SightMate

Josh Joung, Shakthi Angou, Meera Pandya

Department of Electrical and Computer Engineering,
Carnegie Mellon University

***Abstract*—**A system capable of providing an automated wearable navigation system that will alert the user of obstacles in their vicinity along with the optional functionality of detecting the object. The detection will primarily focus on common indoor objects to provide visually impaired people the ability to explore unknown indoor spaces without the need of guide dogs. The system will broadly consist of an object recognition model, text-to-speech engine, and depth sensors tied together by an on-board computer to create a seamless navigation experience that will hopefully be a more accessible alternative to sighted guides.

***Index Terms*—**Distance Estimation (DE) feature, espeak, HC-SR04 Ultrasonic Sensor, NVIDIA Jetson Nano, Object Recognition (OR) model, PCB, pyttsx3, Yolov5

I. Introduction

Using guide dogs is one of the most common methods for visually impaired people to navigate through the obstacles and reach the destination. However, the problem occurs when the users are incapable of raising guide dogs due to the operation cost and unavailability. This project is attempting to address the inefficiency and inaccessibility in using these trained dogs for visually impaired people. Guide dogs are generally very costly to train and maintain, and sometimes unavailable in an indoor setting. In addition, they can be difficult for visually-impaired people to care for. The project aims to offer an accessible alternative for guide dogs, creating a wearable device to aid in maneuvering around obstacles for visually-impaired people. Specifically, the product will be an automated navigation system that indicates to the user when they are approaching objects in front of them. It will let the users notice whatever object is in their pathways, similar to what guide dogs would do but with a much cheaper and easily maintainable solution. This product will be used along with a cane, which is the most commonly used assistive device for the visually-impaired. There will be a restriction to the scope of this project to well-lit indoor spaces with minimal to medium-level object crowding.

II. Use-Case Requirements

There are multiple requirements for the use case for the visually impaired people. Since our device will be an affordable alternative to guide dogs, some of our requirements use guide dog qualifications as a baseline metric.

1. Battery duration

The first requirement is sufficient battery duration. It should operate for a minimum of 4 hours because a guide dog usually takes a break every 4 hours. People also typically spend less than 4 hours exploring around indoor settings.

2. Accuracy of the object recognition model

A relatively high accuracy of the object recognition model is necessary for visually impaired people to utilize the product. The minimum qualification to become a guide dog is 70%, so the project is aiming for the minimum accuracy to be 70%. Yet, the objective accuracy is 80% to ensure the user's safety and usability.

3. Detection distance

The detection distance should be 2 meters to give users enough time and distance to avoid an obstacle once the alert is triggered.

4. Weight of the product

The weight of the product will be no more than 450 grams because the users will be wearing it as a neck device. To further alleviate neck stress, a battery pack may be offloaded to the waist if necessary.

5. Recognition delay

The average walk speed of blind pedestrians is 0.8 m/s, so the upper limit of the recognition delay should be less than 2.5 seconds to permit 2-meter space.

6. Noise detection

Regardless of the audio device in the product, the users should be able to hear surrounding noises regardless to ensure safety and reduce danger concerns.

