Holcombe Department of Electrical and Computer Engineering

Clemson University

Final Report:

Assisting Individuals with Vision Impairments in Navigation Using Machine Learning and Tactile Feedback

Spring 2021

Team SP21- 3 - Digital Guide Dog

Anthony Dimalanta

Griffin Dube

Harry Kenya

Karen Gutierrez

Bradley Selee

Executive Summary (1 page)

In modern society, the ability to quickly and safely navigate an environment is an important aspect of everyday life. While this action is more complex for individuals with visual impairments, many tools are available to support navigation. Some of the most significant, widely accepted mobility aids include walking canes, or white canes, and service animals. These aids are able to provide a basic means for maneuvering the user through an environment, but there are several shortcomings to these state-of-the-art implementations that limit their effectiveness. While service animals are incredibly useful, they come with the added cost of caring for an animal, as well as requiring the owner to place their safety in an animal that may become distracted or confused. On the other hand, white canes are very simple devices which extend the range of the user's knowledge about their environment through the sense of touch. This comes with the limitations that individual objects are not able to be identified, and the range is limited to what the user is comfortable handling when maneuvering an environment.

We address the current limitations of state-of-the-art tools by presenting our solution: The Digital Guide Dog. The Digital Guide Dog is an attachment that augments the traditional capabilities of the average white cane. Our solution fits over the handle of the cane and is able to provide accurate and reliable distance measurement up to three meters, increasing the range of knowledge of the user by 2x using tactile feedback through vibration. As new obstacles come into range, the user is notified of the distance to those objects through a series of vibrations that occur in the handle attachment.

The Digital Guide Dog also provides intelligent audio feedback listing direction and distance to the nearest objects of interest upon user request. Whenever a user wishes to know the location of nearby objects that may be of interest to them (seating, other people, kitchen appliances, etc.) they simply press a button and are provided an audible message describing the location of objects in the field of view of the cane. This feedback can be requested repeatedly in order to aid an individual in successfully navigating their environment in order to reach a desired destination.

By extending the capabilities of the white cane there is great potential for improved quality of life and all of the freedoms that come with the ability to safely move to a desired destination. With these added features, the Digital Guide Dog provides a more robust option to traditional navigational tools for anyone who needs assistance in maneuvering their environment.

Abstract (0.5 page)

Basic means for navigating one's environment are a crucial part of the average individual's daily life. From a young age, people are taught to be mindful of their surroundings and pay attention to where they are walking in order to avoid harm. This is a non-trivial task for individuals with vision impairments. Modern methods require these individuals to rely on apparatus such as a walking cane in order to safely navigate an environment. These aids come with countless shortcomings, leaving the individual unaware of their environment except for their immediate surroundings. Walking canes are unable to identify the types of objects an individual encounters, and are limited in their ability to detect distances greater than 5 feet (1.5 m). We address the shortcomings of modern walking canes with our Digital Guide Dog attachment. Digital Guide Dog is an attachment for common sized canes that provides additional features over traditional canes to aid in navigating environments. Our solution extends the distance at which obstacles in the users vicinity are reliably able to be detected by 2x, while also providing object detection and directional tools to assist the user in safely arriving at their desired location. The Digital Guide Dog allows users to take advantage of a common tool, the walking cane, and extends its capabilities to better suit their needs.

Introduction

Over 12 million people in the United States have some form of visual impairment. About 10% of this population uses a walking cane, or white cane, while the rest use another form of guiding system, such as a guide dog [1]. The white cane is a very simple tool and has potential for many improvements. A more advanced version will benefit current users and allow new users to have a better form of guiding assistance.

Objective & Motivation (0.5-1 page)

While guide canes are used by many, its design is very basic and limited in several ways. It is so limited that around 90% of visually impaired people do not use a cane as their guide assistant [1]. The first issue is that the cane has a limited range of no more than 1.5 meters, any distance greater than that and the user will be unable to sense the obstacles. More so, the user has limited options when it comes to navigating to a specific object or location. The person needs to ask someone nearby or know from prior experience. These limitations are a large reason white canes are not used by all visually impaired people, and other expensive guiding systems are used. Our main objective is to solve these limitations of the standard white cane using machine learning with a depth and camera and various sensors. The depth camera serves to solve the limited range of the white cane, vibration motors will alert the user of obstacles at an extended distance, and machine learning will detect the objects in the users surroundings. A text-to-speech (TTS) program notifies the objects detected with their respective distance and direction to the user. All of this will be integrated into a handle attachment that can be attached to existing white canes.

Curriculum Connections (0.25 page)

- Electronics/Circuits courses (course #s) - All of the circuits necessary to power the motors without causing damage to the hardware
- Big Data and Machine Learning Creative Inquiry
- Microcontrollers - ISRs, getting asynchronous events to occur at any point through the infinite while loop

Problem Definition (1-2 pages)

The shortcomings of modern navigational aids available to the visually impaired community are immediately noticeable when their capabilities are examined closely. While it is very obviously possible that individuals with visual impairments are able to lead normal lives, there are limitations of the current state-of-the-art implementation of the tools at their disposal which makes this much more difficult than it needs to be.

