible drawback is that the limited computing power of the Raspberry Pi may result in a longer response time. Also, the Raspberry Pi may incorrectly recognize irrelevant scenes due to hardware limitations such as camera resolution.

Chapter 3: Model Introduction

3.1 Module Introduction

3.1.1 Arduino

Arduino [10] is an open-source electronics platform designed to provide designers, hobbyists, and engineers with an easy way to create interactive electronic projects. It is based on easy-to-use hardware and software and can be used to control and sense a variety of devices and systems in the physical world. The Arduino platform includes both hardware and software components. On the hardware side, it uses open-source circuit boards based on Atmel AVR microcontrollers. It provides a variety of board specifications and shapes that can be selected according to project needs. On the software side, Arduino uses an integrated development environment based on Processing, which includes a compiler, debugger, libraries, and sample programs. There are many advantages to using Arduino: Arduino has open-source code, and its hardware and software are open-source, which allows anyone to use, modify, and distribute them freely, bringing great convenience for innovation and sharing. It supports multiple programming languages; the Arduino IDE supports the C++ programming language and also supports extensions for other programming languages such as Python, Java, and JavaScript. Arduino supports multiple communication protocols such as serial, I2C, SPI, and radio frequencies, making communicating with other devices and systems accessible. It also supports numerous sensors and actuators; Arduino can be connected to various sensors and actuators such as temperature sensors, humidity sensors, LED lights, motors, and more, enabling a wide range of functions.

Figure 10: Arduino [10]

3.1.2 Ultrasonic Sensor

Ultrasonic sensors [11] commonly measure distance and output distance values. Ultrasonic sensors determine distance by emitting and receiving reflected ultrasonic waves, typically consisting of a transmitter and a receiver. The principle of the ultrasonic sensor is to calculate the distance by emitting ultrasonic waves and calculating the time and speed of the reflected waves. The ultrasonic wave signal emitted by the sensor is reflected, and the sensor calculates the round-trip time of the ultrasonic wave signal, which is the time taken for the wave to travel to the object and back. Then, the distance is calculated based on the speed of sound and the round-trip time. The advantages of ultrasonic sensors are that they can be used indoors and outdoors, are not affected by light, and have high accuracy and stability. They are commonly used in robotics, rangefinders, and smart homes for measuring obstacle distance, controlling smart home devices, and measuring liquid levels. On development platforms like Arduino, ultrasonic sensors can be used with digital IO ports to calculate distance based on the measured echo time. The Arduino platform has dedicated library functions to help users efficiently use ultrasonic sensors, such as the NewPing library. Users only need to connect the sensor's power and signal lines and then write the corresponding program to measure distance and control related devices.

There are also some issues with the use of ultrasonic sensors. Firstly, the working distance of the sensor is affected by environmental factors such as air humidity and temperature, which affect the transmission speed and attenuation of ultrasonic waves. Therefore, adjustments and testing should be performed based on the actual environment. The measurement accuracy and stability of the sensor are affected by the quality and performance of the transmitter and receiver. Generally, ultrasonic sensors of better brands and quality have better measurement accuracy and stability. It is essential to pay attention to the connection method of the sensor's power and signal lines, as the sensor typically requires a 3-5V power supply and a digital IO port for signal transmission.

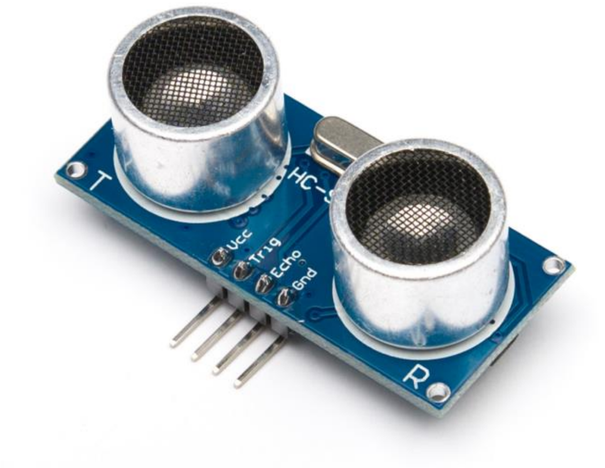


Figure 11: Ultrasonic Sensor [11]

3.1.3 Buzzer

The buzzer [12] is an electronic component designed to produce an audible sound signal. It typically comprises an oscillator and a speaker, with the oscillator playing a crucial role in sound generation. The oscillator, which can be of different types such as electromagnetic and piezoelectric, generates sound waves that are then amplified through the speaker. The buzzer's oscillator usually consists of a core, coil, and diaphragm. When current flows through the coil, the core experiences electromagnetic force, causing it to vibrate and make the diaphragm vibrate, thereby producing sound. Buzzers, such as alarms, gaming consoles, and electronic keyboards, find wide application in various electronic devices and circuits.

Figure 12: Buzzer [12]

3.1.4 Vibration Motor

A Vibration Motor [13] is a small electric motor that can generate vibration force. It usually consists of a motor, an eccentric wheel, a weight, and other components. When the motor is powered on, it drives the eccentric wheel or weight to rotate rapidly, thus producing vibration force. Vibration motors are widely used in electronic devices such as mobile phones, controllers, game consoles, electric toothbrushes, etc. Vibration motors can produce vibration prompts, tactile feedback, and other effects to enhance user interaction with these devices. Vibration motors can implement vibration sensing, alarms, and other functions in robots and other electronic products.

Figure 13: Vibration Motor [13]

3.1.5 Tilt Switch

A Tilt Switch [14] is a type of switch that can sense an object's tilting or tipping state. It typically consists of a pair of metal balls, a metal plate, and a spring. The operating principle of a tilt switch is based on the principle of gravity sensing. When the switch is stable, the metal balls or plate will be separated from the contact point by the spring, breaking the circuit. When the switch is tilted or tipped, the metal balls or plate will meet the contact point, forming a circuit and outputting a signal. Tilt switches are commonly used when it is necessary to detect the tilting state of an object, such as security devices, lighting controls, electronic games, and so on.

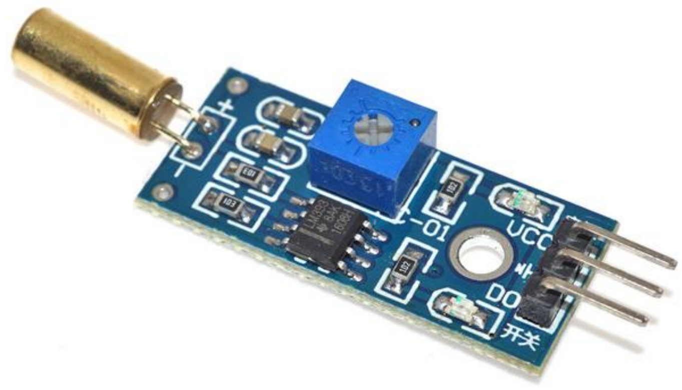


Figure 14: Tilt Switch [14]

3.1.6 Raspberry Pi

Raspberry Pi [15] is a versatile single-board computer that has captured the imagination of hobbyists, students, and professionals since its launch by the Raspberry Pi Foundation in 2012. It is designed to be a low-cost, high-performance small computer, aimed at promoting the popularity of computer science in education and developing countries. Raspberry Pi uses processors based on the ARM architecture and typically runs Linux as its operating system. With a rich set of hardware interfaces and expansion capabilities, such as HDMI, USB, Ethernet, GPIO (General Purpose Input/Output) interfaces, etc., Raspberry Pi can be used in a wide range of applications, from home entertainment to IoT, robotics, and education. Its power, flexibility, and ease of use provide users with a vast creative space to explore and innovate.

The main advantages of Raspberry Pi are as follows: First, it is low-cost. Raspberry Pi is relatively cheap compared to traditional computers, which makes it highly attractive to ordinary consumers, educators, and developing countries. Second, it is easy to use. Raspberry Pi has a rich set of hardware interfaces and an easy-to-use software environment, which enables users to master its use quickly. Third, it has robust expandability. Raspberry Pi has multiple GPIO interfaces, which can connect various expansion boards and sensors, enabling users to apply Raspberry Pi to numerous fields. Fourth, it has low power consumption and can run long without significantly increasing energy costs.

The main disadvantages of Raspberry Pi are as follows: First, its performance is relatively weak compared to traditional computers, with certain limitations. Second, its storage capacity is limited. Raspberry Pi's storage capacity is usually realized through microSD cards with limited storage capacity. Third, its graphics processing capability is weak and may not be suitable for high-performance graphics processing applications. Fourth, its driver and software support are limited. Raspberry Pi's support for some external device drivers and software is relatively limited, which can affect some applications.

Figure 15: Raspberry Pi [15]

3.1.7 Raspberry Pi Camera

The Raspberry Pi Camera [16] is a small camera module based on the Raspberry Pi, which can be used in various Raspberry Pi projects such as smart homes, security monitoring, robots, etc. This camera module can be connected to the Raspberry Pi's CSI interface and is very easy to use.

The main advantages of the Raspberry Pi Camera are high resolution, with the V2 module supporting up to 8 million pixels, allowing for high-quality photos and videos; muscular flexibility, with adjustable focus, allowing for shooting in different scenarios; and affordability, with a very reasonable price compared to other camera modules, making it suitable for Raspberry Pi enthusiasts and manufacturers.