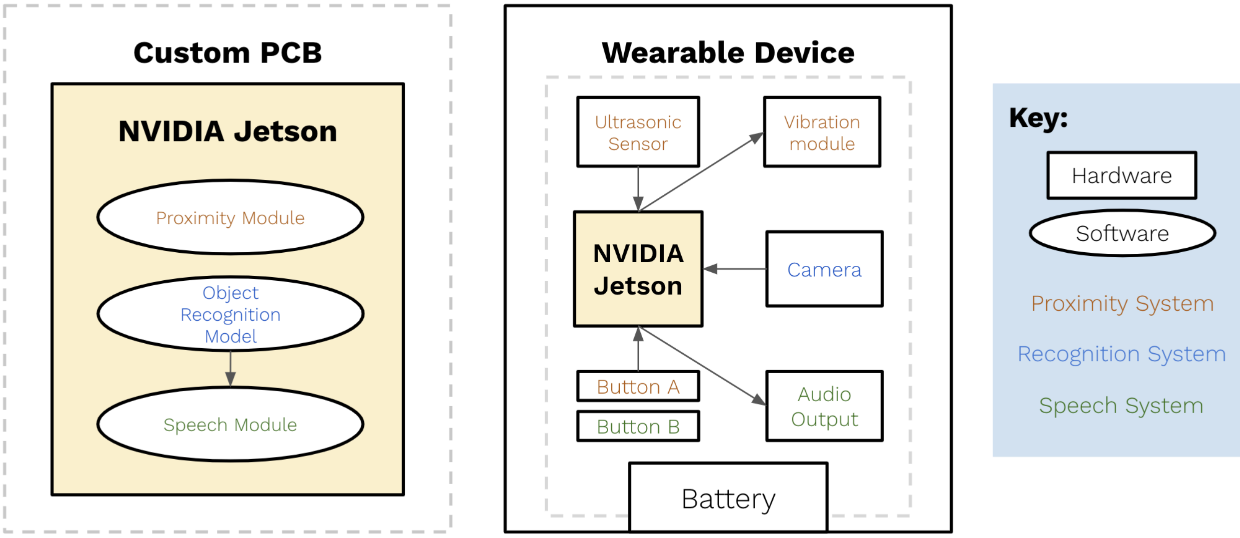


Fig. 1. This example of a block diagram illustrates the device's layout and subsystems.

III. Architecture and/or Principle of Operation

Fig. 1 illustrates the device's layout and subsystems. The device will be worn around the user's neck like a pendant, and the main casing will contain the Jetson Nano that will run our program, as well as the external hardware components, our custom PCB, and the battery pack.

The device is divided into three subsystems: the proximity system, the recognition system, and the speech system. The proximity system will detect obstacles in the user's path using an ultrasonic sensor, and will notify the user of the obstacle using a vibration motor which will be toggled on and off using control button A. The object recognition system will use a camera module sending images to our object recognition model, which will identify obstacles and estimate their distance from the user. Lastly, the speech system will take the OR model's output and convert it into speech output if the user chooses to identify the object using control button B.

The arrows in Fig. 1 indicate the direction of data flow between components. For hardware integration, the camera will be connected to the Jetson using the onboard camera connector, and the audio output will be connected to the Jetson's USB-A ports. Our custom PCB will connect all other hardware components to the Jetson's GPIO pins with circuits to limit current and voltage between the pins and components. For software integration, the OR model will send data to the speech module using serial communication, and the proximity and speech modules will use data from the GPIO connections with hardware components.

IV. Design Requirements

A. Camera to OR model frame rate

The camera of the device captures images and forwards it to the object recognition module at a constant frequency of 2 frames per second (fps). This process of capturing images and directing it to the OR model will happen continuously, regardless of the device modes described above. Given that humans perceive visuals comfortably up to 24 fps, achieving a higher frame rate would undoubtedly improve the experience for the user, in terms of usability. However, for the needs of this device, which is mainly to provide the user with situational awareness of a static environment, along with resource constraints for battery life and other hardware used, 2 fps is the metric we have arrived on. This may be subject to change at later stages of the project depending on the latency of the rest of the system and user feedback on the device. This requirement can ensure that the recognition delay is less than 2.5 seconds.

B. Proximity Module Feedback Frequency

The sensor polling frequency refers to how many times per second the ultrasonic sensor is going to measure for distance, and based on the hardware specification of the HC-SR04 Ultrasonic Sensors we are using, this value is set to 25 Hz. However, when rerouting this sensor data to the vibration motors, a much lower frequency is required so as to not overload the user with sensory output. A more comfortable frequency of vibration is 2-4 Hz, the exact number will be selected based on user testing and overall latency of the device. To ensure that the proximity module has minimal latency, this module is separated from the OR module of the device, with minimal layers from end-to-end. This design can therefore contribute to meeting the recognition delay of 2.5 seconds by alerting the user sufficiently for them to notice the potential danger.

C. *Proximity Detection Distance*

A distance of 2 meters before the user encounters an obstacle is a safe range within which the user can move away from it safely. Given this, we have set 2m to be the threshold of object detection for the proximity module, meaning that when the ultrasonic sensor detects an object under this threshold distance, this triggers the program that will communicate to the user of the object they are approaching, as described in the above section on proximity feedback via vibration motors.

