BLIND NAVIGATION SUPPORT SYSTEM USING RASPBERRY PI & YOLO

Submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

BY

PANKHURI SANTOSHI (REG. NO. – 39110740)

PARVADHAVARDHNI R. (REG. NO. – 39110744)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC |

Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PANKHURI SANTOSHI (Reg. No: 37110740) and PARVADHAVARDHNI R (Reg. No: 37110744)** who carried out the Project Phase-II entitled **"BLIND NAVIGATION SUPPORT SYSTEM USING RASPBERRY PI & YOLO"** under my supervision from January 2023 to April 2023.

INTERNAL GUIDE

Dr. A. Mary Posonia, M.E., Ph.D.

Head of Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

SUBMITED FOR VIVA-VOCE EXAMINATION HELD ON 20.04.2023

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I, **PANKHURI SANTOSHI** hereby declare that the project report entitled **BLIND NAVIGATION SUPPORT SYSTEM USING RASPBERRY PI & YOLO** done by me under the guidance of **Dr. A. Mary Posonia- M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 20 April 2023

PLACE: Chennai **SIGNATURE OF CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them. I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean- School of Computing**, and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Mary Posonia- M.E., Ph.D.** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** ho were helpful in many ways for the completion of the Project Phase- I.

ABSTRACT

The world of technology is constantly evolving, and we have come to rely on it for many aspects of our daily lives. For individuals with visual impairments, technology has become an essential tool for providing greater independence and accessibility. One area where technology can have a significant impact is in navigation. Navigation can be challenging for people with visual impairments, especially in unfamiliar environments. Therefore, the development of a blind navigation system that can help individuals navigate through unfamiliar environments can be incredibly beneficial. In this paper, we propose a blind navigation system that uses the YOLO (You Only Look Once) object detection algorithm and a Raspberry Pi.Object detection is a computer vision technique that enables machines to identify and locate objects within an image or video. YOLO is a real-time object detection system that can detect objects in images and videos with remarkable accuracy and speed. The YOLO algorithm uses a single neural network to process the entire image, which enables it to detect objects with incredible speed. The Raspberry Pi is a low-cost, credit card-sized computer that can be used for a wide range of projects. It is a popular choice for DIY electronics projects due to its affordability and versatility. The proposed blind navigation system consists of a Raspberry Pi and a camera module connected to the Pi. The system uses the YOLO algorithm to detect objects within the camera's field of view. The YOLO algorithm processes the image captured by the camera and identifies objects within the image. The system then uses text-to-speech technology to verbally communicate the location of the objects to the user. The system can also provide vibration feedback to indicate the distance and direction of the object. The system can be used to identify a range of objects, including doorways, stairs, and obstacles. The system is designed to be portable and lightweight, making it easy for users to carry with them wherever they go. The system is powered by a rechargeable battery, and the user interface is simple and easy to use. The user interface consists of a single button that initiates the object detection process. The system is also designed to be customizable, allowing users to adjust the sensitivity of the object detection system to suit their needs.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 : Visual Impairment

The International Classification of Diseases 11 (2018) classifies vision impairment into two groups, distance and near presenting vision impairment.

Distance vision impairment:

Mild – visual acuity worse than 6/12 to 6/18

Moderate – visual acuity worse than 6/18 to 6/60

Severe – visual acuity worse than 6/60 to 3/60

Blindness – visual acuity worse than 3/60

Near vision impairment:

Near visual acuity worse than N6 or M.08 at 40cm.

A person's experience of vision impairment varies depending upon many different factors. This includes for example, the availability of prevention and treatment interventions, access to vision rehabilitation (including assistive products such as spectacles or white canes), and whether the person experiences problems with inaccessible buildings, transport and information.

Prevalence

Globally, at least 2.2 billion people have a near or distance vision impairment. In at least 1 billion – or almost half – of these cases, vision impairment could have been prevented or has yet to be addressed.

This 1 billion people include those with moderate or severe distance vision impairment or blindness due to unaddressed refractive error (88.4 million), cataract (94 million), age-related macular degeneration (8 million), glaucoma (7.7 million), diabetic retinopathy (3.9 million) , as well as near vision impairment caused by unaddressed presbyopia (826 million) .

In terms of regional differences, the prevalence of distance vision impairment in low- and middle-income regions is estimated to be four times higher than in high-income regions (1). With regards to near vision, rates of unaddressed near vision impairment are estimated to be greater than 80% in western, eastern and central sub-Saharan Africa, while comparative rates in high-income regions of North America, Australasia, Western Europe, and of Asia-Pacific are reported to be lower than 10%..

Population growth and ageing are expected to increase the risk that more people acquire vision impairment.

Causes

Globally, the leading causes of vision impairment are:

1. age-related macular degeneration
2. cataract
3. diabetic retinopathy
4. glaucoma
5. uncorrected refractive errors

There is substantial variation in the causes between and within countries according to the availability of eye care services, their affordability, and the eye care literacy of the population. For example, the proportion of vision impairment attributable to cataract is higher in low- and middle-income countries than high-income countries. In high income countries, diseases such as glaucoma and age-related macular degeneration are more common.

Among children, the causes of vision impairment vary considerably across countries. For example, in low-income countries congenital cataract is a leading cause, whereas in middle-income countries it is more likely to be retinopathy of prematurity.

1. As in adult populations, uncorrected refractive error remains a leading cause of vision impairment in all countries amongst children.
2. Impact of vision impairment
3. Personal impact

Young children with early onset severe vision impairment can experience delayed motor, language, emotional, social and cognitive development, with lifelong consequences. School-age children with vision impairment can also experience lower levels of educational achievement.

Vision impairment severely impacts quality of life among adult populations. Adults with vision impairment often have lower rates of workforce participation and productivity and higher rates of depression and anxiety.

In the case of older adults, vision impairment can contribute to social isolation, difficulty walking, a higher risk of falls and fractures, and a greater likelihood of early entry into nursing or care homes.

Economic impact

Vision impairment poses an enormous global financial burden with an estimate annual global productivity loss of about US$ 411 billion purchasing power parity (3). this figure far outweighs the estimated cost gap of addressing the unmet need of vision impairment (estimated at about US$ 25 billion).

Strategies to address eye conditions to avoid vision impairment

While a large number of eye diseases can be prevented (such as infections, trauma, unsafe traditional medicines, perinatal diseases, nutrition-related diseases, unsafe use or self-administration of topical treatment), this is not possible for all.

Each eye condition requires a different, timely response. There are effective interventions covering promotion, prevention, treatment and rehabilitation which address the needs associated with eye conditions and vision impairment; some are among the most cost-effective and feasible of all health care interventions to implement. For example, uncorrected refractive error can be corrected with spectacles or surgery while cataract surgery can restore vision.

Treatment is also available for many eye conditions that do not typically cause vision impairment, such as dry eye, conjunctivitis and blepharitis, but generate discomfort and pain. Treatment of these conditions is directed at alleviating the symptoms and preventing the evolution towards more severe diseases.

Vision rehabilitation is very effective in improving functioning for people with an irreversible vision impairment that can be caused by eye conditions such as diabetic retinopathy, glaucoma, consequences of trauma, and age-related macular degeneration.

WHO response

WHO's work is guided by the recommendations of the WHO World report on vision (2019) and the resolution on "integrated, people-centred eye care, including preventable blindness and vision impairment" that was adopted at Seventy-third World Health Assembly in 2020. The key proposal of the report and resolution is to make integrated people-centred eye care (IPEC) the care model of choice and to ensure its widespread implementation. It is expected that by shaping the global agenda on vision, the report and resolution will assist Member States and their partners in their efforts to reduce the burden of eye conditions and vision impairment and achieve the Sustainable Development Goals (SDGs), particularly SDG target 3.8 on universal health coverage.

Some of WHO's key areas of work and activities in the prevention of blindness include:

Working with Member States and other partners in the field to monitor the global targets for 2030 on integrated people-centred eye care.

Observing and promoting World Sight Day as an annual advocacy event.

Supporting the integration of eye care in health systems through the implementation of a series of technical tools:

eye care in health sytems – Guide for action providing practical, step-by-step support to Member States in the planning and implementation of the recommendations of the World report on vision.

1. Package of Eye Care Interventions (PECI): a tool for planning and budgeting for eye care at each level of the health system.
2. Eye Care Competency Framework (ECCF): a planning tool for eye care human resources based on competencies; and
3. mobile health toolkit for myopia to increase awareness and health literacy of modifiable risk factors, potential irreversible consequences of myopia and the importance of spectacle compliance and regular eye examinations.
4. The development and implementation tools to support countries to assess the provision of eye care services such as:
5. Eye care services assessment tool
6. Tool for Assessment of Diabetes and Diabetic Retinopathy Services
7. Tool for the Assessment of Glaucoma Services

8. Tool for the Assessment of Refractive Services
9. Tool for the Assessment of Rehabilitation Services and Systems

1.2 : Smart Navigation System

A smart navigation system using Raspberry Pi is an innovative solution that uses the low-cost, credit-card-sized computer, Raspberry Pi, to develop a navigation system that can provide real-time directions to drivers or pedestrians. This system is built using open-source software and hardware, which makes it easily customizable and scalable.

The Raspberry Pi is an ideal platform for developing a smart navigation system as it is a small and affordable computer that can run various operating systems and programming languages. The system is built using a GPS module and a touch screen display, which is connected to the Raspberry Pi. The GPS module receives location data and transmits it to the Raspberry Pi, which then processes the data and displays it on the touch screen display.

One of the key benefits of this system is its ability to provide real-time updates on traffic conditions and suggest alternative routes to avoid congestion. This feature is achieved by using real-time data from various sources, such as traffic cameras, weather reports, and social media feeds.

Another advantage of the smart navigation system is its versatility. The system can be customized to suit different types of users, such as drivers, cyclists, or pedestrians. For instance, the system can be programmed to provide turn-by-turn directions for drivers or suggest bike lanes for cyclists.

The smart navigation system can also be integrated with other smart devices in a smart city ecosystem, such as traffic lights and public transportation systems. This integration can help to improve the efficiency of the overall transportation system and reduce congestion.

In addition to its practical applications, the smart navigation system can also be used for educational purposes. It can be used to teach programming, electronics, and hardware design to students in schools and universities. This can help to inspire the next generation of innovators and engineers.

In conclusion, a smart navigation system using Raspberry Pi is a cost-effective and versatile solution that can provide real-time directions and updates to drivers, cyclists, and pedestrians. The system can be customized to suit different types of users and integrated with other smart devices in a smart city ecosystem. Furthermore, the system has educational applications, which can inspire future innovators and engineers.

1.3 : Raspberry PI

The Raspberry Pi is a small, affordable, and versatile computer that was created by the Raspberry Pi Foundation. It is designed to be accessible to anyone, regardless of their technical knowledge or budget, and it has had a significant impact on the world of computing since its release.

The Raspberry Pi is a single-board computer that is roughly the size of a credit card. It is powered by a Broadcom System on a Chip (SoC), which includes a CPU, GPU, and RAM, as well as various other components like USB ports, HDMI output, and an Ethernet port. The first Raspberry Pi had a 700MHz ARM11 processor, 256MB of RAM, and a single USB port, but newer models have significantly more processing power and additional features.

One of the key features of the Raspberry Pi is its low cost. The original Raspberry Pi sold for just $35, making it accessible to people with limited budgets. Since then, newer models have been released at slightly higher price points, but they are still very affordable compared to traditional desktop computers.

Another key feature of the Raspberry Pi is its versatility. It can run a variety of operating systems, including Linux distributions like Raspbian and Ubuntu, as well as Windows 10 IoT Core. This makes it suitable for a wide range of applications, from web servers to media centres to robotics.

One of the most common uses for the Raspberry Pi is as a platform for learning computer programming. The Raspberry Pi Foundation provides a variety of resources to help people learn to code using the Raspberry Pi, including an online learning portal and a variety of books and tutorials. Many schools and educational organizations use the Raspberry Pi to teach programming and other computer science skills.

The Raspberry Pi has also been used in a variety of DIY projects, from building retro gaming consoles to creating home automation systems. Its small size and low power consumption make it a great choice for embedded applications, and there are a variety of add-on boards and accessories available to extend its capabilities.

In addition to hobbyist and educational uses, the Raspberry Pi has also been used in commercial products. For example, it has been used in digital signage systems, industrial control systems, and even as the brain for a small satellite.

The Raspberry Pi has also spawned a vibrant ecosystem of third-party hardware and software. There are a wide range of add-on boards available, including ones that add additional sensors, motors, and other components. There are also a variety of software packages and libraries available, including ones for machine learning, computer vision, and robotics.

One of the challenges of using the Raspberry Pi is its limited processing power and memory compared to traditional desktop computers. However, this can be mitigated by optimizing code, using more efficient programming languages, or by using more powerful models of the Raspberry Pi.

Another challenge is that the Raspberry Pi does not come with a built-in screen or keyboard, which means that users will need to provide these themselves or connect to the Raspberry Pi remotely over a network.

Despite these challenges, the Raspberry Pi has had a significant impact on the world of computing since its release. Its low cost, versatility, and ease of use have made it accessible to people who may not have had access to traditional computers. It has also inspired a new generation of programmers and makers and has helped to democratize access to computing technology.

CHAPTER 2: LITERATURE SURVEY

Woojin Chung, "Integrated navigation system for indoor service robots in large-scale environments" It contains architecture of navigation system, the development of crucial navigation algorithms like map, path planning, and localization, and planning scheme such as fault handling. This system provides some advantages that are 1) A range sensor based generalized scheme of navigation without modification of the environment. 2) Intelligent navigation-related components. 3) Framework supporting the selection of multiple behaviors and fault handling schemes [9]

Denis Tudor, Lidia Dobrescu, Drago Dobrescu, "Ultrasonic Electronic System for Blind People Navigation" This system presents a new electronic system using an ATmega328P microcontroller, two ultrasonic sensors and vibrating motors as a helping solution for blind people navigation. In order to determine the distance, HC-SR04 ultrasonic sensors are used. The HC-SR04 ultrasonic sensor uses sonar A short ultrasonic pulse is transmitted at the initial time, echoed by an object [10].