However, this tiny camera module also has some disadvantages. Firstly, it does not have optical zoom. Although adjusting the focus allows for capturing different types of shots, they cannot be automatically adjusted, which may have specific limitations when shooting scenes that require zooming. Secondly, the lens of the Raspberry Pi Camera module is fixed and cannot be replaced or upgraded, so other camera modules may need to be used in scenes that require specific focal lengths or views. In addition, this camera module has an unavoidable delay. Due to the use of the CSI interface to connect to the Raspberry Pi, there may be an inevitable delay when transmitting images and videos, impacting some application scenarios that require real-time response. Finally, this module is sensitive to environmental light. The shooting effect of the Raspberry Pi Camera module is greatly affected by the ambient light and needs to be used under sufficient light conditions. Otherwise, blur or overexposure may occur.

Figure 16: Raspberry Pi Camera [16]

3.1.8 Bone-Conducted Earphone

Bone-conducted earphones [17] are unique headphones that differ from traditional earbud-style and over-ear headphones. Instead of directly placing a speaker in or over the ear, they transmit sound by vibrating against the bones in the skull. This means the ears are not directly involved in receiving sound; the sound is transmitted through the skull bones. Users of bone conduction headphones can hear ambient sounds around them because they do not block the ear canal but rather transmit sound through vibrations. This makes bone-conducted headphones particularly useful for outdoor activities or situations where awareness of the surrounding environment is essential. They can also provide an alternative for people with hearing difficulties, as they do not rely on traditional ear-based hearing. Bone-conduction headphones are used in outdoor sports, workplaces, and healthcare fields. Their main advantages include being waterproof, breathable, not requiring direct insertion into the ear, and preventing long-term damage to the ear from prolonged use. However, compared to traditional headphones, their sound quality may not be as clear, and they can be more expensive.

The main advantage of bone-conducted earphones is that they preserve ambient sound, allowing users to remain aware of their surroundings. As bone conduction headphones do not need to be inserted into the ear, they enable users to maintain situational awareness, which is helpful for outdoor activities and workplaces where environmental awareness is essential. They are also breathable, unlike traditional earbuds and over-ear headphones, which can block the ear canal and cause discomfort or fatigue. Additionally, they can be helpful for people with hearing problems, as they do not rely on traditional ear-based hearing. Finally, bone conduction headphones are often waterproof due to their design, allowing them to be used in underwater or wet environments.

However, bone-conduction headphones also have some drawbacks. Their sound quality may not be as transparent as traditional headphones due to their method of operation, resulting in inferior sound quality compared to in-ear or over-ear headphones. Additionally, they can be more expensive than conventional headphones.

Users may also require some time to adapt to the auditory experience bone conduction headphones provide. Finally, if the user has a lot of hair or wears a hat, it may affect the effectiveness of bone-conduction headphones, which rely on vibration to transmit sound.

Figure 17: Bone Conducted Earphone [17]

3.2 Final Design

3.2.1 Smart Cane

The Smart Cane detects obstacles overhead; it has many components like an Arduino, ultrasonic sensor, buzzer, LED light, button, tilt switch, and vibration motor.

A smart cane is similar to the traditional white cane; the user uses it when walking. The ultrasonic sensor mainly realizes the detection process. The ultrasonic sensor consists of a transmitter and a receiver. The transmitter emits ultrasonic waves. If the obstacles are ahead of the user, the ultrasonic waves will hit the object and bounce back. The receiver will detect the waves. The receiver then measures the time it takes for the waves to travel back, which can be used to calculate the distance between the sensor and the object. If the distance is below 30 centimeters, the sensor will transform a signal to Arduino. The Arduino will control the LED light, buzzer, and vibration motor work to inform the user of the obstacles.

Another function of this device is the one-button alarm. This means once the user presses the button, the vibration motor, buzzer, and LED light will work to inform the nearby people of the emergency condition. The third function is falling detection. The device is equipped with a tilt switch, and this sensor can detect the angle change of the smart cane. If the user falls, the cane will also fall down. When the cane falls, the tilt switch will detect the angle change; in this way, the vibration motor, buzzer, and LED light will work to inform the nearby people of the emergency. Please see the appendix for the detailed code for the smart cane. The simple principle diagram and the intelligent cane's Arduino code flow chart are here.

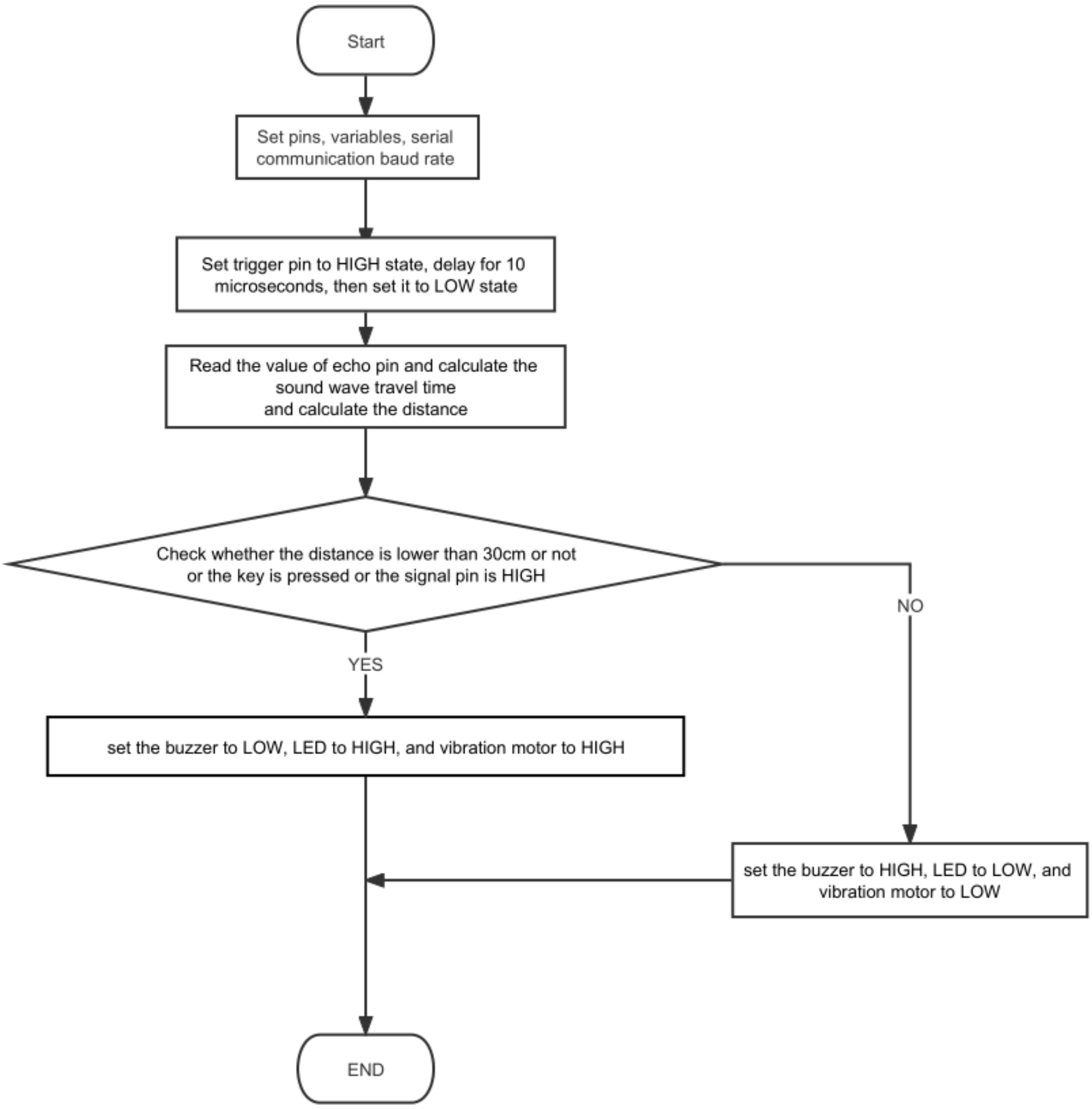


Figure 18: Flow chart of Arduino code

Figure 19: Four views of Smart Cane

On the handle of the intelligent cane, I installed LED lights, a vibration motor, and an alarm switch. When the user holds the cane with their right hand, their thumb will directly face the alarm switch, and the vibration motor will be in the palm position. This makes it very convenient for users to interact with the cane's information.

