

# Indoor Navigation for Blind Individuals Using Computer Vision & Machine Learning

An Interactive Qualifying Project (IQP) submitted to the faculty of  
**WORCESTER POLYTECHNIC INSTITUTE**  
in partial fulfillment of the requirements for the degree of Bachelor of Science

**Submitted By:**

Kelsey Leach  
Sam Mather  
John Winship  
Oliver Chen  
Alex Demirs

**Submission Date:**

April 30, 2022

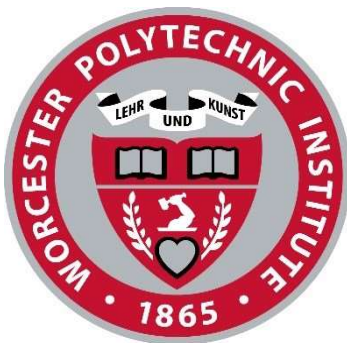
**Submitted to:**

***Project Advisors***

Professor Svetlana Nikitina  
Professor Ivan Mardolovich

***Project Sponsor***

The Financial University under the Government of the Russian Federation  
Professor Anton Losev  
Department of Data Analysis, Decision-Making Theory and Financial Technology



## Table of Contents

Abstract.....	5
Executive Summary.....	6
Project Goal and Objectives.....	7
Methods.....	7
Results.....	8
Conclusions & Final Recommendations.....	9
Acknowledgements.....	10
Authorship Page.....	11
Chapter 1: Introduction.....	12
Figure 1.....	12
Navigating The World Without Vision.....	13
Chapter 2: Background.....	15
2.1 BVI Individuals and Their Navigational Challenges.....	15
Social Considerations.....	15
2.2 Navigational Challenges faced by BVI Individuals in Russia and Germany.....	16
BVI Population of Russia and Germany.....	16
2.3 Limitations of Current Navigational Devices.....	17
2.4 Blind Communities in Russia and Germany.....	17
Rusinovo, Russia.....	17
Marburg, Germany.....	17
2.5 Recent Technological Developments and Research on BVI Navigation.....	18
Hardware-Oriented Navigational Aids.....	18
Wearable Sensors.....	18
Machine Learning Oriented Navigational Aid.....	19
2.6 Review of Previous IQP on Navigator App Development for the Blind Project.....	20
Goal and Outcome.....	20
Challenges and Learning.....	21
Figure 2.....	22
How Our Team Will Move the Project Forward.....	22
2.7 Conclusion.....	23
Chapter 3: Methodology.....	24
Figure 3.....	24
3.1 Interviews with BVI Individuals.....	24
3.2 Evaluation of Existing Navigational Technologies.....	25

3.3 Iterative Design Process.....	26
3.3.1 Step 1 .....	26
3.3.3 Step 2 .....	27
3.3.4 Step 3 .....	27
Chapter 4: Results and Analysis .....	28
4.1 Stationary Object Data Collection .....	28
4.1.1 Interviews Conducted by Kira Kovnat.....	28
Figure 4 .....	29
Figure 5 .....	29
Figure 6.....	30
4.1.2 Interviews conducted by Olga Mironova.....	30
Figure 7.....	31
4.1.3 Compilation of Feature List .....	32
Figure 8.....	33
4.1.4 Observations from Marburg about Adaptations for BVI Individuals .....	33
Figure 9.....	33
Figure 10.....	34
Figure 11.....	34
4.1.5 Observations from Berlin about Adaptations for BVI Individuals .....	35
Figure 12.....	36
4.1.6 Comparing Berlin and Marburg’s Adaptations.....	36
4.2 Moving Object Data Collection.....	36
4.2.1 Comparing Open-Source Code .....	36
4.2.2 Iterative Design Process.....	37
4.2.3 Iteration 1 .....	37
4.2.3.1 Developmental Iteration 1.....	37
4.2.3.2 Developmental Iteration 2.....	38
4.2.3.3 Developmental Iteration 3.....	38
4.2.3.4 Developmental Iteration 4.....	38
4.2.4 Iteration 1 Testing .....	39
Figure 13.....	39
4.2.5 Iteration 2.....	40
4.2.5.2 Developmental Iteration 6.....	40
4.2.5.3 Developmental Iteration 7.....	41
4.2.6 Summary of Findings.....	41

4.2.6 Appearance .....	41
Figure 14 .....	42
Figure 15 .....	42
Chapter 5: Discussion .....	43
Chapter 6: Conclusion and Recommendations .....	44
References.....	47
Appendix A - Sample Questions for Interviews with Russian BVI Individuals (in English).....	50
Appendix B - Sample Questions for Interviews with Russian BVI Individuals (in Russian).....	52
Appendix C – Sample Consent Form for Interviews.....	55
Informed Consent Agreement for Participation in a Research Study .....	55
Appendix D – AARGUS Source Code.....	57
Appendix E – User Guide for in App Usage of AARGUS.....	75

## **Abstract**

Navigating the world with impaired vision is a challenging and hard-to-imagine task. To address this challenge, our team created a smartphone app that would alleviate struggles encountered when traversing indoor spaces. The app was created by utilizing a machine learning model, named D2GO, which helps to detect objects with the use of a smartphone camera. Using D2GO, our team was able to develop an app that can both detect and guide users around other people and obstacles within an indoor space. This app will serve as a framework for a navigational aid that can be further developed by future R&D teams.

## Executive Summary

The goal of this project was to develop an app that allows blind and visually impaired individuals of Russia and Germany to navigate within unknown indoor spaces. Currently, there exists a need for a navigational tool that can work indoors without the use of GPS as most buildings block GPS signal. The blockage may cause additional navigational guides, requiring GPS signal, to become unusable. Thus, the user may have to rely solely on their other traditional navigational guides, such as a white walking cane, to both detect and avoid stationary and moving objects. This is the motivation behind our project to develop a navigational aid for blind and visually impaired users.

While conducting our research, we divided the problem of detecting and navigating around obstacles into two separate parts. Our team focused on the detection of both stationary and moving objects. Through project work, we found that stationary objects were not the main issue, as traditional navigational devices like a white walking cane or seeing-eye dog could detect these objects with high accuracy. Our team then shifted focus to moving objects by developing an app that could both detect and navigate the user around the moving object without causing harm. To detect these obstacles, our team evaluated the 2021 Navigator App for the Blind code, as well as other open-source code including SLAM (*Simultaneous localization and mapping*) and Detectron2. The previous IQP's code ran using a program called OpenCV. While this code ran and detected objects, it was unorganized and lead our team to work with another open-source code.

The next choice was SLAM which is a method widely used for robot navigation and robotic mapping. It is the process of mapping an area whilst keeping track of the location of a device, such as a robot or smartphone, within the area. SLAM makes mobile mapping possible. It allows map construction of large areas in much shorter spaces of time as areas can be measured using mobile robots, drones or vehicles. There are many different types of SLAM algorithms and approaches. The performance of different SLAM algorithms varies depending on the specific use case. For instance, some SLAM technologies are dedicated to drones, where the algorithms are most capable of doing bird-eye view mapping. A navigational device for BVI individuals would require SLAM algorithms that can mimic human eye-level mapping. Additionally, some SLAM schemes are computationally intensive which conflicts with user needs, particularly power consuming and poorly performing apps running on their smartphone. With the proper choice of algorithm and approach, features can be developed to provide some degrees of navigation through environmental mapping and obstacle avoidance for BVI individuals (GeoSLAM, 2022).

Another promising tool was Detectron2. It is Facebook's open-source library for implementing state-of-the-art computer vision techniques using PyTorch. Facebook introduced it in October 2019 as the successor of Detectron and maskrcnn-benchmark. It is especially effective for object detection, segmentation, and other visual recognition tasks. Detectron2 can easily carry out the task of object recognition for a navigational device. By further classification of objects, Detectron2 can provide more robust obstacle avoidance support compared to using SLAM. With the new D2Go extension, FBNet models, that are already optimized for mobile devices, can be created with architectures that can efficiently perform object detection and segmentation tasks where segmentation is the process of dividing an image into different regions based on the characteristics of pixels to identify objects or boundaries to simplify an image so that it can be analyzed more efficiently. (*Facebook - Meld je aan of registreer je*, n.d.-b).

## Project Goal and Objectives

Our goal was to aid the BVI communities of Germany and Russia by working with the Financial University under the Russian Federation to further develop and improve the smartphone enabled object identification software created by the previous Navigator App for the Blind IQP team. To achieve this, our team planned to accomplish the following objectives:

- Identify and document a common set of features of a navigational device that would improve the experience of BVI individuals within indoor navigational contexts.
- Evaluate existing technologies, including the 2021 Navigator for the Blind application code for fitness and reusability.
- Design, test, and revise our navigational device using an iterative design process.
- Finalize the design of our navigational device.

## Methods

Once our team decided to split the challenge of indoor navigation into two distinct parts, the first two objectives were conducted simultaneously. The first objective included conducting interviews with BVI individuals and making observations in the cities of Berlin and Marburg, Germany. The interviews focused on identifying stationary objects that should be detected by our app. To discover this information, our team drafted a series of questions to be asked during each interview that consisted of both open-ended and closed-ended questions. These questions gave us insight into objects that pose the most danger to BVI users, as well as the features they would like implemented into our navigational device. Interviews were paired with visual observations conducted in both Marburg and Berlin, to discover the adaptations these cities possessed for the BVI population. Together, these interviews and observations gave us the necessary knowledge about the issue of stationary obstacles encountered during indoor navigation.

To tackle the moving object aspect of indoor navigation, our team conducted a comparison between the open-source codes mentioned above. This comparison encompassed our second objective, which was conducted alongside our first objective. In this second objective, our team evaluated the previous IQP team's code and compared it to the open-source codes of SLAM (Simultaneous localization and mapping) and Dectectron2. After looking over both options and working with our partners from the Financial University under the Russian Federation, a division of labor was agreed upon. It was decided that the WPI team would work on object detection using Dectectron2, while the Financial University team would work on floor mapping using SLAM.

Once our team decided to proceed using Dectectron2, we moved onto our third objective which covered the iterative design process. This process consisted of three separate but cyclical steps. The first step was the building phase, where our team combined the information learned from the first two objectives into a working object detection prototype. This prototype then entered the second step, which was the testing phase. Here, the prototype was tested using an obstacle course constructed by the team, to see if the device could successfully navigate around objects. In the third step, after testing, the team reviewed and analyzed the results. If the results were not deemed successful, the team cycled back to the first step and addressed the issues that caused the

device to fail. If the device passed the test and was deemed successful, we moved into the fourth objective. In this objective, the team finalized our navigational device, polished errors in the code, and made sure the device was user friendly.

## Results

Due to our team's relocation to Berlin, Germany instead of Moscow, Russia, we were not able to conduct interviews ourselves. Instead, two of our Russian counterparts, Kira Kovnat and Olga Mironova, conducted fifteen interviews on our behalf using the interview questions we drafted. From the responses to the questions, our team was able to analyze the results using the statistical method of coding. By analyzing the results of both the open and closed-ended questions, we were able to develop both a feature list and a list of harmful objects to BVI individuals.

For observations of BVI accommodations in Berlin, Germany, our team was able to take note of observations during our four-week stay. We observed accommodations at different landmarks and museums, including raised maps, three-dimensional tactile replicas of buildings, and high-contrast images. Additionally, we observed chirping crosswalks, although they were neither prevalent nor reliable. The observations in Berlin were compared to our team's day trip to Marburg, Germany. The BVI accommodations in Marburg exceeded those in Berlin. One major difference between the two were the smooth sidewalks and sloped curbs in Marburg that allowed for better feeling by a user's white walking cane.

After our team decided to use Detectron2 for our coding section, we encountered some issues that blocked a fundamental part of our project. Our goal was to design the app for a mobile device, specifically Android devices. We encountered a problem where Detectron2 would not relay information detected fast enough for our app to be useful. This prompted the switch to D2GO, which was a different model of Detectron2 that was designed for mobile use.

To develop the app, we looked at the data compiled within the feature list and the data found within the comparisons of opensource code. With the combination of these initial results, we were able to start on the first iteration of the app. The first iteration consisted of getting the Detectron2 code to work on a mobile Android device, which was achieved with the help of D2GO. After this was successfully achieved, we started the first sub-iterations which included:

- Vibration when an object or person was detected
- Vibration when a specific object was detected
- A voice command that relayed if an object was in front of the user

Once these sub iteration features were added, it was time to test the first iteration. Testing was carried out through the creation of an obstacle course which contained both stationery and moving objects. The goal of this test was to have a user walk across the room without encountering obstacles. While this test was relatively successful, multiple problems were encountered. These issues were taken care of within the second major iteration, as additional sub-iteration features were added including:

- Different vibration speeds based on distance of the object
- Virtual box that represented the user within the code
- A voice command that would indicate if the object was on the right or the left



After these sub iterations were added to solve the issues found in the first iteration testing, the same test was conducted for the second iteration. This test was much more successful than the first, as fewer collisions occurred due to the directional queues and increased time efficiency.

After completing the final iteration of our app, our team proceeded into the fourth objective by polishing and cleaning up our code. Once this was finished, our team decided to name the App and give it a logo. The name was changed to AARGUS, which stands for Applied Augmented Reality Guiding Users Safely. This name related to Greek Mythology, as Argus Panoptes was the Greek titan known to be the watcher of the Greek gods.

## **Conclusions & Final Recommendations**

Through preliminary research and interviews conducted with blind and visually impaired individuals, our team found evidence to suggest that there exists a need for an improved comprehensive navigational aid. To address this need, our team designed, developed, and tested a navigational smartphone application for BVI individuals using an iterative design process. Throughout this process, the team made observations and analyses for each iteration. Every iteration improved upon the previous one and added to the value of this project. Additionally, our team developed a set of recommendations which are intended to help guide both end users and future project developers with our application.

### Users:

- Change screen settings to disable auto lock feature
- Hold steady at chest level
- Use application as a complement to your primary method of navigation

### Developers:

- Develop and train your own custom model
- Implement SLAM or other mapping technology
- Design a device to hold the phone steady
- Increase the degree of vision seen
- Incorporate the application into iOS
- Reduce the latency between detection and relay

Based on the previously outlined recommendations, our team is confident that a future team can pick up where we left off and further develop this application. Our team made great progress in developing a working application for BVI individuals to use as a navigational device. With the future integration of both our object detection and warning systems, along with simultaneous location and mapping software, a cohesive navigational device can be developed for BVI users. Our intent was to make indoor navigation easier, as our application is meant to complement a user's current navigational device, such as a white walking cane or guide dog.

## Acknowledgements

Our group would like to thank the following people for making this project possible and giving the guidance to complete this IQP:

The Financial University under the Russian Federation, for giving us the opportunity to work on this project.

*Kira Kovnat* from the Eurasian Peoples Assembly, for her help in overcoming the language barrier and reaching out to Russian BVI individuals

Our colleagues from the Financial University under the Russian Federation, for their help in allowing us to connect to Russian BVI individuals and their research into blind navigation:

*Olga Mironova*

*Olga Petrova*

*Daniel Zaharov*

*Lena Krapivina*

Our academic advisors, without whom our project would have never made it off the ground:

*Professor Svetlana Nikitina*

*Professor Ivan Mardilovch*

## Authorship Page

<b>Section</b>	<b>Primary Author</b>	<b>Editor(s)</b>
<b>Abstract</b>	Sam Mather	Kelsey Leach
<b>Executive Summary</b>	Kelsey Leach	Sam Mather
<b>Introduction</b>	Sam Mather Kelsey Leach	John Winship Alex Demirs Oliver Chen
<b>Background</b>	Sam Mather Kelsey Leach	John Winship
<b>Methodology</b>	Sam Mather Kelsey Leach	John Winship Oliver Chen
<b>Results and Analysis</b>	Sam Mather Kelsey Leach	John Winship
<b>Discussion</b>	Sam Mather John Winship	Kelsey Leach
<b>Conclusion And Recommendations</b>	Kelsey Leach	John Winship
<b>References</b>	Sam Mather	John Winship Alex Demirs
<b>Appendix A &amp; B</b>	Sam Mather	Kelsey Leach
<b>Appendix C</b>	John Winship Alex Demirs Oliver Chen	Sam Mather
<b>Application Design</b>	John Winship Oliver Chen	Alex Demirs
<b>Application Code</b>	John Winship Oliver Chen	Alex Demirs

## Chapter 1: Introduction

In a world built largely around sight, blind and visually impaired (BVI) individuals face significant hurdles when it comes to navigating daily life. The BVI community consists of those who have a complete loss of sight or severely reduced vision. From getting around the house, to finding one's way through busy city streets, or navigating through an unfamiliar building, BVI individuals have had to adapt to navigating a world that was not built for them. Historically, BVI individuals used walking canes, animal companions, or sighted guides to get around safely and effectively. In today's world of rapidly improving technology and increasingly complex architecture and urban design, these outdated methods, though tried and true, leave much to be desired, such as detecting dynamically moving obstacles and navigating in unfamiliar building layouts. Many solutions have been developed to facilitate outdoor communication through GPS enabled devices, but there is a significantly smaller field of research dedicated to navigation of unfamiliar indoor spaces.

### Figure 1

*Blind Man Walking Inside a Mall with the Use of a White Walking Cane*



*Note:* This 1996 image depicted a blind man who was walking inside a mall with the use of a walking cane. Note the dangerously low staircase with its minimal, to no headroom, beneath which he was passing. Because there was no warning as to the staircase's presence, he was about to crash into the underside of the stairs. Such overhanging objects, or structures with inadequate headroom, are hazards that a person's cane cannot detect. "1996 image depicting a blind man walking inside a mall with the use of a walking cane" by Richard Duncan is licensed under [CC BY-NC-ND 2.0](https://creativecommons.org/licenses/by-nc-nd/2.0/)

Due to the narrow scope of research being done on navigation within unfamiliar indoor spaces, the picture above is a common occurrence. A BVI individual can walk into an objects that are located above their head due to inefficient object detection devices. Whether the location be in Germany, Russia, or anywhere else in the world, this problem occurs too frequently. Our team's goal is to make the navigation of an unknown indoor space both easier and safer, with the help of our application along with a BVI individual's preferred method of navigation.

## Navigating The World Without Vision

Navigational challenges for the blind and visually impaired are significant in Russia, though there have been efforts at improvement. As the largest country in the world, Russia has a population of 142.3 million people ("Population clock: World," 2021). Although the Russian Federation has taken steps to increase accessibility within schools and workplaces, there is a gap in accessibility for BVI individuals in terms of navigation in such contexts. The Russian government has established foundations and support systems for the BVI community. This includes regulatory support for the visually impaired to combat the social inequality these individuals face every day (Pronchev G., 2017). Additionally, there exists a plethora of Russian schools that provide special education for the visually impaired (Oreshkina M., 2009). Like Russia, Germany has many schools for the Blind as well as cities specifically made for BVI individuals, specifically *Blista*. Which is a school that primarily teaches BVI individuals and has been since *WWI*.