Current tools available to those with visual impairments such as blindness include the previously mentioned white cane, or some type of service animal to assist in safe navigation. White canes are an extremely valuable tool for these individuals because their awareness of their environment comes from their ability to sense what is near them using senses such as touch or sound. The white cane extends the user's reach several times over, allowing them to have a greater understanding of their surroundings and make educated decisions about their direction of travel. While this is a cheap, simple tool to provide valuable information to its users, there are glaring limitations. The average length of a white cane varies based on preference. Shorter canes are generally waist-height (around 38 in.) while longer canes range from 50 to 56 inches [2]. The shorter a cane is, the less time an individual has to react to various stimuli, meaning their safety could be compromised if they momentarily lose concentration. Longer canes, on the other hand, are more difficult to maneuver in large crowds, meaning that these individuals need to decide between their own safety and the ease of moving without causing discomfort for others. Further, these canes are unable to provide any sort of information to the user other than the binary information that there either is, or is not, an obstacle in their path. Oftentimes, it is useful to know what type of object is there, in order to understand whether immediate action should be taken to avoid it, or if it is an object to which the user wants to travel. For example, if a user wants to sit in a chair, it is useful to know if someone else is already seated there, but if the cane touches the edge of the chair first, the individual may not become aware of the person seated there until it is too late.

Service animals such as guide dogs provide some improvements over the white cane, however they are not without their own limitations. Guide dogs are able to more intelligently navigate an environment, and correct dangerous behavior if their owner is heading somewhere that will potentially cause them harm. These improve upon the white cane's binary feedback but come with the added challenges of having to also care for an animal. Some people do not have homes capable of accommodating these service animals, or prefer not to take care of an animal. The service animal also presents a challenge for navigating a crowd, where it may be more difficult to navigate through many people without causing others discomfort.

While current solutions are relatively effective and very well accepted by those who make every day use of these tools, there are clearly many avenues for improvement surrounding these techniques for safe navigation of an individual's immediate surroundings.

As such, we are faced with three main challenges:

- How do we provide more comprehensive information to a user about the objects located in their immediate environment?
- How do we extend the range capabilities of current technologies such as the white cane?
- How do we improve the navigation experience while minimizing the impact of the tool on the user's ability to easily maneuver through their environment?

Final Design (4-10 pages)