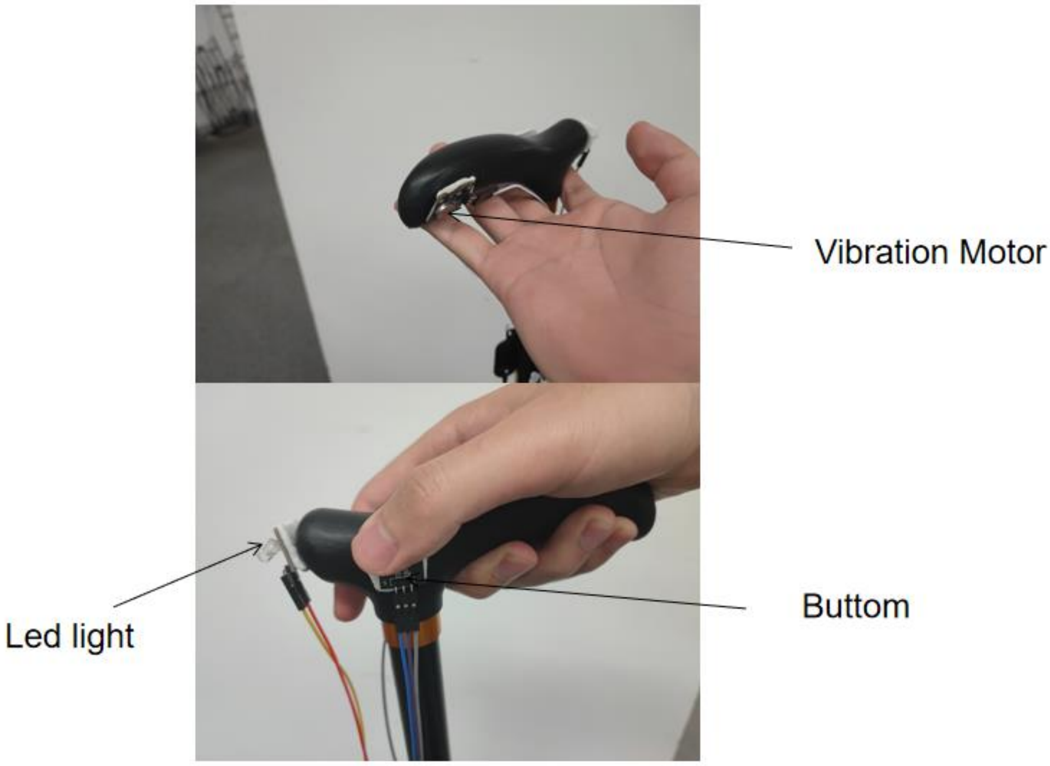


Figure 20: Handle of Smart Cane

As shown in the figure, I have installed an Arduino, battery holder, ultrasonic sensor, buzzer, and tilt switch on the cane shaft.

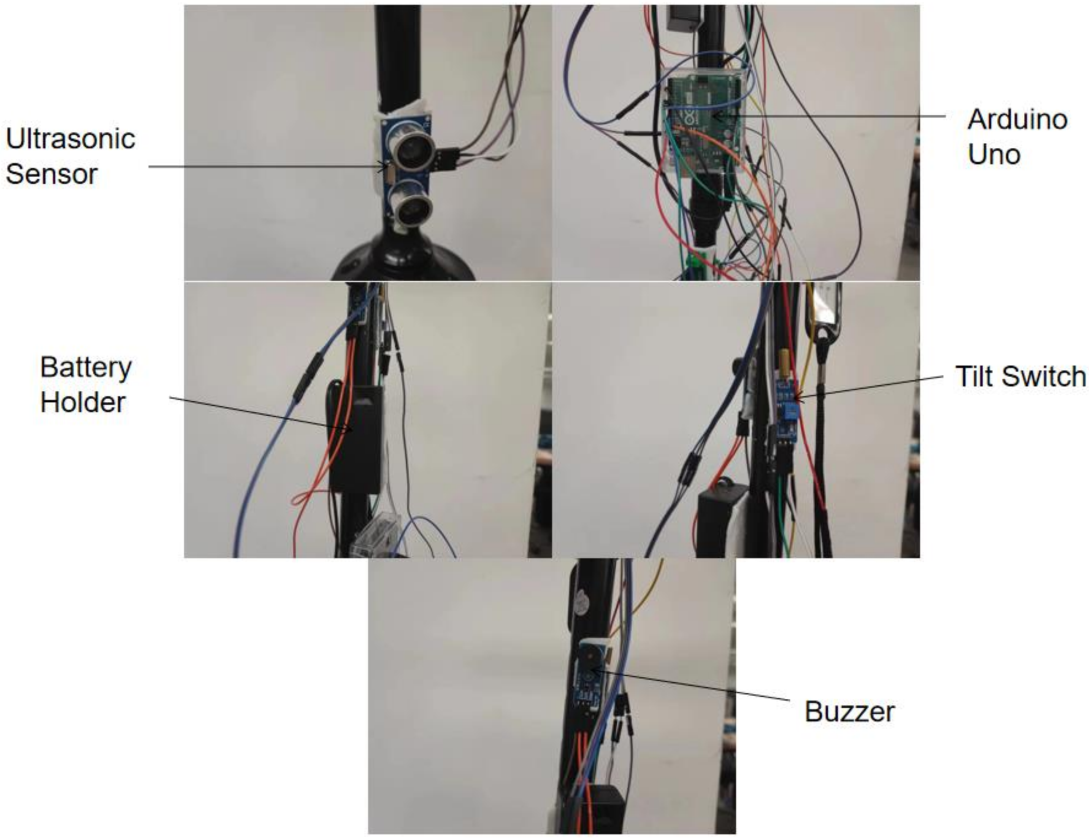


Figure 21: Some Sensors in Smart Cane

3.2.2 Object Identification Device

The object identification device is used to detect nearby scenes. A Raspberry Pi camera and an ultrasonic sensor are placed on the user's head. The ultrasonic sensor and the buzzer realize the function of detecting obstacles from higher positions. The process to achieve this is similar to an intelligent cane's obstacle detection. The ultrasonic sensor's transmitter emits waves; if the waves hit the object and bounce back, they will be detected by the receiver. The receiver then measures the time it takes for the waves to travel back, which can be used to calculate the distance between the sensor and the object. If the distance is below 50 centimeters, the sensor will transform a signal to Raspberry Pi. The Raspberry Pi will control the buzzer to make a noise, reminding the user of the obstacles from a higher position. The graph shows the flow chart of the code of obstacle detection in Raspberry Pi. Please see the appendix for the complete code of this process.

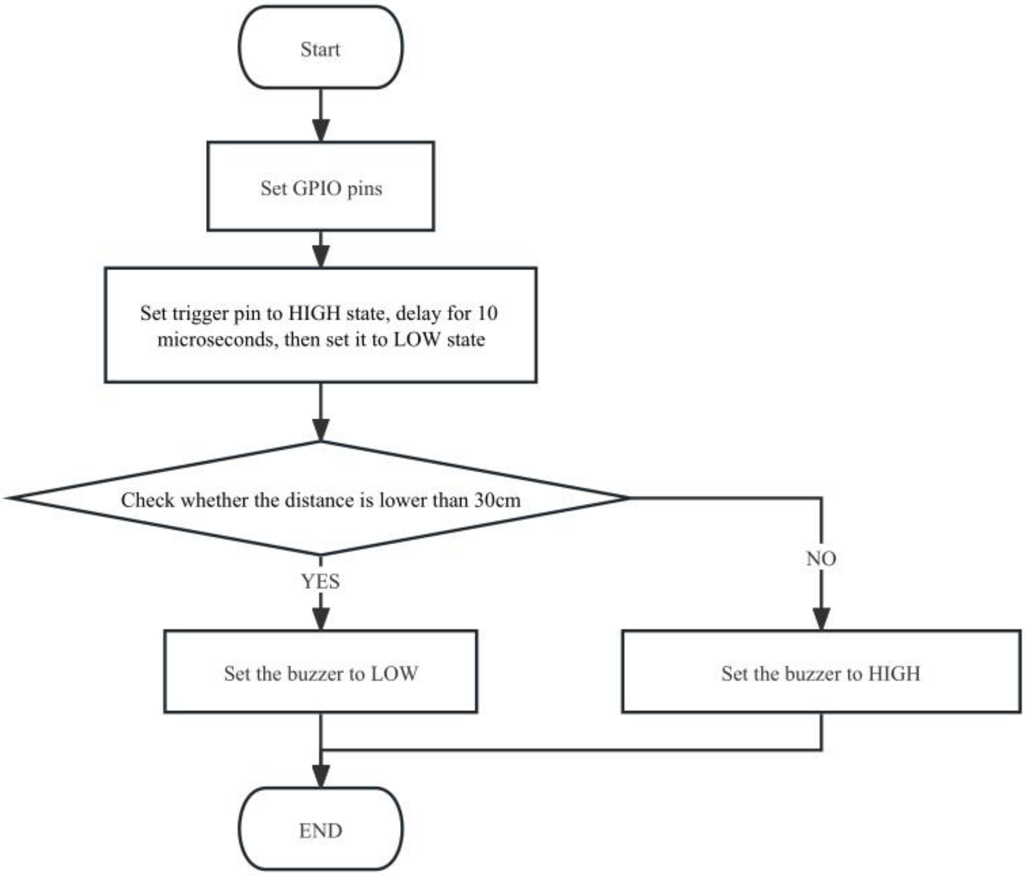


Figure 22: The flow chart of obstacle detection

The camera of the Raspberry Pi is used to capture images of the surrounding environment. OpenCV is pre-installed on the Raspberry Pi, and a model for detecting steps, traffic lights, and pedestrian crossings has been trained on a computer beforehand. A bone-conducting Bluetooth headset is connected to the Raspberry Pi. When the Raspberry Pi detects the corresponding images, it will control the Bluetooth headset to play the corresponding audio, alerting the user to their surroundings. For example, when a user walks on the street, and the Raspberry Pi camera detects a pedestrian crossing, a pre-recorded audio prompt will be played through the headset, indicating that the user is about to cross the pedestrian crossing. The device also detects traffic lights, steps, and other information. In addition, I discovered a new use for the object detection function: the object-finding function that users can use to find objects indoors, such as cups or laptops. The device's appearance, the diagram of the debugging process on the computer, and the flowchart of the object recognition process are presented. Please refer to the appendix for the complete code and Raspberry Pi configuration process.