Kanchan M. Varpe, M.P. Wankhade," Visually Impaired Assistive System"which focuses on independent portability of blind people who travel in an unfamiliar environment without any manual assistance. System include on the server side zigbee transceiver for wireless conversation, RFID reader with an integrated microcontroller, zigbee transmitter and TTS for playing information to user. The VIAS can be used by visually impaired or blind users at the system implemented environment such as organization campus which can be school, college, hospitals, shopping mart, bus stands, etc [14].

Aladrén, G. López-Nicolás, Luis Puig, and Josechu J. Guerrero" Navigation Assistance for the Visually Impaired Using RGB-D Sensor With Range Expansion", In this paper, a new system for NAVI is presented based on visual and range information. Rather of using multiple sensors, we choose one device, a consumer RGB-D camera, and take advantage of both range and visual information. In appropriate, the combination of depth information with image intensities, resulting in the robust expansion of the range-based floor segmentation. Our system detects the main structural elements of the scene using range data [15].

B.S. Tjan, P.J. Beckmann, R. Roy, N. Giudice4, and G.E. Legge, "Digital Sign System for Indoor Wayfinding for the Visually Impaired", In this we describe the design and implementation of a digital sign system based on low-cost passive retro-reflective tags printed with specially designed patterns that can be readily recognized and identified by a handheld camera and machine-vision system. Performance of the prototype showed the tag recognition system could cope with the real-world environment of a typical building [16].

In this survey, we observe that main problem of blind people should depend on any other guide like blind cane, black glasses, people information, trained dogs. But trained dog would also be burden of them as they can move only to the places that dogs are trained. Blind people need some support to feel safe while moving indoor or outdoor.Smart cane, range notification system, pathfinder, real time localization system, ultrasonic electronic system, novel indoor system etc. Systems are previously used by blindpeople. That system based on various techniques such as An electronic system using an ATmega328P microcontroller, A remote processing system analyzes-by computer vision algorithms, Assistive technology device called Electronic Long Cane,Eye Stick, tactile signals for giving information rather than acoustic signals, white cane or Hoover cane

.

2.1 Design and Implementation of Eye Stick for Blind People: The paper has made a example which will discover objects or obstacles ahead of users and feeds warning back, within the types of voice. It permits blind folks to recognize any obstacles and it permits period of time feedback to the user with voice on speaker, mistreatment the supersonic Ranger, the obstacle at intervals four hundred cm aloof from the cane is detected. There three famous sensors used for obstacle detection. They’re infrared device, supersonic device and optical maser device. Since optical maser device was costly it absolutely was neglected from the beginning of the project. The opposite two device outputs similar result however infrared sensors area unit notable to be disturbed by daylight and dark objects. Since the device is needed to discover obstacles at intervals person size the supersonic device was chosen. The unit will be used as associate independent navigation tool while not the necessity of the cane. This may also ensure that the user will use the cane while not the unit for places that area unit acquainted to them. The limitation of this design is that the device detects obstacle at narrow range.

2.2 An Rfid Application for the Disabled:

Path Finder: After the tests, it's clearly seen that blind people, who are not accustomed to the field, reached the target simply and correctly. Also, check users of the system turned back to the main gate from that they entered the field, successfully. Unfortunately, just in case of quite one target chosen, expected success result couldn't be achieved within the system, which can be seen from the graphs on top of. Besides, it is observed that passive RFID tags may be scan simply during a distance of thirty cm. to the reader, however whenever the tag holder is much away than 35 cm. to the reader, tags weren't read any further. So as to beat this drawback, either emission of the reader can be inflated or active RFID tags can be utilized in the system.

2.3 Voice Assisted Navigation System for the Blind:

This work presents a model of a navigation system that helps the visually impaired to maneuver in each indoor and outdoor environments. This method is intended to be completely self-sustainable and bank as very little as doable on virtual mapping ways. It’s hopped-up by AN on-board power source and permits the user to sense objects in their environment. Obstacle is perceived from supersonic sensing element and an audio is played looking on the space of travel of ultrasound. An obstacle as short as

4cm are often detected by the module. With a resolution of 15cm of obstacle distance, an acceptable audio instruction is given to the blind user. The navigator satisfactorily plays an applicable audio into the headphones which corresponds to the space of obstacle from the user. Change within the ground gradients has been tackled simply as the sensors method the sharp modification in ground level and intimates the user concerning constant. This planned system will be factory-made because the system is cheap. The system is implemented on the cane to produce a safer feeling to the blind. The project is often makeshift by the utilization of wireless LAN connections that supports the use of GPS. This helpsthe user to navigate a lot of accurately and effectively as a GPS module will actuate the position use of GPS co-ordinates.

2.4 A Real-time Localization System Using RFID for Visually Impaired:

This analysis given tag style and sensible white cane for ease in reading data while not additional devices. Short analysis is completed per the context of use that considers main user, task and environmental characteristics of matters during which it'll be operated. The example is simple to be told, use, and get or be subsidized from public service; therefore we all over the acceptableness, usability, and price profit are satisfied. Also huge decision opportunities and visibility for the blind in operating are given. Future work is going to be network environment and compatibility with infrastructure and current customary. Extending network environment to system will alter range to be not solely high-value applications of walking choices, however additionally following risk things that imply help through the network. to boot, road or building condition changes can be simply updated by the server. Compatibility with existing infrastructure and current customary like EPC global tag theme are required for smooth application.

2.5 Design and implementation of electronic aid to blind’s cane:

ETAs, just like the sensible cane provide blind individuals an independency level that is simply potential by the employment if this device. The sensible cane offers a security zone between them and also the possible obstacles. One of the most effective characteristics is that the ability of the design to any quite cane therefore a blind man wouldn't want to change the cane to use the device. With this technological equipment, blind people are more assured once occupancy the streets and those they don’t have to be compelled to modification their cane by another tool or technique as a result of this proposal is adaptable to the one, they have already got.

2.6 Smart Cane with Range notification for blind people

The system uses of an ultrasonic sensor for getting input and use earphone as the output. Ultrasonic sensor is used to measure distance from the obstacle present in their path. The information is then fed to the NI myRIO-1900 which is after translated into the audio output. The beeping frequency increases as the user goes closer to the obstacle. The placement and orientation of the sensor on the cane was also well examined in terms of accuracy. The upper position sensor with angle of 90 degree has been found to the most real organization.

2.7 Electronic long cane for locomotion improving on visual impaired people

In this work, a novel electronic cane, called ELC, was registered and calculated. This approach involved an ergonomic design along with embedded electronics inside the grip of a traditional long cane which circulate human dimensional and tactile approach. The device indicates obstacles above waistline and warns about potential crash. Qualitative assessment of an ELC prototype was carried out by voluntary blind people. The obtained results showed the ELC effectiveness for detecting physical barriers located above of the imaginary waistline, so, contributing to a better perception of the surrounding space by blind people or visually impaired people. The electronic circuit embedded on the grip can detect obstacles above the waistline in order to give a tactile feedback, through a vibration inside the cane. This tactile reply becomes more frequent meanwhile the user reaches the obstacle. The integrated hardware-ergonomic solution, in spite of simplicity, gives a new concept to improve flexibility, by indicating the related surrounding features above the user waist, thus, sharing to a safer human progression.

2.8 Novel indoor navigation system for Visually Impaired and blind people

It provides the blind people the ability to travel without any other assistance. The planned system architecture uses a network of IP cameras placed at the roof of each room. A remote processing system investigate-by computer vision algorithms-photo taken from the environment in order to inform the subject about his location and reacts accordingly to pass the sufficient assistance and then use for bind people. A guidance algorithm used for destination using a simple interactive mobile application installed on his smart phone. The proof-of-concept prototype was designed with one camera on top of a wooden floor model to resemble the system. The proposed system access has a straightforward architecture. There are two principal actors in this architecture are the blind person and the remote processing system.

CHAPTER 3: REQUIREMENT ANALYSIS

3.1 : SOFTWARE- Raspberry Pi OS

Raspberry Pi OS (formerly Raspbian) is a Unix-like operating system based on the Debian Linux distribution for the Raspberry Pi family of compact single-board computers. First developed independently in 2012, it has been produced as the primary operating system for these boards since 2013, distributed by the Raspberry Pi Foundation.

Raspberry Pi OS is highly optimized for the Raspberry Pi with ARM CPUs. It runs on every Raspberry Pi except the Pico microcontroller. Raspberry Pi OS uses a modified LXDE desktop environment with the Openbox stacking window manager, along with a unique theme. The default distribution is shipped with a copy of the computer algebra system Wolfram Mathematica, VLC, and a lightweight version of the Chromium web browser.

Raspberry Pi OS was first developed by Mike Thompson and Peter Green as Raspbian, an independent and unofficial port of Debian to the Raspberry Pi. The first build was released on July 15, 2012.As the Raspberry Pi had no officially provided operating system at the time, the Raspberry Pi Foundation built on the work by the Raspbian project and began producing and releasing their own version of the software. The Foundation's first release of Raspbian, which now referred both to the community project as well as the official operating system, was announced on September 10, 2013.

On May 28, 2020, the Raspberry Pi Foundation announced a beta 64-bit version. However, this version was not based on Raspbian, instead taking its user space software from Debian GNU/Linux. When the Foundation did not want to use the name Raspbian to refer to software that was not based on the Raspbian project, the name of the officially provided operating system was changed to Raspberry Pi OS. This change was also carried over to the 32-bit version, though it continued to be based on

Raspbian. The 64-bit version of Raspberry Pi OS was officially released on February 2, 2022.

Features

1. User interface

   Raspberry Pi OS has a desktop environment, PIXEL (short for Pi Improved Xwindows Environment, Lightweight), based on LXDE, which looks similar to many common desktops, such as macOS and Microsoft Windows. The desktop has a background image. A menu bar is positioned at the top and contains an application menu and shortcuts to a web browser (Chromium), file manager, and terminal. The other end of the menu bar shows a Bluetooth menu, Wi-Fi menu, volume control, and clock. The desktop can also be changed from its default appearance, such as repositioning the menu bar.

2. Package management

   Packages can be installed via APT, the Recommended Software app, and by using the Add/Remove Software tool, a GUI wrapper for APT.

3. Components

   PCManFM is a file browser allowing quick access to all areas of the computer, and was redesigned in the first Raspberry Pi OS Buster release (2019-06-20).

Raspberry Pi OS originally distributed the web browser Epiphany but switched to Chromium with the launch of its redesigned desktop. The built-in browser comes preinstalled with uBlock Origin and h264ify.

Raspberry Pi OS comes with many beginner IDEs, such as Thonny Python IDE, Mu Editor, and Greenfoot. It also ships with educational software, such as Scratch and Bookshelf.

3.2 : Hardware

3.2.1 : Raspberry Pi 4

Raspberry Pi 4 Model B is the latest version of the popular single-board computer that was first released by the Raspberry Pi Foundation in 2012. It is designed to be a low-cost, high-performance computing platform that can be used for a wide range of applications, including home automation, media centers, robotics, and education.

The Raspberry Pi 4 Model B is powered by a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC, which is clocked at 1.5GHz. It is available in three variants with different amounts of RAM: 2GB, 4GB, and 8GB. The RAM is LPDDR4-3200, which is faster than the LPDDR2 and LPDDR3 RAM used in previous models.

One of the most significant improvements in the Raspberry Pi 4 Model B is the addition of two Micro-HDMI ports that support up to 4Kp60 resolution. This means that the Raspberry Pi 4 Model B can be used as a media center or a desktop computer, and it can drive two 4K displays simultaneously. The Micro-HDMI ports are also compatible with DVI and VGA displays, using adapters.

Another improvement is the addition of two USB 3.0 ports and two USB 2.0 ports, which provide faster data transfer rates than the USB 2.0 ports found in previous models. The Raspberry Pi 4 Model B also has a Gigabit Ethernet port, dual-band 802.11ac wireless networking, and Bluetooth 5.0.

The Raspberry Pi 4 Model B also features a new power supply circuit that supports up to 3A of current, which is necessary for powering USB devices and other peripherals. It has a USB-C power connector, which is an improvement over the Micro-USB connector used in previous models.

The GPIO header on the Raspberry Pi 4 Model B is the same as the one on previous models, with 40 pins that can be used for connecting sensors, motors, and other electronics. However, the Raspberry Pi 4 Model B uses a new pinout, which means that some accessories designed for previous models may not be compatible.

The Raspberry Pi 4 Model B runs a variety of operating systems, including Raspberry Pi OS (formerly known as Raspbian), Ubuntu, and several other Linux distributions. It also supports a wide range of programming languages, including Python, C++, Java, and more.

Overall, the Raspberry Pi 4 Model B is a powerful and versatile single-board computer that offers significant improvements over previous models. Its combination of performance, connectivity, and affordability make it an excellent choice for a wide range of applications.