D. *Battery Life Analysis*

The goal in terms of battery life is to allow for 4 hours of continuous usage. This is in alignment with the typical length of navigation and aid provided by service dogs before they take breaks, and a reasonable target for duration spent outside for daily activities before access to recharging the device. Given this metric, below is a breakdown of the power consumption of the major components of the device:

Component	**Current (mA)**
NVIDIA Jetson Nano	2000
Camera	491
Ultrasonic Sensor	5
Vibration Motor	85
Audio Converter	70
Total	**2651**
4-Hour usage	**10,604 mAh**

To allow for the 4-hour usage requirement, a ~10,600 mAh battery is needed. The actual battery life per use may vary depending on the size of the OR module and other latencies in our overall system.

E. *OR Module Accuracy*

Mean Average Precision (mAP) is a widely used metric for evaluating the performance of object recognition models (Buhl), particularly in tasks such as image classification or object detection with multiple classes. It calculates the average precision across all classes, providing a comprehensive measure of a model's ability to correctly identify objects of interest in an image. Industry standard considers an mAP of 0.4 - 0.6 to be reasonable to good performance. In the context of guide dogs, which are trained to assist visually impaired individuals, it's noted that their accuracy in guiding is expected to be around 0.7 mAP. To ensure that our OR model can effectively supplement or exceed the capabilities of guide dogs, the aim is to achieve an overall accuracy rate higher than 0.8 mAP. This higher number is in line with our goal to address the need for precise computer vision systems that help visually impaired people move safely and freely. Consequently, the user requirement of having 70% accuracy of OR model can be met.

F. *Wired Bone Conduction Earphone*s

The Jetson will be connected to wired bone conduction earphones for the audio output. They allow the users to hear background noise and device output simultaneously through their structures and designs. Hence, this strategy can meet the use case requirement of the users being able to hear the surrounding sounds to notice a potential danger. The wired option is relatively cheap and easily accessible for the users, so it can also reduce the production cost.

G. *Modules with Minimal Weight for Integration*

The size and weight of individual modules are assessed and considered to alleviate the weight of the product as much as possible. The product consists of a small camera module that is capable of sending real-time image data, and one HC-SR04 Ultrasonic Sensor is used for a proximity measure. The minimum size of PCB that is able to handle the functionality of integration is used to contribute to minimizing the size and weight of the product. This design strategy can meet the weight requirement of 450 grams.

V. Design Trade Studies

A. Hardware

We chose to use an NVIDIA Jetson Nano to run the software and mount the hardware for our device, largely due to its highly performant GPUs which will be capable of continuously running the OR model.

Initially, we considered using a Raspberry Pi 4 as our single-board computer. It has a vast community along with abundant online resources relevant to OR models, camera modules, and ultrasonic sensors, which may potentially save us time in figuring out the integration of modules. However, the problem with RPi4 is overheating of the product. When multiple modules involving a ML program are run, it tends to overheat, causing high latency and potentially even a system shutdown during the run. This issue is relevant because it has a low power video processor, so using a camera module will frequently lead to overheating. To mitigate this overheating risk, we considered hosting the OR model on a server and using the RPi to communicate with the server to send images and receive recognition data. The issue regarding this mitigation plan is that RPi4 needs to be connected to Wi-Fi to send data to the server. Because it would be challenging for visually impaired people to manually connect Wifi in an unknown indoor environment, this plan has been aborted.

The hardware device that can handle the cons of RPi4 is the NVIDIA Jetson Nano. Although it has less online documentation and community support than RPi4, the biggest factor of using Jetson is that it contains higher performance and more powerful GPUs than RPi4. Therefore, Jetson is more suitable in using a ML model like this project. It will have less frequent overheating and allow flexibility in development.

B. OR model version

We have chosen the Yolov5 model that is developed by Ultralytics. It is built on the PyTorch framework, which makes it easier to use and fine-tune for developers. Considering that this project will need a model with a DE feature, using this version can reduce the development time. The Yolo versions generally have relatively high performance in OR. They are also very commonly used for real time data processing, so sending real-time data with a camera module can be easily implemented in this project. Furthermore, there are vast online resources and tutorials, so the learning curve in using this model can be reduced. However, the potential issue can be the accuracy of the model. There are more recent versions that have greater response time and higher accuracy. Another issue is the integration of the DE feature. Because the integration is not provided in the open source library, some time will be designated to integrate the feature onto the Yolov5 model. However, there is open source documentation for the integration of the DE feature to the Yolov4 model, so it will be used as the primary source for implementation.

