IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134175

Step Simple – Guiding the Visually Challenged

Karthik Ganesh[1], Advaith Prasad[2], Mohammed[3], Mrs. Ashwitha Shetty[4]

Student, Computer Science and Engineering, AJIET, Mangalore, India[1-3]

Assistant Professor, Computer Science and Engineering, AJIET, Mangalore, India[4]

Abstract: This project presents the design and implementation of a smart blind stick prototype aimed at enhancing the mobility and safety of visually impaired individuals. The Blind Stick integrates an ultrasonic sensor, USB web camera, speakers, to provide object detection, The prototype leverages an Raspberry pi microcontroller to efficiently manage the sensor data and interactions. The ultrasonic sensor is employed to detect obstacles in the user's path, triggering a speak out to warn the user of potential collisions. The integration of a switch allows the user to initiate an emergency alert. Upon pressing the switch, Additionally, the Blind Stick prototype capitalizes on computer vision techniques through the utilization of the YOLO (You Only Look Once) framework. Connected to a PC, the Blind Stick leverages a USB web camera to capture images. Detected objects are then identified using YOLO, and corresponding audio alerts are relayed to the user through the earphones, enabling the user to understand their surroundings more comprehensively.

Keywords: Embedded system. raspberry pi, ultrasonic sensor, YOLO

I. INTRODUCTION

The project introduces a groundbreaking initiative focusing on the design and implementation of a smart belt prototype named the Blind Stick. This innovative device is tailored to augment the mobility and safety of visually impaired individuals by integrating cutting-edge technologies. At its core, the Blind Stick utilizes a Raspberry pi microcontroller to efficiently manage various components, including an ultrasonic sensor, USB web camera, speakers. The combination of these elements enables the prototype to offer essential functionalities such as object detection, alert generation, and location sharing.

The ultrasonic sensor plays a pivotal role in enhancing user safety by detecting obstacles in the path of the visually impaired individual.

The Blind Stick prototype takes a significant leap forward by incorporating computer vision techniques, leveraging the YOLO (You Only Look Once) framework. Through the use of a USB web camera connected to a PC, the Blind Stick captures images, and YOLO identifies detected objects. The user is then provided with audio alerts through earphones, offering a more comprehensive understanding of their surroundings.

II. PROBLEM STATEMENT

Visually impaired individuals encounter limitations in current blind assistance systems, primarily relying on auditory cues for navigating their environment. The absence of tactile feedback hinders their ability to perceive obstacles effectively, reducing their autonomy and safety. To address this challenge, the project aims to develop a Smart Haptic Feedback System within a blind assistance framework. This system will leverage cutting-edge technologies, including sensors and machine learning algorithms, to provide real-time tactile feedback, empowering visually impaired users to navigate their surroundings with enhanced awareness.

The specific problems to be addressed include existing blind assistance systems predominantly rely on auditory cues, lacking tactile feedback that could significantly enhance the user's spatial awareness. The project must overcome challenges in accurately detecting obstacles and providing precise haptic feedback, ensuring users can interpret the environment with confidence Designing machine learning models that adapt to users' navigation patterns and preferences, enhancing the system's effectiveness over time, presents a significant technical challenge.

Achieving real-time responsiveness in delivering haptic feedback is critical for the system's success, requiring optimizations to minimize latency and ensure timely obstacle detection. The system needs to adapt to diverse environments, including indoor spaces, outdoor areas, and public transportation, presenting challenges in creating a universally applicable solution.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134175

III. OBJECTIVES

The objectives of the blind assistance system is to enhance safety for the visually impaired. Here are key objectives for such a system:

1. Early Obstacle Detection
2. Precision and Reliability in Navigation
3. Real-time Alerts and Intervention
4. Adaptive Environmental Awareness
5. Seamless Integration with Wearable Technologies
6. Enhanced User Interaction and Feedback
7. Inclusive Design and User Empowerment