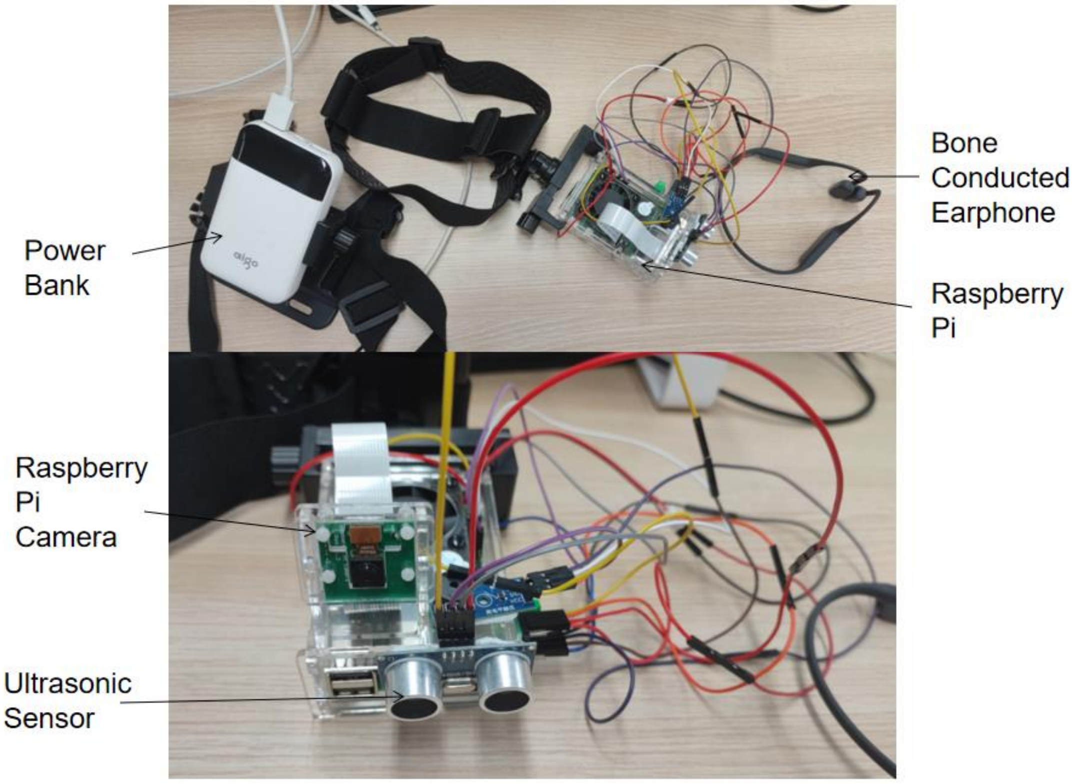


Figure 23: Object Identification Device

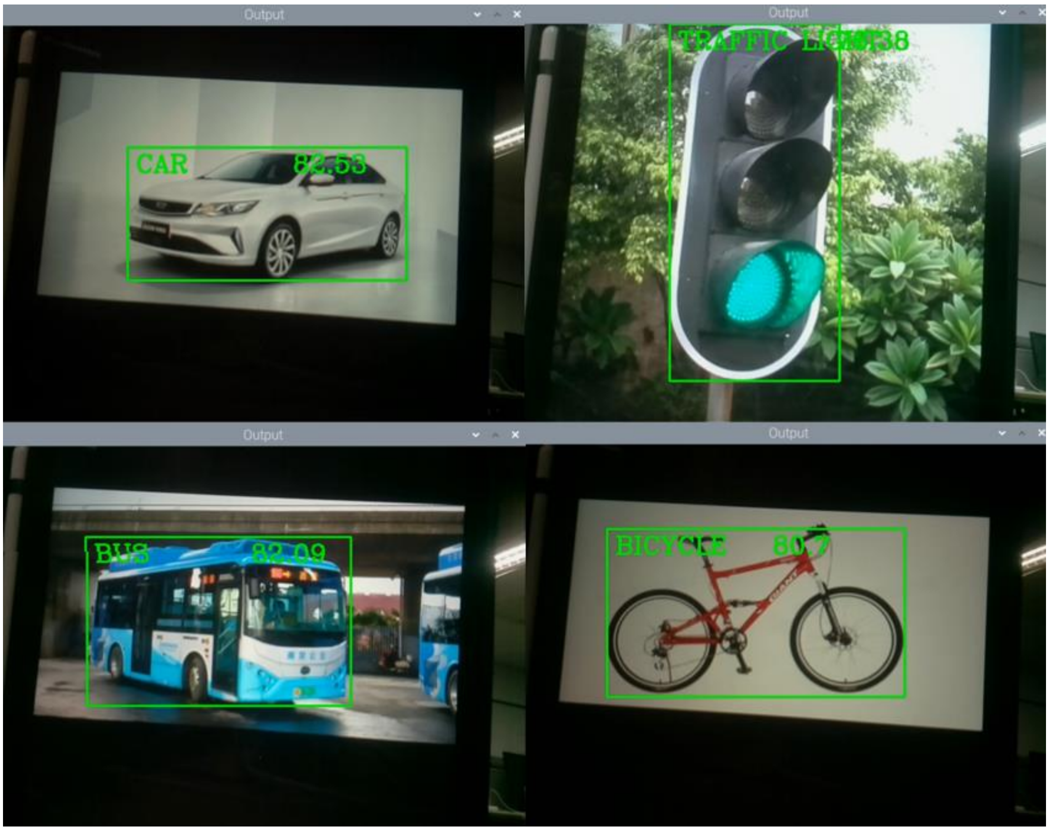


Figure 24: Object Identification on Roads

Figure 25: Object Identification indoors

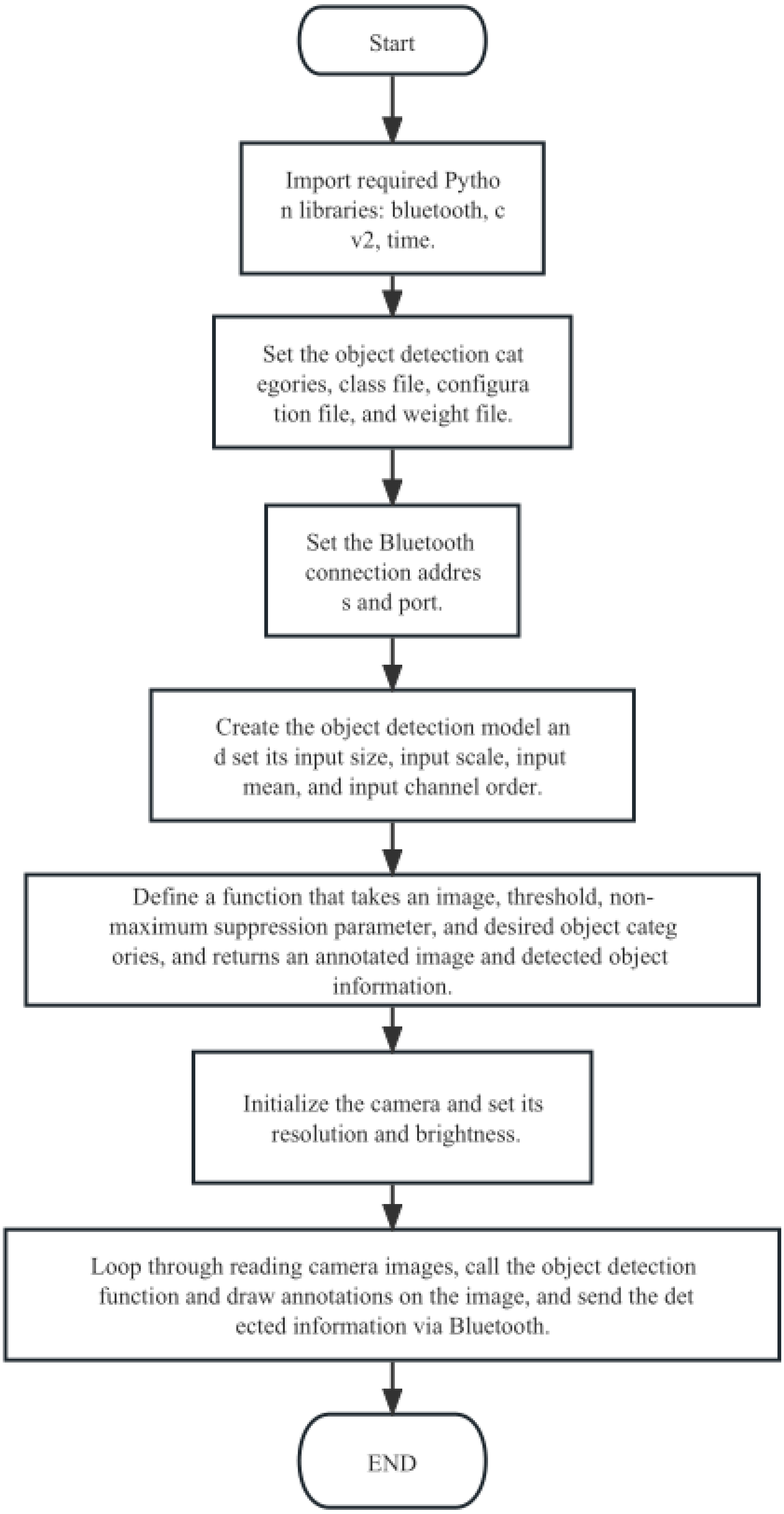


Figure 26: Flow chart of the object identification process

3.2.3 The Connection with Phone

To facilitate the control of the Raspberry Pi and enable the device user's relatives to understand the device status and the user's situation, I have also designed a feature for the interaction between the Raspberry Pi and the smartphone. By connecting the Raspberry Pi and the smartphone through the RaspController software, the user can view the temperature, memory usage, and files in the Raspberry Pi through the smartphone. In addition, the user can also control the Raspberry Pi to run programs through the SSH terminal and call the Raspberry Pi camera to take pictures to view the user's surroundings. The detailed process of connecting the Raspberry Pi and the smartphone is shown in the appendix, and the following figures show some windows displayed after the connection.