Several alternatives to using the Yolov5 model have been considered.

The Detectron2 model, supported by the Facebook research group, provides a modular programming design, so it can be flexible and customizable. It also includes several features including pre-trained models and mixed precision training. However, there is a steep learning curve to the model. Understanding its architecture, configuration, and API's is required to fine tune the model. Additionally, not only it relies on high-end GPUs and large amounts of memory to train the model but also many dependencies. This model can cause compatibility issues or dependency conflicts during the deployment.

An alternative plan is to use the Yolov4 OR model that has a DE feature attached to it. Because the model is available in an open source library, the project can use the code directly, which can reduce a lot of development time. Using the results of the detected object and corresponding distance, only fine tuning the data is needed to output the desired functionality of the product. However, this model is a pretrained model with several objects that are irrelevant to indoor settings. Therefore, it cannot recognize a common indoor object like a table or a sofa. Furthermore, this model cannot be retrained with a personalized dataset anymore because the development team no longer supports a "darknet" module, which has been used to train the OR model. Despite the increase in development time, it is critical to re-train the OR model to an indoor object dataset to meet with the scope of the project. Therefore, a different version needs to be considered.

Another consideration of the model is using Yolov8, which is the most recent version of the OR model. It has the highest detection speed and greatest accuracy among all versions. However, it is not built on the PyTorch framework, which makes this version harder to use and fine-tune for developers. Considering that a DE feature has to be attached to the model too, using Yolov8 can be problematic in spite of its high speed and accuracy.

C. Using headphone jack adapter instead of Bluetooth to connect audio device

Since the Jetson does not have an audio connector onboard, we considered several methods of incorporating audio output into the device. In particular, we had to choose between using a wireless Bluetooth connection or a wired connection. The Bluetooth connection would be preferable for allowing the user greater mobility than the wired connection, but incorporating a Bluetooth module would require an interface for users to pair their device, and our design plan does not include an interface already. Additionally, we were concerned that Bluetooth would draw more power than a wired audio connection, especially if an interface needed to be implemented. We would then require a larger battery capacity to meet our 4-hour usage requirement. Since a larger battery introduces more weight on the user, we chose to implement the wired connection for the sake of the user's comfort.

VI. System Implementation

A. OR module

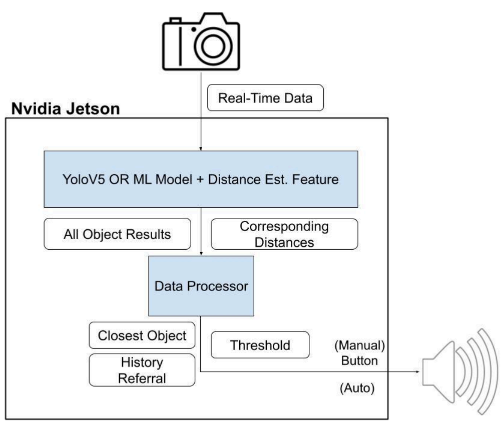


Fig. 2. The OR model architecture connecting a camera module, Jetson, and audio module

The object recognition model is operated by first sending the real-time data from a camera module to the NVIDIA Jetson Nano. Jetson uses these image data as inputs to the YoloV5 OR model with DE feature. The model recognizes all objects and identifies the corresponding distance from the product. The results go through the data processor module, which filters out the relevant data. The filter includes choosing the closest object by comparing the respective distances, referring to the history of detected objects if the recognition is unavailable, and filtering out the probability values that are lower than the chosen threshold. The data processor will send one final processed object and distance to the speech module.