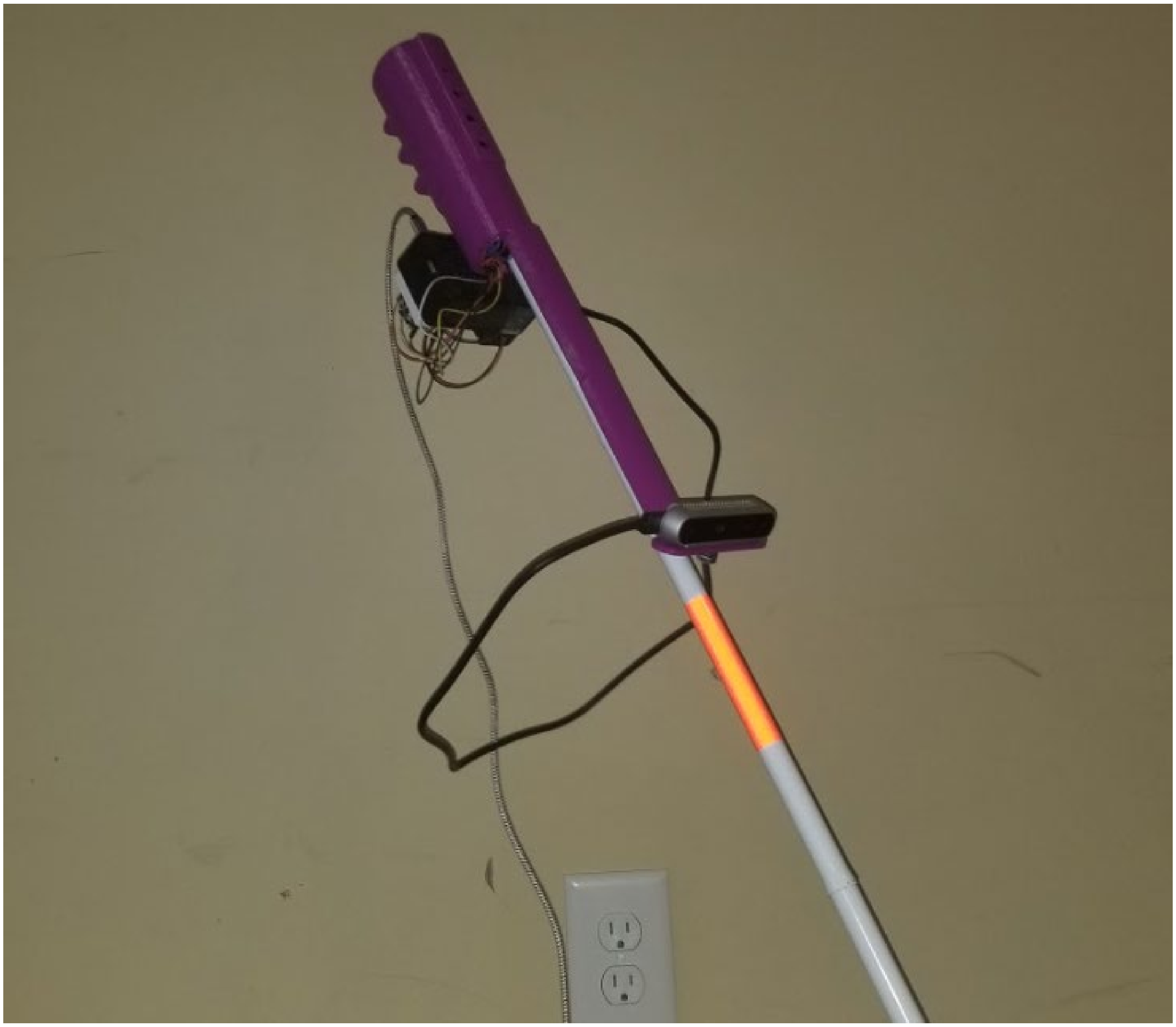

Figure 1: Final Project

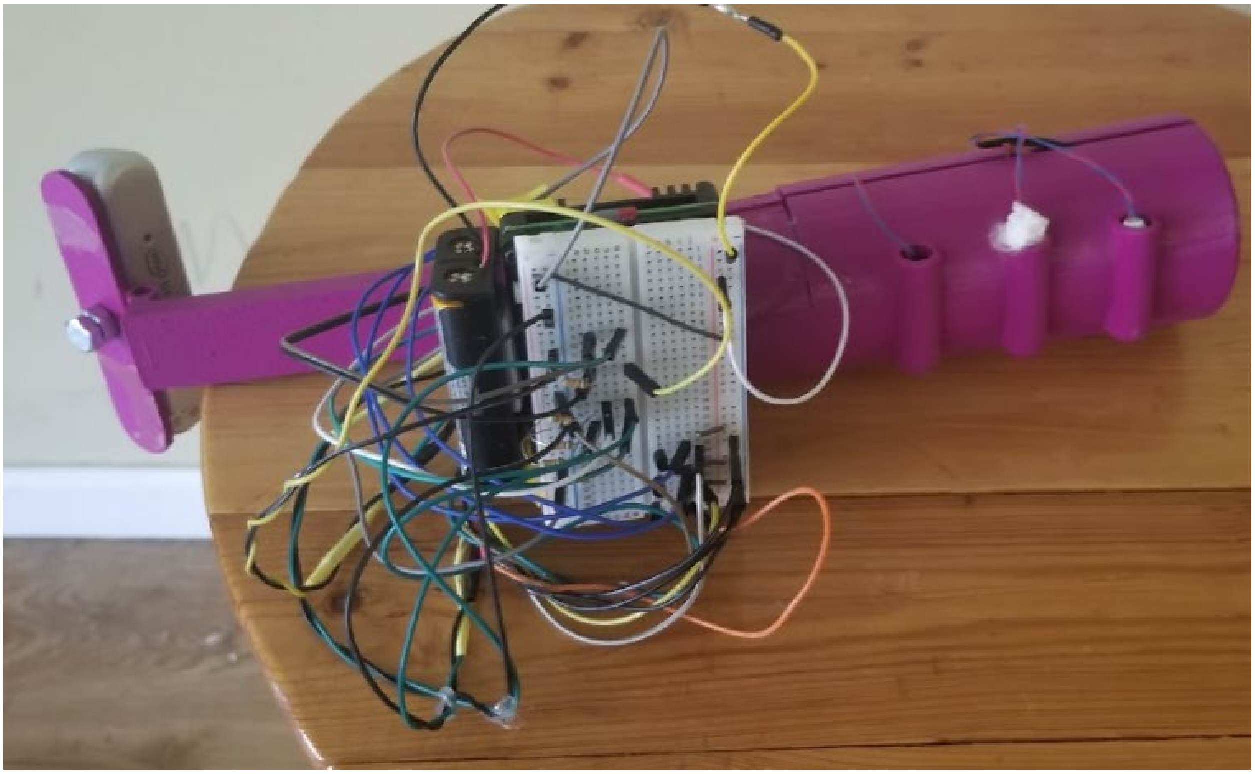

Figure 2: Wiring (as described in Figure 5) and physical cane attachment

General Project Overview

Our project design integrates several hardware components into an adaptable handle that can be fitted onto an existing cane. System has two main components: the physical device, and the software to control everything. The handle was 3D printed and designed to house 4 push buttons, 3 vibration motors and a depth camera. There is a platform to fit the prototype board and the raspberry pi. On the software side, the I/O was interfaced through the raspberry pi user allowing the user to interact with the device at any moment. The environment analysis is done using the depth camera and machine learning techniques to scan the users surroundings. Once objects are recognized, their metrics are processed and presented to the user using text-to-speech.