Navigating indoor spaces poses a challenge to the Russian and German BVI population, as it is difficult to identify and avoid objects in their path. Most public buildings, such as movie theaters, museums, and grocery stores, are not equipped with any special technology to assist the blind and visually impaired with navigating the indoor space.

Many areas of Russian and German infrastructure are not equipped with the technology needed to aid the navigation of BVI individuals. These indoor spaces cause problems for Russian and German BVI individuals because traditional techniques, such as the use of a white cane or a guide dog, can only help guide users around objects that are on the ground or up to waist level. In an indoor space, objects that are positioned above the waist pose a threat to the user and may cause injury. Research has been conducted into incorporating different sensors within navigational aids. These sensors allow for the detection of objects surrounding the user, pointing out objects that could cause harm. While this technology is a significant improvement in the field of navigation, a prominent issue is the use of GPS. Using GPS within a navigational device introduces the possibility for distorted or blocked signals, based on the building material, if used indoors. Researchers have found a way around this by using a camera that is worn by the user. This camera can detect all objects in front of the user and supporting software can process the information gathered by the camera. A drawback of this type of technology is delivery of this information to the user, in a timely manner, so they can avoid rapidly approaching objects.

With the support of various advanced technologies and building on the work of the 2021 IQP cohort's efforts on a Navigator App, our team will proceed further in designing a solution to improve indoor navigation for BVI individuals. As mentioned, an exterior camera can be used to capture information about the surroundings of the user, and the correct management of the data will be crucial to the success of this navigational system. Our group will be focusing on extracting and utilizing information that would help detect and classify various indoor objects so that we can provide obstacle avoidance support within the navigational system. However, the extra cost for a dedicated camera or any other auxiliary sensors is a huge burden for most BVI individuals. A potential workaround for a dedicated camera or processor could be the smartphone. Although the cameras for most smartphones are not ideal for performing object recognition due to their insufficient resolutions, there are promising methods which compensate for hardware limitations. Thus, considering the user convenience and the accessibility of our

navigational system, we will attempt to implement most features into a smartphone where the system would solely rely on the built-in camera and processor of the smartphone. Furthermore, an indoor guidance system will also be explored and experimented.

Our goal is to aid the BVI communities of Germany and Russia by working with the Financial University under the Russian Federation to further develop and improve the smartphone enabled object identification software created by the previous Navigator App for the Blind IQP team. To achieve this, we plan to accomplish the following objectives:

- Identify and document a common set of features of a navigational device that would improve the experience of BVI individuals within indoor navigational contexts.
- Evaluate existing technologies, including the 2021 Navigator for the Blind application code for fitness and reusability.
- Design, test, and revise our navigational device using an iterative design process.
- Finalize the design of our navigational device.

While the previous IQP team identified technology, it was not able to be sufficiently developed. We will continue to improve and refine upon their work, while verifying and validating the design. Our goal is to develop the software and supporting technologies to provide a useful indoor navigational solution for BVI individuals in Russia and Germany. Specifically, we aim to improve the speed and efficiency of current object identification software and develop feedback and communication technology between the software and the individual to improve the existing user experience.

## Chapter 2: Background

In this chapter, we discuss our review of the current research and technological developments in the field of navigation for blind and visually impaired (BVI) individuals. We begin with an overview of the general problems BVI individuals encounter when trying to navigate public spaces and the various navigational aids typically used by these individuals. Next, we discuss the specific challenges faced by BVI individuals in the Russian Federation and Germany due to differences in landscape, architecture, and public perception. After, we review the key research and technological developments from the last 10 years in the field of blind navigation and spatial awareness. This includes conclusions drawn by experts on the benefits and drawbacks of each technology, where such a consensus is available. We finish by reviewing the conclusions drawn by Allen et al. (2021) in the previous phase of the project, and which we are extending. We review the design proposed and prototyped by Allen et al, and discuss the benefits and drawbacks of their implementation.

### 2.1 BVI Individuals and Their Navigational Challenges

In order to fully understand the challenges that BVI individuals face in Russia and Germany, we must understand what an individual must be categorized as to be considered blind or visually impaired. We must also understand the social challenges BVI individuals face to consider how to contribute to improving social conditions.

Out of over seven billion people comprising the world population, there are an estimated 253 million people suffering from blindness or some form of visual impairment worldwide. According to the World Health Organization (WHO), for a person to be considered blind or visually impaired, their eyesight must fall within a certain diminished range of capability. For visual impairment, it is broken down into 3 distinct categories of mild, moderate, and severe. To have mild visual impairment, their eyesight must be between 6/12 and 6/18. For moderate impairment, their eyesight must be between 6/18 and 6/60. For severe impairment, their eyesight must be between 6/60 and 3/60. Anyone who has eyesight worse than 3/60 is considered legally blind (*Vision Impairment and Blindness*, n.d.). In comparison, a person with good vision has 20/20 vision. Following these standards, a total 3.44% of the world population is visually impaired with almost half being blind and 2.95% having moderate to severe visual impairment. This number does not include the 1.1 billion people with functional presbyopia, the gradual farsightedness associated with aging (Ackland P., 2017).

### Social Considerations

Visual impairments can affect all stages and aspects of a person's life and can significantly lower a person's standard of living. Mobility is difficult for BVI individuals, even with assistance. With environments constantly changing, even with directions and tools such as canes navigation remains difficult. Navigation is a constant struggle in a world built to be navigated by those without visual impairments. Crosswalks, lights, shifting obstacles and large crowds make life with BVI difficult. Without the ability to independently navigate BVI individuals can become isolated from others and can struggle to support themselves and their families (Hersh M., 2013).

## **2.2 Navigational Challenges faced by BVI Individuals in Russia and Germany**

### **BVI Population of Russia and Germany**

Russia is one of the most populous countries with 142.3 million citizens, making it the 9<sup>th</sup> highest populated country in the world ("Population clock: World," 2021). There are a total of 551,800 Russian people suffering from blindness, 18,503,762 nearsighted, 3,663,771 with mild visual impairment and 7,992,683 with severe visual impairments (Stewart, 2021).

With the large number of BVI individuals, the Russian Government has set up multiple foundations and support systems to help these people. The first of the programs set up is the Russian Federation regulatory support for the visually impaired (which contains those who are blind as well). The goal of this regulatory system is to “level the playing field of social inequality that this category of citizens face” (Pronchev G., 2017). As mentioned previously, social inequalities in question are access to information as well as access to navigation around many parts of the country. One of the major ways this regulatory system helps to combat social inequality is by utilizing social security and the size of the disability allowance given out by the Russian Federation. This allowance is determined by the severity of the disability as well as the Ministry of Social Security (Pronchev G., 2017). To further improve the everyday lives of those who suffer from BVI, “an active policy of constructing and organizing profitable enterprises which are specifically equipped for the blind to work at have been organized” (Pronchev G., 2017).

One of the major support systems in question is the schooling system in place for children going through lower education (preschool and general (middle / high school)) who need assistance with learning. Of all the students in the Russian Federation, 4.5% or around 1.6 million children require special education (Oreshkina M., 2009). With this large number of students needing special education, there are 10,375 schools that provide lower-level special education (Oreshkina M., 2009). Of those schools, there are 548 schools that provide education for the visually impaired that span the schooling of preschool to general school, including fifteen that are boarding schools (Oreshkina M., 2009). While institutions for special education are available, the teachers are not as readily available as only 10% of teachers have a degree in special education. With the lack of quality teachers, many students develop secondary problems that were not present before starting school such as depression (Oreshkina M., 2009). These situations coupled with the lack of supportive legislation have caused these schools to not succeed in their mission of aiding children with learning disabilities.

The population of Germans that are visually impaired is 350,000 people. This was based on the amount of people who applied for a severe disability pass, which in total is 8 million citizens. This equates to 10% of the German population being disabled and .04% of the population being visually impaired (e.V., 2022).

With the large number of BVI individuals located in Germany, the government has taken action to help aid the financial burdens of these individuals. Firstly, all BVI individuals that are completely blind or severely blind can ride any form of public transport for free as well as accompanying people. Next, many BVI individuals are given a tax break from working wages as well as five extra vacation days. Finally, they are given an allowance based on the severity of their impairment and the location of where they live. The overall largest advocate for the BVI community of Germany is the German Federation of the Blind and Partially Sighted (DBSV). This organization is the oldest consumer organization in all of Germany that “empowers people



who are blind or have low vision to speak with one voice.” This organization lobbies the government, media and housing on behalf of the BVI individuals to improve the situations that they are facing. With the help of this organization a betterment of “Education, braille literacy and information systems” and much more has become possible and only continues to grow (“The Situation of the Blind and Visually Impaired in Germany — An Overview | American Council of the Blind”, 2022).

### **2.3 Limitations of Current Navigational Devices**

Major issues that BVI individuals encounter include high cost of navigational devices and the height of the object in relation to the user. The most common way BVI people navigate is with walking canes or seeing-eye dogs. While the most effective method of navigation is to use seeing-eye dogs, these animals are extremely expensive and difficult to obtain, so their widespread availability to the BVI community is limited (Chen Z., 2021). This leads people to use the more common option, a white walking cane, which is a universal symbol for visual impairment or blindness. Many people opt to use the white walking cane thanks to its “low cost, reliability, efficiency, simplicity and ... direct physical contact with the ground...” (Khan I., 2018, p. 2). With this physical contact, the user can decipher the terrain around them, such as drop-offs (curbs) or obstacles. This is not the only method of navigation as the skill of echolocation can be developed and used. Echolocation “gives the perception about the surrounding environment in the form of the object, its location, dimensions, density, and exploits the variation in sound for the individual” (Khan I., 2018, p. 2). While the white walking cane provides guidance at a lower cost, it has its limitations as well. An example of one such limitation relates to where the object is located, height-wise in relation to the user. Khan, in their study remarked that, “Besides the essential techniques used by the blind people to benefit from the white cane, they are unable to detect trunk, head-level, as well as other over-hanging objects” (Khan I., 2018, p. 3). This causes the white walking-cane to be considered by experts as an adjunct to newer navigational devices, instead of a standalone solution.

### **2.4 Blind Communities in Russia and Germany**

#### **Rusinovo, Russia**

While schooling for the Blind and Visually impaired in Russia is lacking, a town was created to help alleviate some common stresses. This town is Rusinovo, which is located in the Kaluga region. The city was formed in 1948 during the height of the Soviet Union, where people with visual impairment would move to from all over the country. With this high influx of BVI individuals, buildings and factories were specially made to accommodate their needs. By including the needs of BVI individuals and allowing them to work at the factories, they could solve some of the social issues they faced as they could earn double the wages of a seeing engineer. While this situation was going well, the partner factory shut down, causing the factory to diminish in necessity and fall into ruin (“Documentary: Russia's Village of The Blind”, 2022).

#### **Marburg, Germany**

Marburg is a city in central Germany known for being adapted to make life easier for visually impaired and blind individuals. Marburg’s accessibility can be attributed to the *Blindenstudienanstalt* school, also known as Blista. This school was founded in *WWI* and served to provide opportunities for young men who were blinded during war. Blista has developed

many inventions for BVI individuals including a tactile mathematical font. Marburg, the city surrounding Blista, has been shaped to include high levels of accessibility for BVI individuals. Navigation in the city is aided by “beeping traffic lights, pavements and floors with ridges and bumps that act as tactile signals of hazards or barriers” (Hardach S., 2021). The natural hilly landscape of Marburg makes orientation easier, as an individual can simply note if they are walking uphill or downhill. Additionally, most buildings have raised maps and floor plans, including miniature models of major sights to allow BVI individuals to feel each landmark (Schelin et al., 2021).

## 2.5 Recent Technological Developments and Research on BVI Navigation

Our review of the current research reveals that the developing field of blind navigation is divided into two dominating subcategories: hardware-oriented solutions that make use of a variety of sensors for identifying and navigating obstacles, and computer vision algorithms that make use of limited hardware and process images using complex, resource intensive algorithms.

### Hardware-Oriented Navigational Aids

A broad spectrum of hardware-oriented solutions aims to modify and improve existing navigational technologies, such as the ubiquitous white cane, with modern technology. As shown in Figure 2, a variety of these “smart canes” exist on the market today, using technologies such as ultrasound and camera-based distance sensors. The benefit of ultrasound is simple, cheap detection of obstacles in a short range, at the cost of precision in obstacle detection. (Vineeth et al., 2021, p. 1) Camera-based distance sensors offer very precise obstacle detection at the cost of increased processing intensity and cost. These sensors are often embedded in the body or handle of the cane and cause the handle to vibrate when an obstacle is detected.

### Wearable Sensors

An alternative to smart canes that avoids some of the drawbacks inherent with them are wearable sensors. Wearable sensors are used in outdoor navigation and vary in size, purpose, and design. There are many kinds of sensors used in navigational technologies, each with their own useful applications. For example, an accelerometer, which measures the acceleration of something with respect to an *inertial reference frame* (in this case, the earth, which is technically moving through space, but is assumed to be still for purposes of navigating across it) (Tinder, 2007). An accelerometer can be used to measure an individual’s rate of speed at any instant in time, in conjunction with mapping software to determine where in a room a user is, based solely on acceleration data. Many such techniques exist with a variety of different sensors, which can be used in conjunction with each other for incredibly accurate results. The drawback to such methods is the intense processing power required to coordinate multiple sensors at once, making such technology currently infeasible for many consumer devices.

However, sensors commonly used in the field of blind navigation utilize ultrasound or radar to detect obstacles that may lie directly in one’s path. The device conveys this information to the user, usually in the form of vibrations or voice feedback. This technique, though less precise than the previously discussed sensors, is uniquely suited for blind navigation, as they make use of techniques already used by many BVI individuals. When individuals use a cane, they can detect objects in front of them quite easily, as the cane will collide with any potential obstacles in the user’s path. Ultrasound and radar-based sensors give the same feedback, by

detecting objects within a certain distance of the sensor. Adoption of navigational aids with these technologies could reduce the learning curve required for BVI individuals to effectively use the technologies.

Several navigational aids have been designed and prototyped by the scientific and engineering communities. Babu et al. propose a solution that would use an ultrasonic sensor embedded in a belt to detect objects at waist level and deliver a vibration to the user (Babu et al., 2021). Bouteraa proposes a more complicated system composed of “an ROS-based embedded controller, an eyeglass frame containing three ultrasonic sensors with the related vibrator modules for obstacle detection, and a similar accessory that can be worn in the hand for ramp and up–down barrier detection” (Bouteraa, 2021, p. 1). Pradeep Kumar et al. proposes a set of sensors embedded in the user’s shoe, which would guide the user along a predetermined path (Pradeep Kumar et al., 2021). These various approaches to wearable navigational assistance devices each have benefits and drawbacks but have yet to be adopted by the BVI community. We found no studies focused solely on the reasons for this lack of adoption, but one possible theory is that the non-adoption stems from the same reason as for the non-adoption of the smart cane, namely the durability, usability, availability, and cost of the devices.

The key technical drawback with most wearable sensors is their detection accuracy, especially their inability to detect small objects such as thin branches or tripping hazards such as cracks in pavement. Additionally, for those sensors which don’t rely on a binary obstacle detection system which can give feedback through vibration, the voice feedback of the device could be confusing or even misleading if the user comprehends it incorrectly. Although speech instructions are informative and easy to learn, they require more time to inform the user, who risks bumping into an obstacle before the alert message is finished speaking. Moreover, speech distracts the user from the surrounding environment because decoding a verbal message requires a significant cognitive effort (Presti et al., 2021).

For indoor navigation, hardware-dependent designs are the most used and explored solution. However, a big downside for such a solution is the cost. The hardware components are often expensive, and they require regular maintenance to make sure devices can operate properly. Indoor navigation devices indicate a need for a less expensive and more accessible solution that could better address the needs of BVI individuals.

### **Machine Learning Oriented Navigational Aid**

There are a greater number of hardware-oriented solutions than sensor-based solutions, typically powered with complex navigational algorithms. Indoor navigation for the BVI community faces two major challenges: object detection and navigation (the ability to guide users from point A to point B). Luckily, with emerging computer vision and machine learning studies, we can utilize existing tools to help us confront these challenges. Two promising tools are SLAM and Detectron2/D2Go, because both are active, open-source projects.

*Simultaneous localization and mapping (SLAM)* is a method widely used for robot navigation and robotic mapping. It is the process of mapping an area whilst keeping track of the location of a device, such as a robot or smartphone, within the area. SLAM makes mobile mapping possible. It allows map construction of large areas in much shorter spaces of time as areas can be measured using mobile robots, drones or vehicles. There are many different types of

SLAM algorithms and approaches. The performance of different SLAM algorithms varies depending on the specific use case. For instance, some SLAM technologies are dedicated to drones, where the algorithms are most capable of doing bird-eye view mapping. A navigational device for BVI individuals would require SLAM algorithms that can mimic human eye-level mapping. Additionally, some SLAM schemes are computationally intensive which conflicts with user needs, particularly power consuming and poorly performing apps running on their smartphone. With the proper choice of algorithm and approach, features can be developed to provide some degrees of navigation through environmental mapping and obstacle avoidance for BVI individuals (GeoSLAM, 2022).

Another promising tool is Detectron2. It is Facebook's open-source library for implementing state-of-the-art computer vision techniques using PyTorch. Facebook introduced it in October 2019 as the successor of Detectron and maskrcnn-benchmark. It is especially effective for object detection, segmentation, and other visual recognition tasks. Detectron2 can easily carry out the task of object recognition for a navigational device. By further classification of objects, Detectron2 can provide more robust obstacle avoidance support compared to using SLAM. With the new D2Go extension, FBNet models, that are already optimized for mobile devices, can be created with architectures that can efficiently perform object detection and segmentation tasks where segmentation is the process of dividing an image into different regions based on the characteristics of pixels to identify objects or boundaries to simplify an image so that it can be analyzed more efficiently. (*Facebook - Meld je aan of registreer je*, n.d.-b).

In summary, hardware-oriented navigational aid addresses some of the challenges that BVI individuals face, however, they are limited, costly and very cumbersome to set up. A smartphone application could be the ideal mobile navigational device to address such needs. Smartphones are equipped with a CPU, an operating system, various sensors (GPS, accelerometer, gyroscope, magnetometer, pedometer, and compass) and can run mobile apps for data processing (Allen et al., 2021).

## **2.6 Review of Previous IQP on Navigator App Development for the Blind Project**

### **Goal and Outcome**

Engaging with many of the same topics as we have so far discussed, in 2021 an IQP team worked on a “system to help visually impaired persons of all ages and levels of impairment, navigate indoor environments safely” while using an iOS-based smartphone application (Allen et al., 2021, p.21). The team's objectives were to provide orientation to application users, train the program to avoid spatially changing objects where the objects are not still such as walking pedestrian, and determine an output method for directions.