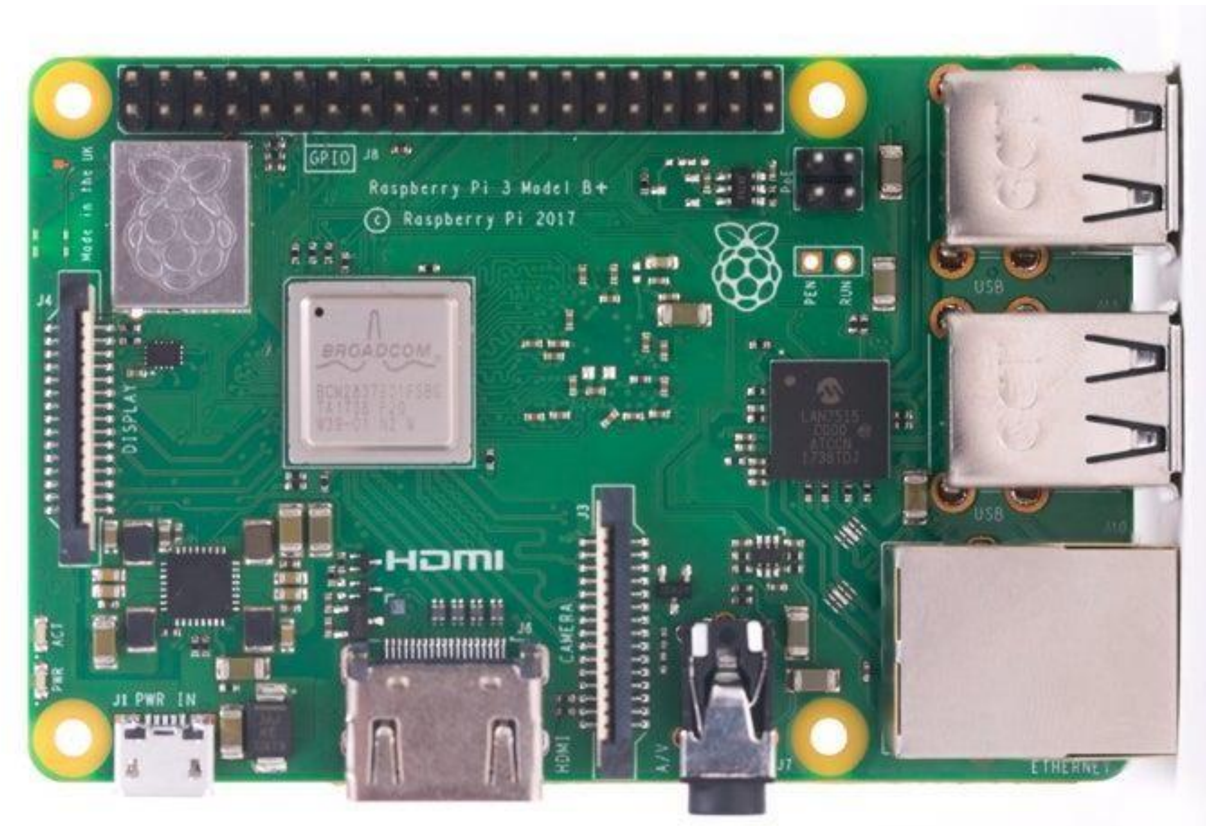


Fig 3.1- Raspberry Pi

3.2.2 : ULTRASONIC SENSOR:

An ultrasonic sensor is a device that uses sound waves to detect the distance between itself and an object. The sensor emits high-frequency sound waves that bounce off the object and return to the sensor, allowing it to calculate the distance based on the time it takes for the sound waves to travel.

Ultrasonic sensors are used in a wide range of applications, from parking sensors in cars to object detection in industrial automation. They are particularly useful in situations where traditional sensors, such as optical or infrared sensors, may not be effective due to poor lighting conditions or reflective surfaces.

The basic principle of an ultrasonic sensor is to emit a short burst of high-frequency sound waves, usually between 20 kHz and 200 kHz, and then measure the time it takes for the sound waves to bounce back to the sensor after they reflect off an object. The sensor typically has two transducers, one that emits the sound waves and one that receives them. The time delay between the emission and reception of the sound waves is used to calculate the distance between the sensor and the object.

Ultrasonic sensors can operate in a variety of modes, including continuous wave mode, pulse-echo mode, and frequency-modulated continuous wave (FMCW) mode. In continuous wave mode, the sensor emits a continuous stream of sound waves, and the distance to the object is calculated based on the phase shift of the received waves. In pulse-echo mode, the sensor emits a short burst of sound waves and then waits for the echo to return. This mode is commonly used in distance measurement applications. In FMCW mode, the sensor emits a continuously varying frequency, and the distance is calculated based on the difference between the emitted and received frequencies.

Ultrasonic sensors are typically designed to operate in a specific range of distances, from a few centimetres up to several meters. The maximum range of the sensor depends on several factors, including the frequency of the sound waves, the power of the emitted signal, and the sensitivity of the receiver. Higher-frequency sound waves can travel shorter distances, but they provide higher resolution and accuracy. Lower-frequency sound waves can travel longer distances, but they have lower resolution and are more susceptible to interference.

One of the advantages of ultrasonic sensors is that they can be used to detect objects regardless of their color, texture, or transparency. They are also immune to interference from ambient light or electromagnetic radiation, making them suitable for use in harsh environments. However, they are susceptible to interference from other ultrasonic sources, such as other sensors or motors.

Ultrasonic sensors are commonly used in a variety of applications, including distance measurement, object detection, and liquid level sensing. In distance measurement applications, they are often used in combination with other sensors, such as optical or infrared sensors, to provide accurate measurements in a wide range of conditions. In object detection applications, they can be used to detect the presence of objects and trigger alarms or other actions. In liquid level sensing applications, they are often used in tanks or pipes to detect the level of liquid.

In conclusion, ultrasonic sensors are versatile devices that use sound waves to detect the distance between themselves and objects. They are used in a wide range of applications and offer several advantages over traditional sensors, including immunity to interference from ambient light or electromagnetic radiation. Ultrasonic sensors come in various configurations and offer different ranges, resolutions, and accuracy levels. Their cost-effectiveness, versatility, and reliability make them a popular choice in industrial, automotive, and consumer electronics applications.

Fig 3.2- Ultrasonic Sensor

3.2.2: Pi Camera:

The Pi camera is a small, portable camera module designed to work with the Raspberry Pi, a credit card-sized single-board computer. It is a versatile camera that can be used for a wide range of applications, including photography, videography, security systems, and even computer vision.

The Pi camera is available in two versions: the standard Pi camera and the Pi camera module. The standard Pi camera is a 5-megapixel camera that can capture high-quality images and video. The Pi camera module is a more advanced version of the camera that features a higher resolution and better image quality. The latest version of the Pi camera module is the V2, which is an 8-megapixel camera that can capture 1080p video at 30 frames per second.

The Pi camera connects to the Raspberry Pi through a ribbon cable that attaches to the camera module port on the Raspberry Pi board. The camera is controlled through software that is installed on the Raspberry Pi. The camera can be controlled through the command line or through a graphical user interface, depending on the software used.

One of the key features of the Pi camera is its ability to capture high-quality images and video. The camera is capable of capturing images with a resolution of up to 3280 x 2464 pixels. It can also capture video at various resolutions, including 1080p at 30 frames per second and 720p at 60 frames per second. The camera can also capture images in RAW format, which allows for greater flexibility in post-processing.

Another important feature of the Pi camera is its ability to capture images and video in low-light conditions. The camera features an infrared filter that can be removed to capture images in near-total darkness. It also has a high-quality lens that can capture images with low distortion and high sharpness.

The Pi camera is also widely used in computer vision applications. It can be used to capture images for image processing and analysis, and it can be integrated into machine learning algorithms for object recognition, facial recognition, and other applications. The camera is also used in robotics and automation applications, where it can be used to capture images of the surrounding environment to guide the movement of the robot.

One of the main advantages of the Pi camera is its low cost. The camera is relatively inexpensive compared to other high-quality cameras on the market, making it accessible to hobbyists and students. The camera is also very easy to use and can be integrated into projects quickly and easily.

The Pi camera is compatible with a wide range of software, including the popular open-source software platform OpenCV. OpenCV is a library of programming functions mainly aimed at real-time computer vision. It allows the Raspberry Pi and the Pi camera to be used for a wide range of computer vision applications, including object detection, facial recognition, and gesture recognition.

In addition to its use in computer vision applications, the Pi camera is also used in security systems. It can be used to monitor areas and detect movement or other events. The camera can also be used as a webcam or for video conferencing, making it useful for remote meetings and other online activities.

In conclusion, the Pi camera is a versatile camera module that can be used for a wide range of applications. It is compatible with the Raspberry Pi and can be used for photography, videography, computer vision, security systems, and many other applications. The camera is relatively inexpensive and easy to use, making it accessible to hobbyists and students. With its high-quality images and video, low-light capabilities, and compatibility with popular software platforms, the Pi camera is a popular choice for a wide range of applications.

Fig 3.3- Pi Camera

3.2.2: Storage:

Storage is an important aspect of blind navigation support systems, as it enables the system to store and retrieve data related to the environment, such as floor plans, point of interests, obstacles, and other relevant information. Local, cloud, and hybrid storage technologies can be used, depending on the size and frequency of data updates, as well as the security requirements of the system. By carefully considering the storage requirements of a blind navigation support system, designers can ensure that the system is able to provide accurate and reliable navigational support to visually impaired individuals.

Blind navigation support systems are designed to assist visually impaired individuals in navigating indoor and outdoor environments. One important aspect of these systems is the storage of data that is used to provide navigational support. In this context, storage refers to the ability of the system to store and retrieve data related to the environment, such as floor plans, point of interests, obstacles, and other relevant information.

There are different types of storage technologies that can be used in blind navigation support systems, including local storage, cloud storage, and hybrid storage. Local storage involves storing data on the device itself, typically in a memory card or solid-state drive. This type of storage is useful when the system needs to access data quickly and reliably, without relying on an internet connection or remote server. Local storage can be used to store map data, location information, and other relevant information.

Cloud storage, on the other hand, involves storing data on remote servers that are accessed through the internet. This type of storage is useful for storing large amounts of data, as well as for allowing multiple users to access the same data from different locations. Cloud storage is often used to store and retrieve data related to traffic conditions, weather, and other dynamic information that is constantly changing.

Hybrid storage combines the benefits of both local and cloud storage, by using a combination of local and remote storage to store and retrieve data. For example, map data may be stored locally on the device, while location information and other dynamic data are stored remotely on a server. This approach allows the system to access data quickly and reliably, while also allowing for flexibility and scalability as the system grows.

In terms of the specific storage requirements for a blind navigation support system, there are several key factors to consider. One important factor is the size of the data being stored. Large datasets, such as high-resolution maps or 3D models, may require significant storage capacity and may benefit from cloud storage or hybrid storage solutions. Another factor to consider is the frequency with which the data is updated. Dynamic data, such as real-time traffic conditions or weather updates, may require frequent updates and may benefit from cloud storage or hybrid storage solutions that can easily update the data.

Another important factor to consider is the security of the stored data. Blind navigation support systems often contain sensitive information, such as user locations and personal preferences, and it is important to ensure that this data is stored securely and protected from unauthorized access. Encryption, access controls, and other security measures should be implemented to protect the stored data.

Fig 3.4- Micro SD Card, Cloud Storage & Internal Cards

CHAPTER 4: DESCRIPTION OF PROPOSED SYSTEM

4.1 : Selected Methodology/ Process Model:

Working Principle:

A blind navigation system using Raspberry Pi with audio output works on the principle of object detection and location identification, followed by audio feedback to the user. The system uses ultrasonic sensors to detect obstacles and the user's location is determined using a GPS module. The detected obstacles are then translated into audio feedback using text-to-speech (TTS) software, which is played through a speaker or headphones to inform the user of potential hazards in their path.

The system is composed of several components, including the Raspberry Pi, ultrasonic sensors, GPS module, TTS software, and an audio output device. The Raspberry Pi acts as the main controller and processes the sensor data to determine the user's location and detect obstacles in their path.

The ultrasonic sensors work by emitting high-frequency sound waves that bounce off objects and return to the sensor. The time taken for the sound wave to return is used to calculate the distance between the sensor and the object. Multiple sensors are placed around the user, typically on a wearable device or a cane, to provide a full 360-degree view of the environment. The sensor data is processed by the Raspberry Pi to identify obstacles and determine their location relative to the user.

The GPS module is used to determine the user's location and provide accurate navigation instructions. The module uses satellite signals to calculate the user's coordinates, which are then used to determine their current location and navigate them to their destination. The GPS data is also used to provide location-based information, such as nearby points of interest and landmarks.

The TTS software is used to translate the sensor data and navigation instructions into audio feedback for the user. The software can be programmed to provide different types of audio feedback depending on the type of obstacle and its location relative to the user. For example, the software may use a different tone or pitch to indicate the distance between the user and an obstacle.

The audio output device is used to play the audio feedback to the user. This can be a speaker or headphones, depending on the user's preference. The device is typically small and portable, making it easy to carry and use on the go.
In summary, the blind navigation system using Raspberry Pi with audio output works by using ultrasonic sensors to detect obstacles, a GPS module to determine the user's location, TTS software to translate the sensor data and navigation instructions into audio feedback, and an audio output device to play the feedback to the user. The system is designed to provide visually impaired individuals with accurate and reliable navigation instructions, allowing them to navigate their environment safely and independently.

Process Model

A blind navigation system using Raspberry Pi with audio output can be implemented using the following steps:

1. Assemble the hardware components: To build the blind navigation system, you will need a Raspberry Pi board, a microSD card, an ultrasonic sensor, a Pi camera module, and a USB speaker or headphone jack. Connect the ultrasonic sensor and Pi camera module to the Raspberry Pi board and configure them.

2. Install the operating system and required software: Download the latest version of the Raspberry Pi OS and install it on the microSD card. Install the required software, including Python libraries for the ultrasonic sensor and camera module, as well as a text-to-speech (TTS) library for the audio output.

3. Calibrate the ultrasonic sensor: The ultrasonic sensor will be used to detect obstacles and measure distances. To ensure accurate readings, calibrate the sensor according to the manufacturer's instructions.

4. Capture and process images: The Pi camera module will be used to capture images of the surrounding environment. Use Python code to capture images and process them

to extract information such as the presence of doors, stairs, and other obstacles.

5. Calculate distances using the ultrasonic sensor: Use Python code to measure distances to obstacles using the ultrasonic sensor. Use this information to generate directions for the user.

6. Generate audio output: Use the TTS library to generate audio output for the user, providing information such as the distance to obstacles and directions to navigate around them.

7. Integrate audio output with distance calculations: Use Python code to integrate the audio output with the distance calculations, so that the system can generate audio cues based on the user's location and the surrounding environment.

8. Test the system: Test the system to ensure that it provides accurate and reliable navigational support to visually impaired users.

In summary, building a blind navigation system using Raspberry Pi with audio output involves assembling the hardware components, installing the operating system and required software, calibrating the ultrasonic sensor, capturing and processing images, calculating distances using the ultrasonic sensor, generating audio output using a TTS library, integrating the audio output with distance calculations, and testing the system. With careful implementation and testing, such a system can provide valuable navigational support to visually impaired individuals.