The Smart Blind Assistance System is designed with a set of key objectives to address the unique challenges faced by individuals with visual impairments. First and foremost, the system aims for Early Obstacle Detection, leveraging advanced algorithms to identify potential hazards promptly. This objective ensures that users receive timely alerts, fostering a proactive approach to navigating their surroundings. The second objective focuses on Precision and Reliability in Navigation, emphasizing the integration of cutting-edge sensors and algorithms to enhance the accuracy and dependability of obstacle detection, ultimately providing users with precise guidance. Real-time Alerts and Intervention form the third objective, ensuring that the system delivers immediate feedback to users about obstacles, allowing for quick and informed decision-making to enhance overall safety. The fourth objective, Adaptive Environmental Awareness, underscores the importance of the system's capability to adapt and comprehensively identify and mitigate potential risks, contributing to accident prevention. Seamless Integration with Wearable Technologies is the fifth objective, emphasizing the importance of a user-friendly and hands-free interaction, particularly with devices like smart glasses or haptic gloves. The sixth objective, Enhanced User Interaction and Feedback, highlights the development of an intuitive user interface that effectively communicates information about the surroundings through auditory or haptic feedback, ultimately enhancing users' awareness and confidence in navigating their environment. Lastly, Inclusive Design and User Empowerment is the seventh objective, emphasizing that the Smart Blind Assistance System aligns with accessibility standards and regulations, promotes inclusivity, and empowers users to navigate the world with increased independence and confidence. Collectively, these objectives aim to empower individuals with visual impairments, providing them with a reliable, adaptive, and user-centric solution for enhanced mobility.

IV. REQUIREMENT SPECIFICATION

Hardware Requirements

1. Computer: Windows 10
2. Processor/RAM: processor (e.g., Intel Core i5, AMD Ryzen)
3. Raspberry Pi board
4. Camera interface
5. Audio Jack

Software Requirements

1. Tool: PyCharm

With its many vital features for Python developers, firmly integrated to produce an environment that's easy for effective Python, web, and data science development, PyCharm is a dedicated Python Integrated Development Environment (IDE).

2. Tool: Jupyter Notebook

Jupyter Notebook is a powerful open-source tool for interactive computing and data analysis. Combining code,visualizations, and narrative text in a web-based environment, it supports various programming languages.

3. Language: Python

Since Python is the most popular programming language among software engineers, analysts, data scientists, and machine learning engineers, it will be the primary programming language used for this project. It also provides several robust libraries for building neural networks and image processing.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134175

V. SYSTEM DESIGN

A. Embedded System Hardware

As with any electronic system, an embedded system requires a hardware platform on which it performs the operation. Embedded system hardware is built with a microprocessor or microcontroller. The embedded system hardware has elements like input output (I/O) interfaces, user interface, memory and the display. Usually, an embedded system consists of:

- Power Supply
- Processor
- Memory
- Timers
- Serial communication ports

Embedded systems use different processors for its desired operation. Some of the processors used are

- Microprocessor
- Microcontroller
- Digital signal processor

B. Embedded System Software

The embedded system software is written to perform a specific function. It is typically written in a high level format and then compiled down to provide code that can be lodged within a non-volatile memory within the hardware. An embedded system software is designed to keep in view of the three limits:

- Availability of system memory
- Availability of processor's speed
- When the system runs continuously, there is a need to limit power dissipation for events like stop, run and wake up.

C. Bringing software and hardware together for embedded system:

To make software to work with embedded systems we need to bring software and hardware together .for this purpose we need to burn our source code into microprocessor or microcontroller which is a hardware component and which takes care of all operations to be done by embedded system according to our code.

Generally we write source codes for embedded systems in assembly language, but the processors run only executable files. The process of converting the source code representation of your embedded software into an executable binary image involves three distinct steps:

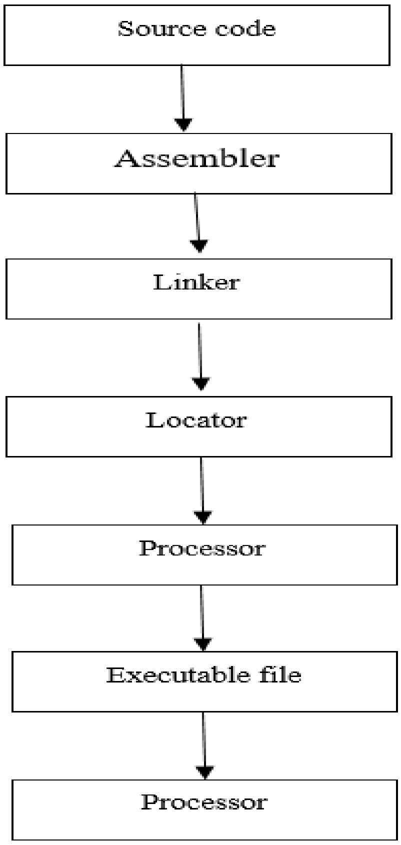
1.Each of the source files must be compiled or assembled into an object file.