B. Proximity module

a. Software

We have set a 2m range for the proximity module, meaning that objects detected within a 2m range will be communicated to the user via the vibration motors if the setting is turned on. Based on the design requirements section highlighting the 25 Hz polling frequency of the ultrasonic sensor, we want to distill the data transfer down to 2-4 Hz before producing vibration feedback, so as to avoid the risk of overloading the user. This process of filtering data before routing to the vibration motors will be handled by this program.

b. Hardware

The proximity module will be implemented using an HC-SR04 ultrasonic sensor communicating with a coin vibration motor using the Jetson Nano. The vibration mode will be toggled on and off with a push button, and all hardware components will interface with the Jetson via our custom PCB. The push button will be directly mounted on the PCB, while the ultrasonic sensor and vibration motor will be connected to the PCB using jumper wires to allow for flexibility in placing the components. The vibration motor will be located on the carrying strap on the back of the user's neck instead of on the main device, so that the vibration can be felt regardless of the user's clothing.

C. Speech module

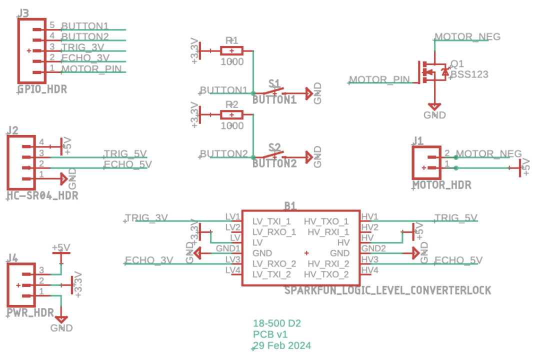
a. Software

This module employs a text-to-speech engine called espeak, along with the wrapper library pyttsx3. Output from the OR module will be intermediately processed based on a threshold we set for the confidence score of the object detected, and the output will be accordingly redirected to the user via the TTS engine and audio device. For example, if we set our confidence threshold to be 0.75, any objects identified with confidence scores lower that this value will undergo the following check: 1) iterate through 10s history of objects detected to see if a higher score was achieved with better camera views, 2) if not, let the user know that object cannot be detected, or use language with lower certainty such as "This object *may* be a stool".

b. Hardware

The speech module output will be implemented using a CM108-based USB-A to headphone jack adapter. Since the Jetson has USB-A ports available on the board but no onboard audio port, this will be the simplest way for us to implement an audio connection, and allows the user to connect the headphones they are most comfortable with. The vibration setting will be toggled on and off by a push button, which will be mounted on the custom PCB.

D. Custom PCB



Designing a custom PCB allows us to safely connect the buttons, ultrasonic sensor and vibration motor to the Jetson Nano's GPIO pins. The J4 PWR_HDR pins connect to the Jetson's 5V, 3.3V, and ground pins to supply power to the PCB, and the J3 GPIO_HDR pins will connect to the Jetson GPIO pins which we will configure for each hardware component.

The two control buttons will each have a pull-up resistor to regulate current flow to the Jetson to prevent damaging the pins (walterfms2i). The BUTTON1 and BUTTON2 pins will be configured as GPIO inputs for the Jetson to read the button value.

The vibration motor we are using is a 3V DC motor. Since the direction and speed of the motor is not necessary for us to control, we are using a N-channel MOSFET to supply current to the motor instead of using an H-bridge or an off-shelf motor driver. The gate of the MOSFET will be driven by a GPIO output pin, and the motor will connect to the PCB via the J1 MOTOR_HDR pins.

Lastly, the ultrasonic sensor will connect to the PCB using jumper wires on the HC-SR04_HDR pins. Since the ultrasonic sensor operates at 5V and the Jetson's GPIO pins can only supply up to 3.3V, we are using a bi-directional logic level converter that is capable of shifting signals between 5V and 3.3V (nemestomi2). The sensor's TRIG pin will be driven by the Jetson and input to the sensor, and the ECHO pin will be output from the sensor and input by the Jetson.

VII. Test, Verification and Validation

A. Tests for OR module

The implementation of OR model can be tested for two features. The first is testing the accuracy of the object recognition. The verification will be done if the OR model is able to recognize the closest item using a built-in computer camera, which is used to simulate the low resolution image data from the camera module. By comparing the value (item name) and the detected object (closest item), the success of each test can be determined. A confusion matrix will be constructed to identify the false positives and assess the potential risk of misidentifying a certain object. From the use case requirement, the minimum accuracy to meet is 70%. If the test result passes this metric, it can be determined that this module is successfully implemented.

