Object Detection Utilizing Mobile Aid App for Visually Impaired

by
Yavuz Furkan Karabıyık

Engineering Project Report

Yeditepe University
Faculty of Engineering
Department of Computer Engineering
2022

Object Detection Utilizing Mobile Aid App for Visually Impaired

APPROVED BY:

Assist. Prof. Dr. Tacha Serif . . . . . .
(Supervisor)

Assist. Prof. Dr. Esin Onbaşıoğlu . . . . . .

Assist. Prof. Dr. Funda Yıldırım . . . . . .

DATE OF APPROVAL: 12/01/2022

ACKNOWLEDGEMENTS

First of all I would like to thank my advisor Assist. Prof. Dr. Tacha Serif for his guidance and support throughout my project. Also I would like to thank Yeditepe University for making this research possible.

Also I would like to thank my friends who supported me while I was doing this project and my family who supported me during my difficult days.

ABSTRACT

Object Detection Utilizing Mobile Aid App for Visually Impaired

In this age where technology is developing day by day, the use of mobile devices is very high. These smart devices that we use every day help us in our daily work. The use of these developing technology products for disabled individuals is also of great importance. It is possible to serve disabled people by using camera, speaker, sensors and similar parts in mobile devices. The number of visually impaired individuals in the society cannot be underestimated. It is estimated that there are 285 million visually impaired people worldwide. These individuals have difficulty in daily tasks such as reading books and distinguishing colors and objects. Using technology in such cases makes their lives easier, but unfortunately the number of technologies developed for the visually impaired remains low.

Depending on these situations, a mobile application that detects objects for visually impaired individuals will be developed in this project. Thanks to the targeted application, it is aimed for visually impaired individuals to find the objects they are looking for in daily life more easily. The object detection method in computer vision will be used in the application. After the objects are detected, the distance of the object and its location will be tried to be determined for the convenience of the visually impaired individual. Then, the names and location information of the identified objects will be sent to the visually impaired individual as a voice message. In addition, the way the application works will be shared with the user via voice message, and in this way, the visually impaired individuals will be able to use it comfortably. Finally, it will be subjected to tests on how accurately the created software can detect objects and its behavior in different conditions.

ÖZET

Görme Engelliler İçin Mobil Yardım Uygulaması ile Nesne Tespiti

Teknolojinin günden güne geliştiği bu çağda, mobil cihaz kullanımı çok yüksek orandadır. Hergün kullandığımız bu akıllı cihazlar gündelik işlerimizde yardımcı olmaktadır. Bu gelişen teknoloji ürünlerinin engelli bireyler için kullanılması da büyük önem arz etmektedir. Mobil cihazların içerisindeki kamera, hoparlör, sensörler ve benzer parçalar kullanılarak engelli bireylere hizmet etmek mümkündür. Toplumda ki görme engelli birey sayısı azımsanamayacak ölçüdedir. Dünya çapında 285 milyon görme engelli olduğu tahmin edilmektedir. Bu bireyler kitap okumak, renkleri ve nesneleri ayırt etmek gibi gündelik işlerde zorlanmaktadır. Bu gibi durumlarda teknolojiyi kullanmak hayatlarını kolaylaştırmaktadır fakat görme engellilere yönelik geliştirilen teknoloji sayısı maalesef düşük kalmaktadır.

Bu durumlara bağlı olarak, bu proje de görme engelli bireyler için nesne tespiti yapan mobil uygulama geliştirilecektir. Hedeflenen uygulama sayesinde görme engelli bireylerin gündelik hayatta aradıkları nesneleri daha kolay bulmaları amaçlanmaktadır. Uygulama içerisinde bilgisayarlı görü içerisinde bulunan object detection yöntemi kullanılacaktır. Nesneler tespit edildikten sonra görme engelli bireye kolaylık olması açısından nesnenin uzaklığı bu bulunduğu konum belirlenmeye çalışılacaktır. Ardından belirlenen nesnelerin isimleri ve konum bilgileri görme engelli bireye sesli mesaj olarak iletilecektir. Ayrıca uygulamanın çalışma şekli sesli mesaj yoluyla kullanıcı ile paylaşılacak ve bu şekilde görme engelli bireylerin rahat kullanımları sağlanacaktır. Son olarak oluşturulan yazılımın nesneleri ne kadar doğru tespit edebildiği ve farklı koşullardaki davranışı gibi konularda testlere tabi tutulacaktır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

Lux	Illuminance intensity
cm	A unit of measure of length, one hundredth of a meter

1. INTRODUCTION

With the development of technology, the use of smartphones is increasing. There are currently 6.4 billion smartphone users in the world (in 2021). If we compare this to the world population, it means that there are over 80 percent smartphone users. A part of this number consists of individuals with disabilities. Technological devices provide convenience and comfort to people with disabilities. Visually impaired individuals use mobile devices and applications to facilitate their daily activities.

The purpose of the application targeted in this project is to detect objects for visually impaired individuals using computer vision. The application will be able to detect the objects around the visually impaired individual and will find the locations of the objects and transmit them to the user with a voice message. In addition, if the user wishes, a special object can be searched around the user. The images of the objects will be reached by the phone camera of the visually impaired individual. The user will be contacted using the device's microphone and speaker. Objects will be detected in the received image and location information will be tried to be calculated.

Accordingly, what is explained in the rest of the report is as follows; In the background part, the related work done before will be explained. The purposes and usage patterns of similar projects will be mentioned. In the Methodologies section, the technologies used in the related works and while creating our application will be explained. In the analysis and design section, the requirements of the application will be mentioned and the working logic will be explained using diagrams.

In the implementation section, the development process of the application will be explained and the technologies used will be mentioned. In the test and results section, the tests of the application in different situations and the results obtained will be mentioned. In the conclusion and future work section, the achieved results and future developments will be mentioned.

2. BACKGROUND

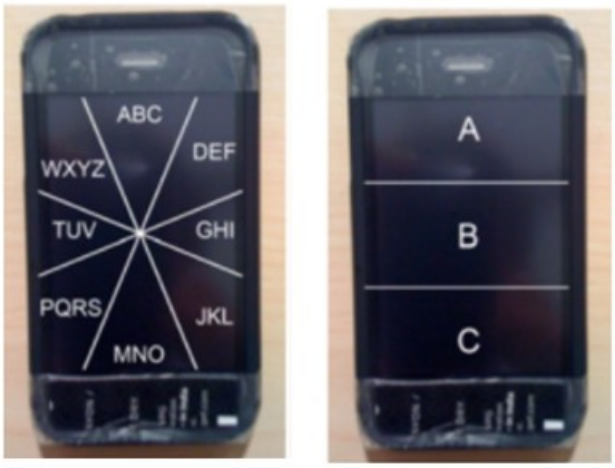
2.1. Previous works

This section reviews and evaluates existing solutions for the visually impaired population. Already produced for the visually impaired, audio devices, specially designed canes called ”white canes” and smart canes (can be seen Wewalk [1]) with sensors are also used. Accordingly the first sub section details projects that have implemented screen enhancements applications which use camera, microphone etc. In some projects, physical objects that divide the screen have been used so that visually impaired individuals may use it. Others are segmented by only symbolic lines on the screen, so such projects are not useful for individuals with high visual impaired. The second subsection will be providing a detailed overview of mobile applications that can be used for image processing, that will act as a third eye for visually impaired people.

The second subsection will be providing a detailed overview of mobile applications that can be used for image processing or the sensors in smartphones, that will act as a third eye for visually impaired people. These applications aim to provide convenience for visually impaired individuals in activities such as finding objects, identifying objects, identifying people and finding directions.

2.2. Screen Enhancements

Recent research has demonstrated that multi-touch events are effective in touch-screen applications for blind people. Bonner, Brudvik, Abowd and Edwards [2] created an eyes-free text entry system in 2010. This system is called No-Look Notes that uses multi-touch for input and audio for output. No-Look Notes was implemented on Apple’s iPhone platform. In the prototype, letters and letter groups grouped according to the predicted frequency of use were arranged in pie menus around the screen. Tested this prototype with five visually impaired participants, each of one entered text for about 1.5 hours. They performed a within-subjects (n = 10) user study of both No-Look Notes and the text entry component of Apple’s VoiceOver, the official accessibility component on the iPhone. No-Look Notes (Figure 2.1) significantly outperformed VoiceOver in terms of speed, accuracy and user preference.



(a) (b)

Figure 2.1. No-Look Notes [2]. (a) is the main screen; (b) is after the 'ABC' target is selected from the main screen.

Frey, Southern and Romero [3] create the BrailleTouch which is an eyes-free entry application for mobile devices. They wanted to use hardware and software together. Actually already there exist a number of hardware and software solutions for eyes-free text entry. However they thought that hardware solutions (An example price list can be seen in Figure 2.2) were expensive and software solutions were insufficient.

Device Name	Braille Display	Price $(USD)
Braille Sense Plus	Yes	6000
Voice Sense	No	2000
VoiceNote BT	No	1900
Refreshabraille 18	Yes	1700
PAC Mate BX 400	No	1500
Braille+ Mobile Manager	No	1400
Maestro	No	1300
Nano	No	1000
EasyLink & Pocketwrite	No	1000
GalaTee	No	400

Figure 2.2. The table lists various hardware solutions for Braille text entry sorted by price in USD. [3]

BrailleTouch provides solutions to both issues. They presented product design rationale and their explorative evaluation of BrailleTouch with HCI experts and visually impaired users. Braille is a 3 by 2 binary matrix that encodes up to 63 characters, not counting the all-null state, which is when no dots are present. In English Braille, a single matrix combination encodes one character.

For example, position 1 (upper left) en-codes the letter “A”, while positions 1 and 4 together (upper left and upper right) represent the letter “C” (see Figure 2.3) [4].

