TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Alish Najafzade 182466IVSM

AI for better integration of people with disabilities (Visual impairment)

Master's Thesis

Supervisor
Sadok Ben Yahia
Ph.D

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Alish Najafzade 182466IVSM

AI puuetega inimeste paremaks integreerimiseks (nägemispuue)

Magistritöö

Juhendaja
Sadok Ben Yahia
Ph.D

Tallinn 2020

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author:	Alish Najafzade
Date:	03.08.2020

Abstract

Considering the capabilities that mobile devices offer, we asked ourselves the following question: Why is it important to help visually impaired people and what can be done to help them? Mobile devices can provide tools for improving the lifestyle of visually impaired users. This study, tries to develop an android program consisting of different modules and objectives in order to improve the quality of life of visually impaired people.

The program developed consists of 7 main modules and aims to help the visually impaired person to perceive the location and intensity of the light source, analyze other people's age, gender and facial expressions, translate text into sound, recognize color and distinguish things like objects and money.

Keywords - visually impaired; mobile App; blind; android platform; object recognition; currency recognition; color information; facial expression; blind face difficulty;

Annotatsioon

Arvestades mobiilseadmete pakutavaid võimalusi, esitasime endale järgmise küsimuse: miks on oluline aidata nägemispuudega inimesi ja mida saaks nende abistamiseks teha? Mobiilsed seadmed võivad pakkuda vahendeid nägemispuudega kasutajate elustiili parandamiseks. Selle uuringu eesmärk on nägemispuudega inimeste elukvaliteedi parandamiseks välja töötada android-programm, mis koosneb erinevatest moodulitest ja eesmärkidest.

Välja töötatud programm koosneb 7 põhimoodulist ja selle eesmärk on aidata nägemispuudega inimesel tajuda valgusallika asukohta ja intensiivsust, analüüsida teiste inimeste näoilmeid, tõlkida teksti heli, ära tunda värvi ja eristada asju, näiteks esemeid ja raha.

Märksõnad - mobiilirakendus; nägemispuudega; pime; androidi platvorm; objektide äratundmine; valuuta tunnustamine; teave värvi kohta; näoilme; pimeda näo raskused;

List of abbreviations and terms

AI	Artificial Intelligence
API	Application Programming Interface
EOA	Electronic Orientation Aids
ETA	Electronic Travel Aids
GPS	Global Positioning System
GUI	Graphical User Interface
ICD	International Classification of Diseases
ICF	International Classification of Functioning
KNN	k-nearest neighbors
ML	Machine Learning
MSE	Mean Squared Error
PLD	Position Locator Devices
PWD	People with Disabilities
SDK	Software Development Kit
VI	Visually Impaired
WHO	World Health Organization

Table of Contents

List of Figures

List of Tables

1. Introduction

According to the figures of the World Health Organization (WHO), there are about one billion disabled people in the world. It means that approximately 15 percent of the world's population is directly or indirectly facing some form of obstacle in their lives [1]. We do not encounter a large number of disabled people in the streets and in our daily lives, not because the disabled people do not exist, but because it is hard for these people to enjoy the outdoors without assistance. According to the studies carried out throughout the world, disabled people can be reintroduced into society by means of services provided.

Today's technology changed drastically due to the introduction of Artificial intelligence (AI). It can automate what people can do, such as recognizing people in photographs, videos, understanding speech, bringing data into an understandable format, and translating between languages. We can also use the power of artificial intelligence to make life easier for people with disabilities. There are many ways to do this and a lot of work has been done in this field. Improving mobility, communication, safety, navigation, and independence are only a few of them.

In International Classification of Functioning, Disability, and Health (ICF), there are three types of disability [2].

- Impairment - Loss of limb, vision loss, and memory loss;
- Activity limitation - walking, difficulty seeing, problem-solving, hearing;
- Participation restrictions - restrictions in normal daily activities such as social activities, working;

Vision is an important sense that we make use of in performing many activities in daily life. We decide on the time, direction, and speed of the activity by interpreting the stimuli we receive from our environment with vision. Our personal experience and research evidence show us that vision plays an important role in allowing one to see the environment; as well as the sensory system. Individuals experiencing the loss of vision cannot get enough information from their environment. The devices related to vision impairment are very expensive and blind people usually cannot afford it. Technology has made available other aids for blind people, including talking calculators, speech- time compressors, computer terminals with speech output, eSight, IrisVision, paperless braille machine, etc. It worth noting important to note that these devices and systems are very expensive for the user. The statistics of WHO estimates that more than 285 million people worldwide suffer from visual impairment (VI) with 39 million blind and 246 million people with low vision [3]. Despite their price, all the mentioned aids, may they be online services, appliances, systems, or digital devices, allow safer mobility for the disabled, providing solutions for conducting their daily tasks.