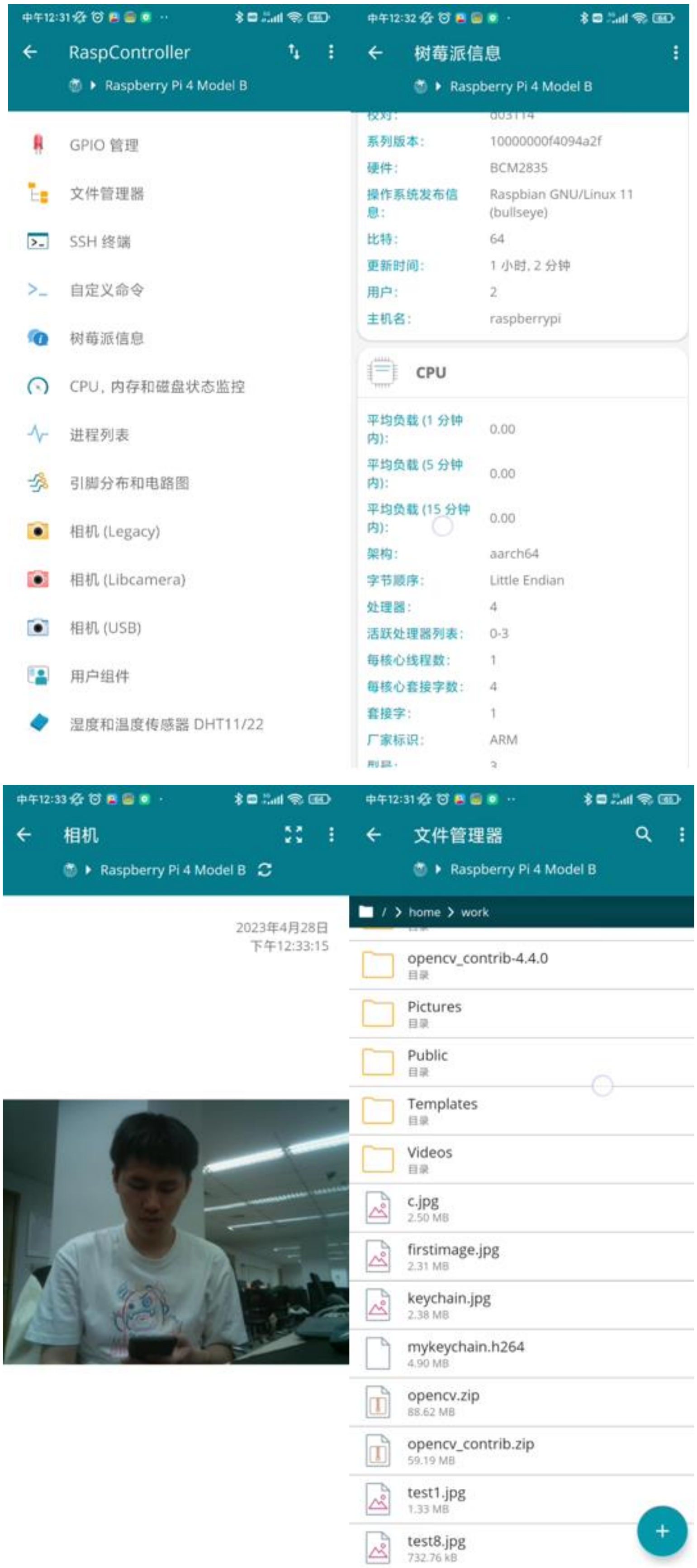


Figure 27: User Interface of the Mobile App

3.3 Cost of two Devices

Name of Component	Quantity Required	Unit Cost (¥)	Total Cost (¥)
Arduino UNO	1	213	213
Ultrasonic Sensor	1	6	6
Buzzer	1	2	2
Vibration Motor	1	3	3
Led Light	1	2	2
Wires	16	0.2	3.2
Battery Holder	1	4	4
Wire Transfer Plate	2	3	6
Cane	1	35	35
Estimated Total Cost (¥)			274.2

Table 1: Cost of Smart Cane

Name of Component	Quantity Required	Unit Cost (¥)	Total Cost (¥)
Raspberry Pi 4B	1	1099	1099
Ultrasonic Sensor	1	6	6
Buzzer	1	2	2
Power Bank	1	99	99
Raspberry Pi Camera	1	35	35
Device Holder	1	49	49
Bone Conduction Earphone	1	619	619
Raspberry Pi Shell	1	35	35
Wires	10	0.2	2
Wire Transfer Plate	1	3	3
Estimated Total Cost (¥)			1949

Table 2: Cost of Object Identification Device

As shown in the tables, the smart cane is cheaper, and by using unofficial channels for Arduino boards, the price of the cane can be further reduced to less than ¥100. Object Identification Device is a little bit expensive because of the rising cost of Raspberry Pi.

Chapter 4 Prototype Testing and Design Evaluation

4.1 Field Test

To test whether the smart cane and head-mounted object detection device can be used in real-world scenarios to provide users with correct direction, I recruited six volunteers to participate in the device testing. To better simulate real-life usage scenarios, I selected six different scenes. All the testers were physically healthy but had their eyes covered during the tests. They were all first-time users of the devices and received some instructions from me before the obstacle avoidance and scene recognition tests. After the test, I asked them to complete a questionnaire to provide feedback on their experience. The image below shows some of the testing scenes.