2.All of the object files that result from the first step must be linked together to produce a single object file, called the e-locatable program.

3.Physical memory addresses must be assigned to the relative offsets within the re-locatable program in a process called relocation.

The result of the final step is a file containing an executable binary image that is ready to run on the embedded system.
The Figure shows the steps that are followed in the methodology of this project.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024
DOI: 10.17148/IJARCCE.2024.134175



The flowchart in Figure above outlines the Flow of burning source code to processor:

1. **Source Code:** This is where the process begins. Source code is the human-readable form of a program written in a programming language like C, C++, Java, etc.

2. **Assembler**: The source code is first passed through an assembler. The assembler converts the source code into machine code, which is a low-level representation of the program that can be understood by the computer's processor. This machine code is often referred to as object code.

3. **Linker:** The object code produced by the assembler may have references to functions or variables that are defined in other parts of the program or in external libraries. The linker is responsible for resolving these references by linking together various object files and libraries to create a single executable file.

4. **Locator**: In some systems, after linking, there might still be unresolved memory addresses or relocation information in the executable. The locator is a hypothetical step representing the process of resolving these addresses and preparing the executable for loading into memory.

5. **Executable File:** The output of the linker is an executable file. This file contains the machine code for the program, along with any necessary data and metadata, in a format that the operating system can understand.

6. **Processor**: Finally, the executable file is loaded into memory and executed by the processor. The processor interprets the instructions in the executable file and performs the necessary computations to run the program.

Each step in this process is crucial for converting the human-readable source code into instructions that the processor can execute. This diagram illustrates the flow of data and transformations as the program moves through each stage of compilation and execution.

D. ALGORITHM USED:

The algorithms used in this project:

- YOLO(You Only Look Once)

YOLO(You Only Look Once):

The YOLO (You Only Look Once) object detection system is a groundbreaking deep learning architecture known for its efficiency and accuracy in real-time object detection tasks. YOLO was introduced by Joseph Redmon and Ali Farhadi, presenting an innovative approach to object detection. Unlike traditional methods that use region proposal networks, YOLO frames object detection as a regression problem, directly predicting bounding boxes and class probabilities from a single neural network.One of the primary advantages of YOLO is its ability to process images incredibly fast, achieving impressive real-time performance. The architecture divides the input image into a grid and predicts bounding boxes and class probabilities within each grid cell. YOLO considers the entire image in one pass and directly outputs the bounding boxes with associated class probabilities, providing a near-instantaneous prediction for each object in the image.The YOLO architecture operates in two stages. The initial stage uses a convolutional neural network to extract features from the input image. Then, these features are utilized to predict bounding boxes and class probabilities. YOLO's output comprises bounding box coordinates, confidence scores, and class probabilities, enabling it to precisely locate and classify multiple objects within an image.

VI. OUTPUT

In Figure 1, it shows the person is detected by the system

Figure 1

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134175

In Figure 2, the page describes the detected object and distance from the system to the object

