Netra: Indoor Navigation Assistance

Siddhartha Srivastava, EE&CSE, Rohan Nandakumar, CSE, Rohan Limaye, CSE and Uche Anonyai, EE

***Abstract*—Visually impaired people encounter many challenges while navigating their surroundings. This problem is further amplified due to the lack of assistance in an indoor environment since all the current technology is built towards assisting people in outdoor navigation. To solve this problem, we are building a device that will utilize image classification and a navigation algorithm and a video which we will receive from the camera to help the user get to his destination. Our design will consist of a batter communication module and camera which will be connected to an app on the phone using LAN which will help visually impaired people navigate indoor environments easily.**

I. INTRODUCTION

People with visual disabilities have incredibly difficult time navigating from one point to another. They can travel more easily in outdoor environments because of all the existing technology and apps that assist with outdoor navigation. Even though apps like BlindSquare [13] help the visually disabled users with outdoor navigation there is still no reliable alternative for indoor navigation. Which is why our device will assist visually disabled people navigate in indoor environments and the need for indoor navigation.

i. Significance

Over the past few years, the number of people with visual disabilities has been rising at a drastic rate. The first global estimate on the magnitude and causes of visual impairment was based on the 1990 world population data (38 million blind). This estimate was later extrapolated to the 1996 world population (45 million blind), and to the projected 2020 world population (76 million blind), indicating a twofold increase in the magnitude of visual impairment in the world by 2020. [2] The number of people with visual impairment or blindness in the United States is expected to double to more than 8 million by 2050, according to projections based on the most recent census data and from studies funded by the National Eye Institute, part of the National Institutes of Health. [1] This shows why technology that will help visually disabled people navigate indoor environments is going to be essential in the coming years. Since as we can see, based on the statistics the number of blind and visually disabled people is going to rise greatly and only having technology that will assist them in outdoor environments is not enough.

ii. Existing Products

There are currently multiple apps and technologies that help make the everyday life of a visually disabled person a little easier. Such as apps for identifying objects like TapTapSee [8], to identify colors such as Color-Blind Pal [9] and even identifying money such as Cash Reader [10]. There is still no app to assist visually disabled people with indoor navigation. One of the most successful devices that is used for assisting visually disabled people is called OrCam [11]. It is an assistance device that allows visually impaired people to understand text and identify objects through audio feedback, describing what they are unable to see. When attached outside eyeglass frames, can read and verbalize text, and supermarket barcodes and also has face recognition. This information is then transmitted to the user through audio feedback. Despite being such a versatile device, it still does not allow visually disabled people identify their current location and does not assist them is navigating from one point to another in an indoor environment. There is an app that assists in indoor navigation, and it's called IndoorAtlas [12]. It is a developer too that utilizes the various sensors present in a smartphone and relative movement of the user to help construct digital maps to estimate the user's location. It requires a user that can see, to setup the beacons that will be used by the app to identify the user's location and the map that this app produces is only useful for a user that is not visually disabled, which is why this app is not suitable for visually disabled users.

iii. Societal Impact

The ones who will benefit from our indoor navigation device our people who are visually disabled or blind. Due to the lack of indoor navigation assistance for visually disabled people they become very dependent on the people around them to reach their destination. Our device will help visually disabled people navigate new indoor environments more easily thereby making them more independent in unfamiliar environments. We have tried to keep the device light and unobtrusive to be able to be used for extended periods of time. The device is also designed to be cheap to be used by a wider user base.

iv. Requirements analysis and specifications

Our project goal is to design a device that works with an app and is able to assist visually disabled people navigate indoor environments. As we can see in Table 1 we are designing the device in such a way that it will be able to function without any data usage and will be able to stream to the app simply using LAN. Our device is also designed to be as small and light as possible in order to make it unobtrusive and easy to use for extended periods of time.