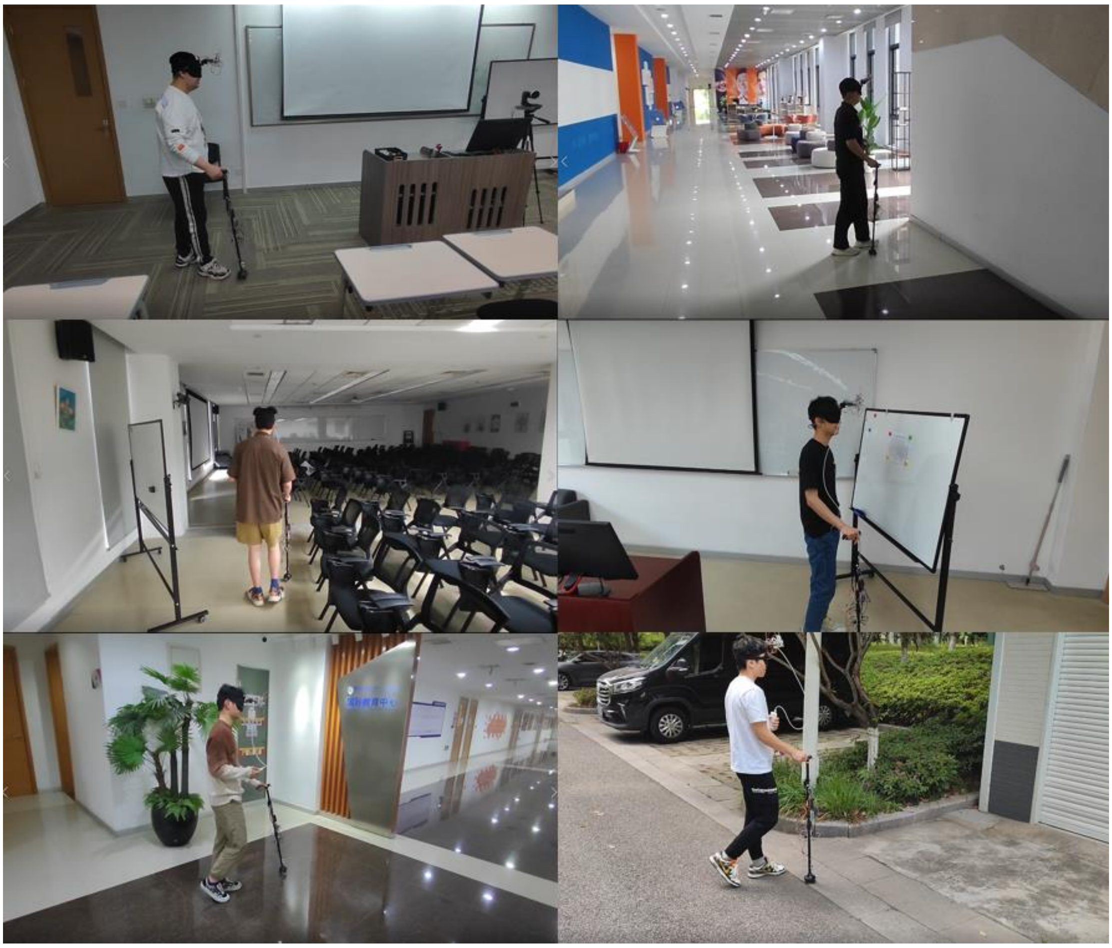
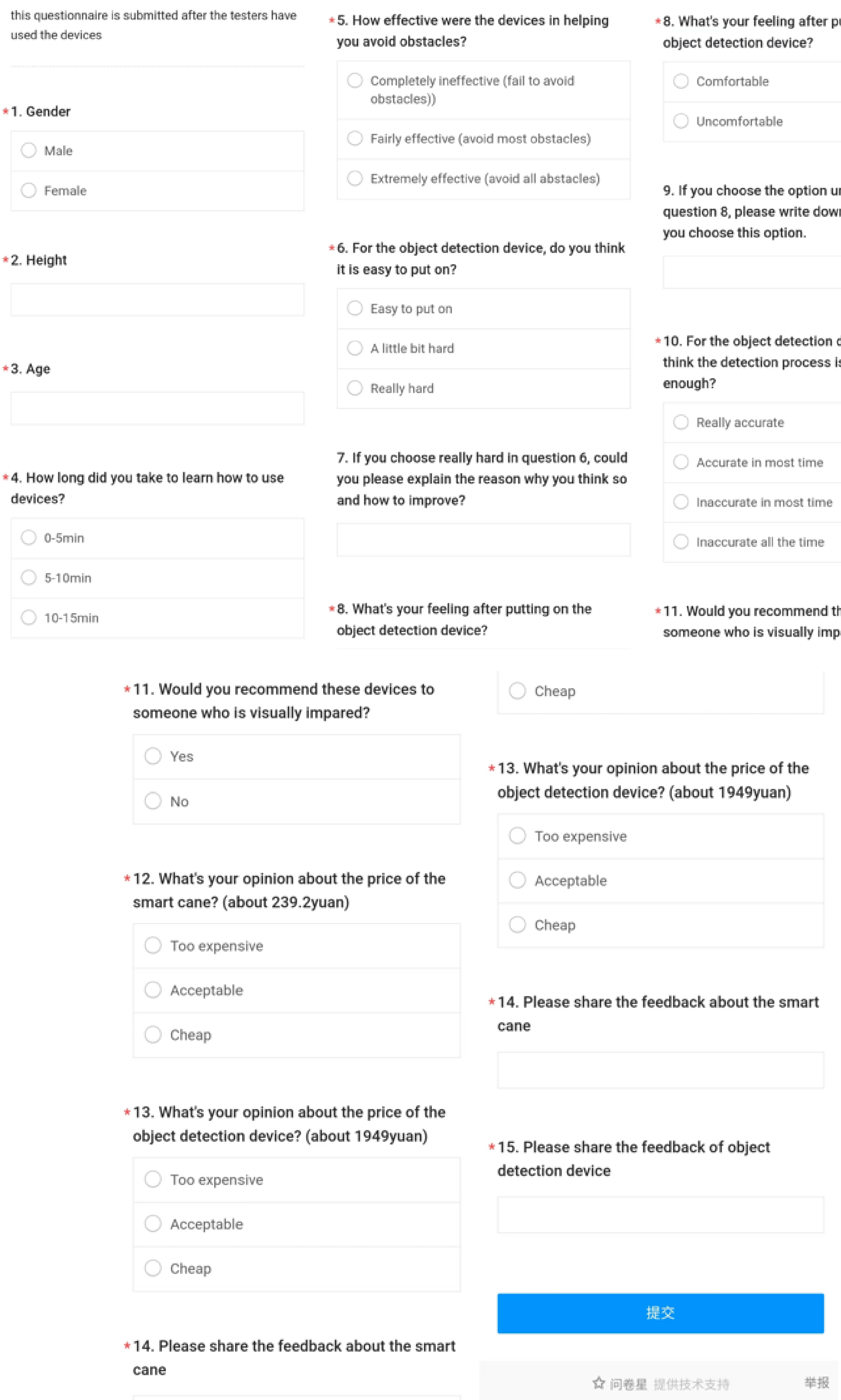


Figure 28: Field Test

4.2 User Questionnaire

To obtain the user's true feelings, I designed a survey questionnaire. The questionnaire consists of 15 questions that explore the device's reliability, convenience, and comfort. The detailed contents of the questionnaire are shown in the figure.



this questionnaire is submitted after the testers have used the devices

*1. Gender
- Male
- Female

*2. Height

*3. Age

*4. How long did you take to learn how to use devices?
- 0-5min
- 5-10min
- 10-15min

*5. How effective were the devices in helping you avoid obstacles?
- Completely ineffective (fail to avoid obstacles))
- Fairly effective (avoid most obstacles)
- Extremely effective (avoid all abstacles)

*6. For the object detection device, do you think it is easy to put on?
- Easy to put on
- A little bit hard
- Really hard

7. If you choose really hard in question 6, could you please explain the reason why you think so and how to improve?

*8. What's your feeling after putting on the object detection device?

*8. What's your feeling after putting on the object detection device?
- Comfortable
- Uncomfortable

9. If you choose the option uncomfortable in question 8, please write down the reason why you choose this option.

*10. For the object detection device, do you think the detection process is accurate enough?
- Really accurate
- Accurate in most time
- Inaccurate in most time
- Inaccurate all the time

*11. Would you recommend these devices to someone who is visually impared?

*11. Would you recommend these devices to someone who is visually impared?
- Yes
- No

*12. What's your opinion about the price of the smart cane? (about 239.2yuan)
- Too expensive
- Acceptable
- Cheap

*13. What's your opinion about the price of the object detection device? (about 1949yuan)
- Too expensive
- Acceptable
- Cheap

*14. Please share the feedback about the smart cane

- Cheap

*13. What's your opinion about the price of the object detection device? (about 1949yuan)
- Too expensive
- Acceptable
- Cheap

*14. Please share the feedback about the smart cane

*15. Please share the feedback of object detection device

提交

☆ 问卷星 提供技术支持 举报

Figure 29: Questionnaire

4.3 Questionnaire Analysis

Through the analysis of the received questionnaires, I have identified some issues that still exist with the device. The complete questionnaire analysis has been included in the appendix.

6. For the object detection device, do you think it is easy to put on?

Options	Quantity	Ratio
Easy to put on	3	50%
A little bit hard	3	50%
Really hard	0	0%
Number of valid responses for this question	6	

Figure 30: Question 6

As shown in the figure, half of the testers think the object identification device is hard to put on. So, I think more work should be done in this field.

8.What's your feeling after putting on the object detection device?

Options	Quantity	Ratio
Comfortable	3	50%
Uncomfortable	3	50%
Number of valid responses for this question	6	

Figure 31: Question 8

As shown in the figure, half of the testers think the object detection identification device is uncomfortable to put on. And the reason why they choose this option is that the device is too heavy.

13. What's your opinion about the price of the object detection device? (about 1949yuan)

Options	Quantity	Ratio
Too expensive	3	50%
Acceptable	3	50%
Cheap	0	0%
Number of valid responses for this question	6	

Figure 32: Question 13

As shown in the figure, half of the testers think about the price of the object identification device. I will try to use some cheaper components to reach the same function.

10. For the object detection device, do you think the detection process is accurate enough?

Options	Quantity	Ratio
Really accurate	2	33.33%
Accurate in most time	3	50%
Inaccurate in most time	1	16.67%
Inaccurate all the time	0	0%
Number of valid responses for this question	6	

Figure 33: Question 10

As shown in the figure, the object identification device's accuracy should also be improved.

Chapter 5: Future Recommendations

During the questionnaire stage, some of the testers found the device uncomfortable to wear, primarily due to the weight of the head-worn device. To address this issue, I plan to move the Raspberry Pi from the head to the vest, keeping only the camera and ultrasonic sensor on the head-worn device, significantly reducing the weight.

In addition, some testers found that the object identification device's accuracy and feedback time needed improvement. To address this issue, I plan to optimize the

Raspberry Pi for a higher-performance, small-sized processor, improve the camera's accuracy, and optimize the recognition algorithm.

During usage, I also found that the wires used to connect the sensors are prone to tangling and can quickly come loose, posing a safety risk, as shown in the figure. To address this issue, I plan to fix the wires and design a casing to enclose all the wire heads.

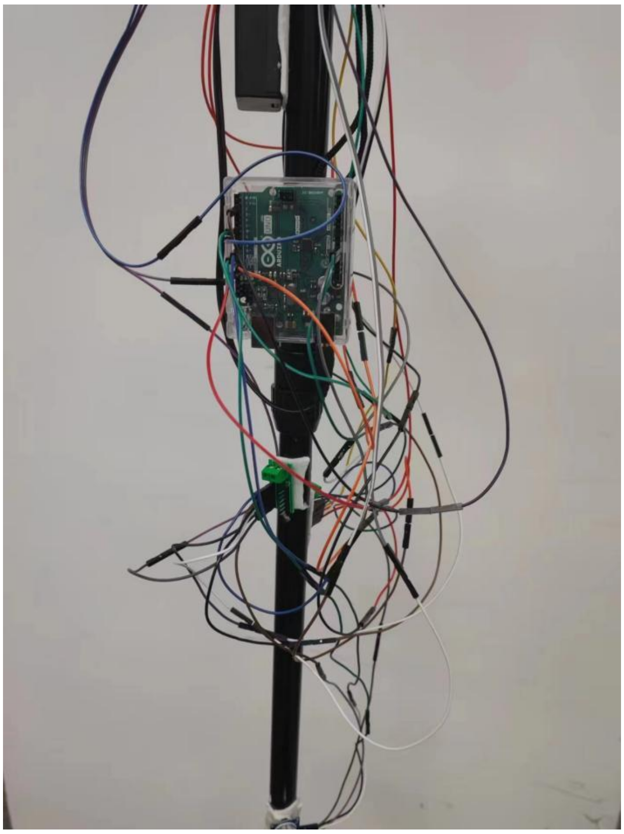


Figure 34: The Tangled Wires

Chapter 6: Conclusion

To address the mobility issues of low vision users, I designed this smart cane and object recognition device that, when used together, can detect high and low obstacles, and assist with environmental recognition and locating objects, significantly improving the quality of life for those with low vision. Currently, the affordable devices available for blind people mainly consist of white canes, which cannot detect high and low obstacles. Very few wearable devices can also recognize objects and environments, and this device will fill that gap. Compared to similar devices, this set of equipment has a lower cost and more function, making it highly competitive if brought to the market.