The outcome of the 2021 IQP team's efforts led to a solution in the form of an iOS application that utilizes training data gathered from building layouts and objects to alert the user of hazards in their path. While providing directions to a user's destination, the application converts the detected hazardous objects, along with directions, into synthesized speech that is then played back to the user through a speaker or headphones. This outcome was made possible through the compilation of background research, feedback from survey participants, and

technical expertise from students at Financial University in Moscow, Russia. The prototype produced for the purposes of this project was able to accurately navigate a user through an unfamiliar indoor space, although the image detection software did not work well in dynamic environments.

### **Challenges and Learning**

The biggest challenge encountered by the group was the ineffectiveness of GPS for the indoor navigational environment. Indoor spaces attenuate the strength of the GPS signal and can distort it. Our group must overcome this challenge as well, as we need to develop a navigational device that does not use GPS. The group overcame this challenge using an iPhone, a smartphone possessing necessary sensors for orientation.

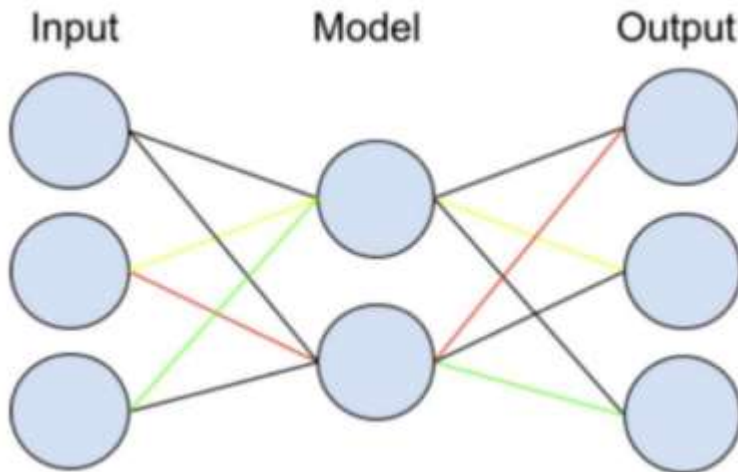
The first objective of the team's project was to provide orientation to its application's users. The team recognized that this approach was multifaceted and that there were many potential solutions to explore. Based on their findings, the team saw the Apple iPhone as the most promising technological solution. They determined that their device should possess the following features:

- Camera with high resolution
- Gyroscope sensor
- Accelerometer sensor

They evaluated the iPhone's camera and sensors and determined they could implement a machine learning system for image processing and object detection. The team found that, "high mobility and resolution are essential camera features for accomplishing position correction" (Allen et al., 2021, p. 21), and thus felt the Apple iPhone was an ideal choice for image data collection since it possesses a high-resolution camera. Along with the iPhone camera, gyroscope and accelerometer sensors can be used to determine an individual's orientation within a building. The team developed a program where, "readings from the integrated phone accelerometer and gyroscope were used to understand where the person is heading with respect to the building" (Allen et al., 2021, p.22).

The second objective of the team's project was to train the program to avoid spatially changing objects. Spatially changing objects are not stationary, meaning they can move position over time. For example, a chair is a spatially changing object because it can be moved by a person and placed in different positions over time. In this objective, the group developed a machine learning system that consisted of a built data set and a neural network. They describe the neural network as, "a machine learning tool that is a collection of functions forming a statistical model that is used to interpret a data set (see Figure 2), a process modeled after the logic of the human brain" (Allen et al., 2021). They then trained this neural network to recognize and avoid spatially changing objects. The group focused on two-dimensional systems that were evaluated using raw pixel values. Their neural network training procedure is as follows: "an image is broken down into individual pixels to find similar groups of curves, similar curves are grouped and assigned to layers, then layers are compiled into logical classifications... layers are datasets that have similar characteristics that are going to be classified to specific items by the training process" (Allen et al., 2021). After the training was complete, the machine learning system could identify and locate objects when given access to a live video feed.

**Figure 2**  
*Simplified Diagram of how a Neural Network Functions*



*Note:* Each blue circle represents a set of data, and each line represents a function that is applied to the data. The blue Input circles represent the images that are used to train the model. The lines between the Input and Model circles represent the system grouping the pixel curves. The Model circles represent the groups of curves that share similar features, and the lines between the Model and Output circles represent the system compiling the layers into classifications. From “Navigator for the Blind”, by Allen et al., 2021. Copyright 2021 by Allen et al. Reprinted with permission.

The third objective of the previous team’s project was to determine the output method of communication with the BVI user. The team first gained an understanding of BVI individuals’ perception of space. They then created a cognitive map for BVI individuals which consisted of, “determining where an object can be found in a physical plane with respect to other objects” (Allen et al., 2021). Their next step was to determine the best way to communicate audio commands to the BVI user. Interviews with BVI individuals helped the team gain a better understanding of how audio commands should be communicated. They made the communications twofold and included both clear verbal directions and variable tones that increased with intensity as the object got closer.

### **How Our Team Will Move the Project Forward**

The previous team outlined future recommendations for this project, which our team plans to consider and expand upon as appropriate in our own methodology. These recommendations are as follows:

1. “Expand the number of objects that can be detected by adding additional reference images.
2. Test navigation system in different indoor environments to better understand accuracy of static and dynamic obstacle detection and avoidance.

3. Make audio instruction more efficient and more specific by combining navigation and obstacle avoidance instructions” (Allen et al., 2021, p. 37).

For example, we will closely review the code from the previous IQP team’s program as a guide to refactoring their routines where possible into SLAM or Detectron2 to increase detection accuracy. Additionally, we will follow the recommendation to increase the number of objects that the program can detect by adding additional reference images to train the machine learning system. After creating a prototype of our own, we will test the navigation system in various indoor environments including static and dynamic obstacles. To decrease user confusion, we will integrate navigational direction and obstacle avoidance instructions into cohesive audio instructions.

## **2.7 Conclusion**

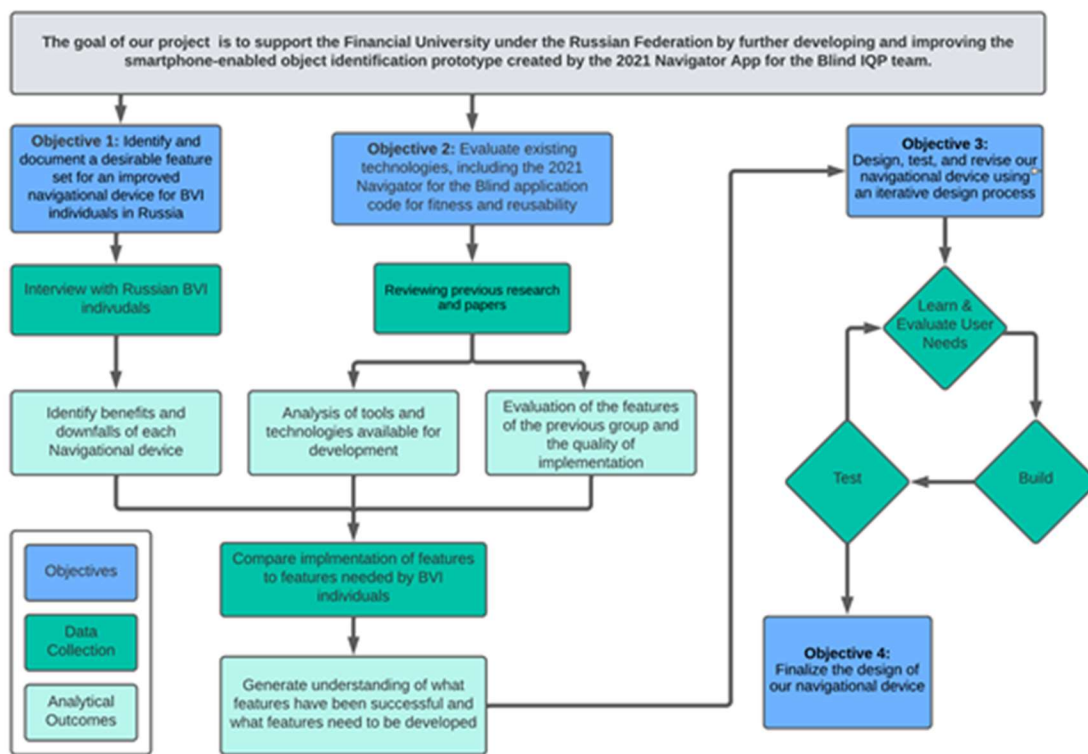
This background has discussed the key areas of interest for our project. First, the navigational problems that BVI individuals face, how these problems are experienced in the Russian Federation and Germany, and finally the technology that is being researched and developed to solve the issue. From our preliminary research, we have found that one of the major problems faced by BVI individuals is indoor navigation in a crowded area containing obstacles. In the Russian Federation and Germany, most of the buildings attenuate the GPS signal (as is typical with most buildings), which causes applications that are used in accompaniment with the white walking cane to be rendered useless and in need of new solutions. To address this problem, researchers are looking towards devices and applications that can be enabled by SLAM or Detectron2 to identify objects, mark their position, and relay that information to the user to provide them with accurate directions. While progress is being made, detection will continue to be improved as technology improves. To contribute to this progress, we will work to improve the accuracy of indoor object detection by validating, debugging, and improving the approach from the previous IQP.

## Chapter 3: Methodology

The flowchart below depicts the methodology that the group will follow in order to complete our deliverable of designing a navigational app for the blind. The blue boxes are our main objectives with the green and teal boxes being the route that we will follow in order to complete our objectives. The main objective that our group worked towards was objective 4 which is the final deliverable of the project, developing an app for the blind. In order to fully develop the app, we needed to understand the problems that BVI individuals face, and solutions already being explored. This is what led to objective 1 and 2 to be started first and lead into objective 3 of testing.

**Figure 3**

*Flowchart Displaying Project Objectives, Data Collection Methods, and Analytical Outcomes*



The flowchart above depicts the methodology that the group will follow in order to complete our deliverable of designing a navigational app for the blind. The blue boxes are our main objectives with the green and teal boxes being the route that we will follow in order to complete our objectives.

### 3.1 Interviews with BVI Individuals

To determine the best features to apply to our navigational device and make it as effective as possible, we directly consulted with the individuals who will be using the device the most. These individuals are the BVI population, specifically those within Moscow, Russia, and



the cities of Marburg and Berlin in Germany. Interviews were conducted by two of our Russian counterparts, Olga Mironova and Kira Kovnat. Olga interviewed BVI individuals from a seeing-eye dog center in Moscow. Kira interviewed BVI individuals from the Izhevsk School for Blind Persons in Russia. By learning from the BVI population, we were able to narrow down the solutions to the most useful core of features, avoiding those that are least effective. The questions asked by both Kira and Olga can be found in Appendix A and B, with the English questions in Appendix A and Russian found in Appendix B.

As for observing German BVI individuals' navigational habits, the group traveled to Marburg, Germany. This town contained the German Competence Center for People with Blindness and Visual Impairment (also known as Blista), which is an area that contains a high population of individuals who are Blind or Visually Impaired. When the group traveled to Marburg, we observed the many precautions that have been taken to aid in the safety of travel and navigation for BVI individuals. This included raised sidewalks, raised maps of buildings, as well as streetlights and crosswalks that emit sounds to help with guidance.

With the interviews being semi-structured, we will have a combination of open-ended and closed-ended questions. The open-ended questions were transcribed by the group and reviewed with the qualitative method of coding. The closed-ended questions were reviewed and placed into a chart for later correlation to our data. This method of coding was reviewed while using inductive methods through statistical and thematic analysis, as well as the development of a codebook. This allowed our group to characterize the benefits and drawbacks of current navigational devices as we develop possible solutions that can be implemented in our final design.

The features being implemented into the navigational device are dependent on both feasibility and interview results, which will be difficult for us to learn given that we are not traveling to Moscow, meaning we will not be able to meet face-to-face with our interview participants. Further, given the likelihood of limitations in technology, we are not able to identify a workable consensus among the groups, although we will not be prompting the subjects in this regard.

With the knowledge gained from these interviews, we have a better understanding of the features desired by the BVI community. Though the limited time frame makes it infeasible to implement every suggestion, we compiled a core set of features that were highly requested. From these features, we designed a set of baseline criteria to be applied to each iteration of our design, which will measure the performance of the design in each desired aspect. Each time the software is iterated it was compared to these criteria and the following iterations attempted to correct the problems of the previous based on that rubric. The primary test for each major version revision was an obstacle course made up of student participants with the goal of navigating through the crowd. Each test was evaluated for speed of navigation, accuracy of detection, and frequency of collisions.

### **3.2 Evaluation of Existing Navigational Technologies**

After getting a better idea of the features that blind and visually impaired individuals want in navigational aid, we began looking at the technologies that already exist in this sector,

and the ones we might be able to use in our design. Notably, we looked at the code from the previous phase of the project, which used the OpenCV (opencv.org) machine learning based computer vision tool to help users navigate indoor spaces. We also looked at the benefits and drawbacks of their approach, and then compared it to similar technologies of SLAM and Detectron2. Though many more solutions exist, we are limited to using open-source software, and we will not be implementing our own algorithms in the prototype phase.

### 3.3 Iterative Design Process

We applied an iterative design process. In each iteration, we learned and evaluated the users' needs, design, and built the deliverable with the best tools available, then tested our prototype with user feedback. If the testing result did not meet the target demands, we repeated the process again with the necessary adjustments.

#### 3.3.1 Step 1

In the first step of the iterative design process, our team learned and evaluated users' needs by using the feature list developed from Objective 1 to select the features our group would implement within the constraints of time constraints and our expertise.

In the second step we designed and built a prototype of our design. Before moving straight into the building portion of our prototype we evaluated the information found in Objective 2 and selected the best technologies that fit our design specifications. Work was delegated to both the WPI and Financial University students. The WPI group worked on the evaluation of Detectron2 while the Financial University students evaluated SLAM. Once we selected our desired technology, Detectron2, we used a version control system, GitHub, for team collaboration between the American and Russian teams.

During the start of the iterative design process our group worked on 2 steps simultaneously, evaluating Objective 1 and Objective 2. To evaluate Objective 1, our team learned and evaluated users' needs by using the feature list developed from the interviews to select the features our group would implement within the constraints of time constraints and our expertise. To evaluate Objective 2 our group researched and selected the best technologies that fit our design specifications. Work was delegated to both the WPI and Financial University students. The WPI group worked on the evaluation of Detectron2 while the Financial University students evaluated SLAM. Once we selected our desired technology, Detectron2, we used a version control system, GitHub, for team collaboration between the American and Russian teams.

During the second step of designing and building our prototype our group worked on training the machine learning aspect of our design. In order to train our model, we used a website called Roboflow which helped compile pictures for the training program. This website took the imported pictures and inverted them, changed the color, highlighted it. These changes were made to train the program to recognize the selected object in any orientation that might appear with great confidence. The website then allowed us to pull the metadata from the images and feed them into the machine learning algorithm we built.

Additionally, in the second step of the iterative design process, our Russian counterparts from the Financial University worked on developing a schema for the operation of the

navigational aspect of the application. This schema was put together to serve as a basis for future convergence of object detection and simultaneous localization and mapping components in a unified navigational aid.

### **3.3.3 Step 2**

In the third step, we tested all features of the deliverable and made sure they were functioning well and met the target demands. This was conducted by using a real time test, specifically an obstacle course. The obstacle course was set up in a way to test every part of our navigational device. This included objects that would be located around head height to test the object detection of that area. Next the course would include people who would walk in front of the camera to test its detection against objects moving around still objects. The course was first tested by blindfolding one of our group members before building the course and having them walk the course without knowing the path. The obstacle course was then photographed to be recreated later to test future iterations of the projects using individuals who were not a part of the team. The performance was examined and evaluated with respect to our testing criteria determined in Objective 1.

### **3.3.4 Step 3**

In the fourth step, we evaluated the test results of the current iteration and determined if it successfully fits the requirements for our design. If our design and test results fell short of the expectations our team set, we cycled back to the first step, identified the areas of failure with the current iteration and brainstormed solutions to improve the next iteration. When it is determined that our section of object detection is complete as well as our Russian counterpart's section of simultaneous localization and mapping components are done, they will be combined to finalize our last iteration.

## Chapter 4: Results and Analysis

The field of indoor navigation can be split into 2 distinct parts, the first part is navigating within stationary objects and the second is navigating among non-stationary objects. Each of these parts are important to focus on and study to solve the dangers of indoor navigation. To solve the distinct issue of navigating stationary objects our group created a set of interviews which our Russian counterparts conducted on our behalf. Their results were used to create a feature list to implement into our device and set of features that were dangerous to BVI individuals. For the second and more prominent issue of navigating around moving objects, our group created our app, testing it at each iteration, to help navigate around these objects safely. Below we will go into each distinct part of the indoor navigation starting with the stationary object problem, followed by the strategy used to solve the moving objects.

With the aftereffects of Covid-19 still lingering in Europe, along with a war occurring between Russia and Ukraine, our group was relocated to Berlin, Germany for our project. As a result, our team was unable to conduct interviews with Russian individuals directly. Instead, we created a set of interview questions for our Russian counterparts to use as they interviewed BVI individuals around Moscow and Izhevsk. Additionally, our group traveled to Marburg, Germany and interviewed BVI individuals there to gather insight for our navigational device prototype. While our group ran into problems over the course of this project, we were able to work around them and adapt.