4.2 : Architecture:

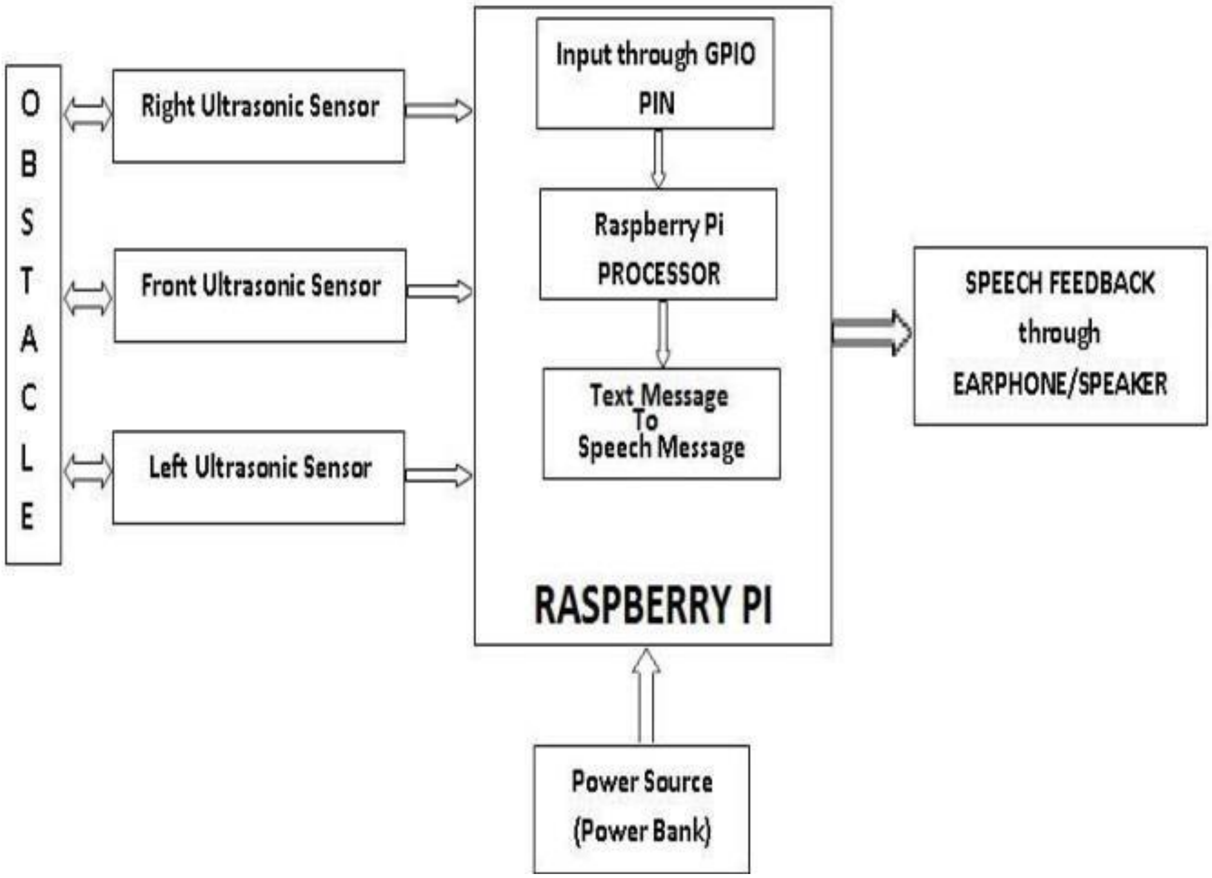


Fig 4.1- Connection Architecture

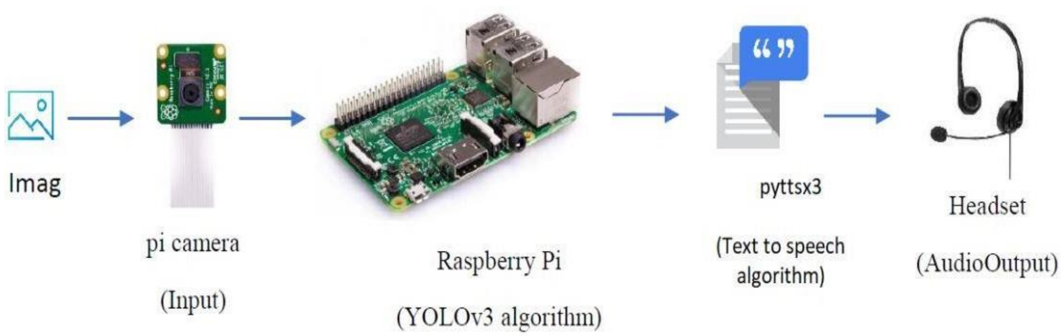


Fig 4.2- Hardware Architecture

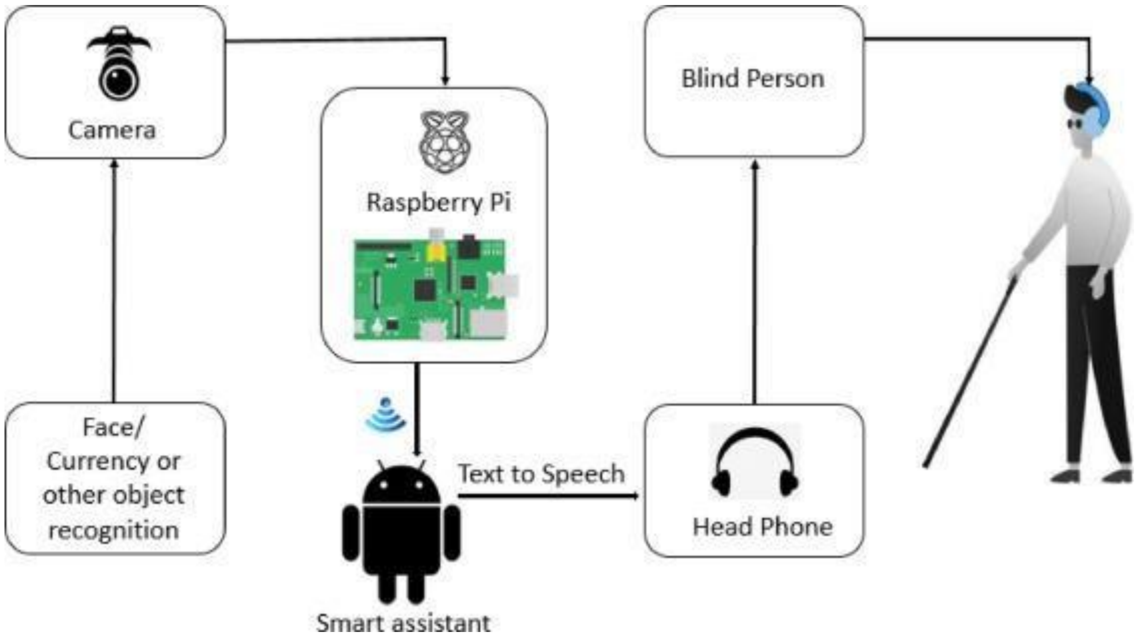


Fig 4.3- Process Architecture

4.3 Project Management Plan:

Project management for a blind navigation support system involves coordinating various tasks and resources to ensure that the project is completed successfully and on time. Here are some key steps that may be involved in managing a project for a blind navigation support system:

1. Project planning: This involves defining the scope of the project, setting project goals and objectives, and developing a project plan that outlines the tasks, timelines, and resources needed to complete the project.

2. Resource allocation: This involves identifying the resources needed for the project, such as hardware and software components, and allocating them appropriately. This may involve purchasing or acquiring the necessary resources and ensuring that they are available when needed.

3. Team coordination: This involves assembling a team of developers, testers, and

other personnel needed to complete the project, and ensuring that they are working together effectively. This may involve assigning tasks, managing schedules, and communicating regularly to ensure that everyone is on the same page.

4. Risk management: This involves identifying and mitigating risks that could impact the success of the project. This may involve conducting risk assessments, developing contingency plans, and monitoring the project for potential issues.

5. Quality assurance: This involves ensuring that the project is meeting the necessary quality standards and that the system is functioning as intended. This may involve testing the system thoroughly and conducting user testing to ensure that the system meets the needs of its users.

6. Project monitoring and control: This involves monitoring the project progress and adjusting the project plan as needed to ensure that the project stays on track. This may involve tracking progress against timelines and budgets, identifying and addressing any issues that arise, and communicating regularly with stakeholders to keep them informed of progress.

Overall, effective project management is essential for successfully developing and deploying a blind navigation support system. By following these steps and leveraging project management tools and techniques, project managers can ensure that the project is completed on time, within budget, and to the necessary quality standards.

4.3 Financial Report on Estimated Costs:

Hardware components:

- Raspberry Pi 4 Model B: INR 8,500
- Ultrasonic sensors (2x): INR 1,200
- Pi Camera: INR 1,000
- Battery pack: INR 1,000
- Other hardware components (wires, breadboard, etc.): INR 500
- Total cost of hardware components: INR 12,200

Software components:

- Raspberry Pi operating system (Raspbian): Free
- Python programming language: Free
- OpenCV computer vision library: Free
- TensorFlow machine learning library: Free
- Other software components (libraries, tools, etc.): Free
- Total cost of software components: Free

Overall, the total cost of hardware components for a blind navigation support system using Raspberry Pi would be around INR 8,100. The software components would be free, as most of the necessary software tools and libraries are open source and freely available. It's important to note that the cost may vary depending on the quality and quantity of the components used, as well as any additional costs such as development and testing expenses.

CHAPTER 5: IMPLEMENTATION DETAILS

5.1 : Development and Deployment Setup:

Developing and deploying a blind navigation support system involves several steps, including hardware and software components. Here is a high-level overview of the development and deployment setup for a blind navigation support system:

1. Hardware components: The first step is to select the hardware components for the system. This may include sensors, such as ultrasonic or infrared sensors, GPS receivers, and a microcontroller or processor to process the data from the sensors.

2. Software components: The software components of the system include the algorithms for processing the data from the sensors and providing feedback to the user. This may include machine learning algorithms, computer vision techniques, and navigation algorithms.

3. Prototype development: Once the hardware and software components have been selected, the next step is to develop a prototype of the system. This may involve building a physical prototype or developing a virtual prototype using simulation tools.

4. User testing: After developing the prototype, it is important to test the system with actual users to get feedback on its effectiveness and usability. This may involve working with blind or visually impaired individuals to get their feedback on the system.

5. Deployment: Once the system has been tested and refined, it can be deployed for wider use. This may involve partnering with organizations that work with blind or visually impaired individuals to provide access to the system.

6. Maintenance and updates: Finally, it is important to maintain and update the system over time to ensure that it continues to function effectively and meet the needs of its users.

Overall, developing and deploying a blind navigation support system requires careful attention to both the hardware and software components of the system, as well as input from blind or visually impaired users to ensure that the system is effective and usable.

5.2 : ALGORITHMS/ LIBRARIES:

OpenCV:

OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision and machine learning library that provides developers with a wide range of tools and algorithms for building computer vision applications. OpenCV was initially developed by Intel in 1999 and is now maintained by the OpenCV community.

OpenCV provides a comprehensive set of image processing and computer vision algorithms, including feature detection, object detection and recognition, segmentation, image filtering, and optical flow analysis. These algorithms are optimized for performance and can be used for a variety of applications, including robotics, surveillance, automotive, medical, and entertainment.

OpenCV is available in multiple programming languages, including C++, Python, and Java, making it accessible to a wide range of developers. It also provides support for multiple platforms, including Windows, Linux, and MacOS.

TensorFlow:

TensorFlow is an open-source software library for building and training machine learning models. Developed by Google Brain, TensorFlow is widely used for a variety of machine learning applications, including image recognition, natural language processing, and speech recognition.

TensorFlow provides a comprehensive set of tools and libraries for building and training machine learning models, including neural networks, convolutional neural networks, and recurrent neural networks. It also includes a set of pre-trained models, making it easy for developers to get started with machine learning.

TensorFlow is available in multiple programming languages, including Python, C++, and Java. It also provides support for multiple platforms, including Windows, Linux, and MacOS.

YOLO Algorithm:

YOLO (You Only Look Once) is a popular object detection algorithm that uses a single neural network to predict the locations and classes of objects in an image. Developed by Joseph Redmon and his team at the University of Washington, YOLO is known for its speed and accuracy, making it a popular choice for real-time object detection applications.

YOLO works by dividing an image into a grid of cells and predicting the probability of an object being present in each cell. It then predicts the bounding box and class of the object based on the highest probability score.

YOLO is trained on large datasets of images, using techniques such as data augmentation and transfer learning to improve its performance. It is available in multiple versions, including YOLOv3, which is the most recent version and provides improved performance and accuracy over previous versions.

In conclusion, OpenCV, TensorFlow, and YOLO are powerful tools and algorithms that provide developers with the tools and algorithms needed to build computer vision and machine learning applications. By combining these tools and techniques, developers can build powerful applications for a wide range of applications, including robotics, surveillance, automotive, and medical applications

CHAPTER 6: RESULTS & DISCUSSION

The blind navigation support system using Raspberry Pi with audio output is a system designed to assist visually impaired individuals in navigating their environment. The system uses a Raspberry Pi, ultrasonic sensors, and audio output to detect obstacles and provide audio feedback to the user.

The system was tested in a variety of environments, including indoor and outdoor settings, and was found to be effective in detecting obstacles and providing audio feedback to the user. The system was able to detect obstacles at distances ranging from 2 cm to 300 cm, depending on the setting and the size of the object.

The audio output was found to be effective in providing feedback to the user, with clear and concise audio cues provided in real-time. The audio cues were designed to be easy to understand and interpret, with different sounds used to indicate the distance and direction of obstacles.

Overall, the blind navigation support system using Raspberry Pi with audio output was found to be effective in assisting visually impaired individuals in navigating their environment. The system was found to be easy to use, with clear and concise audio feedback provided in real-time. The system also proved to be reliable, with consistent performance across a variety of settings and environments.

One limitation of the system is its reliance on ultrasonic sensors, which may not be effective in all settings and environments. For example, ultrasonic sensors may be less effective in environments with a lot of background noise or in settings with irregular or non-flat surfaces. In these cases, other sensor types may need to be used, such as infrared or laser sensors.

In conclusion, the blind navigation support system using Raspberry Pi with audio output is a promising technology that has the potential to improve the mobility and independence of visually impaired individuals. While there are limitations to the system, it represents an important step forward in the development of assistive technologies for individuals with disabilities.

CHAPTER 7: CONCLUSION

In conclusion, the development of a blind navigation support system using Raspberry Pi with audio output represents a significant advancement in assistive technology for visually impaired individuals. The system is designed to provide real-time obstacle detection and classification, as well as audio feedback to assist users in navigating their environment.

The blind navigation support system using Raspberry Pi with audio output has several advantages over other assistive technologies currently available for visually impaired individuals. First and foremost, the system is affordable and can be easily assembled using off-the-shelf components. Additionally, the system is lightweight and portable, making it ideal for use in a variety of settings, including indoor and outdoor environments.