Property	Specification
Data usage	Uses no data, connected via LAN
Classifier Latency	Prediction within 5ms of a change in scene
Stream Frame Rate	30fps
Stream Resolution	1080p
Weight	<25g
Boot-up time	<10s

Table 1: Requirements and Specifications

II. DESIGN

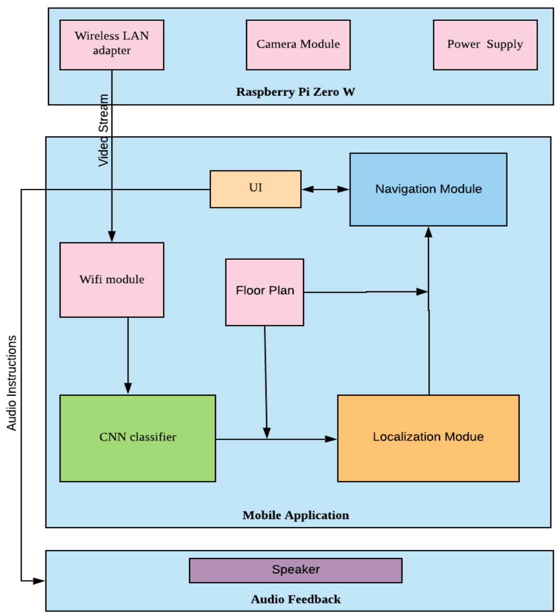

Figure 1. Block Diagram of the device

A. Overview

The biggest concerns we had for the design of the system was the safety and the cost effectiveness of the implementation. This includes both the cost of initial purchase and the running costs of the system. To that end we ensured that there would be no data usage while streaming by transferring the video stream over a local mobile hotspot. The hotspot is used as a connection point between the Raspberry Pi Zero W and the user's mobile device. The second issue to be dealt with was the speed of the classification which would be crucial to smooth and efficient navigation. To achieve the rate of classification required we limited the stream to 720p, this ensured that both the stream and the classifier would perform at an acceptable rate. Since we wanted a device that would be safe and would also be comfortable for the user to wear on their glasses, we decided to not include the battery for the system on the 3D printed frame. Instead we would connect it with a standard Micro USB cable to a portable battery pack which they could comfortably carry around in their pockets.

B. Video-Streaming

One of the biggest challenges during the early phases of the design was to ensure that the device was cost efficient since we realized that a lot of visually impaired people come from a weak financial background. According to the national federation of the blind the average income for a visually impaired person is about $38,500 annually [4]. With data plans skyrocketing, and assistive technology getting very expensive due to excess technology on the devices, our design needed to be inexpensive. Not only that, since we are trying to achieve indoor localization and navigation, we cannot rely on public wifi and good data coverage even if the user has access to it. We also had to use a microcontroller than can leverage a good camera to operate in indoor environment that can swiftly transfer images to the phone for processing. On top of everything, we had to ensure that our wearable can achieve all these requirements while still being light and comfortable for users to adapt to.

For that very reason, the raspberry pi zero W was chosen. Raspberry pi zero W, unlike other raspberry pis, is much lighter. 65mm long x 30 mm wide x 5 mm thick and weighing just 9g in weight, the raspberry pi 0 W was the perfect candidate for this task. For more detailed information, please refer to figure 2. This device has bluetooth, bluetooth LE and 802.11 b/g/n wireless LAN. Our initial attempt to transfer images in real time failed not just because of the limited number of frames being transferred over to the phone but also the latency due to the continuous transfer. With 24 Mbps, with bluetooth we were successfully able to transfer 3-4 images every second with a latency of about 200ms. But since these images needed to be processed and the output had to be conveyed in real time, our project required something more robust and even so, real time. For that purpose, we switched to the 802.11 b/g/n wireless LAN module on the py. This can support bandwidth up to 54 Mbps and has a frequency of 2.4 GHz.

We used Pi Camera Noir V2 as our input source. Other than being ultra-light(3g) and having good resolution(8MP), this camera can also shoot videos in 1080p with a rate of 30 frames per second. We used a public library "\Motion" that helped us stream over a local network, the network being the phone itself. After adjusting frame rates and picture quality in the source code, we successfully got a mobile device connected to the raspberry pi over hotspot and the camera was able to live stream over a local network. The next step was to be able to use an android application to receive the current live stream so we can run our CNN model (read below). Using an app to show the stream was not the most difficult part. However, one interesting thing that we noticed that our server was redirecting us to HTTP from HTTPS, which caused us some major issues since there wasn't an immediate clear solution to that. Upon further investigation, it was established that the cause of the problem was the new Android 9. According to [2], Applications intending to connect to destinations using only secure connections can opt-out of supporting cleartext (using the unencrypted HTTP protocol instead of HTTPS) to those destinations". Since our connection was not a secure one, we had to opt in into cleartext traffic which is disabled by default after API level 27. Adding an xml file with permission certificate status set to true, ensured that we can access non secure networks for our application. The live stream was further transferred to our convolution neural network that helped us recognize a place on the floor plan.