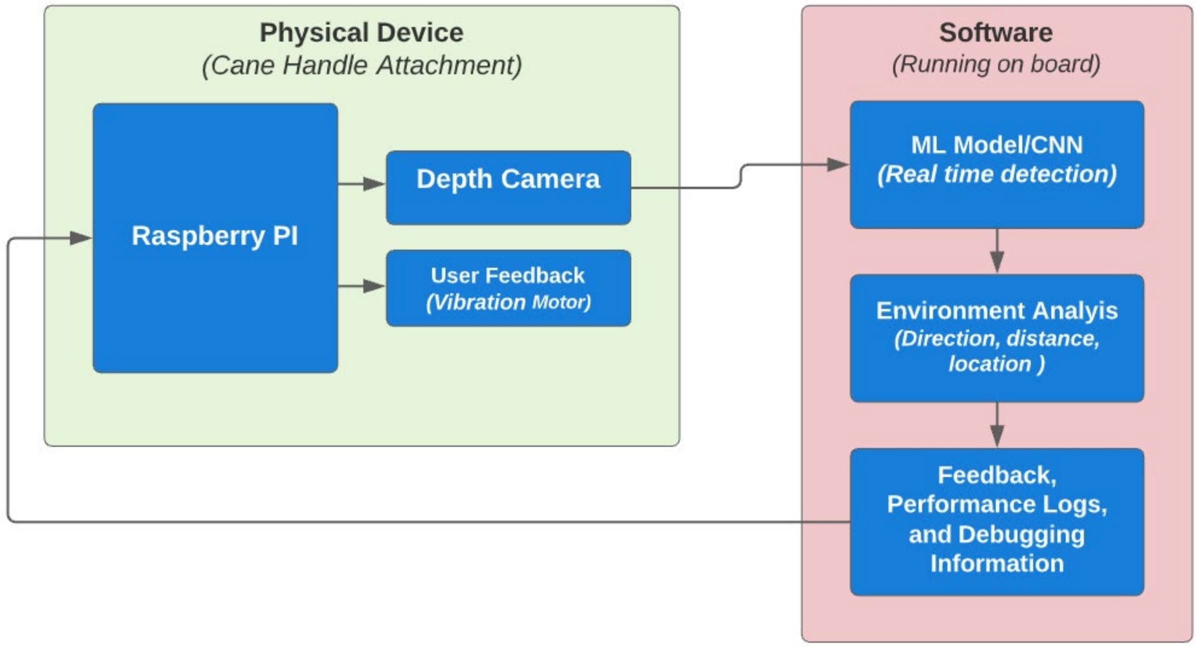

Figure 3: Device Overview Flowchart

Hardware Overview

Compared to the software side, the hardware was a relatively smaller portion. To start, we designed the physical project in CAD software and 3D printed. This design needed to hold 3 vibration motors, 4 push buttons, the depth camera, and the raspberry pi. As Well as storing the hardware, the user experience was also in mind while designing the 3D model. The handle attachment was modular to where it can fit on existing canes and be tightened down. The actual handle itself contains grooves that fit nicely into most hands. The final 3D model is shown below. As for the I/O devices, they were simple 2 wire, power and ground, devices that were soldered onto a perf board then connected to the raspberry pi's GPIO pins. The wiring diagram for the push buttons and vibration motors is also shown below.

Figure 4: Conceptual CAD Model of Cane attachment

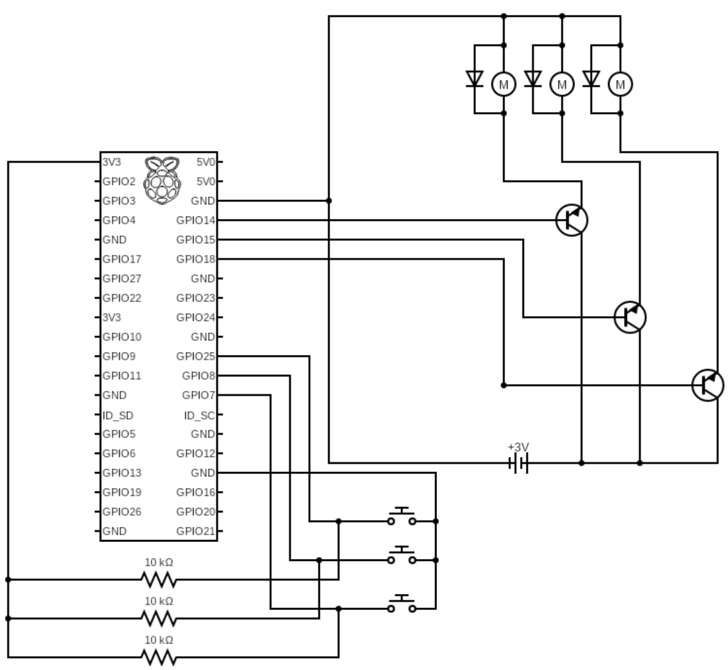

Figure 5: Wiring Diagram showing the motor and push-button connections

Software Overview

The software component of our project is the main control center that defines the hardware logic and workflow progression. The software can be broken up into 3 main sections: machine learning for object detection, post processing of detected objects, and I/O interfacing. Because our project was more software focused, there were many dependencies required by this project to perform all of the necessary tasks. These dependencies are summarized in the table below. Additionally, the OpenCV library was used to display the object detection and analysis for debugging and visualization purposes. OpenCV was not included in the table because when our project is used in production, this library serves no purposes and only slows down inference.