Over the last decade, mobile applications have advanced from senseless programs and mindless games to services that have a huge effect on individuals' life. Approachability shifted to the forefront a few years ago, when giants such as Microsoft and Apple made it a priority. Mobile applications have made life easier for many people living with blindness or visual impairment.

The International Classification of Diseases (ICD) divides visual impairment into two groups [4]:

1. Distance vision impairment:

Type	**Presenting visual acuity**
Mild	worse than 6/12
Moderate	worse than 6/18
Severe	worse than 6/60
Blindness	worse than 3/60

Table 1.1: Distance vision impairment types

Presenting visual acuity worse than 6/12 means, able to see at 6 meters, what the normally sighted person sees at 12 meters. Other lines also follow this rule.

2. Near vision impairment:

Presenting near visual acuity worse than N6 with existing correction.

There are many ways to improve visually impaired people's integration into society:

- Hardware and Visual impairment

  Considering technology from the point of view of people with visual impairment, you can think of two common categories:

  - General technology: such as computers, smartphones, GPS devices, etc.
  - Assistive technology: specially designed to assist visually impaired people or other people with disabilities to assist them. It contains screen magnifiers, screen readers, video magnifiers, and other devices for visually impaired computer users.

- AI and Visual impairment

  One of the biggest blessings of artificial intelligence is that it helps us make our lives easier. The borders in this area are growing day by day and humanity is taking a new step every day in cooperation with artificial intelligence.

  In this context, we see the breakthroughs of giant companies like Microsoft, Google, Huawei and etc. And they seem to be very interested in this area and developing a number of significant projects for the benefit of artificial intelligence for the disabled people.

  There are some technologies that have been developed for blind or visually impaired persons [5].

  - Electronic Travel Aids (ETA):

    These devices gather information from the surrounding environment and transfer it to the user through sensor camera, sonar, or laser sensors;
  - Electronic Orientation Aids (EOA):

    These devices give pedestrians directions;
  - Position Locator Devices (PLD):

    These are devices that determine the precise position of the holder such as devices that use GPS technology;

Why is it important to help people with visual disabilities and what can be done to help them?

- It is very difficult for people with disabilities to be in a social environment. They want to acquaint who the people in their neighborhood, how and how old they look, their facial expressions, and what is their gender.

- Vision allows us to get lots of data from our surroundings. The data for most of the digital aids for visually impaired users is collected through the use of cameras, sensors, laser scanner and sonar system. The information on the user's surroundings is then transmitted by audio, or via tactile format, sometimes the system will use both ways of communicating with the user. Of course, it is not possible to use a device to process all the data that the eye can take up. Nevertheless, if we take as much data as we can and make it somehow understandable, that would make life easier for the visually impaired.

Voice-over features on smartphones are a game changer. They grant the same access to the digital world to non-sighted users as everyone else has. Moreover, there seems to be a shift in consciousness; more service providers are providing an inclusive service, catering for blind people: Netflix has included an audio description technology on its platform, whilst Windows has a more accessible version. And, everywhere we look, we are redefining the definition of “to look”. Digital applications are allowing the visually impaired to become more independent. These help redress the balance, where people are only seen as users, notwithstanding the way in which they “see” things.

The literature shows that very few studies have been conducted on the use of mobile applications of people with visual impairments. One of the most important works in this field was done by Nora Griffin-Shirley, Devender R. Banda, Paul M. Ajuwon, Jongpil Cheon, Jaehoon Lee, Hye Ran Park, and Sanpalei N. Lyngdoh [6]. In this study, the authors investigate the use of mobile applications by visually impaired individuals. A total of 259 participants with visual impairments participated in the survey. Research shows that people with visual impairments use specially designed apps to perform daily activities. Also, people with visual impairment are satisfied with special mobile apps and want to see improvements in current apps and want new ones. As a result, participants rated apps for people with visual impairments as useful and accessible tools.

As can be seen from Table 1.2 which was created on the basis of data from the specified source [6], 95.4% of the participants use smartphones. When we look at the most used applications, we see that the tools for visual impairment come in second place. The most used ones in special applications on mobile devices are related to visual identification.