### 4.1 Stationary Object Data Collection

#### 4.1.1 Interviews Conducted by Kira Kovnat

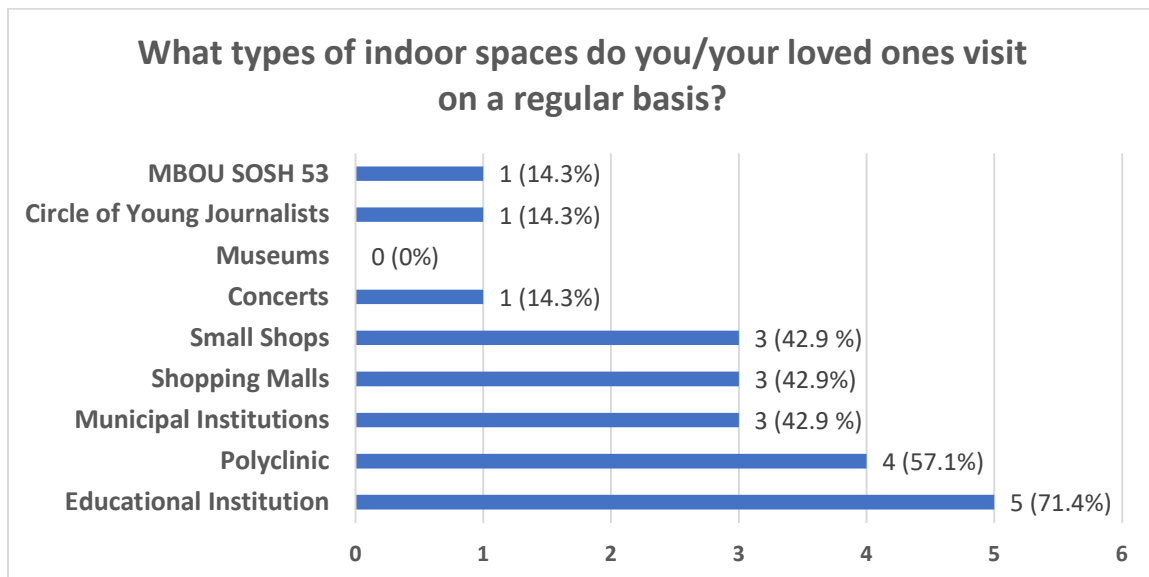
The goal of these interviews was to understand the issues of navigating around stationary objects, specifically which objects were troublesome to detect and move around while using a white walking cane. Our team received access to a Google form which was filled out by eight BVI individuals who answered all questions found in Appendix A. The mixture of closed-ended and open-ended responses we received allowed our team to implement the qualitative method of coding. For the closed-ended questions, the team was able to create pie-chart diagrams that depict the outcomes from the interviews.

The two graphs located below show two of the most important questions asked during the interview, which were "What objects in the room do you/your loved ones avoid with your navigational device?" and "What types of indoor spaces do you/your loved ones visit on a regular basis." These questions give us part of the answers needed to address the stationary object issue of indoor navigation. It helps to address the answer as it highlights the objects to look out for as well as the locations where they walk the most, allowing us to infer other dangerous objects in these locations. With this information our group was able to compile a feature list that was implemented into our app to avoid these objects, such as steps, flooring changes, doorframes, and floor height differences. To determine the best method to deliver the warning to the users of the app if one of these items is detected, we must look at another important interview question. This question was "Which object detection method would be most preferable for you or your loved ones?" This response to this question gave our group most of

our feature list which we made to implement into our app. The key features that were the most requested were vibration, voice command and detection of people within a room.

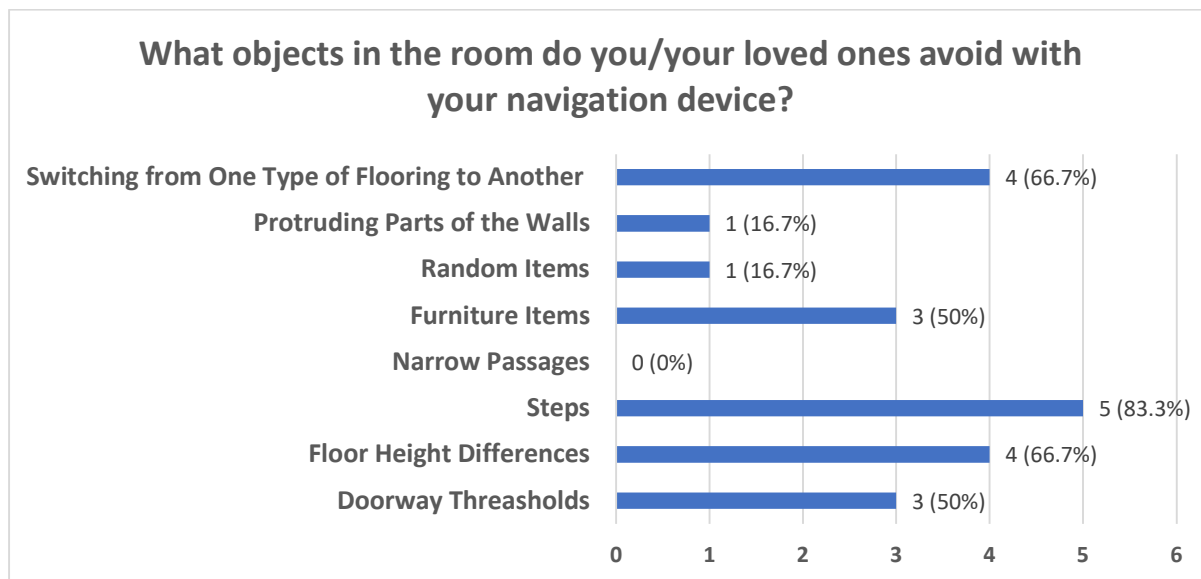
**Figure 4**

*Graph depicting the most visited places by BVI individuals*



**Figure 5**

*Graph depicting the most avoided objects by BVI individuals*

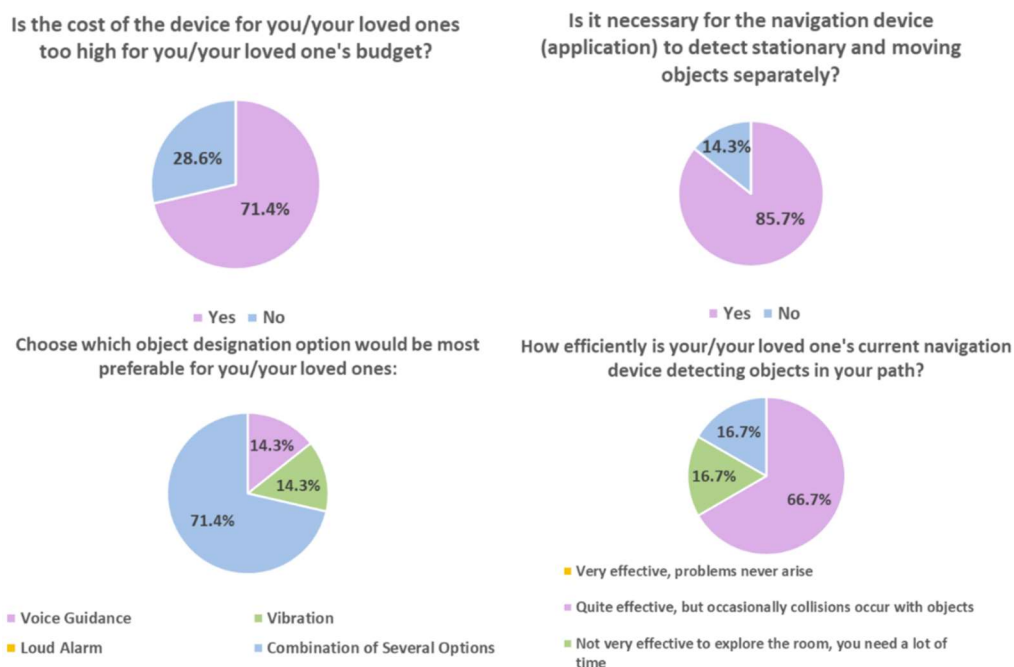


From the qualitative method of coding, our group was able to discover common trends that led us to implementing certain features into our device. The first finding was that the demographic of our interviewees was composed of blind individuals as opposed to the visually impaired. The second finding was that anything over 2000 rubles was too expensive for BVI individuals and their families. These trends helped to answer the lingering questions that we had,

which included,” Can we add extra attachments to the device, such as a harness or another camera?” As well as the biggest possible question” Should we focus on stationary objects or moving objects?” Based on the answers found within the charts below, our group decided to fully focus on moving objects, specifically people.

**Figure 6.**

*Pie-charts depicting the results from Kira Kovnat’s interviews*



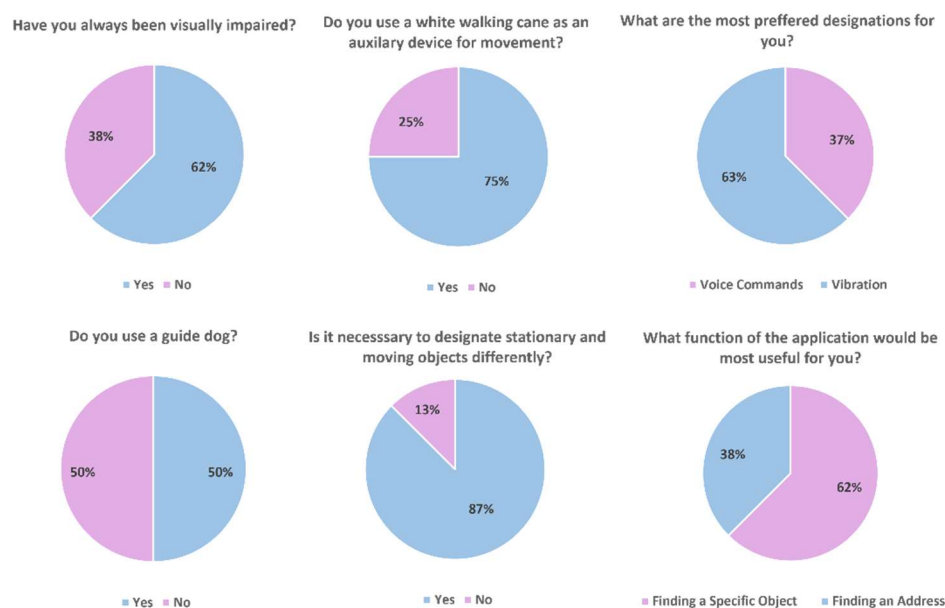
With the information collected by Kira we were able to focus on the walking cane users of the BVI population. To fully round out the BVI population, our Russian counterpart, Olga, interviewed BVI individuals who navigate using seeing-eye dogs.

#### 4.1.2 Interviews conducted by Olga Mironova

Another Russian counterpart, Olga Mironova, provided our team assistance with interviewing blind and visually impaired individuals who are associated with guide dogs.ru, which is an organization in Russia that provides guide dogs for blind and impaired people. The interviews gave our team insight on the current methods of navigation and the challenges encountered by BVI individuals. The interview questions were very similar to those administered previously by Kira Kovnat, with a few minor changes made. Namely, this form of interviewing allowed the participants to give open-ended answer feedback that was in more detail. From both the open-ended and closed-ended questions, our team was able to determine the features of a navigational device that BVI individuals believe to be most useful.

**Figure 7**

*Pie-charts depicting the results from Olga Mironova's interviews.*



As depicted in Figure 7, the interviews conducted by Olga provided insightful results for our team to consider as we developed a navigational app for the blind. The majority of interviewees had always been visually impaired and used a white walking cane as a navigational, while half of the interviewees used a guide dog. Most interviewees believed that it was necessary to designate stationary and moving objects differently and preferred a designation in the form of vibration. The majority also believed that finding a specific object would be the most useful function of the application for them, as opposed to finding an address.

For the open-ended questions, we received a wide range of responses. When asked what function of their current navigational device could be improved, the interviewees answered that improvements could include: increased accuracy, color determinant, higher quality and completeness of cartographic data, larger battery capacity, and a feature that voices stops and stations.

When asked how effective their current assistive device was at detecting objects in their path, interviewees responded with a variety of answers. Some said it was about 50/50, while others believed the guide dog perfectly bypassed obstacles or stops in front of them. A few interviewees stated it was not always effective.

As for the difficulties navigating indoor versus outdoor spaces, interviewees felt that a lot of obstacles, such as crowds, corners, and chairs often got in the way. They believe it is difficult to use a cane because the room is often cramped. When navigating indoors, it is difficult to search for an exit, glass doors, and in office buildings, search for the right office. Some additional factors that increase the difficulty of navigating indoors include the absence of poorly

readable office numbers, poor lighting, and doors that open outwards. One interviewee stated that they are always afraid of getting hit on the forehead by the door. Inside hospitals and offices, the doors are wide open which makes looking for a room number difficult. Additionally, in shopping centers, especially in the newer ones, it is difficult to read the signs as they are high above a person's head and are hardly visible. This also applies to restrooms, as signs indicating them are located high or not in obvious places. The most prevalent response included difficulty with indoor orientation and obstacle avoidance.

When asked what features of their navigational device that they find useful, interviewees responded with very few positive comments, compared to their difficulties. Some interviewees believed their current device was easy to use and gave them the ability to quickly get to the right place. Others found their camera, magnifier, and zoom features to be most helpful. In regard to navigation, some users find their Google maps with route voiceover, GPS navigation, and smartphone useful.

When asked what difficulties they encounter when using a guide dog, interviewees responded saying that their dog was difficult to obtain, was often distracted by other dogs, and needed certain commands to help with various tasks instead of helping on their own. For example, one interviewee said that without a command, their dog does not pick up dropped objects. This was problematic because a user does not always recognize when they drop something or if they lose small items from their pockets. Additional difficulties with guide dogs include all of the feeding, treatment, walking, etc. Lastly, guide dogs are not always welcome at certain shops, cafes, restaurants, hotels, and taxis.

When navigating indoors, the interviewees could not provide a solid answer on what function of their device best helped them navigate indoors. One interviewee responded that nothing helped him navigate indoors, only his hands, feet, and sometimes his forehead. Other interviewees resorted to the help of escorts.

#### **4.1.3 Compilation of Feature List**

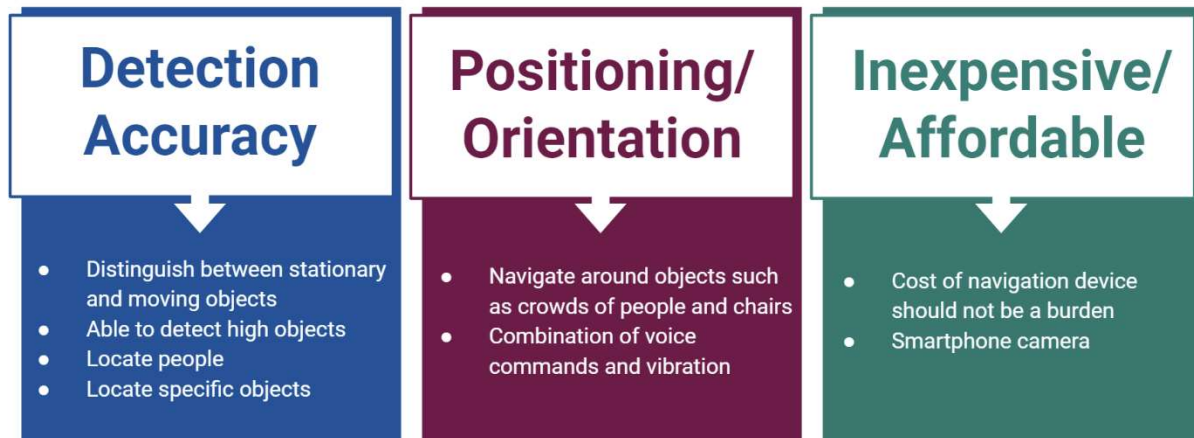
From the interview results, our team constructed a feature list which includes the most desired features of a navigational device. The input from blind and visually impaired individuals in Russia allowed our team to organize these features into three main categories. The first category is detection accuracy. Interview responses stated that our application should be able to distinguish between stationary and moving objects. Other objects that are desired to be detected include high objects located above the head, people, and other specific objects. The second category is positioning and orientation. Interviewees stated they would like our application to provide them with a greater sense of positioning and orientation. They want to be able to navigate around objects such as crowds of people and chairs. To do so, they prefer to be guided using a combination of voice commands and vibration. The third category is cost. Numerous times, it was indicated that an individual's current navigation device was too expensive, and that our application should be inexpensive, affordable, and the cost should not be a burden on the BVI individual or their family. From our preliminary research, our team decided to pursue an application meant to be used on a smartphone. Confirming our prediction, interviewees indicated that they would like our navigational device to use their smartphone camera. This greatly reduces



additional costs, as an individual would not have to purchase any additional equipment to use our application.

**Figure 8**

*Categories that each feature was compared to before being added into the app*



#### 4.1.4 Observations from Marburg about Adaptations for BVI Individuals

When our group traveled to Marburg, we made a plethora of observations about the blind community there. We spent the day walking around the city looking for adaptations that Marburg made specifically for the BVI population. While walking around the city, we were able to find many adaptations for BVI individuals on the ground, including defined and smoothed sidewalks, metal on the edge of stairs to indicate the start and end of sections, as well as rough sections in front of stores to indicate the doorway.

**Figure 9**

*Images depict the walkways and stairs that BVI individuals may encounter in Marburg*



For both informational and navigational purposes, the town of Marburg displayed metal casts of important landmarks for BVI individuals to feel, allowing them to understand the architecture of the area. These casts worked in conjunction with raised city maps to show a detailed overview of the area with each feature having a different texture.

**Figure 10**  
*Map of Blista Campus*



*Note:* This image shows the Blista Campus map, which details the campus using differently textured materials. Objects such as stairs, parking lots, grass areas, entrances to buildings, and wheelchair access points are denoted on the map using various shapes and textures. For example, the grass areas have a dimpled texture. A key is given below the map so that BVI individuals can figure out what each texture or shape represents.

**Figure 11**  
*Tactile Diagrams of buildings in Marburg*



*Note:* These images depict three-dimensional miniature replicas of Marburg landmarks made out of cast metal. These tactile maps are meant for BVI individuals to have the opportunity to “see” these buildings through their sense of touch.

The trip to Marburg allowed our group to identify similarities and differences between Berlin and Marburg. One major similarity was chirping crosswalk signals that indicated when it was safe to cross the street. While both cities possessed these devices, the ones located in Marburg were much more distinctive. Two very different chirping tones were played to signal a red light or a green light. Both cities had raised maps and casts of buildings, however they were more prevalent in Marburg. These maps and tactile figures were harder to find or not displayed as often in Berlin. This trip confirmed that indoor navigation should be the focus, as there are many implementations for outdoor travel already in place, but few detailed for indoor means.

#### **4.1.5 Observations from Berlin about Adaptations for BVI Individuals**

As our team stayed in Berlin, Germany for four weeks, we observed where the city was and was not accommodated for blind and visually impaired individuals. Public transportation is very prevalent in Berlin, and it is very efficient. Our team observed Braille in many transportation locations, namely the “STOP” buttons on public transportation that signified an individual would like to get off at the next stop. Additionally, at all crosswalks, there was Braille on the light pole and a special place for the BVI individuals to press a button to indicate they would like to cross. The crosswalks utilized a chirping noise in two different tones, one for when the signal was red and individuals should not cross, and a higher pitched chirping for when the light turned green, and the individual was permitted to cross the street.

Within buildings in Berlin, specifically museums, our team observed accommodations for blind and visually impaired individuals. Many museum tours included areas for BVI individuals to “see” by feeling a small replica of certain monuments or buildings. An area of discrepancy that the team observed was with the Reichstag building, Germany’s parliament building. On their website, the Reichstag Building tour advertised that an audio guide tour for BVI individuals would be supplemented by twelve tactile reliefs of special objects. However, these tactile portions of the tour were not to be found.