Software (Vers)	**Purpose**
Python 3 (3.7.3)	Main workflow language
TensorFlow Lite (1.15)	Machine learning framework for MobileNet SSD object detection model
PyRealSense (2.2)	Libraries used for Intel Depth Camera
gTTS (2.2.2)	Google's text-to-speech module for speaking to the user (requires internet)
GPIO Zero (1.6.2)	Libraries for interfacing the hardware with the Raspberry PI

Table 1 - Software used

Object detection integrated with the depth camera is a critical piece that makes our project so significant. The concept follows that a machine learning model identifies medium-large household objects that a user would want to navigate to. Knowing the detected objects in the camera frame, the object's centroid can be passed into the depth camera and receive its distance. Object detection was implemented using MobileNet v2 and the TensorFlow Lite framework [3]. This model was pretrained on the Common Objects in Context (COCO) dataset. With over 200,000 labeled images and 80 object categories, this dataset was sufficient to detect the objects we wanted. Because all of this had to run on a raspberry pi, we had to use a very lightweight object detector; MobileNet was quantized from 32 bits to 8 bits to further improve the speed. After implementation, the model was inferring at about 4-5 frames per second. The flowchart below demonstrates the object identification logic.

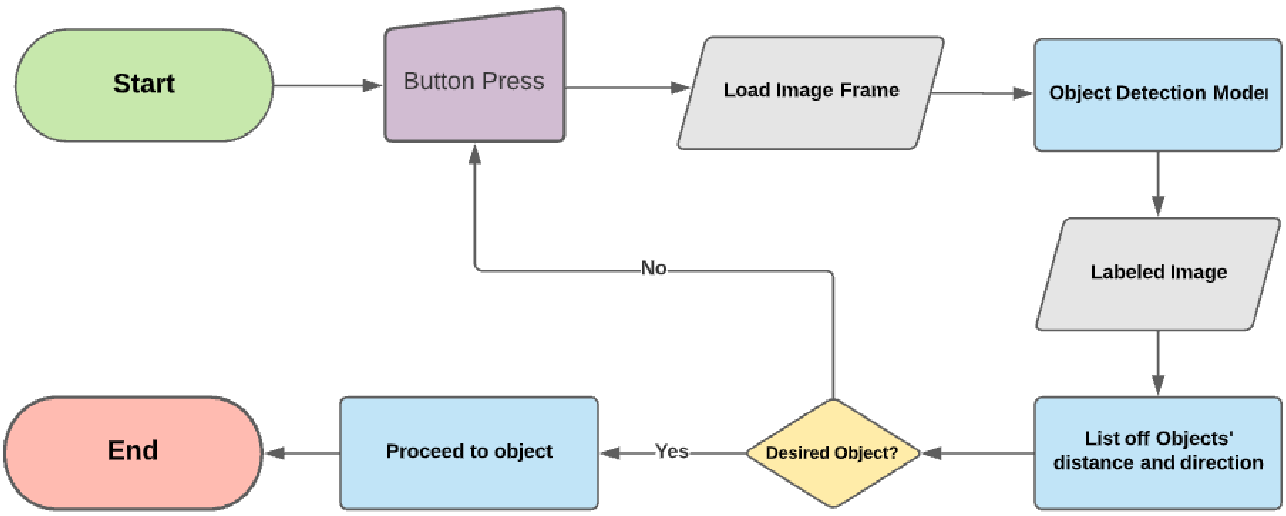

Figure 6: Object Detection Flowchart

While the object recognition algorithm automatically processes the frames and reports the objects metrics, the hardware interfacing allows the user to control these automatic processes. The main I/O hardware components are vibration motors and push buttons. In the main workflow loop, the depth camera is constantly reporting the distance of the center of the frame. Using this distance, the vibration motors are turned depending how close the user is to the center frame. After testing, we found that the best values were at 3 meters away 1 vibration motor turns on, at 2 meters away 2 vibration motors turn on, and at 1 meter away all 3 vibration motors are on. If the user is more than 3 meters away, then no motors will be on implying they are not close to any objects. The vibration motor code is constantly running, but the push button waits for a user input. The push button serves to call the object detection model and recognize certain objects from the frame. From here, these objects are analysed to find their distance and direction as well as filtering out the anomalies picked up by the model. The hardware was interfaced in Python using the GPIO Zero library, making it easy to initialize the raspberry pi pins and read from/write to the I/O. This algorithm is described in the flowchart below.