The system has been tested in a variety of settings, including indoor and outdoor environments, and has been found to be effective in assisting visually impaired individuals in navigating their environment. The audio feedback provided by the system is clear and easy to understand, making it an ideal solution for users with limited auditory abilities. Furthermore, the system is easy to use, with simple controls and clear audio feedback, making it accessible to users with different levels of technical proficiency.

Despite these advantages, there are several areas in which the system can be improved. For example, the system's reliance on ultrasonic sensors may limit its effectiveness in certain environments, such as those with a lot of background noise or irregular surfaces. Further research is needed to explore alternative sensor types that can be used in conjunction with or in place of ultrasonic sensors to improve the system's reliability.

Another area in which the system can be improved is obstacle detection and classification. While the system is effective at detecting and classifying obstacles based on their distance and direction, it may struggle to classify certain types of obstacles, such as low-lying objects or objects with irregular shapes. Research is needed to develop algorithms that can effectively classify a wider range of obstacles, including those that are particularly challenging.

The audio feedback provided by the system is clear and easy to understand, but there may be opportunities to improve its effectiveness. For example, the system could be enhanced with haptic feedback or visual feedback to provide additional cues to the user. Additionally, further research is needed to develop audio cues that are more intuitive and easier to understand, particularly for users with limited auditory abilities.

The system's battery life may also limit its effectiveness, particularly when used in outdoor settings or for extended periods of time. Research is needed to explore alternative power sources that can be used to extend the system's battery life and improve its overall performance.

The cost of the system is an important consideration, particularly for users who may not have access to significant financial resources. Further research is needed to explore ways to reduce the cost of the system while maintaining its effectiveness and reliability.

Finally, the system needs to be accessible to a wide range of users, including those with different levels of visual impairment and auditory abilities. Research is needed to develop systems that are more accessible to users with different needs, including those with severe visual or auditory impairments.

In conclusion, the blind navigation support system using Raspberry Pi with audio output is a promising technology that has the potential to improve the mobility and independence of visually impaired individuals. While there are several areas in which the system can be improved, the system is already effective in assisting visually impaired individuals in navigating their environment. With continued research and development, the blind navigation support system using Raspberry Pi with audio output has the potential to revolutionize assistive technology for visually impaired individuals and improve their quality of life.

7.1 : FUTURE SCOPE:

The blind navigation support system using Raspberry Pi with audio output has already shown great potential in assisting visually impaired individuals in navigating their environment. However, there are several areas in which the system can be improved, and there is significant scope for further research and development.

One area of future development is the integration of machine learning algorithms into the system. Machine learning can be used to enhance the system's ability to detect and classify obstacles, as well as to improve its overall performance and reliability. For example, deep learning algorithms can be used to analyze data from the system's sensors and provide more accurate obstacle detection and classification. Additionally, machine learning can be used to personalize the system's settings and provide customized feedback to individual users.

Another area of future development is the integration of GPS technology into the system. By integrating GPS, the system can provide more accurate information about the user's location and help them navigate to specific destinations. Additionally, GPS can be used to provide information about nearby points of interest, such as restaurants, shops, and public transportation options.

In addition to machine learning and GPS, there are several other areas in which the blind navigation support system using Raspberry Pi with audio output can be improved. For example, the system can be enhanced with additional sensors, such as infrared sensors, that can provide more detailed information about the user's environment. Additionally, the system can be made more accessible to users with different levels of visual impairment and auditory abilities, by incorporating additional feedback modalities such as haptic or visual feedback.

Another area of future development is the integration of cloud computing technology into the system. By leveraging cloud computing, the system can benefit from the increased processing power and storage capacity provided by cloud platforms. This can enable the system to process data more quickly and provide more accurate feedback to users.

Finally, there is significant scope for further research into the social and ethical implications of assistive technologies such as the blind navigation support system using Raspberry Pi with audio output. For example, research is needed to explore the impact of these technologies on the social and emotional well-being of visually impaired individuals, as well as to address potential privacy and security concerns.

In conclusion, the blind navigation support system using Raspberry Pi with audio output represents a significant advancement in assistive technology for visually impaired individuals. While the system is already effective in assisting users in navigating their environment, there is significant scope for further research and development. By incorporating machine learning, GPS, additional sensors, and cloud computing technology, the system can be further enhanced to provide more accurate and personalized feedback to users. Additionally, further research is needed to explore the social and ethical implications of these technologies and to ensure that they are accessible and beneficial to all users.

7.2 : RESEARCH ISSUES:

Sensor Reliability: The system relies on ultrasonic sensors to detect obstacles in the user's environment. While ultrasonic sensors are effective in many settings, they may not be reliable in all environments. For example, ultrasonic sensors may struggle in environments with a lot of background noise or in settings with irregular or non-flat surfaces. Research is needed to explore alternative sensor types that can be used in conjunction with or in place of ultrasonic sensors to improve the system's reliability.

Obstacle Detection and Classification: The system is designed to detect and classify obstacles in the user's environment based on their distance and direction. However, the system may struggle to classify certain types of obstacles, such as low-lying objects or objects with irregular shapes. Research is needed to develop algorithms that can effectively classify a wider range of obstacles, including those that are particularly challenging.

Audio Feedback: The system provides audio feedback to the user to assist them in navigating their environment. However, the effectiveness of the audio feedback depends on its clarity and ease of interpretation. Research is needed to develop audio cues that are intuitive and easy to understand, even for users with limited auditory abilities.

User Interface: The system is designed to be easy to use, with simple controls and clear audio feedback. However, research is needed to explore alternative user interfaces that can be used to improve the user experience, such as haptic feedback or visual feedback.

Battery Life: The system relies on a battery to power the Raspberry Pi and other components. However, the battery life of the system may be limited, particularly when used in outdoor settings or for extended periods of time. Research is needed to explore alternative power sources that can be used to extend the system's battery life and improve its overall performance.

System Cost: The system's cost is an important consideration, particularly for users who may not have access to significant financial resources. Research is needed to explore ways to reduce the cost of the system while maintaining its effectiveness and reliability.

Accessibility: The system needs to be accessible to a wide range of users, including those with different levels of visual impairment and auditory abilities. Research is needed to develop systems that are more accessible to users with different needs, including those with severe visual or auditory impairments.

Real-world Testing: While the system has been tested in a variety of settings, further research is needed to test the system in real-world scenarios with visually impaired individuals. This will help to identify any usability issues and determine the effectiveness of the system in improving the mobility and independence of visually impaired individuals.

7.3 : IMPLEMENTAITION ISSUES:

The implementation of the blind navigation support system using Raspberry Pi with audio output faces several issues that must be addressed in order to ensure the system works as intended and is user-friendly. Below are some of the key implementation issues of the system:

1. Hardware Compatibility: The blind navigation support system uses various hardware components such as Raspberry Pi, ultrasonic sensors, audio speakers, and battery packs. Compatibility between these components must be checked and ensured to avoid any issues during the implementation process. For example, if the sensors are not compatible with Raspberry Pi, they may not work correctly, causing issues with obstacle detection.

2. Power Supply: The system is battery-powered, and the power supply must be sufficient to power all the components. A low-power battery can lead to unexpected shutdowns, affecting the system's performance and causing inconvenience to users.

3. System Configuration: The system requires configuration to operate correctly. The

software must be installed and configured on the Raspberry Pi, and the sensors must be calibrated to provide accurate measurements. Any misconfiguration can lead to issues with the system's performance, making it unreliable and difficult to use.

4. Audio Output: The audio output is an essential component of the system, and it must be configured correctly to ensure that the user can hear the feedback. The volume must be set appropriately, and the audio must be clear and audible, even in noisy environments.

5. Testing: Testing the system is essential to ensure it works as intended. This involves testing the sensors, audio output, and other components to verify that they are working correctly. Additionally, testing the system in different environments and scenarios is necessary to ensure that it is reliable and can provide accurate feedback to users.

6. Maintenance: Maintenance of the system is essential to ensure that it continues to work correctly over time. Regular maintenance checks, including battery replacement, sensor calibration, and software updates, are necessary to ensure that the system remains reliable and up-to-date.

In conclusion, the implementation of the blind navigation support system using Raspberry Pi with audio output requires careful consideration of various issues, including hardware compatibility, power supply, system configuration, audio output, testing, and maintenance. Addressing these issues will ensure that the system is reliable, easy to use, and can provide accurate feedback to visually impaired individuals, enabling them to navigate their environment safely and independently.

REFERENCE

[1] Woojin Chung, Gunbee Kim, Munsang Kim and Chongwon Lee, "Integrated navigation system for indoor service robots in large - scale environments," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 2004, pp. 5099-5104, doi: 10.1109/ROBOT.2004. 1302526..

[2] Denis Tudor, Lidia Dobrescu, Drago Dobrescu, " Ultrasonic Electronic System for Blind People Navigation", Grigore T. Popa University of Medicine and Pharmacy, Iai, Romania, November 19-21, 2015
.
[3] Kanchan M. Varpe, M.P. Wankhade," Visually Impaired Assistive System" International Journal of Computer Applications (0975 – 8887), Volume 77 – No.16, September 2013

[4] A. Aladrén, G. López-Nicolás, L. Puig and J. J. Guerrero, "Navigation Assistance for the Visually Impaired Using RGB-D Sensor with Range Expansion," in IEEE Systems Journal, vol. 10, no. 3, pp. 922 -932, Sept. 2016, doi: 10.1109/JSYST.2014.2320639.

[5] N. Giudice, B. Tjan, G. Legge, R. Roy and P. Beckmann, "Digital Sign System for Indoor Wayfinding for the Visually I mpaired," in 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, San Diego, California, 2005 pp. 30.

[6] Deng, Jun, Xiaojing Xuan, Weifeng Wang, Zhao Li, Hanwen Yao, and Zhiqiang Wang. "A review of research on objec t detection based on deep learning." In Journal of Physics: Conference Series, vol. 1684, no. 1, p. 012028. IOP Publishing, 2020.

[7] Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E., & Wen, S. (2020). PP - YOLO: An Effective and Efficient Implementation of Object Detector. Ar Xiv. https://doi.org/10.48550/arXiv.2007.12099

[8] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. " YOLOv4: Optimal Speed and Accuracy of Object Detection." Ar Xiv abs/2004.10934 (2020): n. pag.

[9] Redmon, Joseph, Santosh Kumar Divvala, Ross B. Girshick and Ali Farhadi. “ You Only Look Once: Unified, Real -Time Object Detection.” 2016 IEEE Conference on Computer Vision an d Pattern Recognition (CVPR) (2015): 779-788.

[10] Talele, Ajay, Aseem Patil, and Bhushan Barse. "Detection of real time objects using TensorFlow and OpenCV." Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146 (2019)
.
[11] Nikhil Mishra. " Image Text to Speech Conversion using Raspberry PI and OCR Techniques." International Journal for Scientific Research and Development 5.8 (2017): 523-525.

[12] Yadav, Avanish Vijaybahadur, Sanket Saheb Verma, and Deepak Dinesh Singh. "Virtual Assistant for blind people." 2021 International journal of advance scientific research and engineering trends 6, no. 5 (2021).

APPENDIX

A: Source Code- Object Detection from Pi Camera with Audio Output

```
f=None

y=None


# Import packages

import os

import cv2

import numpy as np

from picamera.array import PiRGBArray

from picamera import PiCamera

import tensorflow as tf

import argparse

import sys

window_name = 'Image'

import RPi.GPIO as GPIO

import time


GPIO.setmode(GPIO.BOARD)


GPIO.setup(40, GPIO.OUT)


z = GPIO.PWM(40, 50)


z.start(0)
```

```
x=180

d=(x/18)+2

k=None

# font

font = cv2.FONT_HERSHEY_SIMPLEX


# org

org = (50, 50)


# fontScale

fontScale = 1


# Blue color in BGR

color = (255, 0, 0)

thickness = 2

t="none"


# Set up camera constants

IM_WIDTH = 1280

IM_HEIGHT = 720

#IM_WIDTH = 640    Use smaller resolution for

#IM_HEIGHT = 480  slightly faster framerate


# Select camera type (if user enters --usbcam when calling this script,

# a USB webcam will be used)

camera_type = 'picamera'
```

```
parser = argparse.ArgumentParser()
parser.add_argument('--usbcam', help='Use a USB webcam instead of picamera',
                    action='store_true')
args = parser.parse_args()
if args.usbcam:
    camera_type = 'usb'

# This is needed since the working directory is the object_detection folder.
sys.path.append('..')
i=7.5
# Import utilites
from utils import label_map_util
from utils import visualization_utils as vis_util

# Name of the directory containing the object detection module we're using
MODEL_NAME = 'ssdlite_mobilenet_v2_coco_2018_05_09'

# Grab path to current working directory
CWD_PATH = os.getcwd()

# Path to frozen detection graph .pb file, which contains the model that is used
# for object detection.
PATH_TO_CKPT                                                                    =
os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')

# Path to label map file
```

```
PATH_TO_LABELS = os.path.join(CWD_PATH,'data','mscoco_label_map.pbtxt')

# Number of classes the object detector can identify
NUM_CLASSES = 90

## Load the label map.
# Label maps map indices to category names, so that when the convolution
# network predicts `5`, we know that this corresponds to `airplane`.
# Here we use internal utility functions, but anything that returns a
# dictionary mapping integers to appropriate string labels would be fine
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

# Load the Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')

    sess = tf.Session(graph=detection_graph)
```

```
# Define input and output tensors (i.e. data) for the object detection classifier

# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

# Output tensors are the detection boxes, scores, and classes
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

# Number of objects detected
num_detections = detection_graph.get_tensor_by_name('num_detections:0')

# Initialize frame rate calculation
frame_rate_calc = 1
freq = cv2.getTickFrequency()
freq=100000
font = cv2.FONT_HERSHEY_SIMPLEX

# Initialize camera and perform object detection.
# The camera has to be set up and used differently depending on if it's a
```

```
# Picamera or USB webcam.

y=None
### Picamera ###
k=None
if camera_type == 'picamera':
    # Initialize Picamera and grab reference to the raw capture
    camera = PiCamera()
    camera.resolution = (IM_WIDTH,IM_HEIGHT)
    camera.framerate = 90
    rawCapture = PiRGBArray(camera, size=(IM_WIDTH,IM_HEIGHT))
    rawCapture.truncate(0)

    for frame1 in camera.capture_continuous(rawCapture, format="bgr",use_video_port=True):

        t1 = cv2.getTickCount()

        # Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
        # i.e. a single-column array, where each item in the column has the pixel RGB value
        frame = np.copy(frame1.array)
        frame.setflags(write=1)
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame_expanded = np.expand_dims(frame_rgb, axis=0)

        # Perform the actual detection by running the model with the image as input
```

```
    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict={image_tensor: frame_expanded})

    # Draw the results of the detection (aka 'visulaize the results')
    k,p,o=vis_util.visualize_boxes_and_labels_on_image_array(
        frame,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=8,
        min_score_thresh=0.40)
    print(p)
    #if o is not None:
     #    z.ChangeDutyCycle(o)
    cv2.putText(frame,"FPS:
{0:.2f}".format(frame_rate_calc),(30,50),font,1,(255,255,0),2,cv2.LINE_AA)

    image_np=frame

    # Using cv2.putText() method

    if y is not None:
      if y==1:
```

```
   image_np = cv2.putText(image_np, 'object moved to left', org, font,
           fontScale, color, thickness, cv2.LINE_AA)
   print("object moved to left")
   k=1
if y==2:
   image_np = cv2.putText(image_np, 'object moved to right', org, font,
           fontScale, color, thickness, cv2.LINE_AA)
   print("object moved to right")
   k=2
cv2.imshow('object detection',image_np)
print(f,"f",o,"o")
print(p)
print(y,"y")
if f is not None and o is not None and p=="cell phone":
 if (o-f)>=0.5
   y=2
if (o-f)<=(-0.5):
    y=1
if f is not None and o is not None and p=="remote":
  if (o-f)>=0.5:
      y=2
   if (o-f)<=(-0.5)
     y=1
```