Variable	**%**
Mobile devices in current use	
Smartphone	95.4%
Tablet	40.5%
Both smartphone and tablet	37.1%
Other	19.7%
Free or paid apps	
Only free	9.7%
Only paid	0.0%
Both free and paid	90.3%
Most frequently used apps	
E-mail	24.5%
Tools for visual impairment	12.5%
Social networking	11.3%
Entertainment	10.5%
Leisure reading	8.9%
Special apps in mobile device	
Visual identification	29.4%
Screen reading and writing	20.4%
GPS navigation	13.5%
Leisure reading	13.0%
News	7.7%
Special apps wanted to purchase	
Screen reading and writing	37.1%
GPS navigation	22.8%
Visual identification	17.4%
Leasure reading	3.6%
Education	2.4%

Table 1.2: Use of mobile devices and apps

1.1 Research Goal

Thanks to rapid growth in technology, there are today many new available AI systems that can be designed to service the visually impaired. The goal of this thesis is to work towards offering a digital aid for the blind community. They face many problems for carrying out daily activities such as reading, walking, identifying the presence of objects, persons, light, and also emotion recognition. They do not have proper devices to learn and read a text, see the obstacles in their path, which person they are communicating with, etc. This is why this research proposes an android application using Machine Learning techniques for blind or visually impaired people. The goal is to provide an android mobile application that can recognize a person from his/her preloaded picture, make an estimate of age, gender and emotion, recognize and classify objects, distinguish banknotes, perceive light source and intensity. The application performs most of these features through the use of the mobile camera, the camera detects the things the objects, emotions, or light and it will then notify the blind or visually impaired person.

This research concentrates on improving life quality for people with visual disabilities via technological solutions. By doing this, they will be able to improve their quality of life and a little closer to the advantages that people with no disabilities have. Usually, they cannot recognize the people around them, and it significantly affects their communication with society. Furthermore, they cannot recognize the feelings and emotions of the people whom they are talking to. As a result of these problems, visually impaired people become more introverted and they are not willing to spend more time in society [7]. In this research, we try to tackle this problem by developing an android mobile application.

First of all, we try to make the program predict directly; simply by opening the application, the camera will record what it is pointed at. This means that there is no need to actually take a picture manually in order to receive feedback. In this way, there is no need to press any button, the user can instantly receive feedback.

The second goal is that the program is compatible with external tools and/or with finger gestures. As an external device, it works with the connected headset or a third-party camera.

Another functionality is the use of haptic and sound feedback for when any event is changed. For example, if the device camera detects any object, it then gives us haptic and/or sound feedback.

Furthermore, the application detects objects, colors, currencies, light sources, and age, gender, emotion, and faces of people when the camera is pointing at them. It will then give audible feedback about who he or she is and what are the person's emotions during the conversation.

2. Background and Related Work

The following chapter provides an overview of the background concepts used as a basis for this thesis. First, it will examine the classification and the classifying methods for the classification of money and objects that are used in the project. Following, there will be an overview of studies on face recognition, facial expression, and color detections. Finally, there will be an overview of the available applications and their outstanding features.

2.1 Classification

The concept of classification is simply to distribute the data between the various classes defined on a data set. Classification algorithms learn this distribution form from the given training set and then try to classify test data correctly for which the class is not specified comes. The values indicating these classes given on the dataset are called labels and are used to determine the class of the data during both training and testing [8].

The Confusion Matrix (Table 2.1) exists to measure the performance of machine learning classification problems where the output is two or more. It is a table with 4 different combinations with predicted and real values [8].

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 2.1: Confusion Matrix

$$accyracy = \frac{TN+TP}{TP+FP+TN+FN}$$

The formula above is used to calculate accuracy. The latter is a widely used metric to measure the success of a model [8]

$$precision = \frac{TP}{TP+FP}$$

The formula above is used to calculate precision and shows how accurately we estimate from all classes. It should be as high as possible [8]

$$recall = \frac{TP}{TP+FN}$$

The formula above is used to calculate recall and shows how accurately we estimate from all positive classes. A high value of recall is always though after [8].

2.2 ML kit

ML Kit is a mobile SDK that brings Google's machine learning skills to Android and IOS apps in a powerful yet easy-to-use package. One can implement the functionality needed in just a few lines of code. It is very easy to use and enough to know a little about model optimization or neural networks. Moreover, ML Kit provides APIs that help use special TensorFlow Lite models in mobile applications and makes it easy to apply ML techniques in apps using Google's ML technologies in one single SDK. Examples of these are Tensor Flow Lite, Google Cloud Vision API and Android Neural Network API [9].

The features of ML kit are as follows:

- Text Recognition;
- Face Detection;
- Bar-code Scanning;
- Image Labelling;
- Object Detection and Tracking;
- Language Identification;
- Translation;
- Smart Reply;
- Auto-ML model Inference;
- Custom model Inference;

2.3 TensorFlow

TensorFlow is one of the most used libraries in the field of deep learning which has been developed by Google for many years, was first announced in November 2015 and it enables the development of deep learning supported artificial intelligence applications [10].