Over the course of the month, our team observed a handful of individuals using a white walking cane but did not observe any guide dogs in use. Our team also took note of the many uneven surfaces that Berlin presented. Many times, both flat concrete and bumpy cobble stone would be used in conjunction on the streets. This posed a tripping hazard as the cobble stone was not uniform and could easily trip an individual. We observed this as a team member, who was not visually impaired, tripped when the ground surface changed from flat concrete to cobblestone on the sidewalk.

**Figure 12**  
*Various aids for BVI individuals in museums*



*Note:* This image details the raised maps and enhanced pictures that can be found in galleries and museums for those who are blind or visually impaired.

#### 4.1.6 Comparing Berlin and Marburg's Adaptations

From the analysis of our interview results, our group created a list of features to implement into our final navigation device design. This feature list included aspects of detection accuracy, positioning and orientation, and cost. In addition to this feature list, our team traveled to Marburg, Germany to conduct research in the form of observation. From our observations, in conjunction with the feature list, our team was able to analyze the differences between the two. Our interviews came from Russian BVI individuals, whereas our observations were of a German town adapted specifically for blind and visually impaired individuals. Through interviews and observations, our team saw where improvements needed to be made in Russia and where the strengths were in Germany.

## 4.2 Moving Object Data Collection

### 4.2.1 Comparing Open-Source Code

When the team first started looking at the code from the previous year's Navigator App for the Blind, we saw some issues with the code. These issues caused our group to move away from the machine learning software, OpenCV. This narrowed down our code choices to SLAM and Detectron2. After having a meeting with our Russian counterparts, the decision to split the workload was made, with them focusing on the mapping software with our group working on the

object detection. This allowed our group to solely focus on detecting objects which allowed us to implement many of the requests from Objective 1 into our device. While implementing the requests we had to switch to a model of Detectron2 that could relay the information in a quick fashion, therefore D2GO was worked with.

## **4.2.2 Iterative Design Process**

During the initial weeks of the IQP process our group focused on deciding what program we were going to use to develop our app. The initial race was between SLAM and Detectron2 as mentioned before. To decide which program, we would move forward with we had to decide which function of the app we wanted to work with. The option that was chosen with the help of our Russian counterparts was object detection while they would focus on mapping and directions. Once it was decided that we would focus on object detection we moved forward with Detectron2.

When we first started focusing on Detectron2 we ran into numerous problems. The first of these problems was getting the program to run on our computers. After watching many YouTube videos and posting on forums, we found out that setting up the environment using a package manager such as Anaconda is very helpful. While the set up for anaconda is very messy, it allowed us to successfully have the Detectron2 program run on our computer, and we were able access the program without any trouble. This allowed our group to implement code into the already existing program, to limit the number of objects the app can detect that will not cause harm to the user such as a toothbrush or teddy bear. Once the app successfully detected objects and people through the integrated cameras on our laptop, we moved towards putting the program to a mobile android device. An android device was chosen for a couple reasons, the first reason was we can code in Java which is the preferred programming language of our group. The second and most important reason was to program on iOS we would need a developer license which is too expensive and out of the scope of this project. But to transfer Detectron2 to a mobile device we had to switch to a different model that was more optimized and allows for much faster recognition, this model is D2GO.

## **4.2.3 Iteration 1**

### **4.2.3.1 Developmental Iteration 1**

For the first iteration, after successfully getting D2GO to function, our team wanted to implement a vibration when an object or person was detected. This decision was made based on the results obtained from the interviews conducted in Objective 1, as vibration was the most requested feature. After spending a great amount of time with the D2GO code, we were finally able to make the app vibrate when an object was placed in front of the smartphone camera. This feature was tested repeatedly until the app would vibrate when any object within the directory (91 different object classes that d2go provides) was placed in front of it. While testing, we observed that the smartphone would continue to vibrate for an extended period of time after there were no longer objects being detected on the screen. This was a slight problem for our team, however we eventually found this was due to a queue issue, as the device would update and requeue the object that had already been detected. While this initially seemed like a problem, the repetitive warnings were determined to be a non-issue as a combination of human response time

and cellphone vibration lag time caused the repeating vibrations to have no impact on the user. After concluding this phase of testing, the object detection software focused on detecting people, the application would now alert the user to people but not to most other objects. This change was made due to the results from the interviews which had stated that crowded spaces were one of our biggest concerns in terms of navigation.

#### **4.2.3.2 Developmental Iteration 2**

After deciding to solely detect people while consequently implementing the vibration feature, we began to work on the second iteration. This iteration was intended to have the smartphone vibrate when only a person was present in front of the user. This design was tested by having a groupmate first walk towards the user in a straight line, followed by a sporadic zig-zag pattern in and out of the camera frame. Both tests resulted in an overall accurate performance of the vibration based on the location of the tester in the frame. However, a familiar issue reoccurred. Like the first iteration, we observed that the smartphone would continue to vibrate for an extended period after there were no longer objects being detected on the screen. This time around, it was considered a problem because it was not accurate enough when sporadic movements were made. To solve the issue, our team coded in a new line that did not write another call if there was currently one in progress unless the object was detected again after the call ended. This solved the issue and allowed the sporadic movement test to be more accurate. While detecting a person in front of the camera is helpful to the user, it could be further enhanced with the knowledge of how close that person is to the user.

#### **4.2.3.3 Developmental Iteration 3**

During our third iteration, our team added a change in vibration speed based on how close or far the user was from the person in front of them. To test this iteration's design, we had a group member stand far away from the app user and walk towards them, with the user feeling the level of vibration increase as they get closer. From this test, we observed that the vibration increased noticeably, giving a very good indication of how close the user is to the person. This was a large milestone for our group to hit as we gave the feeling of depth-perception to the app. This was an area that we previously believed was going to be a large struggle. While working with solely vibration, we were very concerned about how the user would react without voice commands similar to our testing in objective 1. A combination of voice commands and vibration was the number one recommendation for our app to include.

#### **4.2.3.4 Developmental Iteration 4**

For this iteration we wanted to focus on the other major feature many interviewed individuals wanted to be added from Objective 1. This caused our group to investigate adding a voice function that would go off when a person walked in front of the camera and user. The message that was ultimately picked was "Stop, person in front," we chose this message as we wanted a quick yet descriptive message to play. After the message was successfully added, we needed to test the message to ensure that it would play if a person walked in the frame and if it played in a timely manner. The exact same test as iteration 2 and 3 was performed to test both the criteria mentioned above. While testing the same issue as the vibration came back, this time it was much easier to solve as we just moved the code implemented from the vibration command to the voice command. Besides the queue issue, our group was very surprised with the detection

speed of the program, we surmised it came down to the app refreshing every half second. After finishing this iteration our group decided it was time to move onto our first obstacle course test. This was the entry point into Objective 3 or the iterative design process, as we had completed the initial building process thanks to Objective 1 and 2.

#### **4.2.4 Iteration 1 Testing**

This test was the first test of the app after the first 4 iterations we made. This test was conducted by having 3 group members acting as obstacles, one group member using the app while blindfolded and the final member recording the process. To start the test, the blindfolded member would count down from 5, during the countdown process the 3 members acting as obstacles would move around the course standing in random places, stationary, making as little sound as possible. We then had the blindfolded group member try to cross the room without hitting the other group members. Then after a few tests, we added a walking cane to the test to simulate the usage of our app with another navigational device. After reviewing the results from our test which we recorded, we were impressed with how far we have come, but not satisfied. This leads us back into the first part of objective 3, the building stage. From our test we noted a few issues, the first issue was being overwhelmed with the message of a person being in front of us, but not knowing the direction. This caused us to stop in our tracks, scan back and forth trying to pinpoint where each obstacle is. Another issue we ran into was we would constantly run off course as no person was detected in that area. This was caused by our model being trained to detect people solely, as we believe this is the largest issue inside of an unknown crowded place. But this issue was addressed with the addition of the walking cane which helped greatly to reduce collisions and increase speed of navigation. Another issue would be the angle of the camera, as if it was angled downward the detection of the person would be late or nonexistent. Finally, we ran into a problem where the application would turn off as the screen would not be interacted with during the test, while this is a simple issue that can be fixed within the settings of the phone, it is one that will need to be instructed to the user.

#### **Figure 13**

*First Navigational Test*



## 4.2.5 Iteration 2

### 4.2.5.1 Developmental Iteration 5

Based off the observations from the first test, our team decided to alter the code so that the danger message would only go off if a person filled a specific amount of the screen. That amount of the screen started off at  $1/10^{\text{th}}$  but decreased to  $1/6^{\text{th}}$  after testing finished. To test this iteration, we had a group member stand across the room and walk towards the app user, observing how close we would have to be to trigger the message. When testing, we observed that having  $1/10^{\text{th}}$  of the screen as the trigger point was way too large. We found that a distance of 6ft from the person would trigger the message, which we found to be too early. Once we lowered the distance to  $1/6^{\text{th}}$  of the screen, we noticed that the distance that would trigger the message was much more reasonable for an indoor setting. This iteration was one of the first that challenged us to think of practicality when building our device as we needed to have a reasonable distance between the user and a person to trigger the message. If the distance were too small, two people would run into each other and if it were too large, the message would be triggered constantly and be of almost no use.

### 4.2.5.2 Developmental Iteration 6

To help solve the reasonability issue found in our last iteration, our team created a virtual box located at the exact center of the screen. This virtual box represented the person within the app, so that in some of the edge cases from iteration 5, a danger alert would not be triggered. An edge case refers to a situation where a person fills  $1/6$  of the screen, however, the person is on the side where they are not in the way of the user and therefore not in danger. When testing, we had two people standing on either side of the camera and left one safe passage between them. The tester was blindfolded and tried to find their way through the people. This iteration was used as a steppingstone to build into future iterations. This step was placed within this iteration to help



the app track where the user would be standing behind the camera. As this box was placed within the code and centered at the bottom of the screen, it took up about 1/4<sup>th</sup> of the screen. The area within the box was half of the screen width multiplied by half of the screen height pixel square. This was deemed as the area that needed to be focused on the most as this was the objects/person the user would walk into the most. The areas outside of the box were considered safe, but still monitored in case they moved or shifted into the virtual box or the person. With the help of the virtual box, the detection became more accurate and reliable. Our team could discern both how far the obstacles were, but also whether they were a concern that laid in the way of the user.

#### **4.2.5.3 Developmental Iteration 7**

Once we created the virtual box, we wanted to specify the voice command even more. We did this by splitting the box into three different sections: left (170 pixels wide), middle (200 pixels wide) and right (170 pixels wide). This would not only indicate the front of the user but also the right and left of the user, allowing us to warn the user of what side the person was located on. To test this addition, we had group members stand in random locations in front of the app, to see if it could correctly locate whether the person was on the right or left. From the tests we were able to navigate around and between objects much more efficiently as we had a more detailed sense of their location and distance. This addition combined with the current device used by a BVI individual will allow for a more streamlined navigation of a crowded indoor area.

#### **4.2.6 Summary of Findings**

In the course of research and development, we discovered many of the predominant issues in the field of blind navigation, and the challenges developers face in solving these issues. We found that a major problem in blind navigation is the detection of doorways, stairways, and objects situated off the ground. We found detecting these objects with computer vision to be difficult. Further, we found that with a fast enough inference time, moving objects can be reliably tracked and continually evaluated for danger. We found that objects only needed to trigger alerts if they were within a certain distance of the user, and if they were directly in front of the user. Faraway objects and objects in frame, but to the left or right of the user's path, were found not to be significant hazards to the users. We found that providing the user with directional guidance as to the relative location of a hazard, as well as the relative distance, aided navigation.

#### **4.2.6 Appearance**

For the appearance, there has only been a couple changes that significantly affected the look of the device and most of them came as a result of the iterations mentioned above. The first major visual iteration was the addition of the visual box, this would leave a yellow box on the screen for the majority of the time to represent the individual. The next change that was made was changing the background color of interface to be black to avoid confusion with the picture and the interface which can be found in Figure 15. The final major iteration was the changing of the name, originally, we had no set name, and it was just the navigator device. That changed when the app name AARGUS was chosen, this name has a dual meaning as it is not only the name of the titan who was the watcher for the Greek gods, known as the watcher. But it also spells out Applied Augmented Reality Guiding Users Safely, which is what we hope to achieve with our final product.

**Figure 14**  
*AARGUS Logo*



**Figure 15**  
*Current front page of AARGUS*



## Chapter 5: Discussion

While faced with many challenges, such as relocation to a different country for our project as well as a worldwide pandemic, our group prevailed and developed a smartphone application that can detect objects and navigate a user around them. With the use of our app, our team is confident that we can help the blind and visually impaired communities within Germany, Russia, and the rest of the world by simplifying navigation and preventing collisions with stationary and moving objects within indoor environments. While our product cannot yet act as a stand-alone navigational device, it works brilliantly as a complement to the devices already used by the BVI community.

Our team created an app that scans live video feed from a smartphone and subsequently relays information about obstacles in front of the user using voice cues and vibration. The app was developed with the intention of providing BVI individuals with a tool that makes indoor navigation in unfamiliar spaces easier. The application was envisioned to allow a user to navigate through an unfamiliar indoor space by detecting doorways and stairways in conjunction with advanced mapping software, such as SLAM. The application was also planned to allow for detection of both moving and stationary obstacles within a user's path. While developing and testing our application, our team found the integration of SLAM and Detectron2 to be too difficult to achieve within the time constraints of the project. Shifting our primary focus to Detectron2 and live obstacle detection, we were able to develop an application for Android devices, capable of detecting and alerting users of obstacles in the user's path in real time. At the current level of development, the app allows users to navigate around people using directional voice cues and varying levels of vibration, indicating how close the obstacle is to the user. The time constraints of the project have led to several features we were unable to implement.

A major opportunity for improvement exists within the navigational guidance segment of the app. Ideally, in addition to the person and object detection feature, the app would take input from the user for traveling purposes. The application would decipher where the individual is located and based on the user input, would direct them to their desired destination. This direction to a desired destination was a task delegated to our Russian counterparts. While the Russian team made progress towards this implementation, their work could not be added to the final code. This was primarily due to time limitations, and the technical challenges in the implementation of SLAM or some other localization and mapping software for the Java programming language, which was the primary language used for app development. Several promising implementations have been analyzed, but the technical challenge required to integrate these implementations into our app exceeded the time limitations of the project.

Beyond the integration of SLAM into the application, a few improvements could still be made to the object detection and warning system. Chief among these is the latency of the alerts given to the user. The current model being used is capable of reading objects in the scene several times per second, but the high latency of text-to-speech audio warnings greatly increases the latency between detection of an object and communication to the user. When combined with the latency of the user processing the input, we run into a significant problem with the efficiency of

navigation. While experience in using the app makes this process faster, further improvements could be made to the warning system.

When giving users information about what lies ahead of them, one of the crucial issues that could occur is an information overload. This is caused by a large amount of information being relayed to the user by the app, based on their surroundings. While we want to relay as much information as possible to keep the user safe, we cannot give information to the point where the user cannot digest and understand the directions. This could occur if multiple people and objects are detected around the person, triggering the voice command to not stop altering. To solve this issue, we lowered the number of items stored detected by the app as low as possible. This caused fewer objects to be detected, in turn reducing the flow of information but kept the alerts for important objects to avoid. This would also be solved by creating a custom model to only detect the important objects within a scene. For example, the current model is particularly good at detecting elephants within a user's path but testing of our app has revealed a surprising lack of elephant-based problems in indoor blind navigation.

To help decrease information overload, the smartphone application is focused on the detection and navigation around people, rather than other objects. Certain objects, such as chairs and tables, are currently detectable by the model, and can be issued custom warnings. However, the model used for object detection currently struggles heavily with identifying stairs, doorways, walls, and hanging objects. Training of a custom model with a dataset more appropriately selected for our purposes would increase the efficiency of detection of these objects, as well as potentially lowering the inference time for the model.

One of the original objectives of the project was to build the app for both Android and IOS. Android proved to be the easiest operating system to work with as the development studio for Android is more user friendly and allows for more product testing without needing a developer's license. Android also develops in Java, a language that the group was familiar with while IOS used Swift, something that none of us knew. Xcode, the IOS development IDE was also significantly harder for the group to use than Android studio. With primary development being built of Android Studio it left very little time to do any significant development on IOS. However, developing on IOS is still worth doing as IOS has a few unique merits over Android, the largest being its uniformity. Android is an operating system built for dozens of different phones by dozens of different manufacturers. IOS is just on iPhone. This gives IOS development the unique opportunity to know exactly what hardware the software will be running on. When developing with Android assumptions had to be made, while most factors assumed would be industry standards, such as screen width, it is just simply not the same as the IOS development.

In all cases, we found that the use of a cane in conjunction with our app significantly reduced collisions with objects and increased the speed of navigation. In the future, this project can be built upon with the addition of the recommendations that can be found in the next section.

## **Chapter 6: Conclusion and Recommendations**

Through preliminary research and interviews conducted with blind and visually impaired individuals, our team has found evidence to suggest that there exists a need for an improved

comprehensive navigational aid. To address this need, our team designed, developed, and tested a navigational smartphone application for BVI individuals using an iterative design process. Throughout this process, the team made observations and analyses for each iteration. Every iteration improved upon the previous one and added to the value of this project. Additionally, our team has developed a set of recommendations which are intended to help guide both end users and future project developers with our application.

After research, interviews, creation, and testing of our app, we expected there to be many areas that could be improved upon. Below is a list of recommendations for both users and future developers of the project to look at and consider.

User:

- We recommend that the user changes the settings of their smartphone to disable the screen auto-lock feature in order to ensure the screen is turned on and the application is functional for the entirety of use.
- We recommend that the user holds their phone at chest level, keeping it as still and vertical as possible while trying to avoid angling the camera downwards.
- We recommend the user never uses this application as their primary method of navigation and only as an addition to their current navigational device for safety reasons.

Developer:

Based on time and the scale of the project, our team was not able to accomplish all aspects of the project that we wanted to.

The following recommendations are based on our experiences researching blind navigation, our interviews with BVI individuals, and our experiences developing the application.

- We recommend future application developers train their own model using **Roboflow** to detect objects that pose a hazard to the user such as furniture with sharp edges, door frames, stairs and differences in flooring height.
- We recommend future application developers utilize the SLAM technology in future iterations of the app to map routes that can help the user navigate to specific rooms within a building.
  - Many of the BVI individuals who were interviewed wanted a program that can map out a route to find the bathroom or another space within a building. We recommend looking at this for future iterations of our design, using a program called SLAM.
- When walking it is important to have a detailed view of your entire surroundings to ensure you get to your desired location without harm. Our group recommends implementing a way to have a 180° view to simulate peripheral vision.
  - One way this could be done is by incorporating a set of cameras to allow for this peripheral vision.
- Knowing where an object is located and generally how close it is to you is helpful but knowing the exact location of a dangerous object is leagues better. Our group

recommends implementing a voice prompt that tells you exactly how far away the object is in conjunction with our vibration feature to ensure the safest navigation of indoor spaces.