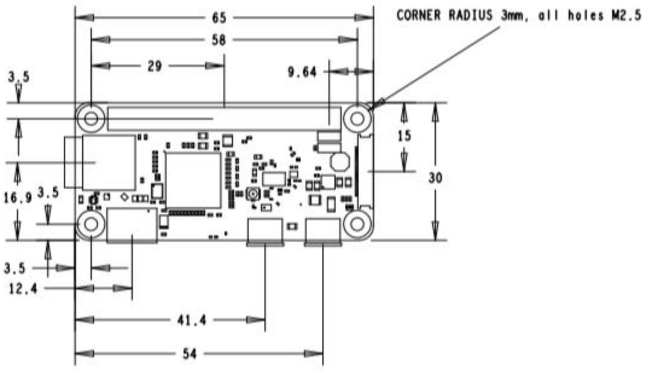

Figure 2. Circuit diagram of raspberry pi zero w

D. CNN and Scene recognition

The scene recognition part of the localization module was done by using a Convolutional Neural Network (CNN). We had a design choice to make at this point, we could either build our own Network or re-train previously existing models such as YOLO and Mobilenets. We chose to go with the former as our classification problem would include recognizing the background itself and not objects placed against a background. All the popular models available for re-training were built with the presupposition mentioned above.

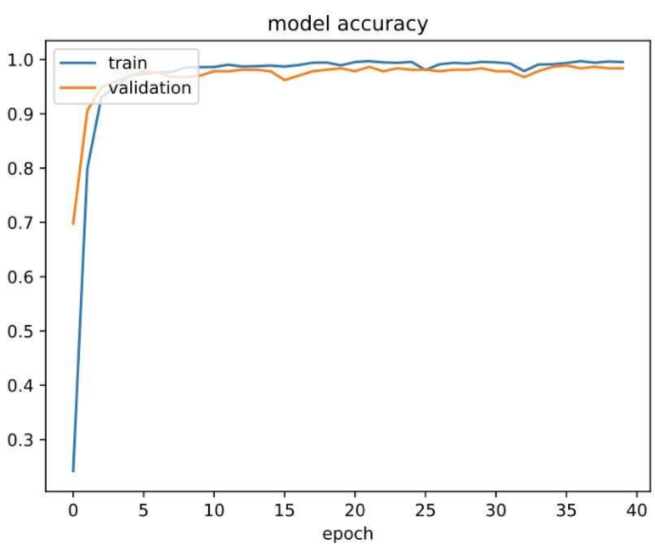

Figure 3. Validation statistics of CNN

The CNN was built, on Tensorflow with Keras, as a simple Sequential model with a relu/softmax Activation Layer and compiled using an Adam optimizer. The training process was very tedious. All members of the team obtained photos of the different points of interest we identified in Marcus Hall using their smartphones. Then pre-processing of the images was conducted where we re-sized the images using bi-linear interpolation and compressed them along with their labels into a pickle file. The training script would then extract the image arrays and fit the model to them. From Figure 3, we can see that the model accuracy curve of the training process indicates a good fit of the model on the data. The whole model was then compressed and saved as a JSON file.

The next challenge was porting this model to work with the android tensorflow package. We realized that although there was a whole page with examples on the Tensorflow website dedicated to running an image classifier on Android, these were all for Classifiers that had been re-trained on existing models. We found little to no documentation on how to port your own model onto the Android platform, after a significant amount of research we came up with a Python script that would freeze the Model graph and add the weights to produce a protobuf file (.pb). The reasons of the above process are twofold. Firstly, it ensures that we have an extremely compact model for deployment (217KB). Secondly, since it contains both the Model graph and the weights, the speed of loading the model and prediction is extremely fast. Half a second for the first and 0.01ms for the second. Another advantage of the above process is that the application and the model are completely functional offline.