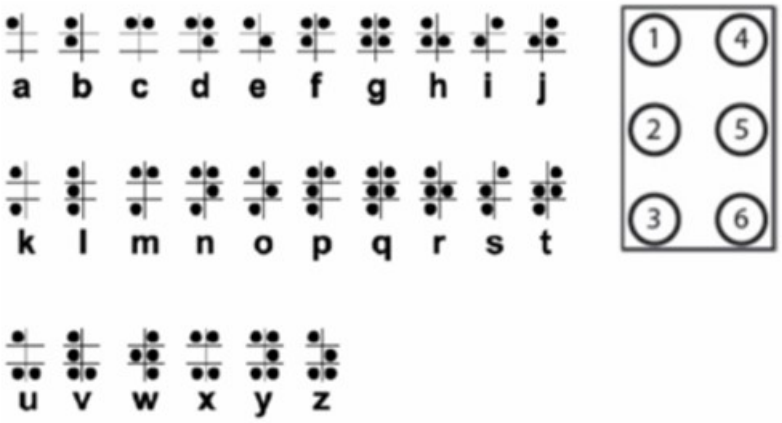


Figure 2.3. The table lists various hardware solutions for Braille text entry sorted by price in USD. [4]

The iPod prototype includes the case to help the user hold the device and position the fingers on the keys. Users hold brailleTouch with two hands and they hold the device with the screen facing away from them (see Figure 2.4). Once placed, the fingers remain in the same position. BrailleTouch allows touch typing on a touch screen using Braille Code.

Figure 2.4. (a) BrailleTouch's back faces the user; (b) BrailleTouch's input surface faces away from the user. [3]

2.3. Visually Impaired Aiding Apps

"Third Eye" is the real-time object detection application for visually impaired people. Developed by Tosun and Karaarslan in September 2018 [5]. Their aim is that the prototype application in this study aims to make the visually impaired people's lives easier with mobile devices. With the designed chest device (see Figure 2.5), the mobile application will be like a virtual third eye that is not very costly and easily accessible.The application has been developed for Android devices. Image processing and machine learning technologies are used. They designed the chest apparatus with a 3D printer to fix the phone to the user's body.

Figure 2.5. SolidWorks Designs for chest strap [5]

The main purpose of the product is to inform the visually impaired audibly. That's why the interface is kept simple as you can see in Figure 2.6. Recognized objects are displayed in a rectangular frame on the mobile device screen.

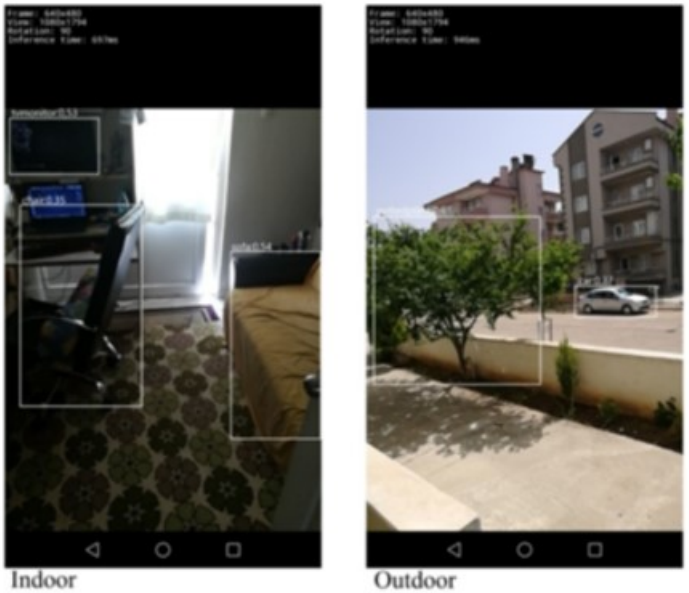


Figure 2.6. Application screenshots [5]

Panëels, Varenne, Blum and Cooperstock [6] implemented a "Walking Straight" application on an existing consumer device, taking advantage of the built-in sensors on smartphones. Their main purpose was to enable visually impaired individuals to walk more comfortably on a straight road since walking straight when deprived of vision is a known problem. Without reference to environmental cues such as the sun, humans have a tendency to walk in circles. The most effective version which they made was tested with nine blind participants. Results demonstrate that Walking Straights (An example photo can be seen in Figure 2.7) significantly reduced the participants' deviation from a straight path as compared to their usual behaviour, e.g., with a guide dog or cane, without affecting their pace. The "Walking Straight" application was developed on a smartphone. They chose the iPhone 4 since the iPhone integrates already accepted accessibility features (VoiceOver) for the blind community, making this a natural choice for the intended user group.

Figure 2.7. The iPhone which using "Walking Straight" is lying on an L-shaped rigid piece of plastic [6]

Another study for the visually impaired individuals was developed by Chaudhry and Chandra [7]. They focused on one of the most difficult tasks faced by the visually impaired individuals which is identification of people. They presented the design and implementation of a face detection and recognition system for the visually impaired through the use of mobile devices. This mobile system is assisted by a server based support system. It features two states of operation; offline and online. These states correspond to the absence and presence of an internet connection respectively. When in the offline state, the application accesses the video streaming from the device's camera. This video stream is then scanned to detect faces. When a face is detected, a temporary image is captured and saved on the device. After the image is saved, it is used to identify the person by searching for a possible match in the application's internal database.

As soon as the person is identified, the result is displayed to the user. In the online state, instead of trying to identify the person, the image is sent to the support servers for identification. After the person is identified, the results are sent back to the application and displayed to the user. (A design of a mobile face recognition system can be seen in Figure 2.8) The system was tested on a custom video database. Experiment results show high face detection accuracy and promising face recognition accuracy in suitable conditions. The system's difficulties include developing better recognition capabilities for adverse lighting and weather scenarios

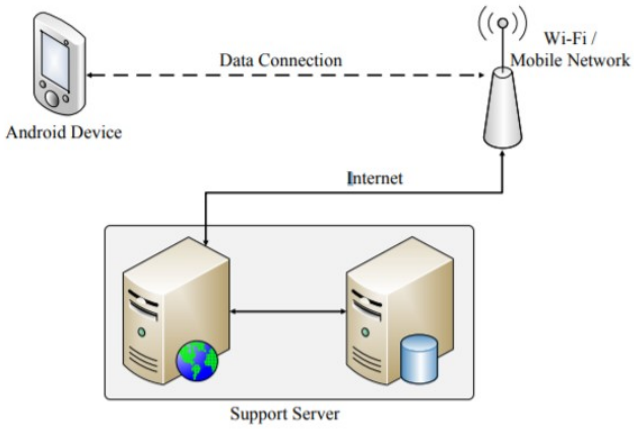


Figure 2.8. Design of the mobile face recognition system. [7]

Long, Karpinsky, Döner and Still [8] designed a mobile application to help visually impaired individuals explore the outdoors. Their aim was to recognize the needs of the visually impaired and to develop assistive applications for partially vision loss individuals. A prototype was created using Axure software. Auditory feedback for each screen option was provided by activating Apple's accessibility settings and VoiceOver. They also made special developments for situations such as checking the battery life of the device, individual falling and needing to call an emergency (Figure 2.9). Functions of the application were tested with visually impaired participants and they completed all tasks in under one minute.

Figure 2.9. Users select the bottom button to begin, select the emergency button, then select Emergency Services (e.g., 112 in the Turkey) [8]

One of the improvements for visually impaired individuals is made by Shaik, Hossain and Yeasin [9]. They presented the design, development and performance evaluation of a Reconfigured Mobile Android Phone for people who are visually impaired. They tried to develop assistive functions for visually impaired individuals such as reading letters, medicine bottles, food containers in the refrigerator; as well as shopping and navigating, walking straight and avoiding collisions, crossing traffic intersections, finding references in open space, etc. Their basic aims were to develop solutions that are low cost and have an intuitive and easy to use interface that can be reconfigured to perform a large number of tasks. They used mobile phone cameras, image capture, text-to-speech (TTS) to develop the proposed application in real time. Empirical analysis under various environments (such as indoor, outdoor, complex background, different surfaces, and different orientations) and usability studies were performed to illustrate the efficiency of the improvement

Nayak and Chandrakala [10] developed an application for the visually impaired that allows schedule management, email and SMS reading completely based on voice command. They wanted to facilitate activities that are difficult for the visually impaired, such as sending and reading emails, program management, wayfinding and sms reading. Speech or text analysis can help improve support for visually-impaired people. These people can speak a command to perform a task. The spoken command will be interpreted by the Speech Recognition Engine (SRE) and can be converted into text or perform suitable actions. The System try to provide blind people to simply speak the desired functionality and be guided there by the system's audio instructions. The proposed and designed app is implemented to support two languages which are English and Hindi. They had an example wayfinding module application: when the user says the outdoor region, it will redirect users to the Direction Screen Module. Visually impaired people must specify the destination address with voice command. Initially it will show a point at our location and when specifying a destination address it will show the way to reach the destination. It will also give voice instructions to reach the destination.

3. METHODOLOGIES

This section highlights the methodologies related to the hardware, software and algorithms of the system. The first subsection describes the methodologies for the hardware of the system. The second subsection describes the technologies that should be used in software development. The last subsection describes the algorithms and libraries used for the software development process.

3.1. Hardware

While developing products for visually impaired individuals, physical devices can be developed to assist in perception. These devices can sometimes be used to segment touchscreens, sometimes they can be used to stabilize mobile devices or take advantage of additional sensors. This subsection describes these related auxiliary equipment.

While developing products for visually impaired individuals, 3D printing is used to create auxiliary objects. 3D printing is the process of producing three-dimensional solid objects from a three-dimensional file prepared in the digital environment. Machines that perform these operations are called three-dimensional printers (Figure 3.1).