TensorFlow is an open-source deep-learning library. Thanks to its flexible structure, it allows regardless of a platform to deploy calculations, using one or more CPUs, GPUs, with a single API. Based on Python, this framework supports many languages such as C ++, Java, C#, Javascript, and R as well as Python [10].

2.4 Currency classification and detection

There are only a few efforts to define banknotes to help visually impaired people. Some initiatives use Computer Vision and “scale-invariant features” and some use Machine Learning. Scale-invariant versions have 70% accuracy, ML versions have a bit higher accuracy. On the ML side, we have to mention Microsoft’s research and the programmer Kshitiz Rimal:

Microsoft announced their version at an AI summit, it would not be wrong to say that this application is one of the best attempts for the visually impaired. Basically, they use Keras and transfer learning, they used a mobile compatible pre-trained net ”MobileNet” to run detection on the device and used 250 images per banknote. They achieved 85% accuracy [11].

Another version is developed by Kshitiz and he took 14,000 images in total and got 93% accuracy. It started with VGG19 and Keras for neural network and “Native Reach” for application. The newest version uses Tensorflow with MobileNetV2 and native applications on each platform. The source code of the application is available in Github [12]. Unfortunately, the dataset used in the application is not available as of July 2020.

2.5 Color Detection

When we scrutiny search the literature, we see that there are many color detectors. However, these color detectors are usually made for color blind people and are not very suitable for blind people. Most of the time, these devices are heavy and hard to carry, and there are some difficulties in predicting color with the phone camera. For example, problems such as low-quality camera, poorly lit environment, light reflected on the camera, etc badly affect the correct perception of color. Nevertheless, there are a few programs that try to solve this problem with a phone camera, but only a few of them are useful.

2.6 Face recognition

Face recognition applications are used quite frequently today and are important. Many new software products are designed using face recognition applications, and this is needed in social security applications and inquiries. Despite the fact that facial recognition is common and is an area that has been studied, the desired results have not been achieved yet. Because of the amount of light that will affect a person's face, perspective, various obstacles (glasses, scarf, hat, etc.), time, aging, multiple expressions, emotions determined on the face cause difficulties in recognizing people. In a successful facial recognition model, despite the face's multi-dimensional and complex structure, the face should be clearly identified among the objects in the photographs taken, and the right person should be able to be predicted even under changing conditions such as light angle, color differences, and expression changes on the face.

One of the best studies on face recognition is Tolga Ensar and Melike Gunay's Comparison of face recognition algorithms. In this study, they analyzed the algorithms used for facial recognition and compare the performance of the algorithms. The methods analyzed are implemented in k-nearest neighbors, Naive Bayes, eigenfaces, principle component analysis (PCA), and k-tools ORL facial dataset. As a result of the analysis, the KNN algorithm and the eigenfaces algorithm are the most successful, and Naive Bayes has the worst performance result [13].

2.7 Aipoly

Moreover, one of the most important applications implemented to help the visually impaired is Aipoly. The latter is a mobile application that uses artificial intelligence and aims to help visually impaired people. The application captures the image and instantly converts it into words and sentences using artificial intelligence with very little delay. Doing so, it allows the program to instantly tell what it sees during video recording, unlike other programs. The main problem this program has is that some features are not accessible for free. Nevertheless, the program is a useful and well thought out initiative for blind people [14].

Aipoy's main features:

- Finds Common Objects (Free);
- Recognizes Food (subscription required);
- Reads Text (subscription required);
- Identifies plants and animals (subscription required);

- Recognizes the US Dollar (subscription required);
- Knows Colors (Free);

2.8 Seeing AI

The next application is “Seeing AI” and it is one of Microsoft’s best ventures. It offers very useful features for the visually impaired. Seeing AI is an intelligent camera application that can provide information about who and what is around you. That is particularly useful for someone with a visual impairment. With the application that turns the camera into a magic eye, you can do the things you can do in multiple applications in one single application. In order for it to work, users must hold up their phones and hear the description, similar to when asking a friend. You can recognize products, people, and other objects like currency. The most crucial part of the program is the scene description: it explains the whole scene with objects in a natural spoken language. The other interesting aspect is that it can easily read handwriting. The only downside to the program is that it needs cloud service power to run and cannot be used offline. The application can be downloaded from the AppStore and it is only running on IOS devices. The version is not available for Windows and Android mobile devices [15].