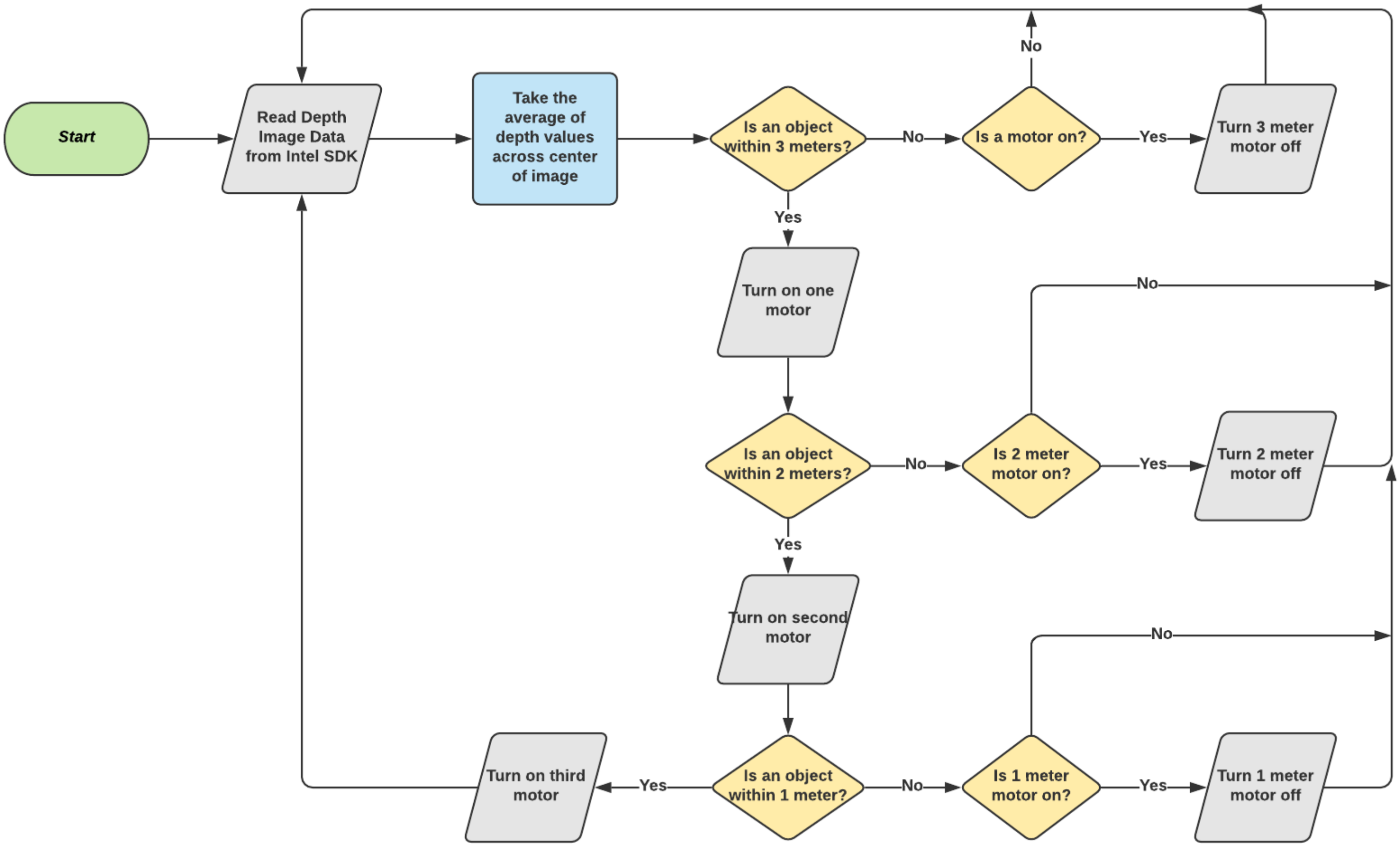

Figure 7: Motor Flowchart

Performance Characterization (1-2 pages)

The performance of the overall project exceeded expectations. The software incorporation and machine learning model was accurate enough in detecting different common objects within an environment, that it allowed the user to feel comfortable enough to be guided by the software. Even though it is very hard to predict which objects will be within an environment when the cane attachment is deployed to different users, the data set that we used included enough pictures that it allowed the training of the neural network to be more accurate and thus give better results when used in “real-life” environments.

When implementing the software to run while mounted on the cane, we were able to run multiple tests to determine the best positioning of the camera. The tuning of the position turned out to be one of the big challenges when not only trying to train the model but for it to be deployed successfully while in movement. Since the cane is an extension of the user we had to be able to maintain a balance between capturing objects in the environment and objects that were directly in the path of the user. We succeeded in not only maintaining the balance and center of the camera but also being able to train the model to only communicate with the user when objects were directly in their path and not just in the environment.

Even though the proposed model depended heavily on the performance of not only the software but also the camera, we had to choose a camera that would have a good enough image quality to allow both object and depth detection. Depth was a key element when successfully deploying the software because what distinguishes our project from others is that we are able to give feedback to the user by announcing how far each object is in relation to their position within the environment. We succeeded at giving the user this information accurately and with a high confidence that the object the camera was identifying was in fact the one it was describing to the user.