The second testing is to test the DE feature of the OR model. The functionality can be verified by checking whether it can detect the distance of the closest item using a built-in computer camera. By comparing the distance of the closest material measured by the model to the actual distance measured by a ruler. If the output is ± 30 cm from the actual distance, the test can be considered a success.

B. Tests for proximity module

The test will consist of checking the vibration if there is an object within 2m in front of the user. It is considered successful if the module passes a test with above 95% accuracy. This element is essential for users to avoid obstacles and relevant to their safety, so this testing will particularly be done rigorously.

C. Tests for speech module

The main form of testing will include user-testing to determine whether the speed of speech outputted by the device is a comfortable one. Along with this, we want to ensure that the device's outputs do not prevent the user from still being able to hear their surroundings as we know that the visually impaired do rely on their hearing. The TTS engine used is one of industry standard and is proven to have high performance accuracy and is customizable to meet our requirements with rate of speech output, volume, accent etc. The quantitative aspect in the testing process might include fine-tuning the length of speech output (how descriptive does the object identification need to be), along with the time given between the outputs in the continuous object identification mode. Hence, testing this module will be mainly user-centric, with some part of it involving quantitative testing to meet overall latency requirements.

D. Tests for device settings

We will test the device's speech and vibration modes in several stages. First, we will test the buttons on the PCB once it has been fabricated by connecting the Jetson's GPIO pins and writing a simple program to ensure that a button press is detected. Failure in this stage would indicate that the PCB may need to be debugged and redesigned. If this stage succeeds, we will use the button presses to set the device's settings to each possible combination of speech and vibration modes and ensure that the device can support each mode. We will consider this a success if every mode combination is supported, and in the case of failure we will likely need to debug the software that defines each mode.

VIII. Project Management

A. Schedule

At the time of submission of this report, we have completed the preliminary and foundational stages of the project. This includes research on the approaches we want to use based on existing industry methodologies as well as our own ideas, interacting with potential users to understand their need, as well as selecting hardware and software tools and components to build our device. The remainder of this project involves the actual implementation of the modules and bringing together the device. An overview of the schedule is attached at the end of this report (appendix A).

B. Team Member Responsibilities

We have separated our overall system into 3 sub-categories that each of the members of our team take the lead on. However, the overarching functionality of the device is ensured by all of us, covering overall integration, the testing and verification process, as well as the outer look (design) of the device. This splits the work as follows:

Team Member(s)	**Responsibility**
Josh Joung	OR Module
Meera Pandya	Hardware Implementation
Shakthi Angou	Proximity and Speech Modules
All	Overall Integration, Device Design, Testing and Verification

C. Bill of Materials and Budget

We have been able to source several components for our device from personal or class inventories. The remaining components that we have purchased from external sources are well within our budget, with some future costs including the customized PCB, strap to hold the device and make it wearable, and 3D printing costs for a case for the device etc. A summary of the items purchased as well as estimated future costs are listed at the end of this report (appendix B).

D. Risk Mitigation Plans

1. Weight of device

Our goal is to keep the overall weight of the device under 400-450g. However, we anticipate that this estimate might still be an uncomfortable one, or that the device might become even heavier, jeopardizing the intended use-case of the device. To address this, we have planned for the back-up of offloading the battery pack to a waist-strap, so as to distribute the total weight and improve practicality. We will make this call during our testing phase, when we can receive feedback from participants who have volunteered to test our device.

2. Connection to peripherals (Custom PCB)

We plan to connect the peripherals (buttons, sensor, and vibration motor) to the GPIO pins of the Jetson, with a custom PCB in between to manage the voltage and current levels. A risk with this approach is that custom PCBs take time to order, and there may not be enough time to redesign a PCB if there are bugs. We plan to manage this risk by first breadboarding the PCB circuit to ensure it is adequate for safely connecting the peripherals before we place the PCB order. Our contingency plan in case the PCB still has bugs is to replace the PCB with an Arduino, which will require us to switch to serial communication between the Jetson and Arduino and will cause us to reevaluate our power and weight requirements.

3. Image classification base model and DE feature

Our current OR module builds off of an existing industry standard model called the Yolov5. This is a change from the initial plan to use the Yolov4, which upon further research we decided to change as that version of the model does not accommodate training with our own dataset, which is a key functionality we wanted our model to have. This has steered us towards the Yolov5 which has improved functionality and support. If we face issues with the accuracy of the model, we have a stretch-goal to upgrade to the Yolov8 as our base model which is the most recent version. The same mitigation method as the OR module will be executed to raise the accuracy of the DE feature.