- When using our app, it is important to hold the device steady, vertically and close to the chest to ensure that the app can accurately scan the users' surroundings. Our group recommends finding a way to ensure the user holds the phone in the correct fashion to maximize effectiveness.
  - One recommendation that we have thought of is developing a harness to hold the phone steady and at chest level. While this is a solution to the issue, it creates an extra cost that could be unaffordable for users. This creates the need for another more cost-effective solution to be developed.
- We recommend future application developers to expand the app to an iOS platform in order to help reach a larger population of BVI individuals who do not use an android smartphone. Building on IOS would also increase the market share that the app is able to reach and would allow for greater uniformity in the development. In this development we recommend this source to assist with the IOS development.
  - Found within the link below is a demo that incorporates D2GO into iOS, which will be a helpful tool for building upon our recommendation and app.
  - <https://github.com/pytorch/ios-demo-app/tree/master/D2Go>

Based on the previously outlined recommendations, our team is confident that a future team can pick up where we left off and further develop this application. Our team made great progress in developing a working application for BVI individuals to use as a navigational device. With the future integration of both our object detection and warning systems, along with simultaneous location and mapping software, a cohesive navigational device can be developed for BVI users. Our intent was to make indoor navigation easier, as our application is meant to complement a user's current navigational device, such as a white walking cane or guide dog.

## References

- Allen, M., Boyea, S., Carling, A., & Miholovich, C. (2021). *Navigator for the Blind*. Worcester Polytechnic Institute. <https://digital.wpi.edu/pdfviewer/pn89d970p>
- Adarsh, M. (2021, October 19). *Bluetooth Low Energy (BLE): A Complete Guide to BLE Beacons*. Beaconstac. Retrieved February 20, 2022, from <https://blog.beaconstac.com/2018/08/ble-made-simple-a-complete-guide-to-ble-bluetooth-beacons/>
- Ackland, P., Resnikoff, S., & Bourne, R. (2017). World blindness and visual impairment: despite many successes, the problem is growing. *Community eye health*, 30(100), 71–73.
- Babu, D. V., Karthikeyan, R., Bharadwaj, S. M., & Sai, B. A. (2021). *Wearable technology for visually challenged people commutation using ultrasonic mapping*. 2040. Scopus. <https://doi.org/10.1088/1742-6596/2040/1/012041>
- Boutera, Y. (2021). Design and development of a wearable assistive device integrating a fuzzy decision support system for blind and visually impaired people. *Micromachines*, 12(9). Scopus. <https://doi.org/10.3390/mi12091082>
- Bikbov, Kazakbaeva, G. M., Gilmanshin, T. R., Zainullin, R. M., Nuriev, I. F., Zaynetdinov, A. F., Israfilova, G. Z., Panda-Jonas, S., Arslangareeva, I. I., Rakhimova, E. M., Rusakova, I. A., & Jonas, J. B. (2020). Prevalence and associated factors of cataract and cataract-related blindness in the Russian Ural Eye and Medical Study. *Scientific Reports*, 10(1), 18157–18157. <https://doi.org/10.1038/s41598-020-75313-0>
- Ciurkot, A. (2022, January 18). *What is beacon and how it work? BLE Bluetooth Beacons Technology Guide*. Kontakt.Io. Retrieved February 20, 2022, from <https://kontakt.io/what-is-a-beacon/>
- Chen, Z., Liu, X., Kojima, M., Huang, Q., & Arai, T. (2021). A Wearable Navigation Device for Visually Impaired People Based on the Real-Time Semantic Visual SLAM System. *Sensors (Basel, Switzerland)*, 21(4), 1536. <https://doi-org.ezpv7-web-p-u01.wpi.edu/10.3390/s21041536>
- Documentary: Russia's Village Of The Blind*. RadioFreeEurope/RadioLiberty. (2022). Retrieved 23 March 2022, from <https://www.rferl.org/a/russia-blind-city-rusinovo/28425942.html>.
- e.V., D. (2022). *Travel in Germany with visual impairment - Germany Travel*. Germany.travel. Retrieved 23 March 2022, from <https://www.germany.travel/en/accessible-germany/disability-friendly-travel-for/visually-impaired.html>.
- Facebook - Meld je aan of registreer je*. (n.d.-b). Facebook. Retrieved February 13, 2022, from <https://www.facebook.com/unsupportedbrowser>
- GeoSLAM. (2022, January 21). *What is SLAM? Simultaneous Localization and Mapping*. Retrieved February 13, 2022, from <https://geoslam.com/what-is-slam/>

- Hardach, S. (2021). The school that created a city for the blind. BBCpage. <https://www.bbc.com/future/article/20210916-the-school-that-change-a-city-into-a-place-for-the-blind>
- Hersh, M. (2013). Deafblind People, Communication, Independence, and Isolation. *Journal of Deaf Studies and Deaf Education*, 18(4), 446–463. <http://www.jstor.org/stable/42659890>
- Vineeth, I., Sharan, Y. H. V. S., Karthik, Y., & Priya, B. K. (2021). Smart Cane for Visually Impaired Person. 2021 International Conference on Intelligent Technologies, CONIT 2021. Scopus. <https://doi.org/10.1109/CONIT51480.2021.9498563>
- Khan, I., Khusro, S., & Ullah, I. (2018). Technology-assisted white cane: Evaluation and future directions. *PeerJ*, <http://dx.doi.org/10.7717/peerj.6058>
- Lilly, E. B., & Van Fleet, C. (2000). Measuring the Accessibility of Public Library Home Pages.
- Midi, N. S., Idris, N. A. M., Yusoff, S. H., & Ripah, N. A. M. (2021). *Evaluation of Energy Harvesting for Smart Cane Application*. 110–114. Scopus. <https://doi.org/10.1109/ICCCE50029.2021.9467150>
- Milallos, R., Tibdewal, V., Wang, Y., Ogueh Udegbe, A., & Oh, T. (2021). “would the smart cane benefit me?”: Perceptions of the Visually Impaired towards Smart Canes. ASSETS 2021 - 23rd International ACM SIGACCESS Conference on Computers and Accessibility. Scopus. <https://doi.org/10.1145/3441852.3476524>
- OpenCV*. (n.d.). OpenCV. Retrieved March 3, 2022, from <https://opencv.org/>
- Oreshkina, M. (2009). Education of children with disabilities in Russia: On the way to integration and inclusion. *International Journal of Special Education*, 24(3), 11.
- Pradeep Kumar, M., Inchara, M. K., Lekhashree, S., Likhith, N. C., & Pavan, U. (2021). *Real Time Assistive Shoe for Visually Impaired People*. 2021 6th International Conference for Convergence in Technology, I2CT 2021. Scopus. <https://doi.org/10.1109/I2CT51068.2021.9417928>
- Presti, G., Ahmetovic, D., Ducci, M., Bernareggi, C., Ludovico, L. A., Baratè, A., . . . Mascetti, S. (2021). Iterative design of sonification techniques to support people with visual impairments in obstacle avoidance. *ACM Transactions on Accessible Computing*, 14(4) doi:10.1145/3470649
- Pronchev, G. B., Goncharova, I. V., Elishev, S. O., & Khodakova, N. P. (2017). Accessibility of virtual social environments for the visually impaired as a means for leveling out social inequality in Russia. *Man in India*, 97(16), 107-116. Retrieved from [www.scopus.com](http://www.scopus.com)
- Schelin, D., Gomes, P. V., & Quandt, V. I. (2021). A smart city for the blind. *Developing and Monitoring Smart Environments for Intelligent Cities*, 112-129. <https://doi.org/10.4018/978-1-7998-5062-5.ch005>



- Stewart, Conor. Vision Impairments in Russia 2020. *Statista*, 17 Feb. 2021, <https://www.statista.com/statistics/1200899/vision-impairments-by-severity-in-russia/#:~:text=In%202020%2C%20there%20were%20almost,affected%20with%20mild%20sight%20loss.>
- Tinder, R. F. (2007). *Relativistic Flight Mechanics and Space Travel: A Primer for Students, Engineers and Scientists*. Morgan & Claypool Publishers.
- The Situation of the Blind and Visually Impaired in Germany — An Overview | American Council of the Blind*. Acb.org. (2022). Retrieved 28 March 2022, from <https://www.acb.org/situation-blind-and-visually-impaired-germany>.
- Vision impairment and blindness*. (n.d.). Retrieved February 12, 2022, from <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- Wu, Y. (2018). Sight-impaired people want to use technology but are excluded by cost and accessibility - new research. *The Conversation*.
- Xie, Babu, R., Lee, T. H., Castillo, M. D., You, S., & Hanlon, A. M. (2020). Enhancing usability of digital libraries: Designing help features to support blind and visually impaired users. *Information Processing & Management*, 57(3), 102110–. <https://doi.org/10.1016/j.ipm.2019.102110>
- Reference & User Services Quarterly*, 40(2), 156–165. <http://www.jstor.org/stable/41241320>

## **Appendix A - Sample Questions for Interviews with Russian BVI Individuals (in English)**

### **1. General Understanding of Visual Impairment**

- 1.1 What type of Visual impairment do you have?
- 1.2 Have you always been Visually Impaired?
- 1.3 If yes to question 1.2... Move to next set
- 1.4 If No to question 1.2, how long have you been Visually impaired?

### **2. Navigational Devices**

- 1.1 General Understanding
  - 1.1.1 Do you use a white walking cane as a navigational device?
  - 1.1.2 Do you use a seeing-eye dog?
  - 1.1.3 If answered both answered no, what navigational device do you use?
  - 1.1.4 Do you use more than one navigational device?
  - 1.1.5 What did the learning process look like for you when you first started using your navigational device(s)?
- 1.2 Use of a White Walking Cane
  - 1.2.1 Have you used any other navigational devices in the past?
  - 1.2.2 If answered no, move onto question 1.2.4.
  - 1.2.3 If answered yes, what device was it?
  - 1.2.4 What features of this device do you find beneficial?
  - 1.2.5 What features of this device do you think can be improved?
  - 1.2.6 How much did this navigational device cost?
    - 1.2.6.1 Was this an economic burden for you?
  - 1.2.7 What difficulty, if any, do you experience when trying to detect objects that are located above you, such as signs or hanging objects?
- 1.3 Use of a Seeing Eye-Dog
  - 1.3.1 Have you used any other navigational devices in the past?
  - 1.3.2 If answered no, move onto the next question
  - 1.3.3 If answered yes, what device was it?
  - 1.3.4 What features of this device do you find beneficial?
  - 1.3.5 What features of this device do you think can be improved?
  - 1.3.6 How much did this navigational device cost, and was this an economic burden for you?
- 1.4 Use of a Navigational Device Other than a White Walking Cane or Seeing Eye-Dog
  - 1.4.1 What features of this device do you find beneficial?
  - 1.4.2 What features of this device do you think can be improved?
  - 1.4.3 How much did this navigational device cost, and was this an economic burden for you?

### **3. Navigating Indoor Spaces**

- 3.1 General Understanding
  - 3.1.1 How would you describe the difficulty of navigating indoor spaces versus outdoor spaces?
  - 3.1.2 What types of indoor spaces do you visit on a regular basis?
  - 3.1.3 What features of a navigational device best help you navigate indoor spaces?

3.1.4 What objects in an indoor space do you use your navigational device to avoid?

3.1.5 How effective is your current navigational device with detecting objects in your path?

### 3.2 Crowded Indoor Spaces

3.2.1 Do you find yourself walking through crowded spaces?

3.2.2 Is it difficult for you?

3.2.3 If yes, why is this difficult for you?

3.2.4 If not, end.

## **Appendix B - Sample Questions for Interviews with Russian BVI Individuals (in Russian)**

### 1. Общее понимание нарушений зрения

1.1 Какой тип нарушения зрения у вас есть?

1.2 Всегда ли у вас были проблемы со зрением?

1.3 Если да вопрос 1.2. , перейти к пункту 2

1.4 Если нет на вопрос 1.2, как давно у вас проблемы со зрением?

### 2. Навигационные устройства

#### 2.1 Общее понимание

2.1.1 Используете ли вы белую трость в качестве навигационного устройства?

2.1.2 Используете ли вы собаку-поводыря?

2.1.3 Если оба ответа были отрицательными, какое навигационное устройство вы используете?

2.1.4 Используете ли вы более одного навигационного устройства?

2.1.5 Как выглядел для вас процесс обучения, когда вы впервые начали использовать свое навигационное устройство (устройства)?

#### 2.2 Использование белой трости для ходьбы

2.2.1 Использовали ли вы в прошлом какие-либо другие навигационные устройства?

2.2.2 Если ответ отрицательный, переходите к вопросу 2.2.4.

2.2.3 Если ответили утвердительно, то какое это было устройство?

2.2.4 Какие функции этого устройства вы считаете полезными?

2.2.5 Какие функции этого устройства, по вашему мнению, можно улучшить?

2.2.6 Сколько стоило это навигационное устройство? Было ли это для вас экономическим бременем?

2.2.7 С какими трудностями, если таковые имеются, вы сталкиваетесь при попытке обнаружить объекты, расположенные над вами, такие как знаки или висящие предметы?

## 2.3 Использование собаки-поводыря

2.3.1 Использовали ли вы в прошлом какие-либо другие навигационные устройства?

2.3.2 Если ответ отрицательный, переходите к следующему вопросу

2.3.3 Если ответили утвердительно, то какое это было устройство?

2.3.4 Какие функции этого устройства вы считаете полезными?

2.3.5 Какие функции этого устройства, по вашему мнению, можно улучшить?

2.3.6 Сколько стоило это навигационное устройство, и было ли это для вас экономическим бременем?

## 2.4 Использование навигационного устройства, отличного от Белой трости для ходьбы или Собаки-поводыря (написать, что за устройство)

2.4.1 Какие функции этого устройства вы считаете полезными?

2.4.2 Какие функции этого устройства, по вашему мнению, можно улучшить?

2.4.3 Сколько стоило это навигационное устройство и было ли это для вас экономическим бременем?

## 3. Навигация По Внутренним Пространствам

### 3.1 Общее понимание

3.1.1 Какие сложности навигации в помещениях по сравнению с открытыми пространствами?

3.1.2 Какие типы внутренних помещений вы посещаете на регулярной основе? (Институт, офис, больница, тц и тд)

3.1.3 Какие функции навигационного устройства лучше всего помогают вам ориентироваться в помещениях?

3.1.4 Каких объектов в помещении вы избегаете с помощью своего навигационного устройства?

3.1.5 Насколько эффективно ваше текущее навигационное устройство обнаруживает объекты на вашем пути?

### 3.2 Переполненные Внутренние Помещения

3.2.1 Обнаруживаете ли вы, что идете по людным местам?

3.2.2 Трудно ли вам это?

3.2.3 Если да, то почему это трудно для вас?

### 4 Улучшение системы навигации приложения

4.1. Хотели бы, чтобы навигационное устройство (приложение) разделяло стационарные и движущиеся объекты?

4.2. Нужно ли обозначать стационарные и движущиеся объекты по-разному?

4.3. Выберите наиболее предпочтительные обозначения для Вас:

•Голосовое сопровождение

•Вибрация

•Громкий сигнал

4.4. Какая функция приложения для вас была бы наиболее полезна?

• Нахождение конкретного объекта (Пример: найти на полках магазина пачку сахара)

• Нахождение конкретного адреса (Пример: найти аудиторию в корпусе института)

## Appendix C – Sample Consent Form for Interviews

### Informed Consent Agreement for Participation in a Research Study

#### Investigator:

Primary Investigator: Svetlana Nikitina

Email: [Svetlana@wpi.edu](mailto:Svetlana@wpi.edu)

Student Investigator: Sam Mather, Kelsey Leach, John Winship, Alex Demires, Hao Chen

Email: [gr-navigator-d22@wpi.edu](mailto:gr-navigator-d22@wpi.edu)

**Title of Research Study:** Developing an Indoor Navigational Aid for Russian BVI Individuals Using Computer Vision and Machine Learning

**Sponsor:** Financial University of Russia

#### Introduction:

You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks, or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation. Participants in this study must be at least 18 years old.

**Purpose of the study:** The purpose of this study is to gain insight into the benefits and drawbacks of typical forms of navigational devices used by BVI individuals. These insights will be used to develop a prototype for navigational aid

The purpose of this interview is to understand what you feel are the best and worst features of the navigational device(s) that you currently use. The questions will focus on duration of your visual impairment, the navigational device that you use, and the indoor spaces that you walk around (whether they are crowded or not). From this information, consideration of applications to be implemented or removed from navigational devices will be made.

**Procedures to be followed:** You will be asked questions by the interviewers, please respond to each question to the best of your ability. If you do not wish to answer any question, please tell the interviewer and that question(s) will be skipped. If, for any reason, you wish to no longer participate in the interview, please tell any of the investigators and the interview will be concluded immediately. If after the conclusion of the interview, you wish to have any responses removed from the record, inform the interviewer and the statement will be removed.

**Risks to study participants:** There are no anticipated risks or sources of discomfort from this interview other than those that can be encountered when using a device online. The interview questions are designed to collect information on the navigational device used, needing little to no personal information. Participants have the right to stop the interview at any time if they feel too much personal information is required and if they are uncomfortable.

**Benefits to research participants and others:** There are no benefits associated with participation of participants in the interview.

**Record keeping and confidentiality:** Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or its designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

**Compensation or treatment in the event of injury:** As this study presents little to no risk of injury or harm to participants, no compensation or medical treatment will be provided. If injury

does occur, contact your emergency medical services or your nearest medical official. You do not give up any of your legal rights by signing this statement.