Seeing AI’s main features:

- Reads short text;
- Scans barcodes to identify products;
- Recognizes and describes people around you;
- Describe the scene;
- Identifies currency bills;
- Generates an audible tone corresponding to the brightness;
- Knows Colors;
- Reads handwritten text;

There are many other devices that are helpful for blind or visually impaired people. However, they are very expensive, meaning that every blind or visually impaired person cannot afford it. Due to the increase in the usage of smartphones, there have been many solutions developed for smartphones for different issues. People don’t have to carry any additional equipment; they have everything on their Smartphones:

this brings ease in their life. The application developed for android in this project helps the visually impaired people so as to avoid buying any new devices or worry about any wearables. All users have to have is their Smartphone and the application installed on it. The application uses the mobile's camera and can help the visually impaired person through a series of features. The device detects the object, person, emotion, light, color, and speaks about it. This way, the visually impaired person could know about any obstacle or object they are finding in their around, etc.

3. Contribution

3.1 Software Details

For the development of the application, Android OS version 9&10, Kotlin, and Python (Django) languages were used for app development. A built-in Android SDK was used for light detection, color detection, and text to speech. Python was used for face recognition, emotion, age, and gender prediction. For other features such as image labeling, and currency detection, ML kit was used. The android version fully supports the ML kit features. The API selections of the ML Kit run in the cloud or on the device, depending on the chosen features, and the APIs on the device can process the data quickly and work even without a network connection. The cloud-based API leverages the power of Google Cloud Platform's machine learning technology to deliver an even higher level of accuracy. Offline features were prioritized in order to make the project work fully offline. The source code of the mobile application is available in the following link: `https://bitbucket.org/Alish32/app-for-visual-impairment`

3.2 Implementation of tool

3.2.1 Architecture

It is possible to call Java from Kotlin and Kotlin from Java. The communication between these two different worlds is very smooth. The compatibility of both languages allowed us to develop new features of the existing Kotlin projects in Java. Since Java is a very popular programming language, it is easy to find libraries written for every need (3rd party jars). The codes written with Kotlin can use all of these libraries. For these above-mentioned reasons, Kotlin was the chosen language for the project's front-end side [16].

The Python Django web framework was used on the back-end. Django is a high-level web framework prepared for Python Programming Language and licensed under the BSD license. It distinguishes itself from other server software and frameworks with its simple installation and usage, detailed error report pages and brand new interface [17].

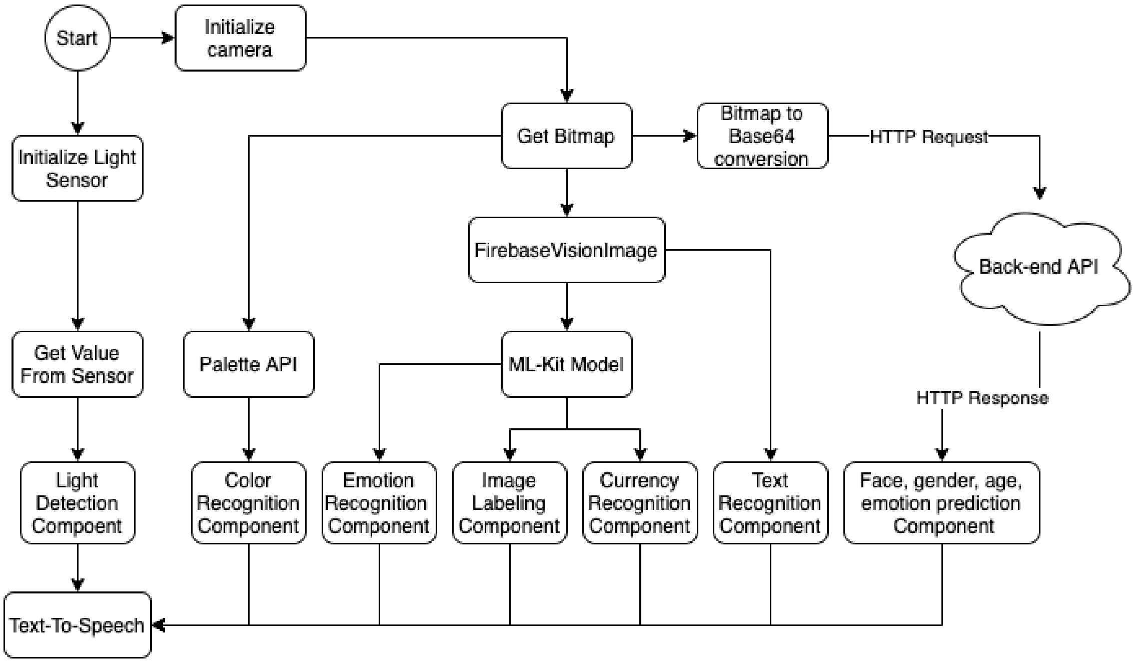

Start
Initialize camera
Initialize Light Sensor
Get Bitmap
Bitmap to Base64 conversion
HTTP Request
Back-end API
FirebaseVisionImage
Get Value From Sensor
Palette API
ML-Kit Model
HTTP Response
Light Detection Component
Color Recognition Component
Emotion Recognition Component
Image Labeling Component
Currency Recognition Component
Text Recognition Component
Face, gender, age, emotion prediction Component
Text-To-Speech