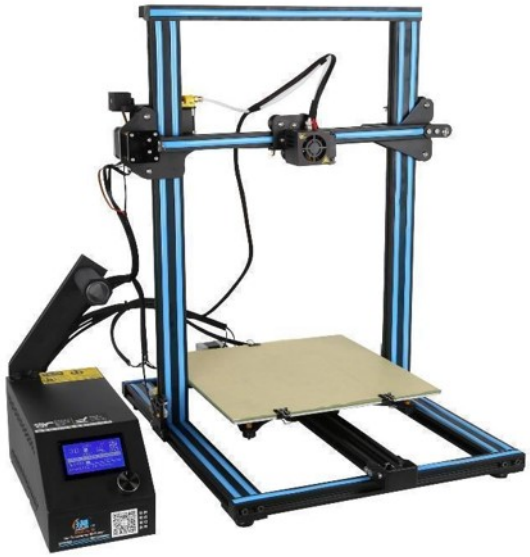


Figure 3.1. An image of a Creality CR-10S 3D Printer

Prints can be made using many types of raw materials. The most used raw materials are hard plastics called PLA and ABS. There are three-dimensional printers that can print using different techniques.The working principle of most three-dimensional printers is based on dividing any three-dimensional object prepared in the computer environment into layers, and printing each layer overlapping by pouring the melted raw material.

Mobile devices contain many sensors and these sensors add functionality to mobile devices. It is thanks to these sensors that phones can count steps or find location and many more features. While developing mobile applications for visually impaired individuals, the sensors of mobile devices are very important and are used actively. In this part of the article, we will explain the four sensors found in mobile devices.

Accelerometers (An example accelerometer can be seen in Figure 3.2) are devices that measure the acceleration applied to a mass. It handles axis-based motion detection and can be found on phones and fitness trackers. Thanks to this device, your smartphone can track your steps. It can also tell the phone software which direction the handset is facing.

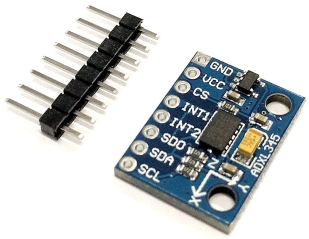


Figure 3.2. An image of ADXL345 3 Axis Accelerometer

The gyroscope is a device used for direction determination. It is integrated in technological devices such as mobile phones and tablets. It is a system that we use a lot in applications, games and 3D software. It is used in applications such as car races, game consoles, camera sensitivity adjustment in photo shoots and navigation that we steer by turning left and right on mobile phones. Gyroscopes inside smartphones are MEMS (Micro-Electro-Mechanical Systems) gyroscopes. The gyroscope sensor is a system that can detect angular velocity. In other words, the rotation direction and speed of a stationary object are determined by comparing the angular ratios in three vertical axes. It processes the data it detects with the help of the processor and converts it into electrical signals. The gyroscope sensor is similar to an accelerometer, but there is a big difference between them: The accelerometer measures the acceleration of the device, while the gyroscope measures the rotational speed of 3 coordinates (X, Y, Z).

Magnetometer (An example magnetometer can be seen in Figure 3.3) is a device used to measure magnetic field strength and direction. It serves to inform the user which road is north by measuring the magnetic field and reporting it to the phone. In map applications, this sensor decides which direction the map should be. It is also used in areas such as the detection of underground resources and safety checkpoints.

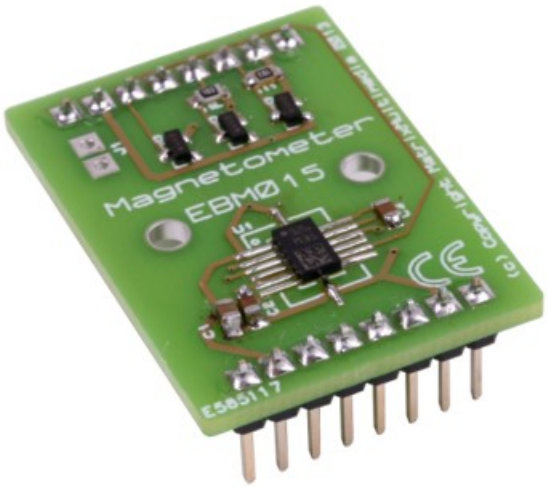


Figure 3.3. An image of EBM015 Magnetometer Sensor Module

One of the most frequently used sensors is GPS (An example GPS module can be seen in Figure 3.4), especially when developing applications for visually impaired individuals. GPS (Global Positioning System); It is a network of satellites that regularly sends coded information and makes it possible to pinpoint the exact location on Earth by measuring the distance to satellites. In fact, they do not use any data from mobile devices, so you can access location information even when your phone loses signal. GPS consumes the battery of mobile devices, so it is recommended to turn it off to save battery.

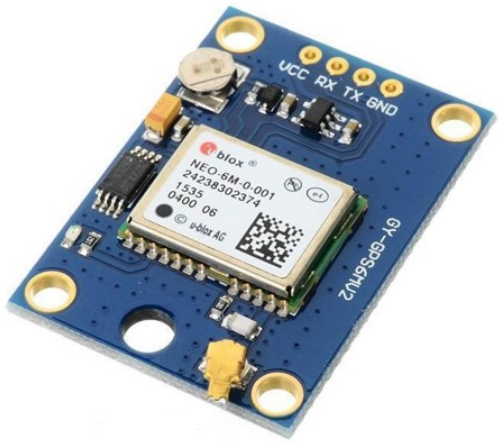


Figure 3.4. An image of GY-NEO6MV2 GPS Module

2D barcodes (Figure 3.5) can be used for visually impaired individuals to recognize objects. A two-dimensional barcode is a graphical image that stores information both horizontally and vertically. A one-dimensional barcode can only store up to 20 characters of information, while 2D codes can store up to 7,089 characters. The visually impaired individual can get information audibly by having his device read the 2D barcode of the product he wants to have information about. The barcode reader interprets the encoded URL(Uniform Resource Locators), which directs the browser to the relevant information on a Web then this information can be converted into sound and transmitted to the visually impaired.

(a) (b) (c) (d)

Figure 3.5. Types of 2D barcodes: (a) QR Code; (b) Datamatrix Code; (c) PDF417; (d) AztecAn image of GY-NEO6MV2 GPS Module

3.2. Software

The programming language chosen is also important when developing software for visually impaired individuals. Java (Figure 3.6) is an open source, object-oriented and high-level programming language. It is one of the most used programming languages. One reason the Java language has not lost its popularity is that compiled Java code can run on all platforms that support Java without the need for recompilation.

Figure 3.6. An image of Java logo

One of the widely used programming languages is C++ which is a general-purpose programming language as an extension of the C programming language. The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features.

Another programming language is Python. Python has a very clean syntax, it is easy to use and read. It is an object-oriented programming language and can run on different platforms such as Unix, Linux, Mac, Windows, Amiga, Symbian. Software can be developed in many areas such as system programming, user interface programming, network programming, web programming, application and database software programming with Python.

In the process of developing applications for visually impaired individuals, mobile devices are preferred because they are very common and accessible. While developing, it is preferred to be developed for android devices as it is free and allows development on computers with different operating systems.

Figure 3.7. An image of Android logo

Android [11] (Figure 3.7) is a linux-based and free operating system developed for touchscreen mobile devices such as smartphones and tablets. The Android supported application extension is ".apk". Android has a large group of developers who write apps that extend the functionality of devices. There are currently over 2.6 million apps for Android.

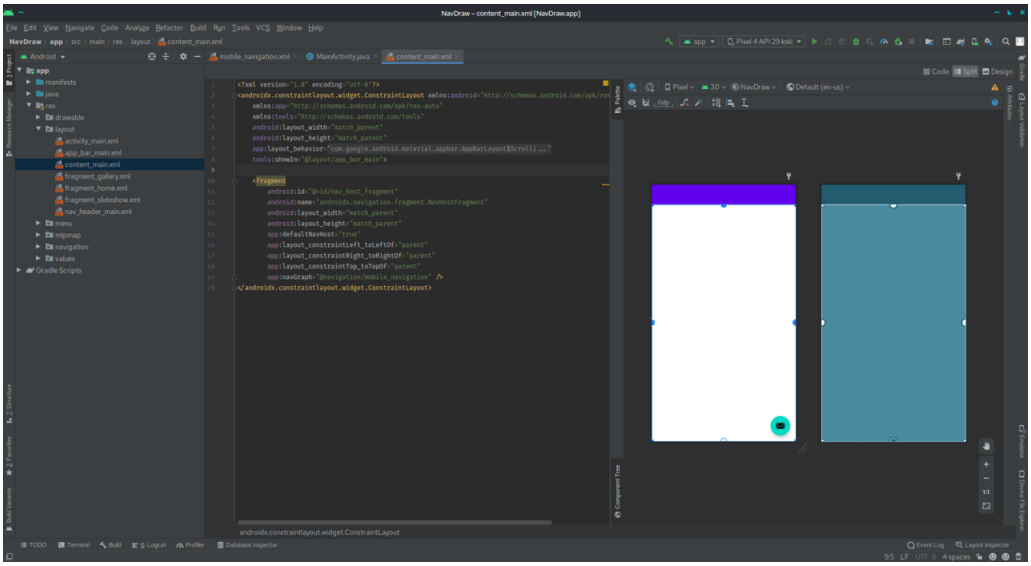


Figure 3.8. A screenshot of Android Studio 4.1

IDE (Integrated Development Environment) aims to enable computer programmers to develop software comfortably and quickly. In addition to organizing the software development process, it also includes technologies that contribute to software development. Android Studio (Figure 3.8) is the official integrated development environment (IDE) for Google's Android operating system. It is based on JetBrains' IntelliJ IDEA software and designed specifically for Android development and is available for download on Windows, macOS and Linux based operating systems. Kotlin, Java and C++ are supported programming languages. It also allows us to perform device tests without using physical devices by downloading emulators into it.

The running time of the application is important in cases where there is audible and instant feedback. Android NDK can be used with Android SDK (Software Development Kit) to speed up the application in some cases. Android NDK is a toolkit that allows us to use your code written in languages such as C and C++ in native Android applications. Likewise, it allows you to use libraries prepared for these languages, these can be both ready-made libraries and APIs that provide high performance.

Eclipse is an open source integrated development environment (IDE) commonly used in software development. Although the focus is on Java and Java-related technologies, it is also used for different languages such as C and Python, thanks to its flexible structure. With its fast interface, elegant design and powerful capabilities, it soon became the most popular development environment among Java developers.

3.3. Algorithms and Libraries

Using algorithms and existing libraries in the software development process speeds up the process and gives the chance to develop the software with much less resources and effort. In this subsection, we will examine commonly used technologies in image processing such as computer vision, the YOLO algorithm, and OpenCV.