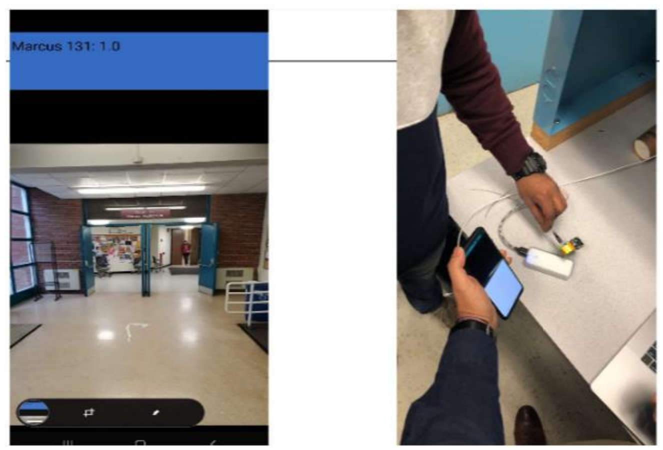

Figure 4. Screenshots of app in use

III. THE PRODUCT

A. Product Overview

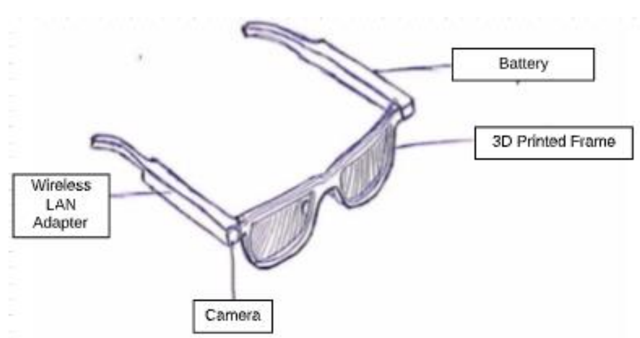

Figure 4. Product Sketch

- Our product is a computer vision-based localization system that utilizes recurrent neural networks to classily real time images.
- Images come to the classifier in a continuous fashion, i.e. a video being streamed over a local area network
- LAN helps avoid connection to regular wifi saving data rates, power consumption while increasing the accessibility of the product.
- The video is streamed at 720p at around 25-28fps and results are stored on a local SQLite server
- SQLite server serving as the backbone of the application in Android Studio utilized Dijkstra algorithm to then determine the shortest possible path
- Treating the floorplan as a "highway system" we then map out the navigated route for the end user
- Using textToSpeech functionality, the route is said out loud to the user

B. Electronic Hardware Component

The electronic hardware component consists of two major parts, the Raspberry Pi zero and the PCB

The Raspberry Pi Zero W mounted as a wearable device attached to sleek power bank enclosed in a 3D printed case printed on Thermoplastic polyurethane (TPU) material which provides durability and flexibility needed to mount it on regular glasses. We ensured the hardware components did not weigh above 30 grams each with the Raspberry Pi weighing 20g and the battery pack weighing 30g. This way the glasses were still light in weight. The battery pack, with a power capacity of 9000 mAh, was used to power the Raspberry Pi which consumed as little as 500 mA. This allowed the Raspberry Pi to run for 18 hours without charge. The nano camera was attached to the Raspberry Pi zero and was responsible for delivering the live stream at 30 frames per second to the mobile android device. The PCB was to be deployed as a wearable pendant which was still under design construction. This would consist of a footprint of TP4056 - Micro USB 5V 1A Lithium battery charger with protection which will allow our PCB to be able to be connected to a PCB battery and be be capable of charging up to 4.2 V. It would also take a 5V input with embedded LEDs to show RED if the charging is in progress and then green LED if charging is finished. These LED connections are also further in our battery level indicator that uses an array of LM339 comparators to check for battery levels of the battery. Piezo buzzer is then used to recognize if the battery is running low and notifies the user if so. The PCB also consists of a footprint of the BMD-350 module which is based on the Nordic nRF52832 SoC. This uses Bluetooth 5 and has an inbuilt ARM Cortex -M4F 32-bit processor with a 512kB embedded flash memory and a 64kB RAM. All of this setup led us to be able to connect to a micro Lidar sensor provided by GARMIN. This sensor is known for its high-performance optical distance measurement as it used an actual laser to measure distance unlike the other fake micro Lidar sensors out there that use Time of Flight (ToF). With a resolution of 1 cm and weight of just 22g this sensor gives a range of 5cm - 40 meters. It used I2C to communicate that could be easily configured in the bmd module and operates at around 5V DC which our battery will be able to provide. No testing on the PCB was performed due to the sudden closings.