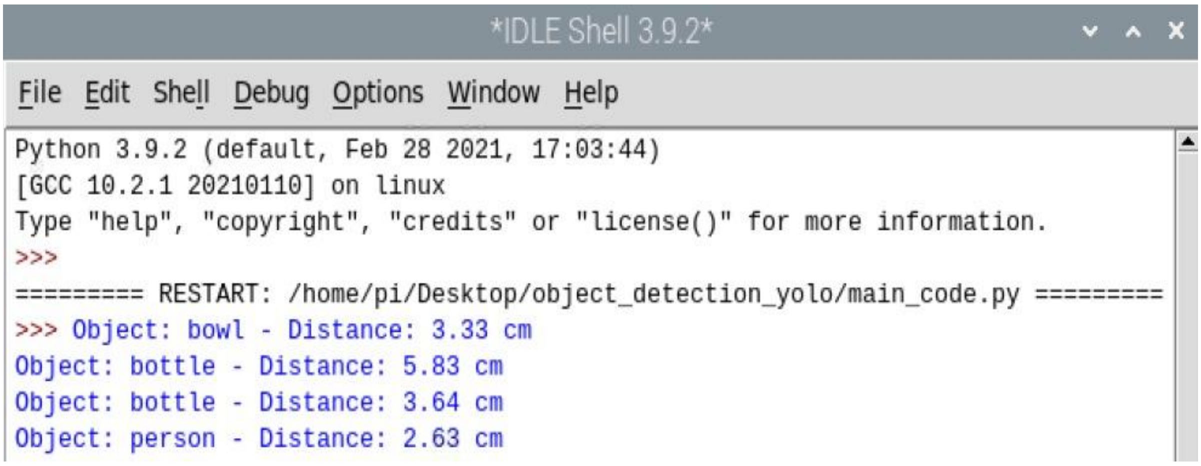


Figure 2

VII. CONCLUSION

In conclusion, the development and implementation of the Blind Stick smart belt prototype represent a significant leap forward in addressing the challenges faced by visually impaired individuals in terms of mobility and safety. By integrating advanced technologies such as ultrasonic sensors, a USB web camera, speakers, and a switch, the Blind Stick offers a multifaceted solution to enhance the independence and security of its users. The ultrasonic sensor plays a crucial role in detecting obstacles, providing real-time alerts through a buzzer to warn users of potential collisions. The incorporation of computer vision techniques, specifically the YOLO framework, adds an extra layer of functionality to the Blind Stick. By capturing and analyzing images from a USB web camera, the prototype can identify objects in the user's surroundings and convey this information through audio alerts, delivered via earphones. This not only improves object recognition but also contributes to a more comprehensive understanding of the environment. The collective integration of sensors and modules in the Blind Stick prototype underscores its potential to make a substantial impact on the lives of visually impaired individuals. The obstacle detection, emergency alert system, challenges faced by the visually impaired community. As with any innovative project, there is room for future enhancements. Potential areas of improvement include refining the prototype for increased usability, optimizing the object detection algorithm for greater accuracy.

VIII. FUTURE WORK

In future work, there are several improvements that can be made to make the system more functional and user friendly. First, faster object detection can be achieved by using a more powerful version of Raspberry Pi so that the system processes faster. Besides, higher resolution cameras or even highly developed camera modules can be upgraded to for better quality and accuracy in detecting objects. This will enable the system detect smaller objects or those at a further distance with more precision. Also various mobility aids may be integrated like a cap-mounted camera which could offer a less intrusive and easier way for people to move around their environments. With this kind of arrangement where the camera is attached to one's hat, they have freedom of natural movement of their heads thus making it comfortable for them while using such assistive technology. Such modifications would foster independence among users as well as wider adoption of these devices. Finally, intelligent decision-making algorithms might also form part of later versions of this system so as to help users know what step they should take next. In terms of spatial relationship between detected objects when analyzed by the program; it can give useful hints or cues that will guide users.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134175

REFERENCES

[1] J. Borenstein and Y. Koren, "Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance," IEEE Transactions on Robotics and Automation, 1985.

[2] S. Durgadevi, K. Thirupurasundari, C. Komathi, and S. Mithun Balaji, "Smart machine learning system for blind assistance," in International Conference on Power, Energy, Control and Transmission Systems, 2020.

[3] Xilin Chen, Jie Yang, Jing Zhang, and Alex Waibel, "Automatic detection and recognition of signs from natural scenes," IEEE Transactions on Image Processing, vol. 13, no. 1, January 2004.

[4] Sunit Vaidya, Niti Shah, Naisha Shah, and Prof. Radha Shankarmani, "Real-time object detection for visually challenged people," in International Conference on Intelligent Computing and Control Systems (ICICCS 2020),2020.

[5] Amit Ghosh, Shamsul Arefeen Al Mahmud, Thajid Ibna Rouf Uday, and Dewan Md. Farid, "Assistive technology for visually impaired using TensorFlow object detection in Raspberry Pi," in 2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020.

[6] Megha P. Arakeri, Keerthana N. S., Madhura M., Anusha Sankar, and Tazeen Munnavar, "Assistive technology for the visually impaired using computer vision," in International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2018.

[7] Sunil Kumar, Dibya Nandan Mishra, Shahid Mohammad Ganie, R. Bharathikannan, and K. Vijayakanthan, "Artificial intelligence solutions for the visually impaired.",2023.

[8] Jiaji Wang, Shuihua Wang, Yudong Zhang, "Artificial intelligence for visually impaired," Volume 77, April 2023.

[9] Risheng Liu, Zhu Liu, Jinyuan Liu, Xin Fan, and Zhongxuan Luo, "A task-guided, implicitly-searched and metainitialized deep model for image fusion," Vol. 14, No. 8, August 2015

[10] E Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, Ping Luo, "Diffusion model for dense visual prediction," IEEE Xplore, pp. 21741-21752, 2023.