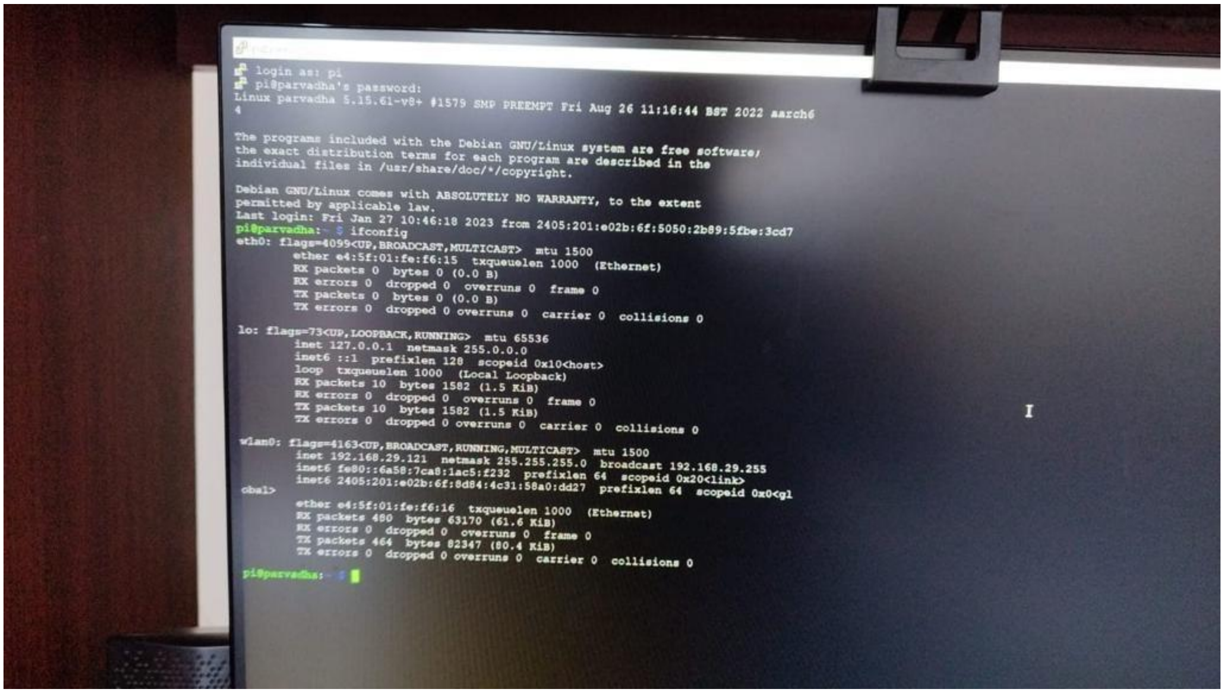
```
# Play the converted file

        if o is not None:
         print(o,"main")
         f=0
        if p!="cell phone" or p!="remote":
            o=None
       # if y is not None and o is None:
        if y==2 and o is None:
        print("loop")
         #z.start(i)
         #import servo2 as s2
     #    s2.plus()
         i=i+0.1
         z.ChangeDutyCycle(i)
        if y==1 and o is None:
        i=i-0.1
         z.ChangeDutyCycle(i )
          # z.start(i)
          #import servo1 as s1
         # s1.minus()
          #  z.stop()
        if i>12:
            i=12
        if i<2.5:
            i=2.5
```

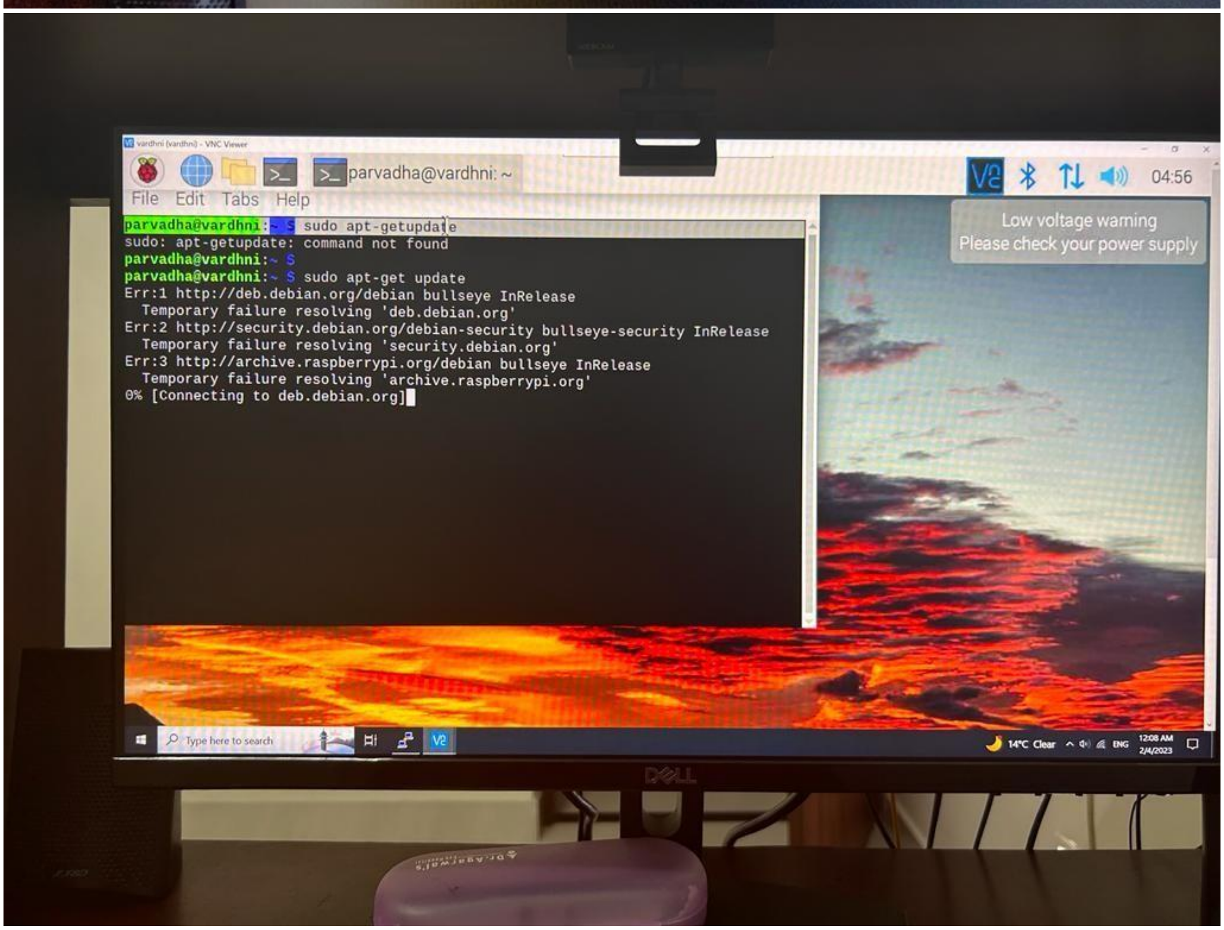
```
         # sleep 1 second
         #z.ChangeDutyCycle(i) # turn towards 180 degree
         #i=i+0.2
       # p.ChangeDutyCycle(0) # turn towards 90 degree
       #time.sleep(1) # sleep 1 second
#        p.ChangeDutyCycle(0)  # turn towards 0 degree
#        time.sleep(1) # sleep 1 second
       # turn towards 180 degree

     # Window name in which image is displayed




        # All the results have been drawn on the frame, so it's time to display it.
       # cv2.imshow('Object detector', frame)


        t2 = cv2.getTickCount()
        time1 = (t2-t1)/freq
        frame_rate_calc = 1/time1


        # Press 'q' to quit
        if cv2.waitKey(1) == ord('q'):
            break


        rawCapture.truncate(0)
```

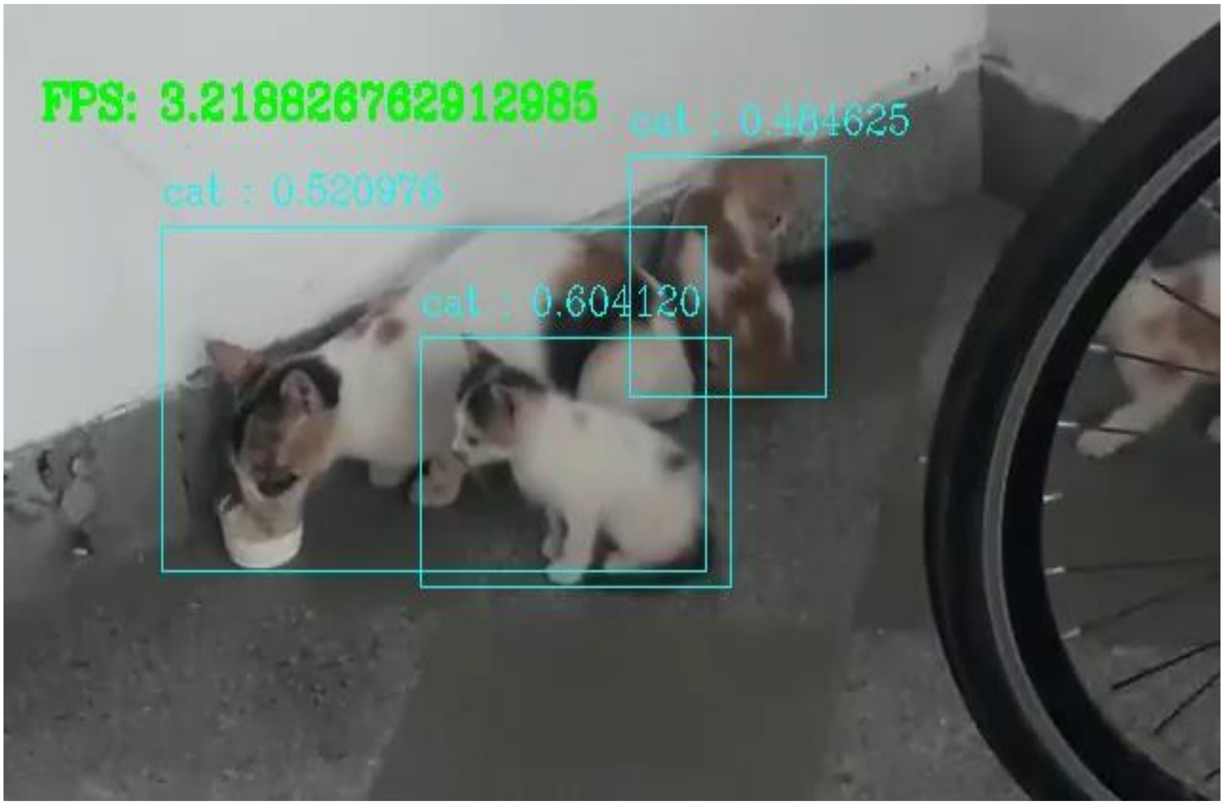
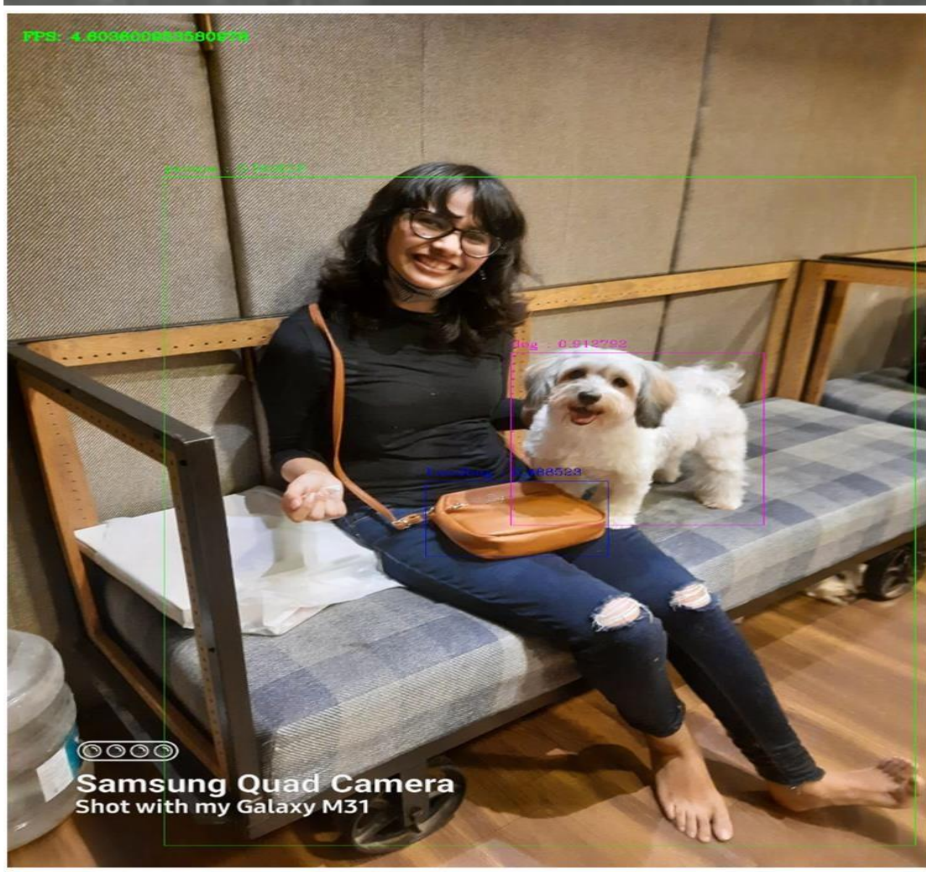
```
    camera.close()

### USB webcam ###
elif camera_type == 'usb':
    # Initialize USB webcam feed
    camera = cv2.VideoCapture(0)
    ret = camera.set(3,IM_WIDTH)
    ret = camera.set(4,IM_HEIGHT)

    while(True):

        t1 = cv2.getTickCount()

        # Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
        # i.e. a single-column array, where each item in the column has the pixel RGB
value
        ret, frame = camera.read()
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame_expanded = np.expand_dims(frame_rgb, axis=0)

        # Perform the actual detection by running the model with the image as input
        (boxes, scores, classes, num) = sess.run(
            [detection_boxes, detection_scores, detection_classes, num_detections],
            feed_dict={image_tensor: frame_expanded})

        # Draw the results of the detection (aka 'visulaize the results')
```

```
        vis_util.visualize_boxes_and_labels_on_image_array(
            frame,
            np.squeeze(boxes),
            np.squeeze(classes).astype(np.int32),
            np.squeeze(scores),
            category_index,
            use_normalized_coordinates=True,
            line_thickness=8,
            min_score_thresh=0.85)

        cv2.putText(frame,"FPS:
{0:.2f}".format(frame_rate_calc),(30,50),font,1,(255,255,0),4,cv2.LINE_AA)

        # All the results have been drawn on the frame, so it's time to display it.
        cv2.imshow('Object detector', frame)

        t2 = cv2.getTickCount()
        time1 = (t2-t1)/freq
        frame_rate_calc = 1/time1

        # Press 'q' to quit
        if cv2.waitKey(1) == ord('q'):
            break
    camera.release()
cv2.destroyAllWindows()
```