3.3.1. Computer Vision

Computer vision [12] is the field of science that deals with how computers can process digital photos and video with the best efficiency. From an engineering perspective, it tries to understand and automate the tasks that the human visual system can do. Computer vision includes methods such as obtaining, processing, analyzing and understanding data from the real world to produce digital information. Subfields of computer vision include scene reconstruction, object detection, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, visual servo rendering, 3D scene modeling, and image restoration.

3.3.2. YOLO: Real Time Object Detection

YOLO [13] is an abbreviation for the term 'You Only Look Once'. YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.

To understand how object detection works, basic topics such as image classification, object localization and object detection should be explained. Image classification: attempts to assign an image to one of several different categories (car, cat, human, etc.). "What's in this image?" It aims to answer the question . Only one category should be assigned to a piece of image. Object localization: allows us to find our object in the image, tries to detect what the object in the image is and where it is. Object detection: finding all objects in an image and making object detection observable by drawing bounding boxes around them. In a real life scenario, multiple objects must be present in a single image and the image must go through the stages (An example image can be seen in Figure 3.9) mentioned above.

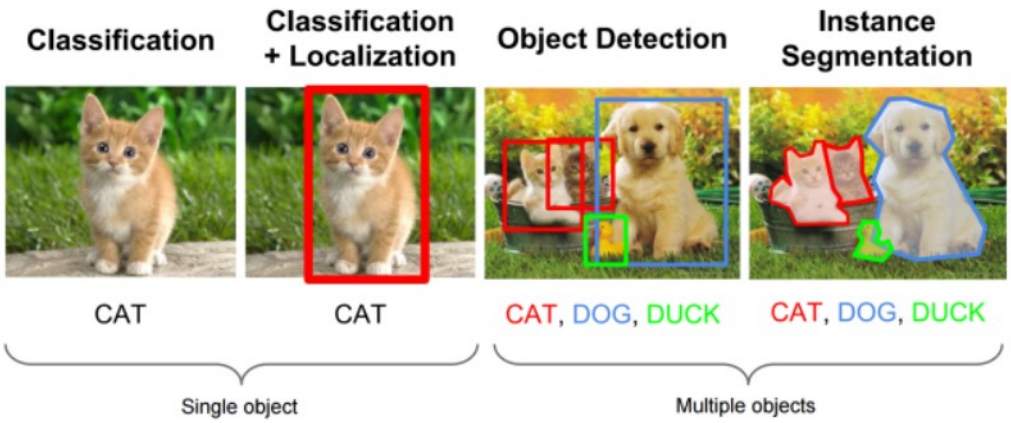


Figure 3.9. An example image of object detection [14]

The YOLO algorithm works by dividing the image into grids, each having an equal dimensional. Each of these grids is responsible for the detection and localization of the object it contains. Accordingly, the algorithm estimates the box coordinates. It predicts the same object with different bounding box predictions and reveals many duplicate predictions. YOLO uses Non-Maximal Suppression (An example of Non Maximal Suppression can be seen in Figure 3.10) to deal with this issue. In Non-Maximal Suppression, YOLO removes all bounding boxes with lower probability scores.

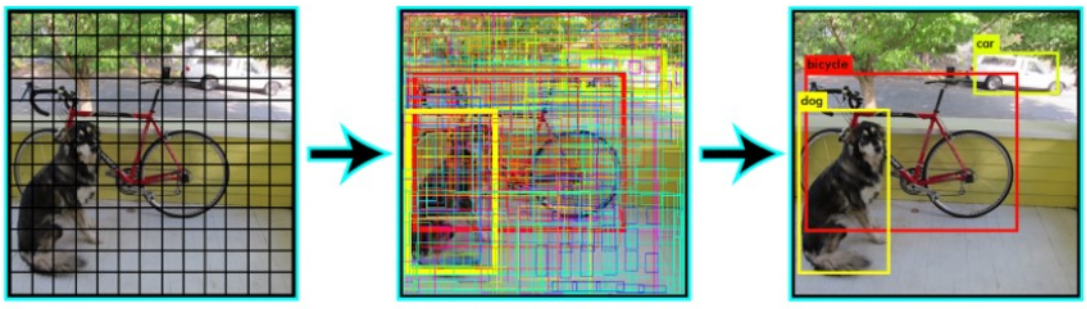


Figure 3.10. An example of Non Maximal Suppression [13]

3.3.3. OpenCV

OpenCV (Open Source Computer Vision Library) [15] is an open source library used in real-time computer vision applications. It has a BSD license, so you can use it for free in any project you want. OpenCV (Figure 3.11) is a platform independent library, so it can run on Windows, Linux, FreeBSD, Android, Mac OS and iOS platforms. There are more than 2500 algorithms for image processing and machine learning in the OpenCV library. With these algorithms, operations such as face recognition, distinguishing objects, detecting human movements, object classification, license plate recognition, processing on three-dimensional images, image comparison, optical character identification OCR (Optical Character Recognition) can be easily performed.

Figure 3.11. An image of OpenCV logo

3.3.4. TensorFlow

TensorFlow [16] is an open source software library for machine learning. It is a symbolic math library based on dataflow and differentiable programming to handle a variety of tasks including deep neural network training and inference. It allows developers to create machine learning applications using various tools such as libraries and community resources. TensorFlow applications can be run on a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. TensorFlow allows you to build dataflow graphs and structures to define how data moves through a graph by taking inputs as a multi dimensional array called Tensor. TensorFlow (Figure 3.12) consists of two main components, tensor and graphs. The graph depicts the operations and relationships that exist between the nodes. Graph framework is used by TensorFlow. Each connection or edge between nodes is a multidimensional vector or matrix that can represent any data format called tensor. Tensors are used in computation of tensorflow.

Figure 3.12. An image of TensorFlow logo

TensorFlow Lite has been prepared for you to use the TensorFlow library in your mobile applications or IoT projects. If you are developing an iOS application and want to include TensorFlow in your application, you need to add the following library. Similar usages apply on the Android side as well. If you want to use TensorFlow in your application, you should add the neccesary gradle packages to your project.

As can be seen, there are many algorithms and libraries for computer vision. Each of them has its advantages, some of them are more advantageous on desktop computers and some seem more suitable for use on mobile devices. TensorFlow Lite is one of the most advantageous libraries for object detection on mobile devices. It is preferred today due to its ease of use, effective employee algorithm and other features.

4. ANALYSIS AND DESIGN

This section highlights the requirements and design of the project. The first subsection details the functional and non-functional requirements. In line with the identified functional and non-functional requirements, the second subsection presents the relevant UML diagrams and describes them and details.

4.1. Requirements

This section highlights both functional and non-functional requirements. It explains the issues to be considered while making decisions, that is how to design the system.

4.1.1. Functional Requirements

Functional requirements define the functions that the system must provide. In other words, functional requirements describe the functionality of a software system or module. Seven functional requirements are defined in this section.

The application must understand what the searching object is from the speech of the visually impaired person. The user has to indicate to the application which object he is looking for. It will be difficult for the visually impaired person to type the name of the object using the keyboard. Therefore, the searched word must be specified by speech and the application must learn the name of the object using the phone's microphone.

The application should try to find the sought object using computer vision. The application should search for the specified object with the object recognition algorithm and the user's phone camera. Identifying and specifying the location of the object will provide convenience for the visually impaired user.

The application should give instructions (information) to the visually impaired user about the location of the object found by object recognition. After determining the location of the searched object relative to the smartphone, voice messages should be given to the user by using the phone's speaker. In this way, the user's vision loss is eliminated and the application achieves its purpose.

The application should be able to work on different smart devices. Visually impaired users should be able to run the application from android devices with different hardware or different versions. Moreover, the user should be able to use the application when he changes his smartphone or by installing the necessary software on another person's phone. In this way, more users will be able to access and benefit from the application.

The application should give information to the visually impaired user about which button does what. Partially visually impaired users can understand the location of the buttons while using the application, but they may not be able to see the texts written on the button. It can happen to blind users remembering the button location and forgetting its function. Therefore, the application should indicate the directions and functions of the buttons to the user using voice messages.

If the application cannot understand the name of the object indicated by the sound or if it cannot find the specified object in the given image, it should give voice information to the user. The visually impaired user may not understand by looking at the screen whether the object indicated by the voice is understood by the application. For this reason, if the said object is incomprehensible, the user should be asked to say it again. In addition, If the application cannot find the object with computer vision, it should inform the user with sound in order to inform the user and change the angle of the camera. In this way, the desired result can be obtained by providing interaction between the user and the application.

4.1.2. Non-Functional Requirements

Non-functional requirements define constraints on the functions and services that the system offers. Examples include details such as time constraints, security, development constraints or standards. Seven non-functional requirements are defined and explained in this section.

The application should detect the sound within a maximum of 2 seconds when the searched object is spoken. When the visually impaired individual says the name of the object he is looking for, his voice should be identified in a short time, and if the searched object is understood, it should be transmitted to the computer vision. If the name of the object is not understood, a voice notification should be sent to the user for it to be said again. The fact that the sound detection is fast will allow the user to continue without spending too much time.

The application should search for the object of the visually impaired individual with a maximum of 3 keystrokes. The application should be easy to operate and produce results with a few keystrokes so that users can use the application with less vision or blindness (learning from a different person) without getting bored. In addition, the interface should be created with simple and distinctive colors.

The application should be able to detect objects at illuminance levels of 40 lux and above. Visually impaired individuals can be found in places that do not have high light levels. The object detection application should be able to work in such environments and assist the user.

Buttons in the application should be designed to be used with one hand. Visually impaired individuals can use the phone with one hand and use the other hand to search for objects or to maintain their balance by touching. For this reason, the buttons in the application should be used comfortably with one hand.

The application should delete the images provided by the visually impaired user during object detection. In order not to violate the personal rights of the user and for security purposes, the images taken for object detection in the user's home, vehicle or any area should never be shared with third parties and should be deleted after the object is detected. In this way, a reliable service is provided to the user.