Discussion (1-2 pages)

Our ultimate goal of our project was to create a cane attachment that performs object avoidance and detection for those who have vision impairments. The cane we hoped to improve is formally known as a white cane. This is the device in which people with vision impairments use daily to navigate. Although this tool is used world wide, there are a few drawbacks within its functionality. The cane is used only for immediate distance (close proximity). This means you are only able to understand your surroundings by making contact with your surroundings within a very small distance. The white cane is very simple. As you walk, you swing the cane and hope to make contact with any possible obstacle that may be in your way. This means there’s no guidance system, and there is no way to identify which object you may encounter. Given these issues, we decided to integrate object detection and object avoidance within our version of the white cane. This would allow the well known movement, the panning back and forth of the cane, to remain the same while adding features that would make the tool much more versatile for every day use. Within this goal we were confined to a time constraint of four months and a precalculated budget of $400.

In our final hardware design, we implemented a Raspberry Pi, an IntelRealSense D415 Depth Camera, 3 3V vibration motors, a push button, 3D printed Enclosures, a white cane, and 5V battery pack of batteries. As for the software, we used Python 3 (3.7.3), TensorFlow Lite (1.15), PyRealSense (2.2), gTTS (2.2.2), and GPIO Zero (1.6.2). After integrating both hardware and software, we were able to successfully create a fully functional object detection and avoidance white cane. Within the 4 months of working on this project, we faced many challenges. Although the task of creating our goal was difficult in itself, we found that the greatest challenge was working on the project remotely rather than together in one location. Oftentimes, we would find ourselves checking and testing specific portions of the code and functionality of the hardware sequentially which made understanding the full scope of our project very difficult. Despite our struggles, we were able to meet our goals and ultimately revolutionize the idea of the white cane.

Cost Accounting (0.5 page)

Our budget for this project was $400.00, however we were able to achieve the final product for $275.76, well below our initial budget. This cost includes all purchased materials including a Raspberry Pi 4 Model B, an Intel Realsense D415i Depth Camera, and waterproof DC vibration motors described in greater detail in Appendix A. The price of $275.76 is our Total Development Cost. We did not request any reimbursement, therefore that value is not applicable. Since several of the items we used were previously purchased and held on hand at the beginning of the project, our Expected Cost of Artifact is slightly higher at $307.76. Complete information on all required parts is found in Appendix A, and Table X, which displays the parts ordered during the semester and their corresponding prices.

Component	**Price**
Intel Realsense D415i Depth Camera	$199.89
Raspberry Pi 4 Model B (4 GB)	$61.88
Tatoko DC Coreless Motor (Vibration)	$10.99
3D printed components	$3.00

Table X: Components acquired throughout the semester along with pricing information.

Project Postmortem

Technical Postmortem (1-3 page)

The overall design of our project included strong points in the design that ended up in the final design. These components that worked well from the beginning formed the basis for the rest of our design process. Many of the weaker points of our design came out of the smaller more detailed parts and didn't encapsulate the entire scope of the product. While there were a few design flaws that would have required us to rework the entire design, in the end we found solutions to most of these problems.

The design and 3d printing of the presented us with a few issues. Since none of us have experience with 3d modeling, this resulted with a few of our early prototypes failing to print properly. This was due to not accounting for enough tolerances in the printing process. We also initially had the issue of print times being too long as our infill was set too high giving the printer a higher chance of failing during the printing process. Eventually we found a design that worked well enough that only had a few criticisms from someone more experienced in CAD modeling but they said it looked fine overall. The printed model had no points for us to use screws to anchor each piece into place. This was not a problem since the tolerances were tight enough that it held together on its own. Since the model was made out of ABS plastic, this resulted in the product being very light and easy to hold despite its awkward design.

Trying to implement object detection took a lot of research in understanding the modern machine learning object detection models. This field is still highly researched and there is by no means a perfect solution. Some object detection models have very high accuracy but low inference speed, while other models have much higher inference speed but lower accuracy. We explored 3 main object detection models for our project: Yolo v4, Mask R-CNN, and MobileNet SSD v2. Initially, we tried to implement Yolo, however we quickly realized that it was too open source and lacked documentation and support to solve errors. Next, we moved on to Mask R-CNN. This model detected objects very well with high accuracy, however, the inference speed was incredibly slow and would not suffice for our project. Finally, we decided to accept the lower accuracy model , about 65% confidence interval, for a large increase in speed and chose to use the MobileNet SSD v2 model. With this model, we achieved about 4 frames per second on the raspberry pi and it was accurate enough for our project.

In finding the distance to an obstacle, we had to determine what the most useful way of calculating the depth should be. We went over a few options which each had their pros and cons. We considered grabbing the average across a region in front of the user. Region averaging benefits the user by providing a cleaner distance to the closest obstacle that would fluctuate less. There were two issues with this however, it wasn't always accurate due to the averaging causing losses of precision. The other issue with it was that it added too much performance overhead which we needed for the intensive task of object detection. We ended up using the center point of the depth image to determine when to notify the user of an obstacle. This provided a high enough framerate that the small fluctuations were negligible.