**For more information about this research or about the rights of research participants, or in case of research-related injury, contact:**

See contact information of the primary and student investigators above

Worcester Polytechnic Institutes IRB chair: Professor Kent Rissmiller

Tel. 508-831-5019      Email: [kjr@wpi.edu](mailto:kjr@wpi.edu)

University Compliance Officer: Michael J. Curley

Tel. 508-831-6919      Email: [mjcurley@wpi.edu](mailto:mjcurley@wpi.edu)

**Your participation in this research is voluntary.** Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

**By signing below,** you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

\_\_\_\_\_

Study Participant Signature

**Date:** \_\_\_\_\_

\_\_\_\_\_

Study Participant Name (Please print)

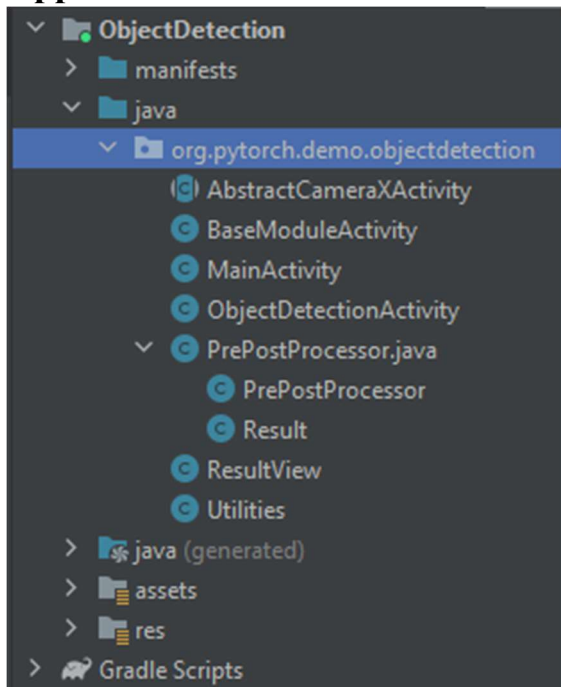
**Date:** \_\_\_\_\_

\_\_\_\_\_

Signature of Person who explained this study



## Appendix D – AARGUS Source Code



The following is all the java source code for our application.

Our public repository on GitHub: <https://github.com/jwwinship/navigator-test-app>

```
/**
 * abstract class AbstractCameraXActivity
 * handles all activities (Creation, request, setup and other API calls) of the camera
 module
 * extends from BaseModuleActivity
 * Friendly Reminder: This class is a standard modular setup, you don't wanna touch this
 unless necessary
 * @param <R>
 */
public abstract class AbstractCameraXActivity<R> extends BaseModuleActivity {
    private static final int REQUEST_CODE_CAMERA_PERMISSION = 200;
    private static final String[] PERMISSIONS = {Manifest.permission.CAMERA};

    private long mLastAnalysisResultTime;

    protected abstract int getContentViewLayoutId();

    protected abstract TextureView getCameraPreviewTextureView();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(getContentViewLayoutId());

        startBackgroundThread();
    }
}
```

```

    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.CAMERA)
        != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(
            this,
            PERMISSIONS,
            REQUEST_CODE_CAMERA_PERMISSION);
    } else {
        setupCameraX();
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
    if (requestCode == REQUEST_CODE_CAMERA_PERMISSION) {
        if (grantResults[0] == PackageManager.PERMISSION_DENIED) {
            Toast.makeText(
                this,
                "You can't use object detection example without granting CAMERA
permission",
                Toast.LENGTH_LONG)
                .show();
            finish();
        } else {
            setupCameraX();
        }
    }
}

private void setupCameraX() {
    final TextureView textureView = getCameraPreviewTextureView();
    final PreviewConfig previewConfig = new PreviewConfig.Builder().build();
    final Preview preview = new Preview(previewConfig);
    preview.setOnPreviewOutputUpdateListener(output ->
textureView.setSurfaceTexture(output.getSurfaceTexture()));

    final ImageAnalysisConfig imageAnalysisConfig =
        new ImageAnalysisConfig.Builder()
            .setTargetResolution(new Size(480, 640))
            .setCallbackHandler(mBackgroundHandler)
            .setImageReaderMode(ImageAnalysis.ImageReaderMode.ACQUIRE_LATEST_IMAGE)
            .build();
    final ImageAnalysis imageAnalysis = new ImageAnalysis(imageAnalysisConfig);
    imageAnalysis.setAnalyzer((image, rotationDegrees) -> {
        if (SystemClock.elapsedRealtime() - mLastAnalysisResultTime < 500) {
            return;
        }

        final R result = analyzeImage(image, rotationDegrees);
        if (result != null) {
            mLastAnalysisResultTime = SystemClock.elapsedRealtime();
            runOnUiThread(() -> applyToUiAnalyzeImageResult(result));
        }
    });
}
}

```

```

        CameraX.bindToLifecycle(this, preview, imageAnalysis);
    }

    @WorkerThread
    @Nullable
    protected abstract R analyzeImage(ImageProxy image, int rotationDegrees);

    @UiThread
    protected abstract void applyToUiAnalyzeImageResult(R result);
}

-----
/**
 * class BaseModuleActivity
 * handles various activities from the base module
 * extends from AppCompatActivity
 * Friendly Reminder: This class is a standard modular setup, you don't wanna touch this
 unless necessary
 */
public class BaseModuleActivity extends AppCompatActivity {
    protected HandlerThread mBackgroundThread;
    protected Handler mBackgroundHandler;
    protected Handler mUIHandler;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mUIHandler = new Handler(getMainLooper());
    }

    @Override
    protected void onStart() {
        super.onStart();
        startBackgroundThread();
    }

    protected void startBackgroundThread() {
        mBackgroundThread = new HandlerThread("ModuleActivity");
        mBackgroundThread.start();
        mBackgroundHandler = new Handler(mBackgroundThread.getLooper());
    }

    @Override
    protected void onDestroy() {
        stopBackgroundThread();
        super.onDestroy();
    }

    protected void stopBackgroundThread() {
        mBackgroundThread.quitSafely();
        try {
            mBackgroundThread.join();
            mBackgroundThread = null;
            mBackgroundHandler = null;
        }
    }
}

```

```

        } catch (InterruptedException e) {
            Log.e("Object Detection", "Error on stopping background thread", e);
        }
    }
}

-----
/**
 * class MainActivity
 * handles all main application activities (create, UI ,and etc.)
 */
public class MainActivity extends AppCompatActivity implements Runnable {

    static {
        if (!NativeLoader.isInitialized()) {
            NativeLoader.init(new SystemDelegate());
        }
        // loading pytorch and torchvision
        NativeLoader.loadLibrary("pytorch_jni");
        NativeLoader.loadLibrary("torchvision_ops");
    }

    private int mImageIndex = 0;
    private String[] mTestImages = {"test1.png", "test2.jpg", "test3.png"};

    private ImageView mImageView;
    private ResultView mResultView;
    private Button mButtonDetect;
    private ProgressBar mProgressBar;
    private Bitmap mBitmap = null;
    private Module mModule = null;
    private float mImgScaleX, mImgScaleY, mIvScaleX, mIvScaleY, mStartX, mStartY;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (ContextCompat.checkSelfPermission(this,
            Manifest.permission.READ_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
                String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 1);
        }

        if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) !=
            PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
                String[]{Manifest.permission.CAMERA}, 1);
        }

        setContentView(R.layout.activity_main);

        try {

```



```

        else if (options[item].equals("Choose from Photos")) {
            Intent pickPhoto = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.INTERNAL_CONTENT_URI);
            startActivityForResult(pickPhoto , 1);
        }
        else if (options[item].equals("Cancel")) {
            dialog.dismiss();
        }
    }
});
builder.show();
});

// handles "Live" button
final Button buttonLive = findViewById(R.id.LiveButton);
buttonLive.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        final Intent intent = new Intent(MainActivity.this,
ObjectDetectionActivity.class);
        startActivity(intent);
    }
});

mButtonDetect = findViewById(R.id.detectButton);
mProgressBar = (ProgressBar) findViewById(R.id.progressBar);
mButtonDetect.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        mButtonDetect.setEnabled(false);
        mProgressBar.setVisibility(ProgressBar.VISIBLE);
        mButtonDetect.setText(getString(R.string.run_model));

        mImgScaleX = (float)mBitmap.getWidth() / PrePostProcessor.INPUT_WIDTH;
        mImgScaleY = (float)mBitmap.getHeight() / PrePostProcessor.INPUT_HEIGHT;

        mIvScaleX = (mBitmap.getWidth() > mBitmap.getHeight() ?
(float)mImageView.getWidth() / mBitmap.getWidth() : (float)mImageView.getHeight() /
mBitmap.getHeight());
        mIvScaleY = (mBitmap.getHeight() > mBitmap.getWidth() ?
(float)mImageView.getHeight() / mBitmap.getHeight() : (float)mImageView.getWidth() /
mBitmap.getWidth());

        mStartX = (mImageView.getWidth() - mIvScaleX * mBitmap.getWidth())/2;
        mStartY = (mImageView.getHeight() - mIvScaleY * mBitmap.getHeight())/2;

        Thread thread = new Thread(MainActivity.this);
        thread.start();
    }
});

try {
    // try quantized d2go model

```

```

mModule = PyTorchAndroid.LoadModuleFromAsset(getAssets(), "d2go.pt");

BufferedReader br = new BufferedReader(new
InputStreamReader(getAssets().open("classes.txt")));
String line;
List<String> classes = new ArrayList<>();
while ((line = br.readLine()) != null) {
    classes.add(line);
}
PrePostProcessor.mClasses = new String[classes.size()];
classes.toArray(PrePostProcessor.mClasses);
} catch (IOException e) {
    Log.e("Object Detection", "Error reading assets", e);
    finish();
}
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode != RESULT_CANCELED) {
        switch (requestCode) {
            case 0:
                if (resultCode == RESULT_OK && data != null) {
                    mBitmap = (Bitmap) data.getExtras().get("data");
                    Matrix matrix = new Matrix();
                    //matrix.postRotate(90.0f); //TODO: Uncomment to rotate image
                    mBitmap = Bitmap.createBitmap(mBitmap, 0, 0, mBitmap.getWidth(),
mBitmap.getHeight(), matrix, true);
                    mImageView.setImageBitmap(mBitmap);
                }
                break;
            case 1:
                if (resultCode == RESULT_OK && data != null) {
                    Uri selectedImage = data.getData();
                    String[] filePathColumn = {MediaStore.Images.Media.DATA};
                    if (selectedImage != null) {
                        Cursor cursor = getContentResolver().query(selectedImage,
filePathColumn, null, null, null);
                        if (cursor != null) {
                            cursor.moveToFirst();
                            int columnIndex =
cursor.getColumnIndex(filePathColumn[0]);
                            String picturePath = cursor.getString(columnIndex);
                            mBitmap = BitmapFactory.decodeFile(picturePath);
                            Matrix matrix = new Matrix();
                            //matrix.postRotate(90.0f); //TODO: uncomment this line
to rotate image

                            mBitmap = Bitmap.createBitmap(mBitmap, 0, 0,
mBitmap.getWidth(), mBitmap.getHeight(), matrix, true);
                            mImageView.setImageBitmap(mBitmap);
                            cursor.close();
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    break;
}
}
}

@Override
public void run() {
    Bitmap resizedBitmap = Bitmap.createScaledBitmap(mBitmap,
PrePostProcessor.INPUT_WIDTH, PrePostProcessor.INPUT_HEIGHT, true);

    final FloatBuffer floatBuffer = Tensor.allocateFloatBuffer(3 *
resizedBitmap.getWidth() * resizedBitmap.getHeight());
    TensorImageUtils.bitmapToFloatBuffer(resizedBitmap,
0,0,resizedBitmap.getWidth(),resizedBitmap.getHeight(), PrePostProcessor.NO_MEAN_RGB,
PrePostProcessor.NO_STD_RGB, floatBuffer, 0);
    final Tensor inputTensor = Tensor.fromBlob(floatBuffer, new long[] {3,
resizedBitmap.getHeight(), resizedBitmap.getWidth()});

    final long startTime = SystemClock.elapsedRealtime();
    IValue[] outputTuple = mModule.forward(IValue.listFrom(inputTensor)).toTuple();
    final long inferenceTime = SystemClock.elapsedRealtime() - startTime;
    Log.d("D2Go", "inference time (ms): " + inferenceTime);

    Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE); //TODO:
Vibrator initialization.

    final Map<String, IValue> map = outputTuple[1].toList()[0].toDictStringKey();
    float[] boxesData = new float[]{};
    float[] scoresData = new float[]{};
    long[] labelsData = new long[]{};
    if (map.containsKey("boxes")) {
        final Tensor boxesTensor = map.get("boxes").toTensor();
        final Tensor scoresTensor = map.get("scores").toTensor();
        final Tensor labelsTensor = map.get("labels").toTensor();
        boxesData = boxesTensor.getDataAsFloatArray();
        scoresData = scoresTensor.getDataAsFloatArray();
        labelsData = labelsTensor.getDataAsLongArray();

        // set up virtual box
        DisplayMetrics displayMetrics = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
        int height = displayMetrics.heightPixels;
        int width = displayMetrics.widthPixels;
        Utilities helper = new Utilities();
        Rect VB = helper.setupVirtualBox(height,width);

        final int n = scoresData.length;
        float[] outputs = new float[n * PrePostProcessor.OUTPUT_COLUMN];
        int count = 0;
        for (int i = 0; i < n; i++) {

```





```

    }
}

@Override
protected int getContentViewLayoutId() {
    return R.layout.activity_object_detection;
}

@Override
protected TextureView getCameraPreviewTextureView() {
    mResultView = findViewById(R.id.resultView);
    return ((ViewStub) findViewById(R.id.object_detection_texture_view_stub))
        .inflate()
        .findViewById(R.id.object_detection_texture_view);
}

@Override
protected void applyToUiAnalyzeImageResult(AnalysisResult result) {
    Rect VB = getVirtualBox();
    mResultView.setBox(VB);
    mResultView.setResults(result.mResults);
    mResultView.invalidate();
}

private Bitmap imgToBitmap(Image image) {
    Image.Plane[] planes = image.getPlanes();
    ByteBuffer yBuffer = planes[0].getBuffer();
    ByteBuffer uBuffer = planes[1].getBuffer();
    ByteBuffer vBuffer = planes[2].getBuffer();

    int ySize = yBuffer.remaining();
    int uSize = uBuffer.remaining();
    int vSize = vBuffer.remaining();

    byte[] nv21 = new byte[ySize + uSize + vSize];
    yBuffer.get(nv21, 0, ySize);
    vBuffer.get(nv21, ySize, vSize);
    uBuffer.get(nv21, ySize + vSize, uSize);

    YuvImage yuvImage = new YuvImage(nv21, ImageFormat.NV21, image.getWidth(),
image.getHeight(), null);
    ByteArrayOutputStream out = new ByteArrayOutputStream();
    yuvImage.compressToJpeg(new Rect(0, 0, yuvImage.getWidth(),
yuvImage.getHeight()), 75, out);

    byte[] imageBytes = out.toByteArray();
    return BitmapFactory.decodeByteArray(imageBytes, 0, imageBytes.length);
}

@Override
@WorkerThread
@Nullable

```

```

protected AnalysisResult analyzeImage(ImageProxy image, int rotationDegrees) {
    if (mModule == null) {
        mModule = PyTorchAndroid.LoadModuleFromAsset(getAssets(), "d2go.pt");
    }
    Bitmap bitmap = imgToBitmap(image.getImage());
    Matrix matrix = new Matrix();
    matrix.postRotate(90.0f);
    bitmap = Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(),
matrix, true);

    //TEXT TO SPEECH SECTION

    mTTS = new TextToSpeech(this, new TextToSpeech.OnInitListener() {
        @Override
        public void onInit(int status) {
            if (status == TextToSpeech.SUCCESS) {
                int result = mTTS.setLanguage(Locale.US);

                if (result == TextToSpeech.LANG_MISSING_DATA
                    || result == TextToSpeech.LANG_NOT_SUPPORTED) {
                    Log.e("TTS", "Language not supported");
                }
            } else {
                Log.e("TTS", "Initialization failed");
            }
        }
    });

    //END TEXT TO SPEECH DEFINES

    final FloatBuffer floatBuffer = Tensor.allocateFloatBuffer(3 * bitmap.getWidth()
* bitmap.getHeight());
    TensorImageUtils.bitmapToFloatBuffer(bitmap,
0,0,bitmap.getWidth(),bitmap.getHeight(), PrePostProcessor.NO_MEAN_RGB,
PrePostProcessor.NO_STD_RGB, floatBuffer, 0);
    final Tensor inputTensor = Tensor.fromBlob(floatBuffer, new long[] {3,
bitmap.getHeight(), bitmap.getWidth()});

    IValue[] outputTuple = mModule.forward(IValue.listFrom(inputTensor)).toTuple();
    final Map<String, IValue> map = outputTuple[1].toList()[0].toDictStringKey();
    float[] boxesData;
    float[] scoresData;
    long[] labelsData;

    Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);

    if (map.containsKey("boxes")) {
        final Tensor boxesTensor = map.get("boxes").toTensor();
        final Tensor scoresTensor = map.get("scores").toTensor();
        final Tensor labelsTensor = map.get("labels").toTensor();
        boxesData = boxesTensor.getDataAsFloatArray();
        scoresData = scoresTensor.getDataAsFloatArray();
    }
}

```

```

labelsData = labelsTensor.getDataAsLongArray();

final int n = scoresData.length;
int count = 0;
float[] outputs = new float[n * PrePostProcessor.OUTPUT_COLUMN];
for (int i = 0; i < n; i++) {
    if (scoresData[i] < 0.6) //TODO: Modify score for precision (Original
precision was 0.4)
        continue;

    outputs[PrePostProcessor.OUTPUT_COLUMN * count] = boxesData[4 * i];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 1] = boxesData[4 * i +
1];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 2] = boxesData[4 * i +
2];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 3] = boxesData[4 * i +
3];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 4] = scoresData[i];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 5] = labelsData[i] - 1;
    count++;
}

// getting rid of the shifting box by giving up the scaling
float imgScaleX = 1;  //(float) bitmap.getWidth() /
PrePostProcessor.INPUT_WIDTH;
float imgScaleY = 1;  //(float) bitmap.getHeight() /
PrePostProcessor.INPUT_HEIGHT;
float ivScaleX = (float) mResultView.getWidth() / bitmap.getWidth();
float ivScaleY = (float) mResultView.getHeight() / bitmap.getHeight();

final ArrayList<Result> results =
PrePostProcessor.outputsToPredictions(count, outputs, imgScaleX, imgScaleY, ivScaleX,
ivScaleY, 0, 0);

// getting the size of total objects
final int n_label = results.size();
for (int i = 0; i < n_label; i++)
{
    // for each object, check if it obeys the "conditions"
    float boxSizeRatio = getObjectSizeRatio(results.get(i), 0);
    if (boxSizeRatio > 0.17 && withinBox(results.get(i))) //If result is
greater than 1/6 of total screen
    {
        // vibrate accordingly
        v.vibrate(VibrationEffect.createOneShot((int)(1000 * boxSizeRatio),
(int)(255*boxSizeRatio)));
        // speak accordingly
        speak(getDirection(results.get(i)));
    }
}
}