The application must indicate within a maximum of 4 seconds whether the searched object is in the presented image or not. In order not to disrupt the work of the visually impaired person and to ensure that he does not get bored, the image obtained should be scanned in a short time and the user should be informed whether the object is found.

The application should repeat the sound notifications at appropriate intervals while describing the location of the found object. If the voice notifications indicating the location of the determined object are repeated very rapidly or slowly, the visually impaired individual may have difficulty in understanding. Therefore, the algorithm should search for the location of the object every 2 seconds on average and guide the user with a voice notification again.

Voice notifications transmitted by the application to the user should be understandable. Sound notifications are important for the visually impaired user and should not be misunderstood. If the notification sound is too loud, too low or incomprehensible, it may disturb the user or cause misunderstanding.

The application should not occupy more than 100 megabytes of disk space on the smart device. Different applications may be installed on the smartphones of visually impaired users. The object identification application taking up too much space may cause the phone to slow down or be unable to load different applications. Therefore, it is important that the application takes up little space on the disk.

4.2. Design

4.2.1. Use Case Diagram

This section shows the use case diagram of the object detection application developed for visually impaired individuals (Figure 4.1).

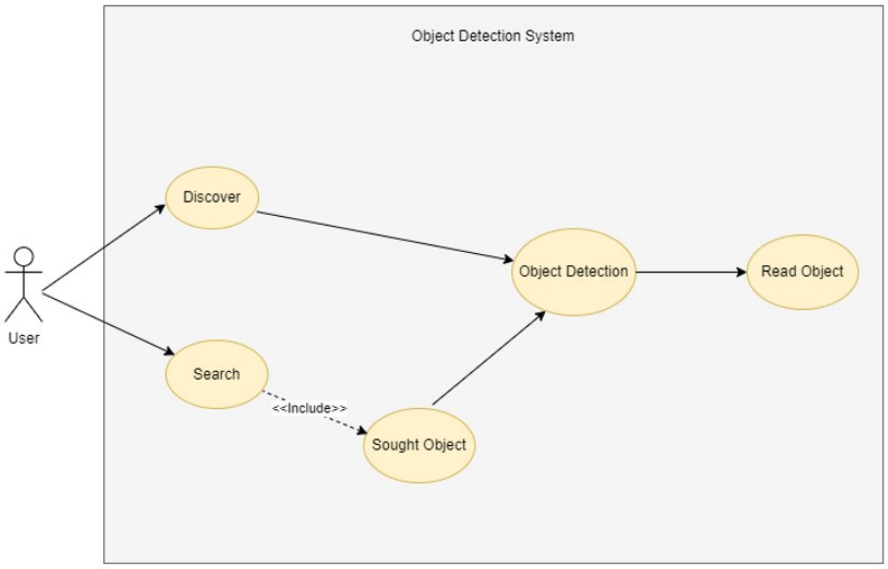


Figure 4.1. The use case diagram of the object detection system

In the use case diagram (Figure 4.1), it is seen that the actor in the system is the user. The user can choose between discover or search functions. The Discover option was created with the aim of reading the names of the objects found by the application. The Search selection was created to find the object specified by the user. After selecting the search option, the user must specify the searched object. Attempts are made to capture the ”specified” or ”discovered” object(s). Afterwards, the results are transmitted to the visually impaired user with sound.

4.2.2. Sequence Diagram

This section demonstrates the sequence diagram of the system (Figure 4.2) and describes the interaction details.

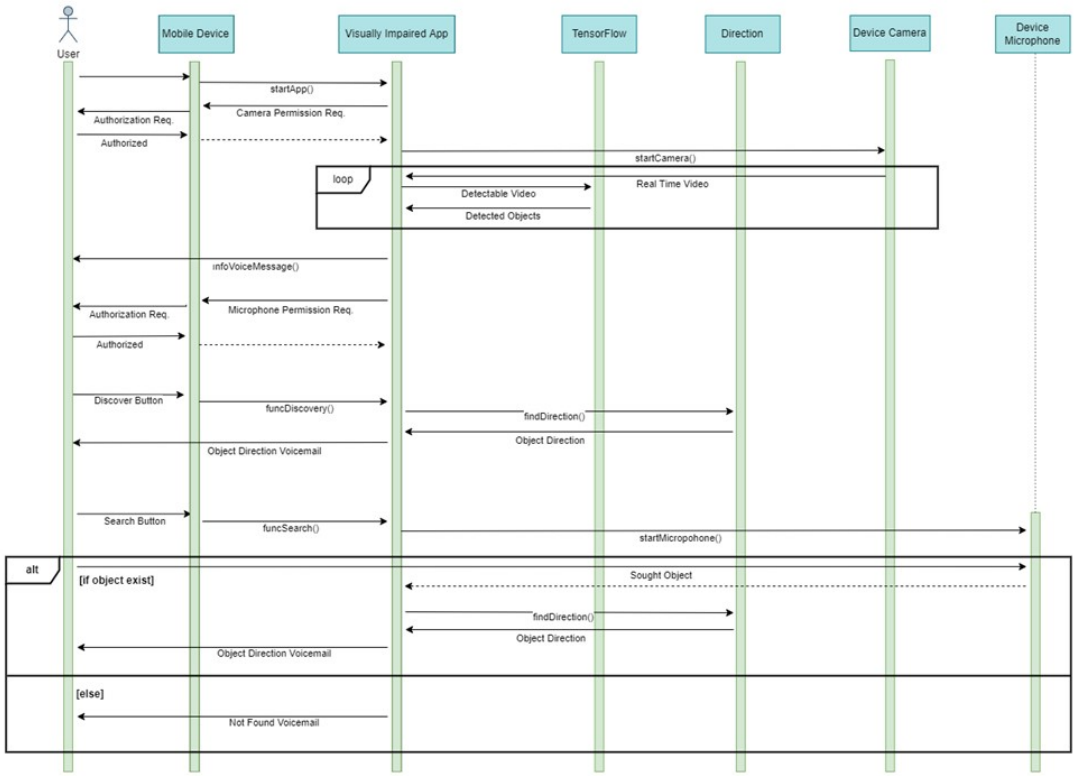


Figure 4.2. The sequence case diagram of the object detection system

In the sequence diagram we will try to describe interactions between user and application. As the diagram shows, the camera, speaker and microphone of the mobile device are required for the application to work properly. In addition to these, the TensorFlow library and the direction section developed to predict the direction of the object are also important. Figure 4.2 shows these system parts clearly.

Let's explain the user interactions and communication between components shown in the sequence diagram. First the application is started by the user. Then, the user is asked for permission to use the device's camera and microphone. If allowed, the process continues, otherwise the application terminates.

It starts taking images from the camera and this image is processed using the TensorFlow library and objects are detected. The processed image starts to show on the application screen. With the operation of the application, a voice message is given to the visually impaired user. This message gives information about the location and functions of the buttons. There are two functions that the user can select with the buttons, these are search and discovery operations. If the visually impaired user presses the explore button, the direction of the objects detected from the camera image is determined first. Then, the object name and direction information is transmitted to the user with a voice message. If the user presses the search button, the searched object is requested to be said using a voice message. When the user says the name of the object he is looking for, this object is searched among the detected objects. If found, direction information is generated and the user is told the name and direction of the object using a voice message. If the searched object is not found among the detected objects, it is conveyed to the user via a voice message that it cannot be found.

5. IMPLEMENTATION

This section explains the implementation stages of the application developed for the visually impaired. The environments used during development, the programming languages and libraries used will be explained. Then, the preparation process of the application, the preferences and the details of interaction with the user will be mentioned. The developed software will be explained using pseudo codes.

5.1. System Architecture

In this section, how the application is designed and the infrastructures used are explained. The application has been developed using the necessary software running on the computer. It is aimed for visually impaired individuals to use the application using their smartphones. Smartphones work on different operating systems, and it was preferred to develop devices with android operating systems because they have more users and are accessible. Let's briefly talk about the software development environment used. Android [11] (Android and related technologies explained in chapter 3), is a free linux-based operating system developed for touchscreen mobile devices such as smartphones and tablets. Android Studio is the official Integrated Development Environment (IDE) for Android app development and it offers even more features that enhance your productivity when building Android apps.

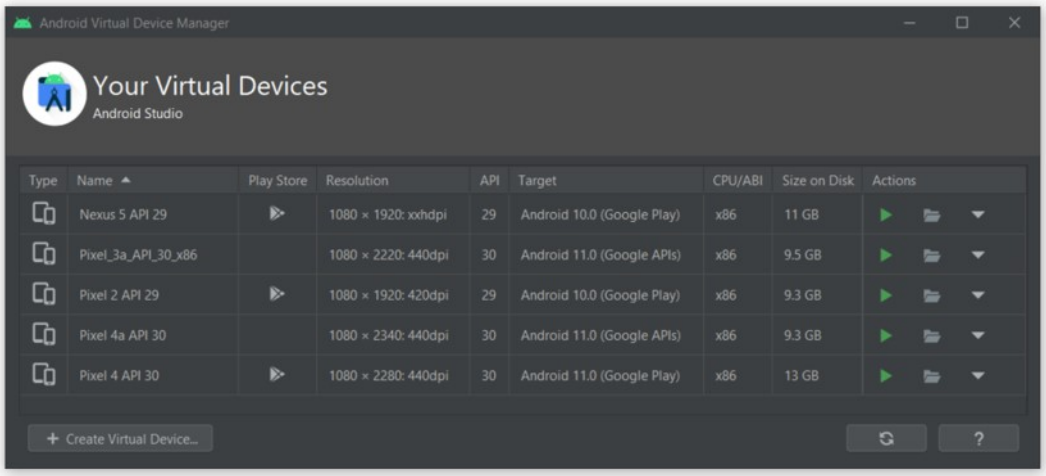


Figure 5.1. The screen shot of the virtual devices

As we mentioned, Android Studio was used as the development environment. During development, emulators were used for device testing. The emulator list we use in the application can be seen in Figure 5.1.

If we want to explain an emulator: it is a piece of hardware or software that allows one computer system (called a host) to behave like another computer system (called a guest). In most cases, an emulator allows the host system to run software designed for the guest system or use peripheral devices. The ability of a computer program in an electronic device to simulate another program or device is called emulation. In addition, emulators allow us to test our application on devices of different brands and models.