4. Ultrasonic detection range

Ultrasonic sensors have a range of approximately 2 centimeters to 4 meters, with a sensing cone of 30 degrees (“Ultrasonic Distance Sensor”). If we experience failures to detect an obstacle within range, we will include additional sensors as needed to cover the entire necessary sensing range. If the sensors falsely indicate the presence of obstacles in the 2-meter range, we plan to reduce the detection threshold on the software end to prevent erroneous sensing.

IX. Related Work

During the initial stages of our project, we drew inspiration from a device named Theia that was built by Anthony Camu (Camu) , a final year Industrial Design student at Loughborough University in 2021. Their device has a similar purpose of aiding navigation for the visually impaired via a handheld device. It uses LIDAR technology in combination with a control moment gyroscope to construct a 3D image of the user’s surroundings and determine the safest path to take. Our device modifies most parts of this project but we certainly were inspired by this student-lead project to provide a cheaper and more accessible alternative to guide dogs.

X. Summary

The neck-wearable device will essentially guide visually impaired users in an indoor environment to avoid obstacles and identify an item right in front of them. It will encourage visually impaired people to go to an unknown environment, such as visiting an acquaintance's house, with less safety concerns. This project consequently motivates them to connect with more people and explore the world around them.

An upcoming challenge in this project is the implementation of individual components and the integration of them. The accuracy of an OR model and DE feature, fine-tuning of the model for data processing, the connectivity of the speech module and Jetson, and the functionality of the custom made PCB are the most critical parts of this project, and therefore will be spent most time on. The designated slack time in between each milestone will be used to work on improving the success of the individual components.

Glossary of Acronyms

DE	Distance Estimation
LIDAR	Light Detection and Ranging
OR	Object Recognition
PCB	Printed Circuit Board
RPi4	Raspberry Pi 4

References

Buhl, Nikolaj. “Mean Average Precision in Object Detection : A Comprehensive Guide.” *Encord*, 5 November 2023, https://encord.com/blog/mean-average-precision-object-detection/. Accessed 1 March 2024.

Camu, Anthony. “Introducing Theia: The Handheld Robotic Guide Dog – Polytec Personnel Engineering and Scientific Staff Cambridge Cambridgeshire.” *Polytec Personnel*, https://www.polytec.co.uk/whats-new/introducing-theia-the-handheld-robotic-guide-dog.html. Accessed 1 March 2024.

nemestomi2. “Using HC-SR04 Ultrasonic Sensor with Jetson Nano?” NVIDIA Developer Forums, 11 June 2020, forums.developer.nvidia.com/t/using-hc-sr04-ultrasonic-sensor-with-jetson-nano/78861/3. Accessed 01 Mar. 2024.