```

```

        if (results.stream().noneMatch(result -> result.classIndex == 0))
mTTS.stop(); //Stop speech if no more person detected.

        return new AnalysisResult(results);
    }
    return null;
}

/**
 *
 * Determines the ratio of the size of the detected object to the size of the device
screen. Used to approximate how close an object is to the user.
 * @param result The object prediction who's size we want to evaluate
 * @param classToMatch The class of the object we want to test against. Will be
obsolete once code is changed to only recognize useful classes.
 * @return the ratio of the object size to the screen size. Will be a float value
between 0 and 1. Returns -1 if something goes wrong.
 */
protected float getObjectSizeRatio(Result result, int classToMatch) {

    DisplayMetrics displayMetrics = new DisplayMetrics();
    getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
    int screenHeight = displayMetrics.heightPixels;
    int screenWidth = displayMetrics.widthPixels;

    int height = result.rect.height();
    int width = result.rect.width();

    // calculate the box area
    float totalArea = height * width;
    System.out.println("Total Area: " + totalArea);

    // calculate the screen area
    float screenSize = (float) screenHeight * screenWidth;

    if (result.classIndex == classToMatch) {
        return totalArea / screenSize;
    }
    return -1f;
}

/**
 * check if the center point of the result object falls within the virtual box
 * @param result the post processed result for the object
 * @return true when the center point of the result object is within the box, false
otherwise
 */
private boolean withinBox(Result result){
    // check if the object is within the box
    // set up the virtual box
    Rect VB = getVirtualBox();

```

```

    // getting the center points of the result rect
    int resultCenterX = result.rect.centerX();
    int resultCenterY = result.rect.centerY();

    return VB.contains(resultCenterX,resultCenterY);
}

/**
 * determine the object location relative to the user
 * @param result the post processed result for the object
 * @return 0: if the object lies ahead of the user; 1: if the object locates on the
left of the user; 2: if the object locates on the right of the user
 */
private int getDirection(Result result){
    // set up the virtual box
    Rect VB = getVirtualBox();
    int boxCenterX = VB.centerX();

    // getting the center points of the result rect
    int resultCenterX = result.rect.centerX();

    // set up middle section width (the middle area is set to 100x2 pixels wide)
    int middleLeft = boxCenterX - 100;
    int middleRight = boxCenterX + 100;

    // check if the center is on the left side or right side of the box or in the
middle
    // 0: middle 1: left 2: right
    if (resultCenterX >= middleLeft && resultCenterX <= middleRight){
        return 0;
    }
    else if (resultCenterX < middleLeft) return 1;
    else return 2;
}

/**
 * handle the instructional speech
 * @param direction the location of the object relative to the user
 */
private void speak(int direction) {
    //String text = "Person";
    CharSequence text = "";
    // right in front
    if (direction == 0) text = "Stop, person ahead!";

    // on left
    else if (direction == 1) text = "person on left!";

    // on right
    else text = "person on right!";

    float pitch = 1f;

```

```

    if (pitch < 0.1) pitch = 0.1f;
    float speed = 1f;
    if (speed < 0.1) speed = 0.1f;

    mTTS.setPitch(pitch);
    mTTS.setSpeechRate(speed);
    if (!mTTS.isSpeaking()) mTTS.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
}

/**
 * a getter for virtual box
 * to reduce redundancy
 * @return the virtual box
 */
private Rect getVirtualBox(){
    Utilities helper = new Utilities();
    DisplayMetrics displayMetrics = new DisplayMetrics();
    getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
    int screenHeight = displayMetrics.heightPixels;
    int screenWidth = displayMetrics.widthPixels;
    return helper.setupVirtualBox(screenHeight,screenWidth);
}

}

-----
/**
 * class Result
 * to store the object class index, detection score and the bounding box
 */
class Result {
    int classIndex;
    Float score;
    Rect rect;

    public Result(int cls, Float output, Rect rect) {
        this.classIndex = cls;
        this.score = output;
        this.rect = rect;
    }
};

/**
 * class PrePostProcessor
 * handle the result processing
 */
public class PrePostProcessor {
    // for yolov5 model, no need to apply MEAN and STD
    public final static float[] NO_MEAN_RGB = new float[] {0.0f, 0.0f, 0.0f};
    public final static float[] NO_STD_RGB = new float[] {1.0f, 1.0f, 1.0f};
}

```

```

// model input image size
public final static int INPUT_WIDTH = 640;
public final static int INPUT_HEIGHT = 640;
public final static int OUTPUT_COLUMN = 6; // left, top, right, bottom, score and
label

static String[] mClasses;
static int[] classesToDetect = new int[]{0,61,68}; // 0:person, 61:chair, 68:potted
plant

// finalize the results
static ArrayList<Result> outputsToPredictions(int countResult, float[] outputs, float
imgScaleX, float imgScaleY, float ivScaleX, float ivScaleY, float startX, float startY) {
    ArrayList<Result> results = new ArrayList<>();
    for (int i = 0; i < countResult; i++) {

        int finalI = i;
        if (Arrays.stream(classesToDetect).noneMatch(n->
n==outputs[finalI*OUTPUT_COLUMN +5]))
            continue;
        /*if (outputs[i* OUTPUT_COLUMN +5] != 0 ) //TODO: This is where we decide
which classes we want to detect.
            continue;*/

        float left = outputs[i* OUTPUT_COLUMN];
        float top = outputs[i* OUTPUT_COLUMN +1];
        float right = outputs[i* OUTPUT_COLUMN +2];
        float bottom = outputs[i* OUTPUT_COLUMN +3];

        left = imgScaleX * left;
        top = imgScaleY * top;
        right = imgScaleX * right;
        bottom = imgScaleY * bottom;

        /*float horShift = 50; //Boxes are drawn inaccurately, usually shifted
significantly to the left. We need to manually adjust until we can figure out why boxes
are shifted
        left += horShift;
        right += horShift;*/

        Rect rect = new Rect((int)(startX+ivScaleX*left), (int)(startY+top*ivScaleY),
(int)(startX+ivScaleX*right), (int)(startY+ivScaleY*bottom));
        Result result = new Result((int)outputs[i* OUTPUT_COLUMN +5], outputs[i*
OUTPUT_COLUMN +4], rect);
        results.add(result);

    }
    return results;
}
}

```

---



```

/**
 * class ResultView
 * handles the display of all results
 * extends from View
 */
public class ResultView extends View {

    private final static int TEXT_X = 40;
    private final static int TEXT_Y = 35;
    private final static int TEXT_WIDTH = 260;
    private final static int TEXT_HEIGHT = 50;

    private Paint mPaintRectangle;
    private Paint mPaintText;
    private ArrayList<Result> mResults;
    private Rect VB;

    public ResultView(Context context) {
        super(context);
    }

    public ResultView(Context context, AttributeSet attrs){
        super(context, attrs);
        mPaintRectangle = new Paint();
        mPaintRectangle.setColor(Color.YELLOW);
        mPaintText = new Paint();
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);

        if (mResults == null) return;

        // drawing the virtual box
        drawBox(canvas,VB,"Virtual Box");

        // drawing the bounding boxes for each result along with its scores and class
type
        for (Result result : mResults) {
            drawBox(canvas,result.rect,String.format("%s %.2f",
PrePostProcessor.mClasses[result.classIndex], result.score));
        }
    }

    /**
     * a helper function to draw a box with desired texts (with the same style as well)
     * @param canvas target canvas
     * @param box the bounding box
     * @param text desired text
     */
    private void drawBox(Canvas canvas, Rect box,String text){

```

```

mPaintRectangle.setStrokeWidth(5);
mPaintRectangle.setStyle(Paint.Style.STROKE);
canvas.drawRect(box,mPaintRectangle);

Path mPath = new Path();
RectF mRectF = new RectF(box.left, box.top, box.left + TEXT_WIDTH, box.top +
TEXT_HEIGHT);
mPath.addRect(mRectF, Path.Direction.CW);
mPaintText.setColor(Color.MAGENTA);
canvas.drawPath(mPath, mPaintText);

mPaintText.setColor(Color.WHITE);
mPaintText.setStrokeWidth(0);
mPaintText.setStyle(Paint.Style.FILL);
mPaintText.setTextSize(32);
canvas.drawText(text, box.left + TEXT_X, box.top + TEXT_Y, mPaintText);

}

public void setResults(ArrayList<Result> results) {
    mResults = results;
}
public void setBox(Rect box){VB = box;}
}

```

---

```

/**
 * class Utilities
 * a helper class that holds some of the helper functions
 */
public class Utilities {

    /**
     * setup a virtual box in the middle of the screen for detection purpose
     * The box takes up 1/4 of the total screen size and is located in the exact center
     of the screen
     * @param screenHeight
     * @param screenWidth
     */
    public Rect setupVirtualBox(int screenHeight, int screenWidth){
        int centerX = screenWidth/2;
        int centerY = screenHeight/2;

        int topLeftX = centerX - centerX/2;
        int topLeftY = centerY - centerY/2;
        int bottomRightX = centerX + centerX/2;
        int bottomRightY = centerY + centerY/2;

        return new Rect(topLeftX,topLeftY,bottomRightX,bottomRightY);
    }
}

```

## Appendix E – User Guide for in App Usage of AARGUS

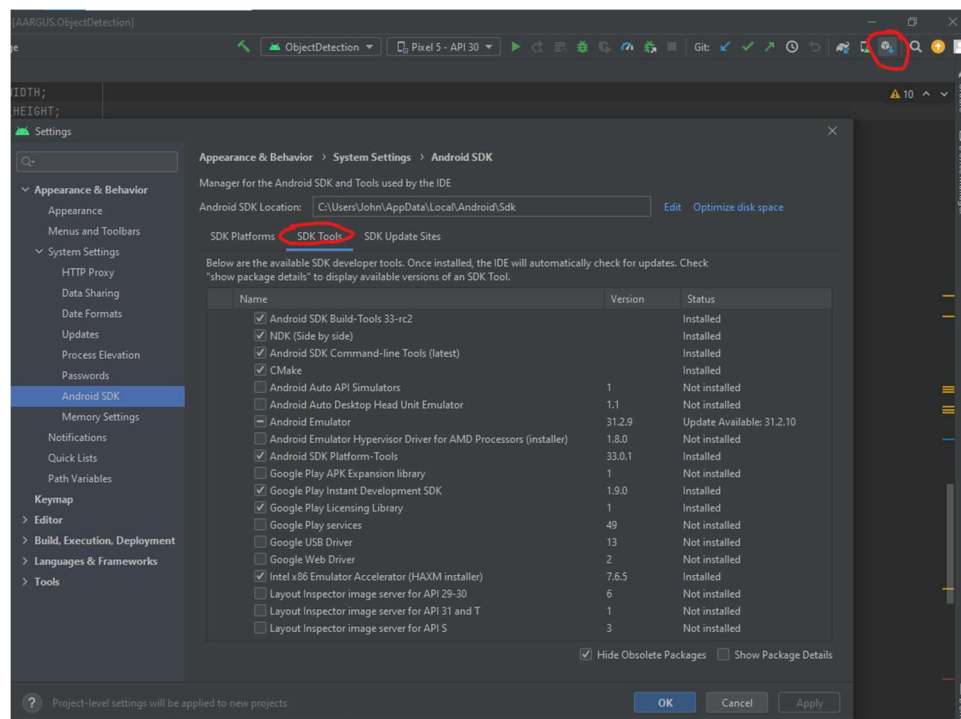
Hello, intrepid developer!

If you're reading this document, then you're attempting to further develop on the AARGUS navigational aid designed by my team. Kudos to you for choosing such a wonderful project.

I'm putting this document together to make the process as seamless as possible for you. A major problem my team ran into is that half of our development time was spent just trying to get a basic app to work, and then we only had a few weeks to properly develop that app. With this guide, you'll be able to quickly get started with development. This should be a complete and total guide to getting started.

### Preliminary Steps

1. To start, I'd recommend downloading the [Anaconda package manager](#). It's my favorite package manager, and makes it easy to keep track of all dependencies for any obscure software. This project shouldn't technically need it for the base version, but if you want to do any work in training your own models, it will be good to have
2. You will want to download [Android Studio](#) Bumblebee to develop the app. If for whatever reason you are trying to develop on iOS, good luck. You are on your own there.
  - a. After setting up Android Studio, you'll have to install the necessary Android SDK tools. I'm not sure exactly which ones are mandatory (I know *Android SDK Platform-Tools* is), but here are the ones I have installed:



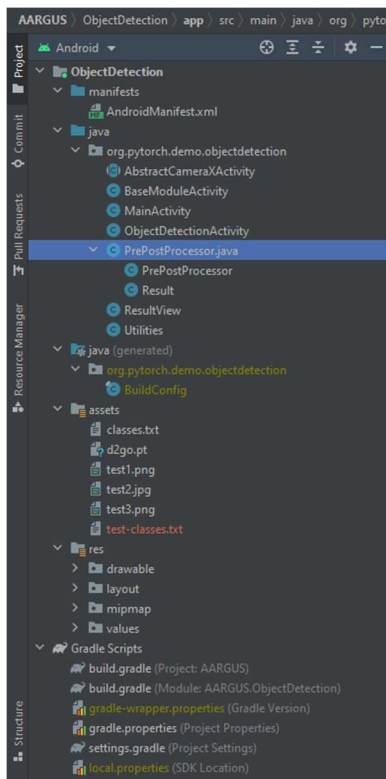
If for whatever reason this isn't all the necessary tools, the error messages Android Studio gives you should be very clear about what you need to do.

3. It's a good idea to follow the instructions to download the base [D2go demo app](#) that AARGUS is based on. It will walk you through installing Detectron2, Pytorch, Torchvision, and the other key software that makes all this possible. Steps 1-3 will let you create the model on your own, and will be an important step to reference if you're trying to do a custom model.
  - i. **NOTE:** I'm not actually sure if this step is necessary to just run the app. I couldn't get a VM to work on my computer to try a fresh installation, but I'm fairly certain all you need to do is just clone the git repository and run it in Android studio. Installing Detectron2, Pytorch, and Torchvision should only be necessary if you're training your own models.
  - ii. In this case, also make sure you have *Pywin32* installed. No tutorials tell you to do it, but I found it to be necessary

## AARGUS Walkthrough

There are a few important parts that make this app work. If you're familiar with android app development, a lot of this should be old news to you. I'm including it just so anyone who doesn't know how android app development works can keep up to speed.

When you boot up the project in android studio, you should see a project structure that looks like this:



We'll go from the top down, explaining the important files.

**AndroidManifest.xml** describes essential information about your app to the Android build tools, the Android operating system, and Google Play. You shouldn't have to change much here, but if for whatever reason you want to change the app logo, or add system permissions (like vibration), or even if you need to add a new activity to the app (I'll discuss activities soon), this file needs to change. The IDE should do some of this stuff for you, but just keep an eye on it.

## Java Files

The Java folder contains the most important code, and the core functionality of the app.

I won't go in to depth on activities, you can read all about them [here](#). In short, an Activity is the way android apps manage different functionalities (or Activities) that an app performs. Here, the main two are **MainActivity** and **ObjectDetectionActivity**. MainActivity handles the front page of the app, which can detect objects from a still image. ObjectDetectionActivity is where most of AARGUS's core functionality lies, as it handles object detection through the live video feed. You can make your own activities pretty easily, if that's appropriate for your development.

MainActivity.java doesn't do much we're interested in right now. The main thing we care about is that the model file used is defined in this file. Currently, we're using the d2go.pt model, but you can use whichever model you like. The "classes.txt" file is a text file holding the names of all of the object classes in the model to be displayed on the screen.

```
182         try {
183             mModule = PyTorchAndroid.loadModuleFromAsset(getAssets(), assetName: "d2go.pt");
184
185             BufferedReader br = new BufferedReader(new InputStreamReader(getAssets().open( fileName: "classes.txt")));
```

The other thing to note in this file is the section that leads to the ObjectDetectionActivity:

```
152         final Button buttonLive = findViewById(R.id.liveButton);
153         buttonLive.setOnClickListener(new View.OnClickListener() {
154             public void onClick(View v) {
155                 final Intent intent = new Intent( packageContext: MainActivity.this, ObjectDetectionActivity.class);
156                 startActivity(intent);
157             }
158         });
```

This shows how you can switch over to any activities you might want to create. It might be useful.

**ObjectDetectionActivity.java** is where most of the magic happens. Most of the methods are just setup for the live object detection. The important part happens in the *analyzeImage* method. Without going line by line, the method takes the raw image from the camera feed, and turns it into a tensor, which is basically a multidimensional array. This tensor is then passed into the complicated detection algorithm using PyTorch's builtin functions:

```
133         IValue[] outputTuple = mModule.forward(IValue.listFrom(inputTensor)).toTuple();
134         final Map<String, IValue> map = outputTuple[1].toList()[0].toDictStringKey();
```

The output is then split up into 3 pieces of information: box data, score data, and label data. Box data includes the information about the box drawn around each detected object, such as size and position. Score data contains a number between 0-1 that corresponds to how "sure" the detection algorithm is of its answer. A score closer to 1 means that the algorithm is more certain that it's correct. Label data contains the value of what the algorithm thinks a detected object is.

This information from all detected objects that meet a score threshold is then all shoved into one giant array called “outputs”. The way it’s structured is somewhat strange, as there’s no obvious delineation between one object and another. Each object takes up 6 spaces in the array, 4 for each corner of the box, one for the score, and one for the label (which is passed as a number for now, and later turned into text by referencing the “classes.txt” file).

```
final int n = scoresData.length;
int count = 0;
float[] outputs = new float[n * PrePostProcessor.OUTPUT_COLUMN];
for (int i = 0; i < n; i++) {
    if (scoresData[i] < 0.6) //TODO: Modify score for precision (Original precision was 0.4)
        continue;

    outputs[PrePostProcessor.OUTPUT_COLUMN * count] = boxesData[4 * i];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 1] = boxesData[4 * i + 1];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 2] = boxesData[4 * i + 2];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 3] = boxesData[4 * i + 3];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 4] = scoresData[i];
    outputs[PrePostProcessor.OUTPUT_COLUMN * count + 5] = labelsData[i] - 1;
    count++;
}
```

Once the output array is fully populated, the array is then passed into the *PrePostProcessor* file for some further analysis, and final object prediction. The only method we care about here is *outputsToPredictions*:

Because we were working with an inefficient model with far too many objects in the dataset, we had to specify which classes of object we actually wanted to detect. In the picture above, you can see that the *classesToDetect* array only contains 3 values. Those values correspond to people, chairs, and desks. The value for each class is simply the index of the class name in the “classes.txt” file. If you’re making your own model, you’ll have to change this function to include every class you want to include. This is as simple as removing a few lines which check for the class before continuing. Remove these lines, and you’ll be able to detect every class in the model you’re using. Alternatively, add or change the classes in the array to detect different objects from the model we used.

The outputs are then conveniently reformatted into the *Result* object.

This class contains most of the information we need. `ClassIndex` references the type of object, `score` is the same as before, and `rect` is a `Rectangle` that describes the bounds of the object detected.

Now, back to `ObjectDetectionActivity`

-