A software library is a set of program codes and data that software developers use when developing programs. Software libraries assist programmers and compilers of programming languages to develop executable programs. Software libraries usually contain pre-made code, classes, procedures, scripts, and configuration data. While developing software, a developer uses software libraries to make his software more functional or to add various functions. There are different libraries and trained models for computer vision, which is the main function of the visually impaired application. TensorFlow lite was used because of its support for use on mobile devices, it's up-to-date and effective. TensorFlow lite is the mobile version of the TensorFlow library. The TensorFlow Lite architecture can be seen in Figure 5.2 .

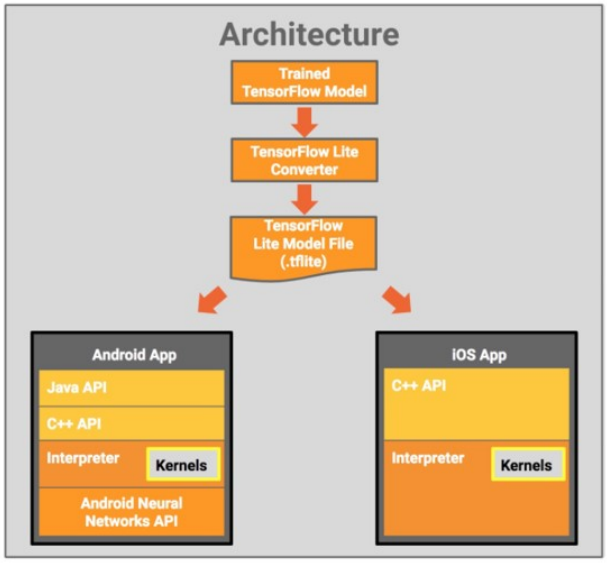


Figure 5.2. The figure of the TensorFlow Lite architecture [16]

In order for the computer vision to detect the objects, the data of the objects is needed and in these cases datasets are used. Dataset is a collection of data. One of the most frequently used datasets in object detection operations is the COCO dataset, and we performed operations on this dataset in our application. The COCO (Common Objects in Context) dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. The dataset consists of 328000 images. Using this dataset, we can detect 80 known objects. (The name of these objects can be seen in Figure 5.3.)

```
labels = ["person","bicycle","car","motorcycle","airplane","bus","train","truck","boat",
          "trafficlight","firehydrant","stopsign","parkingmeter","bench","bird","cat",
          "dog","horse","sheep","cow","elephant","bear","zebra","giraffe","backpack",
          "umbrella","handbag","tie","suitcase","frisbee","skis","snowboard","sportsball",
          "kite","baseballbat","baseballglove","skateboard","surfboard","tennisracket",
          "bottle","wineglass","cup","fork","knife","spoon","bowl","banana","apple",
          "sandwich","orange","broccoli","carrot","hotdog","pizza","donut","cake","chair",
          "sofa","pottedplant","bed","diningtable","toilet","tvmonitor","laptop","mouse",
          "remote","keyboard","cellphone","microwave","oven","toaster","sink","refrigerator",
          "book","clock","vase","scissors","teddybear","hairdrier","toothbrush"]
```

Figure 5.3. The labels of the COCO Dataset

5.2. Software Development Process

In this section, the software development process of the application created for the visually impaired is explained. How the technologies described in the previous chapter are used, the methods followed during the development process and the preferences will be mentioned.

First the basic GUI was created. To explain it, a GUI (Graphical User Interface) is a graphical operating system interface that uses icons, menus, and a mouse (to click the icon or pull down the menus) to manage interaction with the system. Image of empty Android Studio empty screen can be seen in Figure 5.4.

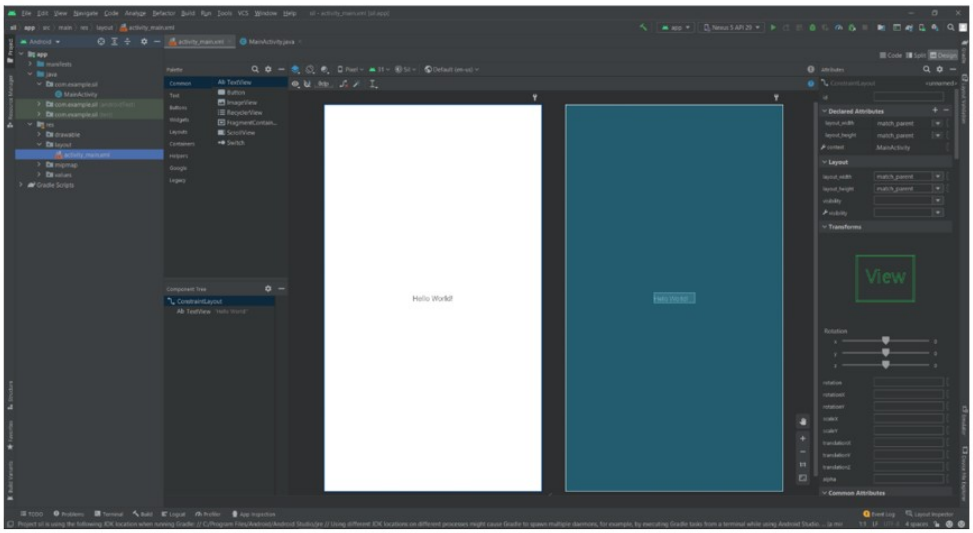


Figure 5.4. The screen shot of the empty Android Studio GUI

Only two buttons (Figure 5.5) have been added to the interface for the visually impaired and it has been kept simple. The buttons are painted white so that people with partial vision can easily perceive them.

The button on the right is the discover button. It is designed to be used when visually impaired individuals want to know what objects are in front of the phone. Objects are detected using computer vision when the application starts running. When the user clicks the right (discover) button for the first time, the direction of the detected objects should be determined and the name and direction of the object (left, right, middle) should be reported to the visually impaired user using the device's loudspeaker. If the user presses the right button for the second time, the voice information system should be terminated. The button on the left is the search button. It is designed for the visually impaired user to be able to use it when he wants to find a certain object. When this button is pressed, the device's microphone is used to get the name of the object that the visually impaired user is looking for. This object is searched among the detected objects. If found, its direction is determined and this information is transmitted to the user using the device's loudspeaker.

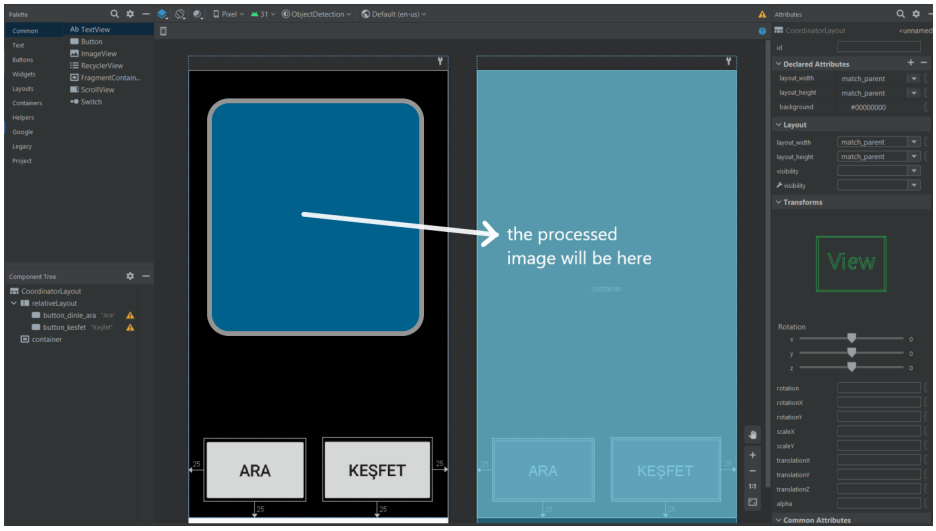


Figure 5.5. The screen shot of the Android Studio with related buttons

As we mentioned, TensorFlow Lite API and library are used for object detection. In order to use these technologies, the imported Gradle dependency can be seen in Figure 5.6.

```
dependencies {
    // Other dependencies

    // Import the Task Vision Library dependency
    implementation 'org.tensorflow:tensorflow-lite-task-vision:0.3.0'
    // Import the GPU delegate plugin Library for GPU inference
    implementation 'org.tensorflow:tensorflow-lite-gpu-delegate-
plugin:0.3.0'
}
```

Figure 5.6. A code of implementation dependencies

After that, camera operations can be started. First, the user was asked for permission to use the camera (Figure 5.7). If the camera permission cannot be obtained, the application cannot work. The front camera of the device is insufficient for our object detection so we use the back camera. If allowed, the camera engine is started and images are taken continuously.