References

[1] World Health Organization. (2022). Visual impairment and blindness: Key facts. From https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] Hart, L. A. et al. (2018). Assistance dog placement in society: A review of the evidence. Frontiers in Veterinary Science, 5, 1-12.

[3] Aydogdu, M. et al. (2021). Investigating the effectiveness of a smart cane in daily life of people with visual impairments. International Journal of Industrial Ergonomics, 84, 103103.

[4] Borges, R. et al. (2019). OrCam MyEye device: usability, satisfaction and potential impact on quality of life of Brazilian individuals with visual impairment. International Journal of Technology and Inclusive Education, 1(2), 91-98.

[5] iPhone/iPad: Low Vision Features Series. From https://hadley.edu/workshops/iphone-ipad-low-vision-features-series

[6] Fiksdal, T. et al. (2017). Comparison of wayfinding devices for people who are blind or visually impaired: Google Maps, Trekker Breeze, and Sendero GPS. Journal of Visual Impairment & Blindness, 111(3), 219-227.

[7] HumanWare Trekker Breeze+ Handheld Talking GPS. From https://www.corada.com/products/humanware-trekker-breeze-handheld-talking-gps

[8] Ballesta, M. et al. (2020). Using BlindSquare and iBeacon technologies to enhance accessibility for people with visual impairment in museums. Sensors, 20(20), 1-18.

[9] Buzzi, M. et al. (2020). Accessibility evaluation of Seeing AI for blind and visually impaired users. Computers in Human Behavior, 103, 198-207.

[10] Schwartz, M. (2017). Arduino for Beginners: Essential Skills Every Maker Needs, 2nd Edition. San Francisco, CA: Maker Media.

[11] Li, J. et al. (2018). Application of Ultrasonic Sensors in Medicine: A Review. IEEE Sensors Journal, 18(23), 9601-9611.

[12] Huang, H. et al. (2020). Design and implementation of an automatic irrigation system based on soil moisture sensors and buzzer alarm. Journal of Physics: Conference Series, 1461(1), 012091.

[13] Liu, H. et al. (2019). Design and evaluation of a novel dual-mode ultrasonic-vibration motor for haptic feedback. IEEE Transactions on Industrial Electronics, 66(2), 903-911.

[14] Liu, X., Zhu, Y., Wang, J., & Guo, Y. (2019). Design and Analysis of a High-Performance Tilt Switch Based on MEMS Technology. Micromachines, 10(2), 107.

[15] Shovic, J. (2019). Raspberry Pi Electronics Projects for the Evil Genius: Control Your World with Simple Circuit Programs. McGraw-Hill Education.

[16] Li, R., Li, H., & Gao, R. (2018). Raspberry Pi Camera-Based Real-Time Motion

Detection System. Journal of Sensors, 2018, 7293084.
[17] Bone conduction. From https://en.wikipedia.org/wiki/Bone_conduction

Appendix

Appendix.1 Codes

Appendix.1.1 Arduino codes of Smart Cane

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 13;
const int keyPin = 7;
const int sigPin = 6;
const int vibPin = 2;
// defines variables
long duration;
int distance;
int safetyDistance;
boolean sigState = 0;
void setup() {
pinMode(keyPin, INPUT);
pinMode(sigPin, INPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);
pinMode(vibPin, OUTPUT);
Serial.begin(9600);
}
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
```

```
distance= duration*0.034/2;
safetyDistance = distance;
boolean Value=digitalRead(keyPin);
sigState = digitalRead(sigPin);
Serial.println(sigState);
if (safetyDistance <= 30||Value ==LOW||sigState == HIGH){
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, HIGH);
  digitalWrite(vibPin, HIGH);
}
else{
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, LOW);
  digitalWrite(vibPin, LOW);
}
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

Appendix.1.2 Codes of Obstacles Detection of Raspberry Pi

```
import RPi.GPIO as GPIO
import time

# set the GPIO pins
GPIO_TRIGGER = 23
GPIO_ECHO = 24
BUZZER_PIN = 18

GPIO.setmode(GPIO.BCM)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
GPIO.setup(BUZZER_PIN, GPIO.OUT)

def distance():
    # Send ultrasonic signal
    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
    # Receive reflected signal
    start_time = time.time()
    stop_time = time.time()
    while GPIO.input(GPIO_ECHO) == 0:
        start_time = time.time()
```

```

    while GPIO.input(GPIO_ECHO) == 1:
        stop_time = time.time()

    # calculate distance
    elapsed_time = stop_time - start_time
    distance = (elapsed_time * 34300) / 2
    return distance
try:
    while True:
        dist = distance()
        print("Distance: %.2f cm" % dist)
        if dist < 30:
            GPIO.output(BUZZER_PIN, True)
        else:
            GPIO.output(BUZZER_PIN, False)
        time.sleep(0.1)
except KeyboardInterrupt:
GPIO.cleanup()
```

Appendix.1.3 Codes of object identification process of Raspberry Pi

```
import bluetooth
import cv2
import time
#thres = 0.45 # Threshold to detect object
classNames = []
classFile = "/home/work/Desktop/Object_Detection_Files/coco.names"
with open(classFile,"rt") as f:
classNames = f.read().rstrip("\n").split("\n")
configPath = "/home/work/Desktop/Object_Detection_Files/ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt"
weightsPath = "/home/work/Desktop/Object_Detection_Files/frozen_inference_graph.pb"
net = cv2.dnn_DetectionModel(weightsPath,configPath)
net.setInputSize(320,320)
net.setInputScale(1.0/ 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)

#connect bluetooth
bd_addr = "E0:08:71:52:26:7F"
port = 1
sock = bluetooth.BluetoothSocket( bluetooth.RFCOMM )
sock.connect((bd_addr,port))
```

```
def getObjects(img, thres, nms, draw=True, objects=[]):
    classIds, confs, bbox = net.detect(img,confThreshold=thres,nmsThreshold=nms)
    #print(classIds,bbox)
    if len(objects) == 0: objects = classNames
    objectInfo =[]
    if len(classIds) != 0:
        for classId, confidence,box in zip(classIds.flatten(),confs.flatten(),bbox):
            className = classNames[classId - 1]
            if className in objects:
                objectInfo.append([box,className])
                if (draw):
cv2.rectangle(img,box,color=(0,255,0),thickness=2)            cv2.putText(img,classNames[classId-1].upper(),(box[0]+10,box[1]+30),
cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)            cv2.putText(img,str(round(confidence*100,2)),(box[0]+200,box[1]+30),
      cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
#send the information to Bluetooth
message = classNames[classId-1].upper() + " -Confidence: " + str(round(confidence*100,2)) + "%"
sock.send(message)
return img,objectInfo
if __name__ == "__main__":
    cap = cv2.VideoCapture(0)
    cap.set(3,640)
    cap.set(4,480)
    cap.set(10,70)
    while True:
        success, img = cap.read()
        result, objectInfo = getObjects(img,0.45,0.2, True, ["person"] )
        #print(objectInfo)
        cv2.imshow("Output",img)
           break
    cap.release()
    cv2.destroyAllWindows()
    sock.close()
```

Appendix.2 Detailed Survey Result

1. Gender

Options	Quantity	Ratio

Male	6	100%
Female	0	0%
Number of valid responses for this question	6	

2. Height

180cm 176cm 170cm 186cm 179cm 172cm

3. Age

22 22 21 22 22 22

4. How long did you take to learn how to use devices?

Options	Quantity	Ratio
0-5min	6	100%
5-10min	0	0%
10-15min	0	0%
Number of valid responses for this question	6	

5. How effective were the devices in helping you avoid obstacles?

Options	Quantity	Ratio
Completely ineffective (fail to avoid obstacles))	0	0%
Fairly effective (avoid most obstacles)	0	0%
Extremely effective (avoid all abstacles)	6	100%
Number of valid responses for this question	6	

6. For the object detection device, do you think it is easy to put on?

Options	Quantity	Ratio
Easy to put on	3	50%
A little bit hard	3	50%
Really hard	0	0%
Number of valid responses for this question	6	

7. If you choose really hard in question 6, could you please explain the reason why you think so and how to improve?

Nobody chooses this option.

8. What's your feeling after putting on the object detection device?

Options	Quantity	Ratio
Comfortable	3	50%
Uncomfortable	3	50%
Number of valid responses for this question	6	

9. If you choose the option uncomfortable in question 8, please write down the reason why you choose this option.
(1) Too heavy. (2) Weight (3)The device is too heavy
10. For the object detection device, do you think the detection process is accurate enough?