Figure 3.1: Application Flow (Android side)

As can be seen from Figure 3.1, when the program first starts, it is generally divided into two parts, while one part only uses the light sensor and the other part needs the camera. After the camera is initialized, a bitmap is taken from the image. The bitmap is then sent to the Palette Api (to find the dominant color) or FirebaseVisionImage class depending on the selected feature. If a feature using the ML kit model is selected, the output from FirebaseVisionImage is sent to the ML kit model, otherwise, the output is sent to the text recognition component. The only part that the application needs the internet is the face (name, age, gender, emotion) recognition function. To do this, the bitmap is converted to base64 format and the output is sent back for processing. All outputs at the end are sent to the TextToSpeech class and converted to sound.

3.2.2 Technologies used

Android Studio

Android Studio is the official programming tool for developing Android apps, with high-end features and recommended by Google. Android Studio was used in our project because of its gradle-based, flexible project building system, fast and feature-rich emulator, multiple APK output for different features and versions, code templates to help creating general application features and import sample code and GitHub integration, NDK support, Rich editor with drag-and-drop feature that facilitates screen designs, application performance, usability, test tools and frameworks that can be controlled in different versions, easy and secure APK signing, and most importantly, to add Google services to the application without the need for additional effort [18].

PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, especially for the Python language. Developed by the Czech company JetBrains. To taking advantage of an IDE and focuses on building a better code and not worrying about dependencies, Pycharm was chosen for backend development. [19].

ML-kit implementation

ML Kit is a mobile SDK that offers Google's machine learning expertise to Android and iOS apps in a powerful and easy-to-use package. The ML Kit provides an easy way to detect faces from various image types such as ByteBuffer, byte[], media.Image, Bitmap. First, we need to create a FirebaseVisionImage object from the image types and put it into the model as shown in Figure 3.2 [20].

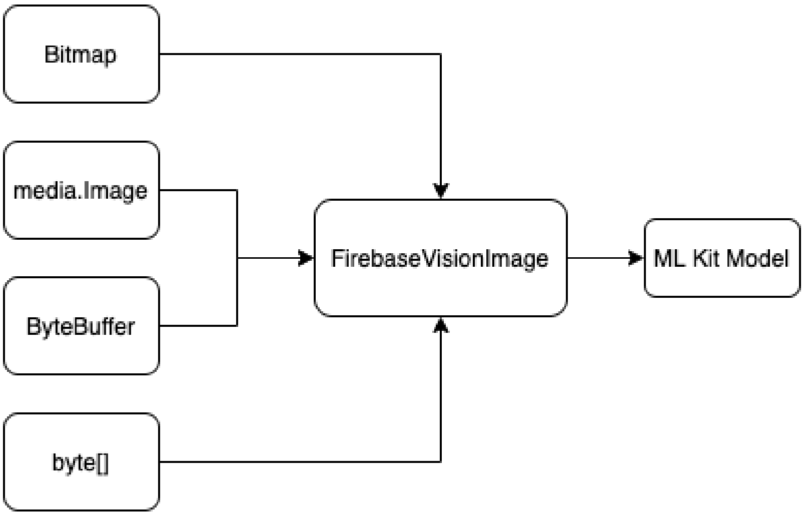


Figure 3.2: Creating FirebaseVisionImage object from different image types

In this project, Bitmap was used to create FirebaseVisionImage objects.

AutoML implementation

AutoML is a tool that allows creating custom models on Google Cloud Platform. A project was created in the Google Cloud Console to create the custom model. In this project, AutoMl was used to recognize euro banknotes. To train the model, it must have a minimum of 10 images per tag. At least 100 images for each tag were uploaded as suggested by AutoML to get more accurate results. Since this is an experimental model, 1 hour of free model training was used, enough for this case scenario. It is also possible to get more training time as the model grows. In the next sections how the model was used will be explained [21].

3.2.3 Features of the Application

The android application for blind or visually impaired people uses different features in order to provide them ease. As follows is the list of all the available features:

- Generates an audible tone corresponding to the brightness;
- Recognizes emotion;
- Extracts text from image;
- Knows Colors;
- Image Labeling;
- Identifies Euro bills;
- Recognizes and describes people;

Light Detection

The app has a user-friendly light detector, which comes in handy for a completely blind person. For the light detection feature, Android SDK is used. When the light sensor of the device detects a change in light intensity, the program is informed about the change and initiates a DeviceLightEvent event. The event gives information about the light intensity of the environment of the device. The app increases or decreases the intensity of the sound coming from the speaker according to the light intensity from the sensor. In this way, it is possible to easily estimate the amount of light in the environment. In addition, it is also useful for checking lighting conditions before taking photos or leaving the house.