In order for the received image to be processed, the dimensions must first be edited. Frames are assigned coordinates and rotation is calculated. Rotation is important to follow the image as the device moves. Dimensions are adjusted and fixed according to these data. Then, object detection processes are started.

```
Program CameraActivity()
      // Ask user
      boolean cameraPermission = getCameraPermission()
      // We don't use front facing camera
      camera = chooseCamera()

      if ( cameraPermission == false ) then
            stopActivity()
      endif
      if( camera == CAMERA.BACK ) then
            startCameraEngine()
      else
            changeCamera()
      endif
End

     Program RealImage()
          image = getImageFromCamera()
          imageWidth = getImageWidth()
          imageHeight = getImageHeight()
          fragmentProcess(image, imageWidth, imageHeight);

          switch ( getRotation() )
               case Surface.ROTATION_270;
                    return 270;
               case Surface.ROTATION_180;
                    return 180;
               case Surface.ROTATION_90;
                    return 90;
               default:
                    return 0;
     End
```

Figure 5.7. A pseudo code of camara activities and source image

We use TensorFlow Lite's trained model from the file. In addition, we use a file containing the names of the objects in the COCO Dataset when detecting an object. The algorithm detects the same object multiple times and sometimes fails to detect objects. In such cases, we pay attention to the truth value called confidence. For example, if an object in the same location is detected multiple times, it is considered the one with the highest confidence (Figure 5.8). If the accuracy rate is below a certain number, we will not accept that determination. Confidence value was accepted as 50% in our application.

```
Program Detection()
      // TensorFlow Lite model for object detection
      TF_MODEL_FILE = "detect.tflite";
      // COCO Dataset list
      TF_LABELS_FILE = "labelmap.txt";
      // Min detection confidence to track a detection
      MIN_CONFIDENCE = 0.5f;

      Detector detector; // imported from TensorFlow Lite
      detector =  TFLiteObjectDetectionModel.create(
                        TF_MODEL_FILE, TF_LABELS_FILE );
      Frame ( xValue, yValue, width, height );
      List result = detector.recognizeImage();
End
```

Figure 5.8. A pseudo code of detection procedures

To summarize, the acquired image is measured and transmitted to the object detection algorithm. Then detected acquired objects are obtained. The structure created for this obtained object can be seen in Figure 5.9.

```
Program Recognition()
      // A unique identifier for what has been recognized
      String objectID;
      // Display name for the recognition
      String objectTitle;
      // A score for how good the recognition is relative to others
      Float confidence;
      // The location of the recognized object
      RectF location; // RectF holds four float coordinates for a rectangle
End
```

Figure 5.9. A pseudo code of recognition structure

Then we take the found objects into border boxes. The center point, width and height data provided by the object recognition algorithm are used to create the border boxes. We print the object titles on the border boxes. This processed image starts to be displayed to the user as soon as the application is opened.

Then, the direction information of the object is tried to be determined. First of all, we try to perceive whether it is left, right or in the middle. For this, the objects in the area from the center point of the screen to the left of forty five units and to the right of forty five units are called middle. Objects to the left of this center area are called left, and objects to the right of it are called to the right. This structure is shown in Figure 5.10.

```
Program findLocation ()
            // locationX = object x coordinate
            if (locationX < centreOfScreen - 45) then
                  objectLocation = LEFT;
            elseIf (locationX > centreOfScreen + 45) then
                  objectLocation = RIGHT;
            else
                  objectLocation = MIDDLE;
            endIf
End
```

Figure 5.10. A pseudo code of finding a location

Improvements were made to ensure that the objects are close or far away so that the visually impaired person can reach them more easily. Since the distance of the object to the phone is based, the user can find the object more easily by moving the phone back and forth. The general logic is to determine whether the image of the detected object is near or far in proportion to the screen size. First of all, the ratio of the detected object to the screen has been reached, this process can be seen in Figure 5.11.

```
Program
      // each object shoul be processed separately
      List objectToRecognize = "laptop", "keyboard", "mouse",  ...

      // Calculating what percentage of the screen the object covers
      canvasArea = screenWidth * screenHeight;
      objectArea = objectWidth * objectHeight;
      objectPercentage = (objectArea / canvasArea) * 100;

      if (objectName.equals(objectToRecognize[index])) then
        if(objectPercentage >= 50) then //value changes according to the object
            return CameraActivity.NEAR;
        else
            return CameraActivity.FAR;
      endIf
End
```

Figure 5.11. A pseudo code of distance test

Then, measurements were made by changing the distance between the object and the phone, and a ratio was determined for each object. By comparing this ratio and the ratio of the object to the screen, it was determined whether it was close to the user.

After determining whether the detected object is close or not, audio information sharing with the user begins. Our application serves in Turkish, but the object detection algorithm works in English. Therefore, when voice information is received from the user, it should be translated into English and when information is given to the user, it should be translated into Turkish. Android's feature is used in our project for text-to-speech translation and voice recognition. These processes can be seen in Figure 5.12.

```
Program Contact()
      TextToSpeech TTs;
      SpeechRecognizer recognizer;
      String engine = "com.anroid.tts";
      TTs = new TextToSpeech(
                  TTs.setLanguage(language:"tr", country:"TR"),
                  TTS.speak("Welcome Message"),
                  engine
      );
End
```

Figure 5.12. A pseudo code of translation processes

The object detection project for the visually impaired has been completed with the processes described above. Object detection starts automatically when the application is started. Then the user is expected to press the discover or search button. If the Explore button is pressed, all defined objects are started to be transmitted audibly together with their location information. In this way, it is aimed for the visually impaired individual to learn about the objects around them. If the search button is pressed, the user is expected to say the name of the object they are looking for by voice.

Then, the searched object is searched among the detected objects and only the information of that object is transmitted to the user. In this way, it is aimed to facilitate the user to find a fixed object they are looking for. The final version of the application for the visually impaired can be seen on Figure 5.13.

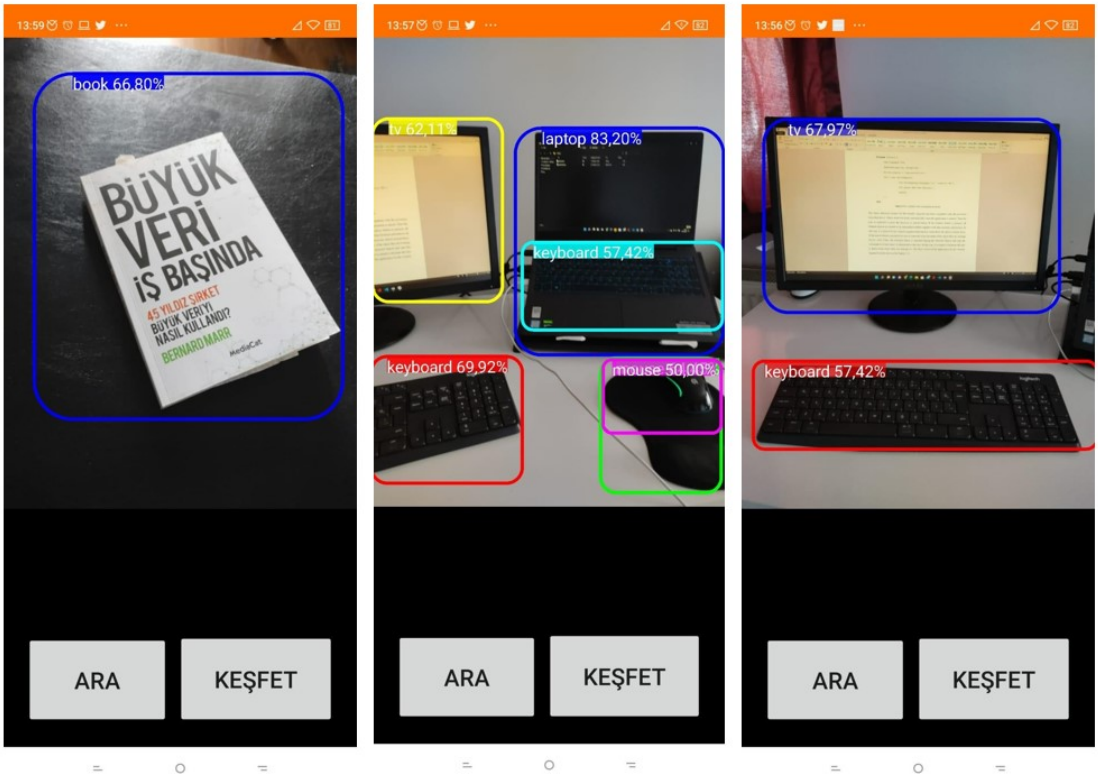


Figure 5.13. Visually impaired app screenshots

6. TEST AND RESULTS

This section highlights the tests and shows whether the results meet the system requirements. It has been tested whether the application developed for visually impaired individuals works as intended. Two main variables were used while conducting the tests. These are the distance and the amount of light. By changing these two variables, the accuracy of object detection was observed. Three tools were used during the tests, these are smart bulb, smart laser meter (range meter) and smartphone.

Since image processing is used in our application, the amount of light in the environment is important. The unit used to measure illuminance is lux. Some examples of illuminance provided under various conditions can be seen in Table 6.1.

Illuminance (lux)	Surfaces illuminated by
0.0001	Moonless, overcast night sky
0.002	Moonless clear night sky with airglow
0.05–0.3	Full moon on a clear night
3.4	Dark limit of civil twilight under a clear sky
20–50	Public areas with dark surroundings
50	Family living room lights (Australia, 1998)
80	Office building hallway/toilet lighting
100	Very dark overcast day
150	Train station platforms
320–500	Office lighting
400	Sunrise or sunset on a clear day
1000	Overcast day, typical TV studio lighting
10,000–25,000	Full daylight (not direct sun)
32,000–100,000	Direct sunlight

Table 6.1. The example of illuminance values in different conditions [17]

The smart bulb provides the possibility to choose the amount of light and the color of the light. The smart laser meter was used to measure distance with the help of a laser. We used a mobile application to measure the amount of lux when the sun goes down or when the brightness is changed with a smart bulb. The photograph of the mobile device on which this application is running and the smart laser meter mentioned above can be seen in Figure 6.1.