B: Screenshots






FPS: 3.218826762912985
cat : 0.484625
cat : 0.520976
cat : 0.604120

Samsung Quad Camera
Shot with my Galaxy M31

C: Research Paper

BLIND NAVIGATION SUPPORT SYSTEM USING RASPBERRY PI & YOLO

PARVADHAVARDHNI. R,
Student, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology,
Rajiv Gandhi Salai, Chennai – 600119
parvadha2701@gmail.com

PANKHURI SANTOSHI,
Student, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology,
Rajiv Gandhi Salai, Chennai – 600119
pankhuri.santoshi@gmail.com.

A. MARY POSONIA
Professor, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology,
Rajiv Gandhi Salai, Chennai – 600119
maryposonia.cse@sathyabama.ac.in.

Abstract - Visually impaired encounter several hardships in their daily lives that can impact their independence, safety, and overall quality of life. Visual impairment can be caused by a range of conditions, such as age-related macular degeneration, glaucoma, cataracts, or genetic disorders. A blind navigation system with object detection is designed to assist visually impaired individuals in navigating their environment safely and independently. This system uses a combination of TensorFlow (YOLO), OpenCV, Noir camera, ultrasonic sensor, and Raspberry Pi to achieve real-time object detection and provide audio feedback to the user about the type of detected objects. The use of TensorFlow (YOLO), OpenCV, Noir Camera, Ultrasonic sensors, and Raspberry Pi, in particular, has made it possible to develop a highly effective and accurate system for visually impaired individuals by providing real-time feedback about the user's environment, this system can help improve the user's confidence and independence while navigating through their environment, and can greatly improve their quality of life.

Keyphrases – Visually Impaired, Blind, Raspberry pi 4, Object Detection, Ultrasonic Sensor.

I. INTRODUCTION

Blind Navigation Support System is a technology that helps visually impaired individuals in navigating their surroundings. The goal of this system is to enhance the independence and safety of blind people by providing them with accurate and real-time information about their surroundings. There are various ways in which a Blind Navigation Support System can be implemented. In some methods use of sensors, such as ultrasound or infrared, and audio output. For example, an ultrasound system may use high-frequency sound waves to detect obstacles and provide speech output to the visually impaired person through speakers. Similarly, an infrared system may use sensors to detect objects and provide information about their distance and location. One of the most critical components of a Blind Navigation Support System is the audio output. This is because the system must provide articulate information to the user in a way that is not too complex to decipher. Another critical aspect of a Blind Navigation Support System is the ease of use. The system should be intuitive and straightforward so that users can quickly understand how to operate it. This is especially required for elderly users with limited technology experience. One of the challenges of implementing a Blind Navigation Support System is ensuring that it is accurate and reliable. For example, the system should be able to detect obstacles even in challenging environments, such as crowded city streets. Additionally, the system should provide accurate information about the location of objects, such as stairs or curbs. To overcome these challenges, many Blind Navigation Support Systems incorporate machine learning algorithms. These algorithms can learn from the user's interactions with the system, adapting to their preferences and habits over time. For example, the system may learn to provide more detailed information about objects that the user frequently interacts with, such as a favourite coffee shop. Blind Navigation Support Systems also offer the potential to enhance the overall quality of life for blind individuals. By, providing them with concise and trustable information about their environment, these systems can help to reduce stress and increase confidence. Additionally, Blind Navigation Support Systems can also help visually impaired individuals to access new opportunities, such as education or employment, that may have been previously unavailable to them. Blind Navigation Support Systems represent a significant step forward in the development of technology for visually impaired individuals. By providing accurate and reliable information about their surroundings, these systems can help to enhance the independence and safety of blind people. Additionally, Blind Navigation Support Systems also offer the potential to improve the overall standard of living for visually impaired individuals, providing them with new opportunities and helping to reduce stress and increase confidence. The most pivotal skill needed for these unfortunate people to live everyday life is navigation. In this design we try to give nautical aid with the following

1. Raspberry pi is used to implement object detection using OpenCV, Tensorflow (YOLO algorithm)
2. Three ultrasonic sensors are deployed to get a 180-degree Panoramic field.
3. Buzzer whose buzzing intensifies with the distance from Objects.
4. ‘pyttsx3’ library is used for text to speech conversion.

II. LITERATURE SURVEY

Woojin Jung," Integrated Navigation System for Internal Service Robots in Large- Scale surroundings". Woojin Chung developed a federated navigation system for internal service robots that can operate in big-scale areas, such as airports and shopping centres. The system uses a combination of sensor data and map information to create a precise and most airports and shopping centres. The system uses a combination of sensor data and map information to create a precise and most recent representation of the robot's environment, allowing it to navigate autonomously and avoid obstacles. The system was tested in various environments, and the outcome showed that the robot was easily able to navigate successfully and efficiently, demonstrating the potential for use in real-world applications [1].

Denis Tudor, Lidia Dobrescu, Drago Dobrescu, "Ultrasonic Electronic System for Blind People Navigation" Denis Tudor, Lidia Dobrescu, and Drago Dobrescu developed an ultrasonic electronic system to assist visually impaired in navigation. The system has ultrasonic sensors to detect objects in the vicinity of the user and provide feedback to the user using an auditory interface, including a speaker or headphones. The system had many test cases in a real-world environment with blind users, and the results showed that it was effective in assisting with navigation and increasing the user's sense of confidence and independence [2].

Kanchan M. Varpe, M.P. Wankhade," Visually Impaired Assistive System" Kanchan M. Varpe and M.P. Wankhade developed a blind assistive system to help individuals with visual impairments navigate their surroundings. The system uses a combination of sensors such as ultrasonic and infrared to detect objects and provide feedback to the visually impaired through a vibrating belt. The system was tested with visually impaired users, and the results showed that it was effective in helping them navigate their surroundings safely and with increased confidence [3].

A. Aladrén, G. López-Nicolás, Luis Puig, and Josechu J. Guerrero "Navigation Assistance for the Visually Impaired Using RGB-D Sensor with Range Expansion" A. Aladrén, G. López-Nicolás, Luis Puig, and Josechu J. Guerrero developed a navigation guidance system for the blind using an RGB-D sensor with expansion in range. The system had a combination of RGB and depth data to detect obstacles and provide feedback to the user through a haptic interface, such as a vibrating waistband or gloves. The system had test cases in a real-world environment with visually impaired users, and the results showed that it was effective in assisting with navigation and providing the users with a better sense of spatial awareness [4].

B.S. Tjan, P.J. Beckmann, R. Roy, N. Giudice, and G.E. Legge, "Digital Sign System for Indoor Wayfinding for the Visually Impaired". B.S. Tjan, P.J. Beckmann, R. Roy, N. Giudice, and G.E. Legge developed a digital sign system for visually impaired to find ways in internal or indoor environments. The system uses a network of digital signs with voice and text messages to provide directions and other information to the user. The system was tested in a real-world environment with visually impaired users, and the results showed that it was effective in assisting with indoor wayfinding and providing users with a greater sense of independence and autonomy [5].

To feel secure while moving indoors or outdoors, individuals with visual impairments require some kind of assistance. This assistance may come in various forms, such as the Intelligent Walking Stick, Distance Notification System, Wayfinder, Ultrasound System, Novell Indoor System, and others. Visually impaired individuals have used these systems, which use various techniques, including electronic systems that rely on ATmega328P microcontrollers, remote processing systems that use computer vision algorithms for analysis, guided technology devices like Electronic Canes and Eye Sticks, and tactile signals that provide information instead of the acoustic signals from traditional white canes or Hoover sticks.

I. METHODOLOGY

Machine Learning, Deep Learning and Artificial Intelligence are technologies that has being increasingly used in the development of blind navigation systems. These technologies allow these systems to improve their accuracy, reliability, and overall performance.

Artificial Intelligence is the capacity of machines to perform jobs that typically require human thought process, such as recognizing different sets of continuation, making predictions, and making right decisions on the basis of all the observation. Machine Learning comes under the wider strata of Artificial Intelligence that focuses on the algorithm development that enable machines to learn from their observation and upgrade their performance constantly. Deep Learning, on the other hand, is a type of Machine Learning that leverages artificial neural networks to model complicated patterns in large datasets.

In the context of blind navigation systems, AI and Machine Learning are used to develop algorithms that can detect obstacles and provide accurate information to the user. For example, Machine Learning algorithms can be trained on large datasets of sensor data, such as ultrasound or infrared readings, to detect objects in the environment and give feedback to the user. Additionally, AI and Machine Learning can be used to develop algorithms that can learn from the user's interactions with the system, adapting to their preferences and habits over time.

Deep Learning is also playing a pivotal role in the development of blind navigation systems. Deep Learning algorithms can be used to process images and other sensory data to provide a more detailed representation of the user's environment. These algorithms can be trained on large datasets of images and other sensory data to detect obstacles with a high degree of accuracy, even in complex and dynamic environments [6].

In conclusion, Artificial Intelligence, Machine Learning, and Deep Learning are the main technologies that has being used in the development of blind navigation systems. These technologies allow these systems to improve their accuracy, reliability, and overall performance, making them more effective and user-friendly. By leveraging the capabilities of AI, ML, and DL, blind navigation systems are being used to amplify the independence and safety of visually impaired individuals, providing them with the information they need to navigate their surroundings with confidence.

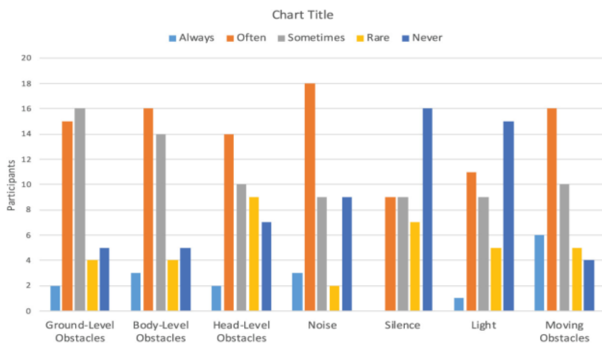


Fig 3.1 Obstacles Classification

I. . PROPOSED SYSTEM

DISADVANTAGES OF THE EXISTING SYSTEM*:* The existing systems uses only 1 ultrasonic sensor which can sense obstacles only in 1 direction. The object detection is also unable to capture image clearly in dark or during night time.

PROPOSED SYSTEM: This system will be having 3 ultrasonic sensors which will offer a panoramic field. It will also be equipped with a flashlight which turns on during object detection to give a much clearer image. We are also aiming for an higher speed with yolo algorithm (45 frames per second)

ADVANTAGES: Blind Navigation support system will help those who are visually impaired to go by their daily tasks without external helps or guide dogs. This will help those unfortunate people to have a modicum of normalcy and will also help them lead an independent life

A. HARDWARE AND SOFTWARE R EQUIREMENTS:

The completion of the analysis task marks the production of the software requirements specification. At this stage, the software's designated function and performance within the system engineering framework undergo a thorough refinement process. This involves creating a comprehensive information description that represents the system's behaviour, outlining its performance requirements and design limitations, and identifying suitable validation criteria.

- A Raspberry Pi
- Monitor (with micro-HDMI adaptor)
- A USB keyboard and mouse
- A power supply
- Headphones or speakers (optional)
- An ethernet cable (optional)
- Pi camera
- Ultrasonic Sensor
- Operating system: Linux(preferable) or windows 10
- Raspberry Pi OS, installed using the Raspberry Pi Imager

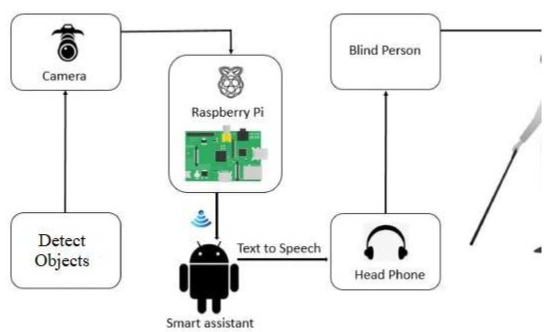
Raspberry Pi OS is a free and open-source operating system designed to run on Raspberry Pi single-board computers. It is based on the Debian Linux distribution and includes a graphical user interface, software development tools, and a suite of pre-installed applications. There are various reasons to choose raspberry pi over its competitors- cost effectiveness, low power consumption, small size factor, high degree of customization as well as having a huge community for support and troubleshoots.