Emotion Recognition

Facial emotions are a way humans have of communicating non-verbally. Nowadays, computers are improving the way of reading emotions. For offline emotion recognition, a machine learning kit is used. As mentioned on the firebase.google.com "With ML Kit's face detection API, one can detect faces in an image, identify key facial features, and get the contours of detected faces. With face detection, one can get the information needed to perform tasks like embellishing selfies and portraits or generating avatars from a user's photo [9]". For Emotion detection, FirebaseVisionImage was integrated to create Bitmap objects. If the application can do face detection successfully, the listener will get a list of FirebaseVisionFace objects. Each FirebaseVisionFace object represents a detected face with all the information about the detected face.

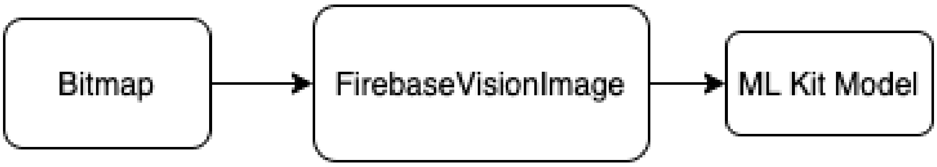


Figure 3.3: creating FirebaseVisionImage object from bitmap

val image = FirebaseVisionImage.fromBitmap(bitmap)

A FirebaseVisionImage creates an object from the bitmap which is one of the Image types (Figure 3.2) and passed it to the ML Kit model.

In the next step, the Face Detection model has been prepared. At the final stage, the picture will be transferred to the Face Detection model.

If the application can handle face detection successfully, the listener will get a list of FirebaseVisionFace objects. Each FirebaseVisionFace object represents a detected face with all the information about the detected face.

Thanks to smilingProbability, we find the possibility of smiling. We found the percentages below by trial and error.

```
for (visionFace in visions) {
            val smileProbability =
            visionFace.smilingProbability
            Log.d("aaa", visionFace.toString())
            if (smileProbability < 0.1)
                speakText("serious face")
            else if (smileProbability >= 0.1 &&
            smileProbability < 0.8)
                speakText("Person looks happy")
            else speakText("Person is smiling")
        }
```

More detailed emotion recognition method will be mentioned in the face recognition section. For this, python library is used and data is exchanged with the rest API.

Extract text from image

First, the picture retrieved from the camera was converted to bitmap. The bitmap that was received is sent to firebaseVision.onDeviceTextRecognizer (onDeviceTextRecognize is used to make the program work offline), and it extracted the text on the bitmap, and then the application took it and turned it back into sound. For text to speech, Android SDK is used, and it allows to convert the text into voice. Not only it allows to convert it, but it also allows to speak text in a variety of different languages. Android provides TextToSpeech class for this purpose.

Color Recognition

Android's Palette class was used to detect and define color from an image. Thanks to these classed class's getDominantColor method, it was possible to get the dominant color from the bitmap without using any additional algorithm.

First, the bitmap is taken from the image, and the width and height of the bitmap were divided into 3 and 5 sections accordingly and the middle part was taken. In this way, it is easier to find out what color in the middle of the screen.

Then, the bitmap sent to the Palette class, and the class returns a color in RGB format.

```
private ArrayList<ColorName> initColorList() {
    ArrayList<ColorName> colorList = new ArrayList<>();
    colorList.add(new ColorName(”Black”, 0x00, 0x00, 0x0
    colorList.add(new ColorName(”Blue”, 0x00, 0x00, 0xFF)
    .
    .
    .
    colorList.add(new ColorName(”Cyan”, 0x00, 0xFF, 0xFF))
    }
```

In order to be able to say what the dominant color returned above, compares it to the colors previously defined (140 colors were predefined for this purpose). For this, the getColorNameFromRgb method was written and found the color closest to the returned color and it finds its name from the colorList array and sends it to the speaker. The MSE formulation shown in equation 3.1 was used to find the closest color.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i}) \tag{3.1}$$

Image Labeling

For object detection, the system categorizes the things based on the machine learning kit. An object detection model can identify multiple objects within an image or a video stream. It can identify which objects are present and may also provide information about their position in the image. It Recognizes 80 different classes of objects. An object detection model is trained to detect the presence and location of multiple classes of objects. When sent to the ML Kit images, ML Kit returns, for each image, a list of up to five detected objects in the image. When detecting objects in video streams, each object has an ID that it can use to track the object between images. Optionally, it can also enable coarse object classification that labels objects with broad category descriptions.