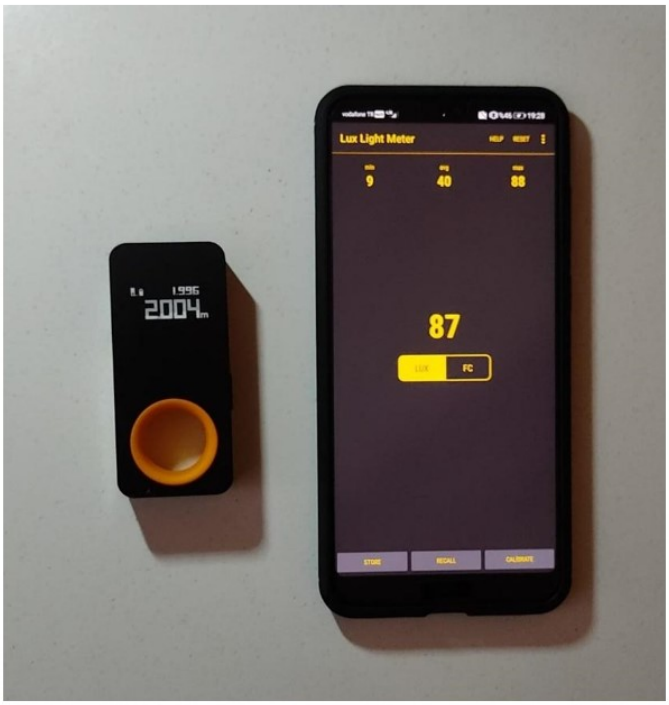
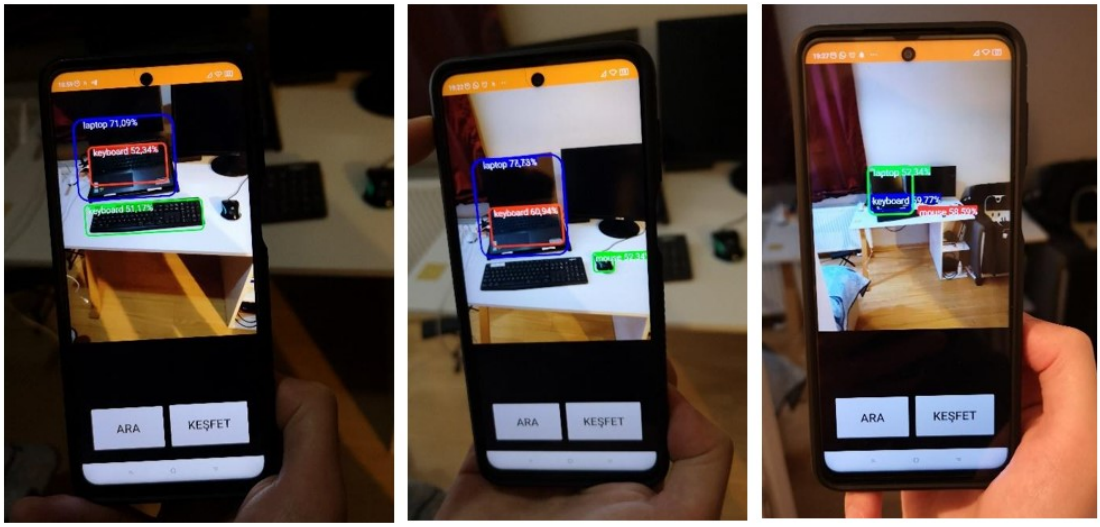


Figure 6.1. The photo of light meter and distance meter

As mentioned above, an average of 350 different measurements were made at different distances and different illuminance values. To be fair, the same objects were used during the measurements. Data from these measurements are clustered by condition, and the calculated average data is shown below using a table and graph.

In the tests, we observed that our object recognition application has a high level of accuracy even at low illuminance levels. No errors were observed in the voice prompt message when the application was started. No significant problems observed in the functions of the buttons and voice responses. Depending on the size of the detected object, whether it is on the left or right, whether it is far or near, is detected and transmitted to the user without delay.



(a) (b) (c)

Figure 6.2. The test photos in different conditions; (a) low illuminance and average distance; (b) average illuminance and distance; (c) high illuminance and distance

Integrated devices are the first of the situations that cause inaccurate measurements. Although the laptop device is also detected as a single object, the keyboard of the laptop is also detected as a separate object. In terms of the application, it does not seem like a problem to be fixed since the image is also the keyboard. One of the objects displayed during the test is the laptop. In close range object detection tests, although there are three objects in the image, the application detects four objects on average. The mentioned results are indicated with a bold line in Table 6.2.

Number of Objects	Number of Detected Objects	Illuminance (Lux)	Distance (cm)
3	2.5	15	50
3	4.2	15	60
3	3.4	15	80
3	2.8	15	100
3	2.3	15	120
3	1.9	15	150
3	2.6	40	50
3	4	40	60
3	3.4	40	80
3	2.9	40	100
3	2.4	40	120
3	2	40	150

Table 6.2. The table containing the average values of tests performed in low illuminance

In the tests, it was observed that the application had difficulty in detecting small-sized objects such as mice at distances of 120cm and more. In order to be suitable for daily situations, measurements were made at a distance between 50cm and 150cm and at an illuminance between 15lux and 180lux.

Number of Objects	Number of Detected Objects	Illuminance (Lux)	Distance (cm)
3	2.6	4500	50
3	4	4500	60
3	3.6	4500	80
3	3.2	4500	100
3	2.4	4500	120
3	2.3	4500	150

Table 6.3. The table containing the average values of tests performed in sunlight (not direct)

Since sunlight has a very high illuminance value, it was also observed and the data obtained are shown in Table 6.3. Despite the very high illuminance of daylight, it does not provide a clear improvement in the detection rate. As shown in figure 6.3, object detection varies according to distance. Maximum efficiency is obtained from an average distance of eighty centimeters.

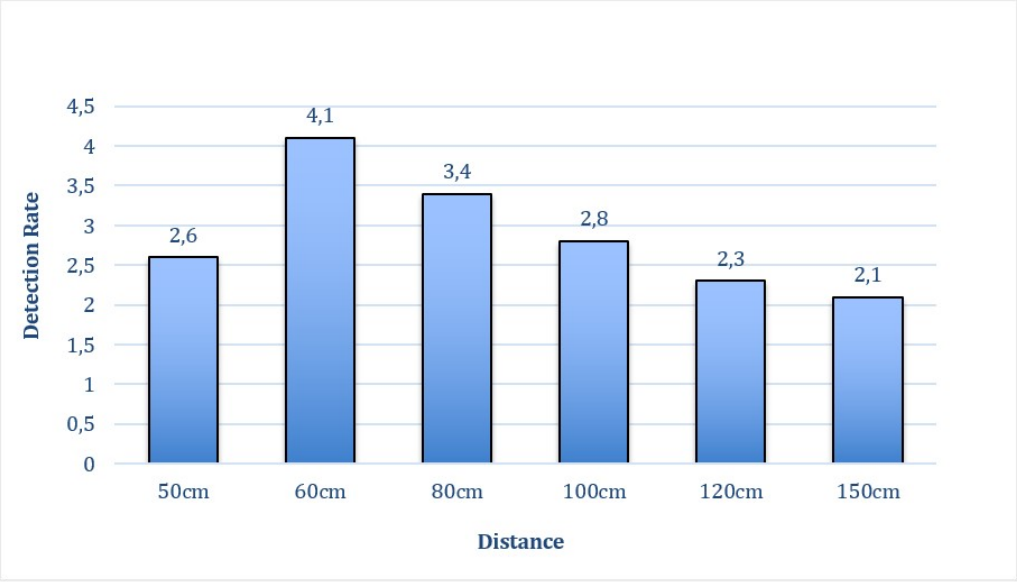


Figure 6.3. The graph showing detection rate varying with distance

7. CONCLUSION FUTURE WORK

This section highlights the conclusion and future work. It will explain why the development was made, how the results were obtained, in which aspects it was tested and how it could be further improved.

It is important to use rapidly developing technology for disabled individuals. In this study, we developed an application to help visually impaired individuals. The purpose of the application is to develop a mobile application that detects objects using image processing technology. Thanks to the developed application, visually impaired individuals will be able to get support by using the application when they are looking for any object (if available in our dataset).

It has been observed that the developed application performs highly accurate object detection. While testing the application, the illuminance value of the environment and the distance of the mobile device to the objects were taken as reference. According to the tests performed at different illuminance values, it has been observed that the application can detect objects with 95% accuracy, even in low light such as 15lux. In the tests performed at different distance values, we observed that high accuracy results were obtained at a distance between 50cm and 100cm but the accuracy value decreased to 63% at a distance of 120cm and higher.

7.1. Future Work

With future developments, the application for the visually impaired can be made more effective and usable. First of all, unlike desktop computers, object detection algorithms used for mobile devices have a lower accuracy rate. By making improvements in these algorithms, more accurate object detection can be achieved. In addition, our dataset currently consists of eighty objects. By expanding this data set, more objects can be detected. In this way, the application can be transformed into a better assistive technology for visually impaired individuals.

Bibliography

[1] Wewalk. (2021), [Online]. Available: https://wewalk.io/en/?force_switch=1.

[2] M. N. Bonner, J. T. Brudvik, G. D. Abowd, and W. K. Edwards, "No-look notes: Accessible eyes-free multi-touch text entry," pp. 409–426, 2010.

[3] B. Frey, C. Southern, and M. Romero, "Brailletouch: Mobile texting for the visually impaired," pp. 19–25, 2011.

[4] E. Foulke, "Braille," pp. 231–246, 2013.

[5] S. Tosun and E. Karaarslan, "Real-time object detection application for visually impaired people: Third eye," pp. 1–6, 2018.

[6] S. A. Paneels, D. Varenne, J. R. Blum, and J. R. Cooperstock, "The walking straight mobile application: Helping the visually impaired avoid veering," 2013.

[7] S. Chaudhry and R. Chandra, "Design of a mobile face recognition system for visually impaired persons," *arXiv preprint arXiv:1502.00756*, 2015.

[8] S. K. Long, N. D. Karpinsky, H. Döner, and J. D. Still, "Using a mobile application to help visually impaired individuals explore the outdoors," pp. 213–223, 2016.

[9] A. S. Shaik, G. Hossain, and M. Yeasin, "Design, development and performance evaluation of reconfigured mobile android phone for people who are blind or visually impaired," pp. 159–166, 2010.

[10] S. Nayak and C. Chandrakala, "Assistive mobile application for visually impaired people," 2020.

[11] Android. (2021), [Online]. Available: https://www.android.com/.

[12] C. Vision. (2021), [Online]. Available: https://en.wikipedia.org/wiki/Computer_vision.

[13] Y. Algorithm. (2021), [Online]. Available: https://www.v7labs.com/blog/yolo-object-detection.

[14] ——, (2021), [Online]. Available: https://appsilon.com/object-detection-yolo-algorithm/.

[15] OpenCV. (2021), [Online]. Available: https://opencv.org/.

[16] TensorFlow. (2021), [Online]. Available: https://www.tensorflow.org.

[17] Wikipedia. (2021), [Online]. Available: https://en.wikipedia.org/wiki/Lux.