C. Product Functionality

As seen in the block diagram, our device first identifies the user's current location by receiving visual input from the camera module sent to the android application using the wireless LAN adapter. This input is then sent to the CNN classifier, which is based on a simple sequential model with a RELU and Softmax activation layers, followed by the localization module alongside the floor plan giving the android application the user's current location. After this, the user can provide his desired destination through audio input. Our navigation module utilizes a graph structure derived from the floor plan to make it easier to navigate through. This tree is then stored in a SQLite server so that it can be easily accessed by the navigation module in real time. The navigation module uses Dijkstra to identify the shortest path the user would have to take to go from his current location to desired destination and sends it to the user as audio instructions.

D. Product Performance

- The product as mentioned in the design section uses Motion Eye OS to stream the video over a local hotspot and therefore just uses the mobile devices hotspot as a connection point and ends up not using data as according to the network usage statistics on the android device.

-As was demonstrated in the CDR meeting the Classifier returns a prediction for the image being displayed almost instantaneously. The accuracy is affected by motion of the user but once the motion ceases the classifier can recognize the image within 2ms.

- Since we downgraded the quality of the stream to 720p we were able to comfortably get a frame rate of around 35fps.

- We were able to achieve the desired 1080p but instead decided that maintaining the stream at 720p would put the lesser load on the classifier resulting in more consistent and accurate output

- The device was around 20g since we did not include the battery for the system on the 3D printed frame. Instead we would connect it with a standard Micro USB cable to a portable battery pack which they could comfortably carry around in their pockets.

- The device had an average boot up time of 8s to start Motion IOS service and start streaming to the application.

IV. CONCLUSION

Right now, our project comprises of two components, the classifier which tells the user where he is and an application which displays a live stream from the Raspberry Pi camera over a LAN network. These two components combine to form our localization subsystem. We were able to achieve this by working together as a team and doing research on how to work with a raspberry pi and how to build a classifier entirely on our own. Our plans can be seen in our Gantt chart above. We anticipate having to do a lot more research as we go on with our project. A difficulty we anticipate facing is implementing our navigation system to work with our localization system. Also, there would be some difficulties in designing our PCB and integrating it with the rest of our system as it is a wearable device, so it has to be very light. By following our proposed gantt diagram, we will be able to meet the deadlines for each subsystem as that provides a thoroughly planned timeline of how long each thing would take.

References

[1] Visual Impairment, Blindness Cases in U.S. Expected to Double by 2050." National Institutes of Health, U.S. Department of Health and Human Services, 19 May 2016, https://www.nih.gov/news-events/news-releases/visualimpairment-blindness-cases-us-expected-double-2050
[2] Global Trends in the Magnitude of Blindness and Visual Impairment." World Health Organization, World Health Organization, 3 July 2017, https://www.who.int/blindness/causes/trends/en/.
[3] Network Security Configuration: Android Developers." Android Developers, https://developer.android.com/training/articles/securityconfigCleartextTrafficPermitted.
[4] Blindness Statistics." National Federation of the Blind, https://www.nfb.org/resources/blindness-statistics.
[5] Network Security Configuration: Android Developers." Android Developers, https://developer.android.com/training/articles/securityconfigCleartextTrafficPermitted.
[6] Image Classification: TensorFlow Lite." TensorFlow, https://www.tensorflow.org/lite/models/imageclassification/overview.
[7] Getting Started with the Keras Sequential Model." Guide to the Sequential Model - Keras Documentation, https://keras.io/getting-started/sequential-model-guide/.
[8] https://taptapseeapp.com/
[9] https://colorblindpal.com/
[10] https://cashreader.app/en/
[11] https://www.orcam.com/en/
[12] https://www.indooratlas.com/
[13] https://www.blindsquare.com/