The motors used to notify the user were the perfect size and strength for our application. They were advertised as motors for electric toothbrushes which seems to be exactly what we got. They were small enough to fit into our 3d model without taking up too much space while also having enough strength to be noticeable by the user. These benefits of the motors greatly outweighed the drawbacks. The only issue we had with incorporating them into the project was due to the fact that the motors were only 3V max. Since no motor drivers we had were able to properly drive motors of such low voltage, we had to default to more fundamental components. To power the motors, we went with 2 AA batteries in series which produced our required 3V power source. To logically control the motors, we went with the 3904 transistor and operated it in saturation mode.

When wiring the final device together, the original plan was to use a perfboard and solder all connections together where applicable. Considering a member of our team has some experience with soldering, we figured this part would not be a problem. This ended up being an issue that we wouldn't figure out what was wrong until it was too late to rectify and we only had a couple days until the demo. The main issue with this was the soldering iron we have was not calibrated properly resulting in difficulties getting the solder to melt as well as burning components from being held on the iron for too long. Once we deduced what the issue was, we performed a factory reset on the soldering iron which solved that problem. With only a few days until the demo we attempted to solder the rest of the connections together, we realized the solder mask on the perf board was preventing us from making good connections across the board. Because of these issues with soldering, we had to use a breadboard which had more loose wires that were likely to get caught and come unplugged.

Nontechnical Postmortem (1-3 page)

Given the limitations of our working environment and resources available due to COVID-19, we were still able to achieve a reasonably successful implementation of our original concept, that still provides a useful alternative to traditional white canes. One of the most difficult portions of the project was finding a remote work schedule which allowed us to collaborate on the physical project without geographical restrictions coming into play.

Not only did we have to cope with social distancing and common COVID-19 safety protocols, but also geographical distance. For the majority of the semester, none of our team members were located in the same city, which means an important challenge we faced was how to handle code development for hardware that was not readily available to us. This required some ingenuity by using SSH to connect our development system (Raspberry Pi 4) to the Apollo machines on Clemson's network, then using our individual machines to SSH into the Apollo machines, and then from there into the Raspberry Pi just in order to work on the software. This did present several problems to the group. Each time the Raspberry Pi was rebooted, this connection had to be manually restarted, putting additional pressure on the individual in physical possession of the Raspberry Pi. There could have been a solution to this problem that would have been easier in the long run, but this quick fix worked well enough that it fit our needs without requiring a significant amount of additional work.

One of the more difficult portions of the project was to figure out the hardware design and testing. Since only one of us at a time had possession of the hardware, it made rigorous testing something very challenging. Luckily, a few of us were able to meet a handful of times in person to make use of 3D printing tools and work on hardware design, but most of our communication happened through a Discord server we set up specifically for this project. This also served as our primary meeting spot, which we made use of quite frequently.

One strategy for coordinating meeting times that worked very well for our group was to extend the weekly meetings we held with Dr. Raza and Dr. Groff by meeting both before and after to discuss our weekly progress and plan ahead for upcoming milestones. This allowed us to stay organized and achieve consistent progress throughout the semester. This also avoided scheduling conflicts because we already had the required time set aside for these weekly meetings anyways.

The challenge that set us back most of the time was that of geographical separation. We believe that if we were all located more centrally to one another, we would have been able to achieve a higher quality final product without many of the difficulties we faced. Although these meetings would have still required compliance with COVID-19 specific safety guidelines, the ability to work in person would have saved time debugging our development setup and made hardware integration much, much easier.

References

1. “Fast Facts of Common Eye Disorders,” *Centers for Disease Control and Prevention*, 09-Jun-2020. [Online]. Available: https://www.cdc.gov/visionhealth/basics/ced/fastfacts.htm#:~:text=Approximately%2012%20million%20people%2040,due%20to%20uncorrected%20refractive%20error. [Accessed: 17-Apr-2021].
2. Ambrose-Zaken, Grace, “Knowledge of and Preferences for Long Cane Components: A Qualitative and Quantitative Study,” JVIB, vol. 99, no. 100, pp. 633-645, Oct, 2005, doi: 10.1177/0145482x0509901007.
3. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *arXiv.org*, 21-Mar-2019. [Online]. Available: https://arxiv.org/abs/1801.04381. [Accessed: 17-Apr-2021].

Appendix A - Cost Accounting

Item Description	Part Number	Vendor	Quantity	Total Cost	Payment Type	Used in Final Project
Raspberry Pi Board	Pi 4 Model B (4 GB)	Raspberry Pi Foundation	1	$61.88	PP	Yes
DC Coreless Vibration Motors	N/A	Tatoko	3	$10.99	PP	Yes
Realsense Depth Camera	D435	Intel	1	$199.89	PP	Yes
Generic Headphones/ Speaker	N/A	N/A	1	$20.00	PP	Yes
Breadboard	N/A	N/A	1	$8.00	PP	Yes
Perfboard	N/A	N/A	1	$2.00	PP	No
Electronic Components	N/A	N/A	1	$5.00	PP	Yes

Table A: Components list along with pricing information, vendor, and description of part.