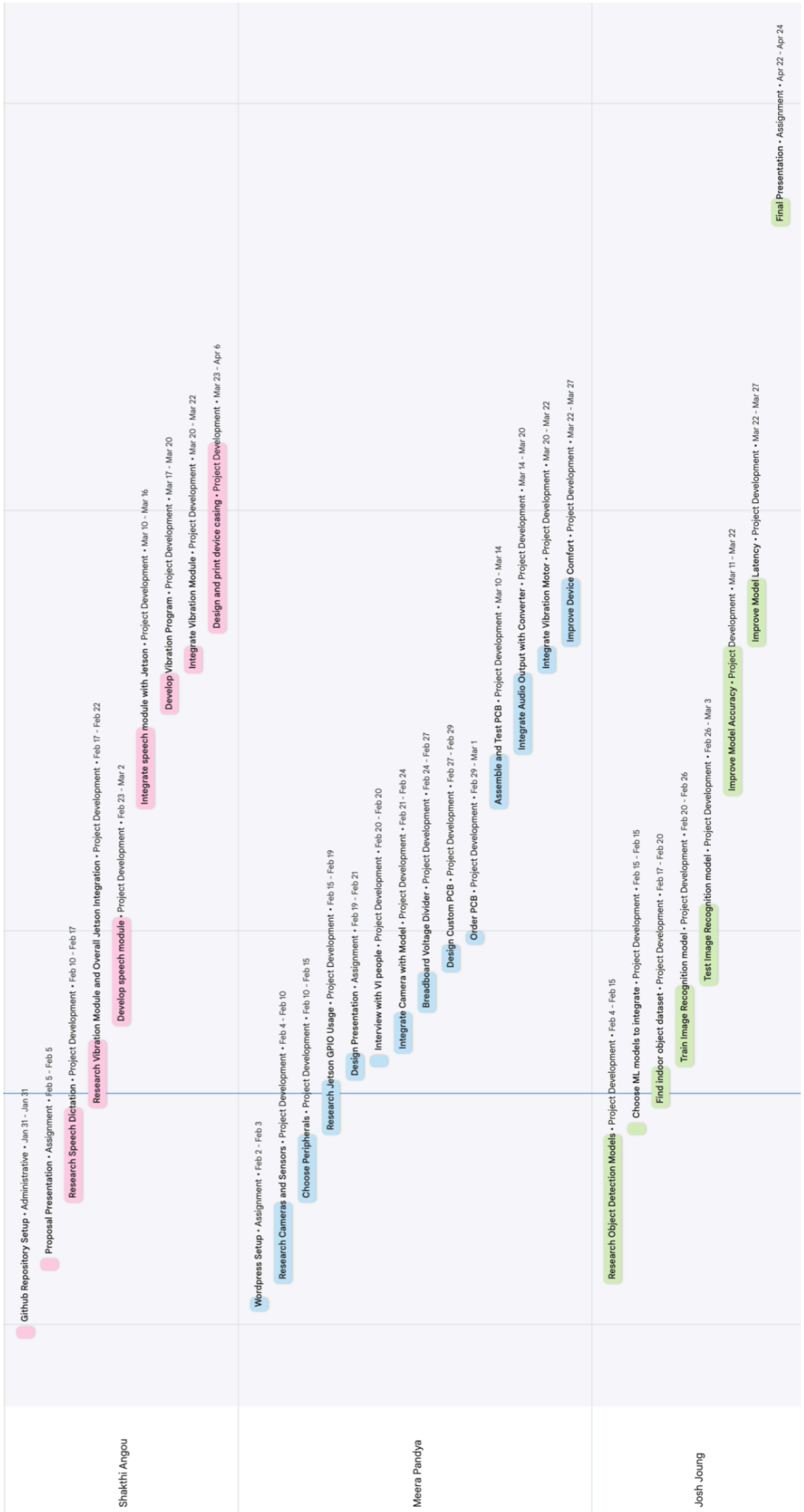
“Ultrasonic Distance Sensor (HC-SR04).” PiSupply, uk.pi-supply.com/products/ultrasonic-distance-sensor-hc-sr04#:~:text=The%20HC%2DSR04%20sensor%20works,to%20the%20nearest%200.3cm.

User, Deci. “Overview and Comparison of Neural Network Training Libraries.” Deci, 4 Apr. 2022, deci.ai/blog/neural-network-training-libraries-tools/.

walterfms2i. "Adding Buttons to Jetson Nano." NVIDIA Developer Forums, 19 Sept. 2019, forums.developer.nvidia.com/t/adding-buttons-to-jetson-nano/81942. Accessed 01 Mar. 2024.

"YOLOv8 vs. YOLOv5: Choosing the Best Object Detection Model." Www.augmentedstartups.com, www.augmentedstartups.com/blog/yolov8-vs-yolov5-choosing-the-best-object-detection-model.

Appendix A: Schedule



Appendix B: Bill of Materials

Purchased Items	Quantity	Cost
IMX219-160 Camera	1	$28
Zyamy Micro Flat Vibration Motor	10 count	$7.59
64 GB MicroSD Card*	1	$13.99
5V 2A Power Supply*	1	$9.95
Ultrasonic Sensors	10 count	$9.99
Total		$69.52

Inventory Items	Quantity
Nvidia Jetson Nano	1
Breadboard, Resistors, Capacitors	Variable

Future Costs	Estimate
Custom PCB Iterations	$50 - 100
Battery Pack	$50
Wearable Device Straps and Casings	$30

* May not come from project budget , paid for by 18-500