Appendix

A. Design Alternatives

Earlier in the semester, we tested an ultrasonic sensor to detect the distance to a point of interest. However, this design did not work as the sensor informed us whenever we were close to any object and not just the points of interest. This allowed us to alter our design as we were able to take out the sensor component of our hardware allowing our microcontroller to weigh less and be more comfortable for the user to wear. To calculate the distance from a point of interest, we decided to use the cameras to obtain this information.

B. Technical Standards

1. LAN Standards - IEEE 802.11 is part of the IEEE 802 set of LAN protocols, and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) Wi-Fi computer communication in various frequencies, including but not limited to 2.4 GHz, 5 GHz, and 60 GHz frequency bands.

2. Battery Standards for Raspberry pi- IEEE 1625. This standard establishes criteria for design analysis for qualification, quality, and reliability of rechargeable battery systems for multi-cell mobile computing devices. It also provides methods for quantifying the operational performance of these batteries and their associated management and control systems including considerations for end-user notification. This standard covers rechargeable battery systems for mobile computing. The battery technologies covered are limited to Li-ion and Li-ion polymer, but future versions of this standard may include technologies that are not in general use at present.

3. Battery Standards for android phone - IEEE 1725. This standard establishes criteria for design analysis for quality and establishes criteria for reliability of rechargeable Li-Ion and Li-Ion Polymer batteries for mobile telephone applications. Also included in the standard are battery pack electrical and mechanical construction, packaging technologies, pack and cell level charge and discharge controls, and overall system considerations.

4. Audio Standard when telling the user what direction to go- This standard is the second generation of IEEE 1857.2, which provides more flexible and high efficiency coding tool sets for compression, decompression, and packaging of the audio data, which should double the coding efficiency of IEEE std 1857.2-2013. The target applications and services include but are not limited to the audio accompanying video and other audio services and applications, such as the audio accompanying Internet video, TV audio system, digital audio storage, audio broadcasting and communication, virtual reality and panoramic audio, etc.

C. Testing Methods

Experiments have been carried out on our classifier and live stream. In regard to the live stream we conducted experiments to ensure that the stream could be obtained completely offline. We did this by turning off the wifi and mobile hotspot on the mobile device and we noticed the stream was still available. This showed that our live stream was meeting our desired specifications. Experiments conducted on our classifier involved testing it in the Marcus Hall and taking note of the confidence level of the classifier and checking that against the text returned by the label. We tested to see if the confidence rating corresponded with the text output we were getting. This experiment helped us discover we needed to continue to train our classifier and continue to conduct more tests until we meet an accuracy of 90 percent

D. Team Organization

The team has worked together well during this period. Our work ethic as a team has been very essential for the project to progress as well as it has done. We try to have at least two people at the scene when a component of the project is being worked on, so they have needed support whenever it becomes necessary. We decided that each member of the team will have a quarter of the semester where they act as the leader of the group. Each member has shown leadership qualities by overseeing our weekly meetings with a new leader being appointed every month. We had periods during the semester where it became hard to schedule meetings because of transportation to campus issues and midterm examinations. However, we handled this by scheduling video calls instead so we could check in on each other even though some could not make it to campus or would not have been able to get transportation if we decided to meet in person at the time of the video call.

E. Beyond the Classroom The project has been challenging for all members of the team and this has led to us having to learn different skills in order to get parts of the project completed. Rohan Nandakumar has had to learn how to build convolution neural networks, Rohan Limaye has had to learn how to build and design android applications, Siddhartha has learnt to work with a raspberry pi and Uche has learnt how to design PCBs. We have used all the resources available to us like Google and those that work at M5 to improve our skills and to help us whenever we are at stand stills with our projects. Skills gained working on this project, from the technical skills to those that have to do with simply relating with people and working as a team, can be translated to our life as professionals. We are aware that as professionals, we will have to teach ourselves a lot of skills by consulting online resources and by reaching out to those around us like we have done with this project.