Options	Quantity	Ratio
Really accurate	2	33.33%
Accurate in most time	3	50%
Inaccurate in most time	1	16.67%
Inaccurate all the time	0	0%
Number of valid responses for this question	6	

11. Would you recommend these devices to someone who is visually impared?

Options	Quantity	Ratio
Yes	5	83.33%
No	1	16.67%
Number of valid responses for this question	6	

12. What's your opinion about the price of the smart cane? (about 239.2yuan)

Options	Quantity	Ratio
Too expensive	0	0%
Acceptable	5	83.33%
Cheap	1	16.67%
Number of valid responses for this question	6	

13. What's your opinion about the price of the object detection device? (about 1949yuan)

Options	Quantity	Ratio
Too expensive	3	50%
Acceptable	3	50%
Cheap	0	0%
Number of valid responses for this question	6	

14. Please share the feedback about the smart cane

(1) sensitive，a good product; (2)too heavy; (3)not; (4)Very handy, more like a cane with alarm to me; (5)comfortable and effective; (6)Good.

15. Please share the feedback of object detection device

(1) too heavy to wear; (2)too big; (3)no; (4)Works fine when meeting obstacles; (5)easy to wear but maybe can be lighter; (6)Good.

Appendix.3 The detailed configuration process of Raspberry Pi

Appendix.3.1 Flashing the Raspberry Pi OS

After installing the Raspberry Pi Imager software on the computer, this software will help flash the Raspberry Pi OS. And insert your microSD card into your computer's card reader. Ensure the microSD card is at least 8GB and has been formatted to FAT32.

Once the burning process is complete, eject the microSD card from your computer and insert it into your Raspberry Pi. Power on your Raspberry Pi and wait for the system to boot up. Depending on the specific image you installed, you may need to configure the system settings.

Once the system is running, you can begin using your Raspberry Pi as intended.

Figure 35: Raspberry Pi Imager

There are precautions in this process. In the version before 2020, Raspberry Pi's account will be raspberry, and the password will be Pi automatically. However, in a version later than 2020, we should set the account and password in this process. Without this process, the Raspberry Pi will be out of work.

Appendix.3.2 Install OpenCV in raspberry pi

Open the Terminal by pressing the Terminal Button found on the top left of the button. Copy and paste each command into your Pi's terminal, press Enter, and allow it to finish before moving on to the next command. If prompted, "Do you want to continue? (y/n)" press Y and then the Enter key to continue the process.

sudo apt-get update && sudo apt-get upgrade

sudo nano /etc/dphys-swapfile

Then change the number on CONF_SWAPSIZE = 100 to CONF_SWAPSIZE=2048. Having done this press Ctrl-X, Y, and then Enter Key to save these changes. This change is only temporary, and you should change it back after completing this. To have these changes affect anything, we must restart the swapfile by entering the following command into the terminal. Then we will resume Terminal Commands as normal.

```
1. sudo apt-get install build-essential cmake pkg-config
2. sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-de
   v
3. sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv
   4l-dev
4. sudo apt-get install libxvidcore-dev libx264-dev
5. sudo apt-get install libgtk2.0-dev libgtk-3-dev
6. sudo apt-get install libatlas-base-dev gfortran
7. sudo pip3 install numpy
```

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.4.0.zip
Wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.4.0.zip
unzip opencv.zip
unzip opencv_contrib.zip
cd ~/opencv-4.4.0/
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-4.4.0/modules \
-D BUILD_EXAMPLES=ON ..
make -j $(nproc)
```

This command will take over an hour to install, and there will be no indication of how long it will take. Once complete we will resume terminal commands.

sudo make install && sudo ldconfig

sudo reboot

At this point the majority of the installation process is complete, and you can now change back the Swapfile so that the CONF_SWAPSIZE = 100.

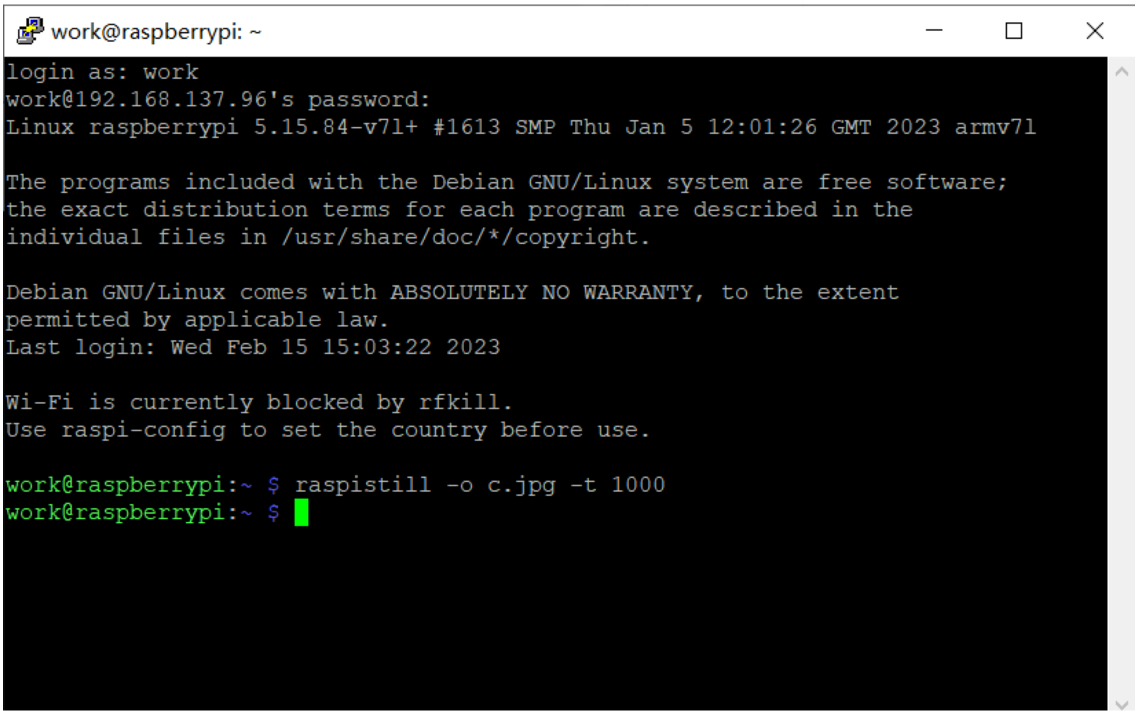


Figure 36: the Operation window of Raspberry Pi

Appendix.4 Detailed process for connecting phone to a Raspberry Pi

Download the RaspController app from the Google Play Store or the Apple App Store and install it on your mobile device.On your Raspberry Pi, ensure you have enabled SSH access and installed any necessary software such as Apache, Samba, or FTP. You may also need to configure your network settings to allow remote access.

Open the RaspController app on your mobile device and click the "+" button to create a new connection. Enter the IP address of your Raspberry Pi and choose the connection type, such as SSH. If you connect via SSH, enter your Raspberry Pi username and password. Once connected to your Raspberry Pi, you can access various controls and functions through the app. For example, you can execute terminal commands, view system information, manage files, and control GPIO pins. To add more functionality to RaspController, you can install plugins from within the app. For example, you can install a plugin to control your Raspberry Pi camera or to monitor system temperature.

And in this process, you need to gain the IP address of Raspberry Pi. You can reach this aim through this process. Make sure your Raspberry Pi and smartphone are connected to the same network.

Download and install a network scanner app on your smartphone, such as Fing or Network Analyzer. Open the network scanner app and start a scan of your network. Once the scan is complete, look for the device named "Raspberry Pi Foundation" or a similar name. The IP address of your Raspberry Pi should be displayed next to the device name. Note down the IP address and use it to connect to your Raspberry Pi through SSH or VNC, depending on your preference.

Appendix.5 Detailed process for connecting Bluetooth headphones to a Raspberry Pi

Make sure your Raspberry Pi has Bluetooth capabilities. If not, you must add a Bluetooth dongle to one of the USB ports.
Put your Bluetooth headphones into pairing mode. The method for doing this may vary depending on the model of your headphones, so refer to the user manual if necessary.
On your Raspberry Pi, open a terminal and run the following command to install the necessary software
sudo apt-get install bluetooth bluez pulseaudio-module-bluetooth
Next, run the following command to pair your headphones with the Raspberry Pi:
sudo bluetoothctl
This will open the Bluetooth control panel in the terminal.
In the Bluetooth control panel, run the following commands
power on
agent on
default-agent
scan on
This will make your Raspberry Pi discoverable and start scanning for Bluetooth devices.
Wait for your headphones to appear in the list of discovered devices. When they do,

note down the MAC address.
Run the following command to pair your headphones with your Raspberry Pi:

```
1. pair MAC_ADDRESS
```

Replace MAC_ADDRESS with the MAC address of your headphones.
You may be prompted to enter a PIN code. If so, refer to the user manual for your headphones to find the correct PIN code.
Once your headphones are paired, run the following command to connect them to your Raspberry Pi:

```
1. connect MAC_ADDRESS
```

Replace MAC_ADDRESS with the MAC address of your headphones.
Finally, run the following command to make your headphones the default audio output device for your Raspberry Pi:
sudo raspi-config
Navigate to "System Options" > "Audio" > "Headphones" and select "Yes".