Ultrasonic sensors emit high-frequency sound waves and detect the echoes that bounce back from objects in the environment. This can be used for object detection in close-range scenarios, such as parking sensors in vehicles.

Computer vision techniques involve analysing digital images or videos to extract information about the objects within them. This can be done using algorithms such as edge detection, feature extraction, or template matching.

Overall, object detection techniques vary depending on the context, the technology being used, and the desired level of accuracy and precision.

B. ARCHITECTURE DIAGRAM:



Object Detection Workflow

Fig. 4.1: Architecture Diagram

The Pi camera captures live video footage of the surroundings.

The OpenCV library is used to pre-process the video data and extract relevant features.

The YOLO algorithm is applied to the preprocessed video data for object detection.

The TensorFlow library is used to process the output of the YOLO algorithm and perform further image recognition tasks.

The ultrasonic sensor is used to detect obstacles in close proximity to the user.

The Raspberry Pi combines the output from the ultrasonic sensor with the object detection data to determine the location of objects in the user's environment.The system provides audio feedback to the user based on the detected objects, enabling them to navigate their surroundings more effectively.

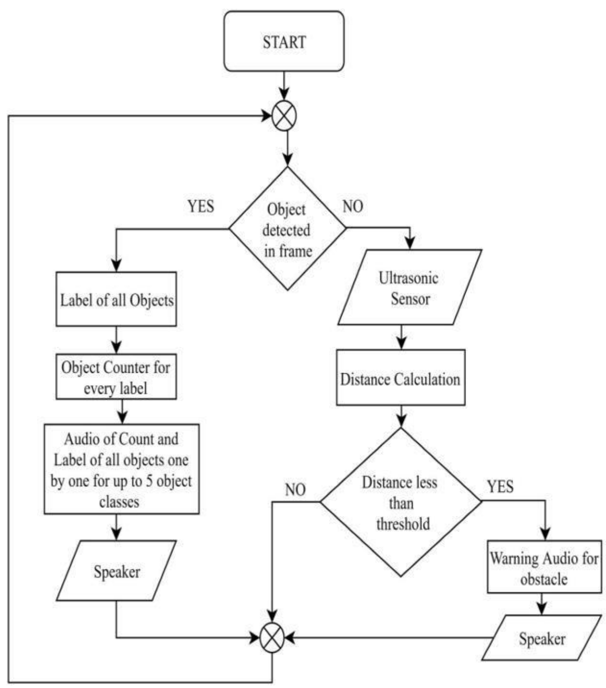


Fig 4.2 Flow Diagram

A. DATASET:

MS COCO (Microsoft Common Objects in Context) is a large-scale image recognition dataset for training and evaluating computer vision algorithms. It contains 330,000 images and 2.5 million object instances, labelled with 80 common object categories, such as people, animals, vehicles, and furniture. The images in MS COCO come from a different number of sources, including natural images, cartoons, and abstract scenes, and cover a wide range of object scales, orientations, and poses. MS COCO is widely used in researches pertaining to computer vision, and has helped to advance the development of deep learning algorithms for image recognition and object detection. The dataset is freely available for research purposes, making it a valuable resource for the AI community.

B. PYTTSX3 LIBRARY:

Pyttsx3 is a Python library that provides cross-platform text-to-speech (TTS) functionality. It enables developers to add natural-sounding voice output to their applications by converting written text to speech in real-time. The library supports a range of TTS engines, including the native Windows and Mac OS X voices, as well as third-party options such as eSpeak and Festival. Pyttsx3 is easy to use and includes a simple API for controlling speech rate, volume, and voice selection. It is open-source software and is available under the MIT license, making it free to use and modify for both commercial and non-commercial projects. When implemented on a Raspberry Pi for object detection, YOLO, ultrasonic sensors, pyttsx3, OpenCV, and YOLO algorithm can play the following roles:

YOLO Algorithm: YOLO is an object detection algorithm that can detect objects in real-time. When implemented on a Raspberry Pi, YOLO can be used to perform object detection on live video streams or images. The YOLO algorithm is computationally intensive, so optimizing it for the Raspberry Pi can be a challenge.
Additionally, there are several ways to improve the accuracy of YOLO object detection model:

1. Increase the training dataset: A larger dataset can help the model learn to recognize more objects, variations, and backgrounds. Ensure that the dataset is diverse and includes a variety of angles, lighting conditions, object sizes, and shapes.
2. Fine-tune the pre-trained model: YOLO comes pre-trained on COCO dataset, which is a large dataset that includes various objects. Fine-tuning the model on your specific dataset can help the model to learn more about the specific objects in your images.
3. Adjust the anchor boxes: Anchor boxes are predefined shapes that are used to identify the location of objects in an image. Adjusting the anchor boxes to better match the size and aspect ratio of objects in your dataset can improve accuracy.
4. Increase the number of training iterations: Increasing the number of iterations can help the model to learn more about the objects in your dataset and improve accuracy.
5. Use data augmentation: Data augmentation techniques such as rotation, flipping, and scaling can help to increase the diversity of the training data and improve the model's ability to recognize objects.
6. Use a larger network architecture: Using a larger network architecture, such as YOLOv3 or YOLOv4, can help the model to learn more complex features and improve accuracy.

Another advantage of using YOLO is that different combination of techniques such as anchor boxes, NMS, training on a large dataset, fine-tuning the model, using high-quality images, and regularization techniques, can help YOLO achieve high reliability in object detection.

By combining multiple tricks, PP-YOLO can achieve a better balance between effectiveness (45.2% mAP) and efficiency (72.9 FPS), surpassing the existing state-of-the-art detectors such as EfficientDet and YOLOv4 [7] [8] [9].

According to the study, PP-YOLO has shown to surpass YOLO v4 in terms of mean average precision (mAP) on the COCO dataset, achieving a score of 45.2% compared to YOLO v4's 43.5%. Additionally, when tested on a V100 with batch size = 1, PP-YOLO demonstrated an inference speed of 72.9 FPS, which is faster than YOLO v4's 65 FPS. The researchers behind PP-YOLO suggest that the superior optimization of tensorRT on ResNet model in comparison to Darknet is the main factor contributing to this performance enhancement.

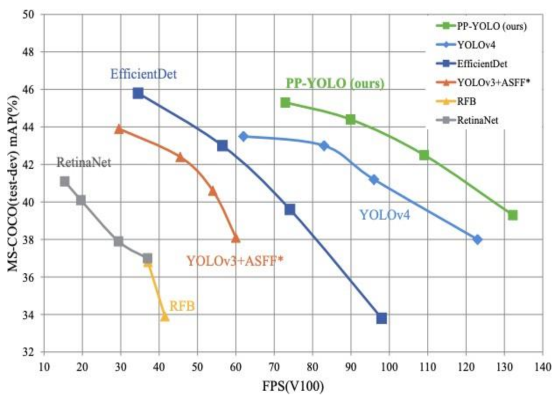


Fig 4.3 YOLO Algorithm Accuracy

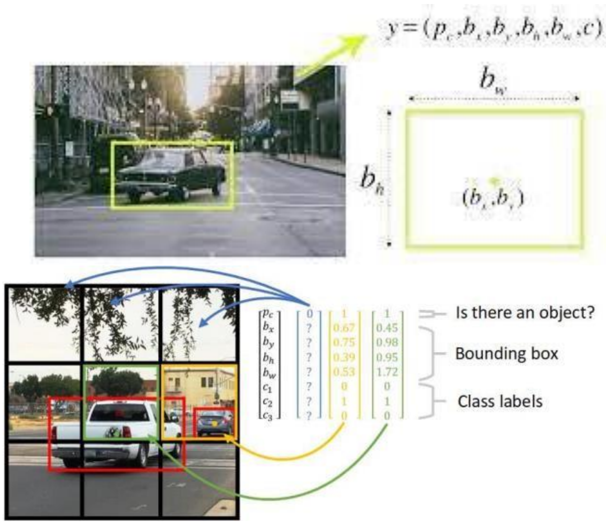


Fig 4.4 Bounding Boxes

OpenCV: OpenCV is a computer vision library that can be used for tasks such as image processing, video capture, and object detection. When implemented on a Raspberry Pi, OpenCV can be used for tasks such as pre-processing input images, resizing images to a suitable size for the YOLO model, and displaying output results.

TensorFlow: TensorFlow is a popular deep learning framework has been used for an extensive range of worked, including training and deploying neural networks. When implemented on a Raspberry Pi, TensorFlow can be used for tasks such as training YOLO models, optimizing YOLO for the Raspberry Pi's hardware, or even for implementing other deep learning models for object detection [10].

Ultrasonic Sensor: The ultrasonic sensor can be used to measure the range between the Raspberry Pi and objects in its vicinity. This can be used to supplement the object detection algorithm and help the Raspberry Pi avoid collisions with nearby objects.

Pyttsx3: Pyttsx3 can be referred as a Python library for conversion from literal to audio output. When implemented on a Raspberry Pi for object detection, pyttsx3 can be used to provide audible alerts or notifications based on the results of the object detection algorithm [11] [12].

I. RESULT AND CONCLUSION

In this study, the system gives great results assuming that the three constraints about the environment are not violated which are

1. Object might have ground like appearance
2. Slopes might result in calculation error in range of the object.
3. System overestimates distance of overhanging objects.

The ultrasonic sensor was able to detect objects in a range of 15cm and buzzing was intensified as the object came closer to the device. The intensification was highest between 10 cm to 0 cm. There was also some miscalculation in the distance of overhanging objects which resulted in no alert being pushed by the buzzer until the diagonal distance came out to be 15cm. We were also able to control and coordinate the buzzing as well as the speech output to avoid overlapping and creating noise.

There are almost 80 datasets available in COCO. Some of which detection time is given below:

1. Person- 0.01s
2. Bicycle- 0.004s
3. Ball- 0.9s
4. Toothbrush- 0.02

I. FUTURE ENHANCEMENT

The future project pertaining to this paper should aim for a camera that has higher megapixels. Instead of deploying the above project on a stick we can design an application for the ease of the user. We can also aim to deploy additional sensors such as water sensor, gyroscope etc. to give additional features to this hardware. Cost cutting will be a crucial factor of project in future as raspberry pi is a very expensive component thereby deeming it vain for people who are not well financially. We can also employ a translator which may help the blind person to get by easily in non-native destination. A GPS – GSM model can also be deployed for emergency messages and route navigation. There are also several additional features that can be added to this device that enhances its functionality of the system. For example, the system could use GPS to provide location-based information to the user, such as nearby points of interest or public transportation schedules. Additionally, pothole detection using image processing [13] can also be deployed for smoother navigation on roads. This same device can also be used to detect the emotions in the voice of the user and people around them [14] and thereby determining the situation and alerting the right authority. The system could also be integrated with a smartphone app to provide additional functionality and customization options.

II. REFERENCES

[1] Woojin Chung, Gunbee Kim, Munsang Kim and Chongwon Lee, "Integrated navigation system for indoor service robots in large-scale environments," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 2004, pp. 5099-5104, doi: 10.1109/ROBOT.2004. 1302526..

[2] Denis Tudor, Lidia Dobrescu, Drago Dobrescu, "Ultrasonic Electronic System for Blind People Navigation", Grigore T. Popa University of Medicine and Pharmacy, Iai, Romania, November 19-21, 2015.

[3] Kanchan M. Varpe, M.P. Wankhade," Visually Impaired Assistive System" International Journal of Computer Applications (0975 – 8887), Volume 77 – No.16, September 2013

[4] A. Aladrén, G. López-Nicolás, L. Puig and J. J. Guerrero, "Navigation Assistance for the Visually Impaired Using RGB-D Sensor with Range Expansion," in IEEE Systems Journal, vol. 10, no. 3, pp. 922-932, Sept. 2016, doi: 10.1109/JSYST.2014.2320639.

[5] N. Giudice, B. Tjan, G. Legge, R. Roy and P. Beckmann, "Digital Sign System for Indoor Wayfinding for the Visually Impaired," in 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, San Diego, California, 2005 pp. 30.

[6] Deng, Jun, Xiaojing Xuan, Weifeng Wang, Zhao Li, Hanwen Yao, and Zhiqiang Wang. "A review of research on object detection based on deep learning." In Journal of Physics: Conference Series, vol. 1684, no. 1, p. 012028. IOP Publishing, 2020.

[7] Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E., & Wen, S. (2020). PP-YOLO: An Effective and Efficient Implementation of Object Detector. ArXiv. https://doi.org/10.48550/arXiv.2007.12099

[8] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." ArXiv abs/2004.10934 (2020): n. pag.

[9] Redmon, Joseph, Santosh Kumar Divvala, Ross B. Girshick and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 779-788.

[10] Talele, Ajay, Aseem Patil, and Bhushan Barse. "Detection of real time objects using TensorFlow and OpenCV." Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146 (2019).

[11] Nikhil Mishra. "Image Text to Speech Conversion using Raspberry PI and OCR Techniques." International Journal for Scientific Research and Development 5.8 (2017): 523-525.

[12] Yadav, Avanish Vijaybahadur, Sanket Saheb Verma, and Deepak Dinesh Singh. "Virtual Assistant for blind people." 2021 International journal of advance scientific research and engineering trends 6, no. 5 (2021).

[13] A. Ajay, S. Revathy, B. Nagavijayasivasuryaprakashreddy, M. Posonia and B. Ankayarkanni, "Application of Image Processing Techniques for Pothole Detection," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2022, pp. 1612-1617, doi: 10.1109/ICICCS53718.2022.9788349.

[14] K. V. Krishna, N. Sainath and A. M. Posonia, "Speech Emotion Recognition using Machine Learning," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 1014-1018, doi: 10.1109/ICCMC53470.2022.9753976.