Device-based API supports more than 400 tags and cloud-based API supports 10,000+ labels. We chose the device-based API considering the offline use in the project. It is worthy to mention the cloud-based version is free for the first 1000 uses per month.

Firstly, we created a FirebaseVisionImage object from the image from the camera. The image labeler runs faster when using a Bitmap so we used Bitmap for this purpose.

We pass the FirebaseVisionImage object to the FirebaseVisionImageLabeler's processImage method to tag objects in an image.

Next, we pass the image to the processImage() method. If the image labeling operation succeeds, then a list of FirebaseVisionImageLabel objects will be passed to the success listener. Each FirebaseVisionImageLabel object represents something that was labeled in the image. For every label, it will receive the label's information. And in the end, it turns these outputs into voice and sends them to the speaker.

Currency recognition

This service recognizes what denomination a euro banknote is (5, 10, 20, 50, 100, 200, 500). Google machine learning kit was used for the classification of banknotes. Based on the image recognition technique, the images of currency were saved in the application to train the model. Whenever the camera detects any currency it compares it to the images saved in the model and then it speaks about the currency. The purpose of the research is to help blind people, so we took care to create real-life situations such as holding banknotes in a hand, sometimes folded, and sometimes covered in some parts. To do this, an average of 100 pictures of each banknote that we wanted to recognize were taken and subjected the model to a 1-hour training period with the Google ML Kit. An example of the images taken is shown in Figure 3.4.

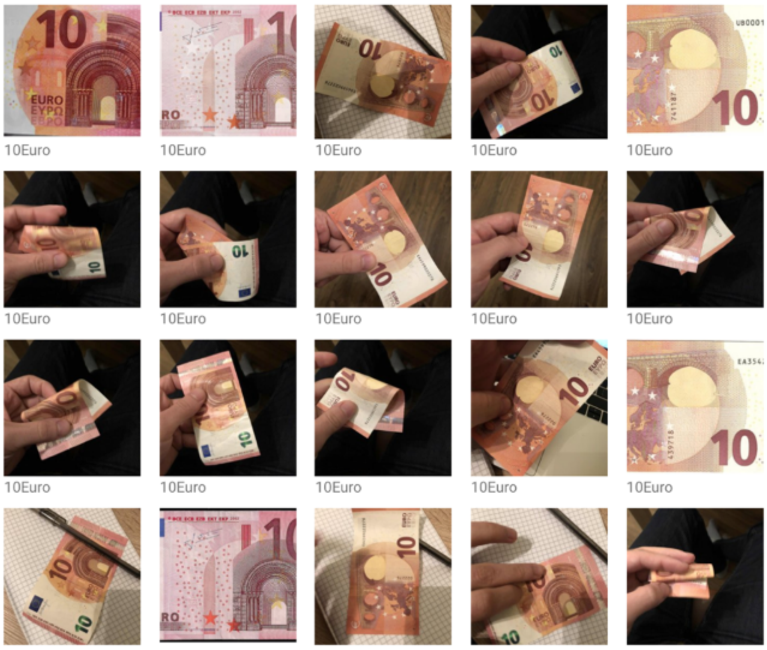


Figure 3.4: 10 Euro Dataset

The picture is taken from the camera, converted into the bitmap, and sent to the currencyOnDeviceAutoMLImageLabeler method, and if there is no empty value, then the speaker says the nominal value of the banknote. The score threshold of 0.5 was selected, which is the minimum confidence that the model must have to assign a label to an image. If the application can do currency detection successfully, the listener will get a list of currencyOnDeviceAutoMLImageLabeler objects. If the returned data is equal to any data in currencyMap (It was predefined to match the currency name and the label) that name is thus converting to voice and sending it to the speaker.

Face Identification

Recognizing face, age, gender, and facial expression are natural abilities in humans. Nowadays, computers are improving the way of reading biometric data. Face Recognition can be used to improve access and security like the latest Smartphones.

Face recognition solutions are still doing R&D and their free solutions are limited, in a way that users will provide them with more and more data. Firstly, we wanted to use a phone facial recognition system. But phone facial recognition used for face ID is using specialized hardware and libraries which are not available to third-party Apps. Face ID is not only restricted to other apps but furthermore, it is restricted to recognize one face only.

Since there are not many free SDKs or APIs on the Android for face recognition, we decided to develop our own API. For this, we used Python programming language, considering the richness of the libraries used in facial recognition.

Three different libraries were used at the Python side. These are libraries for predicting emotion, age and gender, face recognition respectively. First of all, the face picture taken from the phone camera on Android is converted to base64 format. After the conversion, this output and the name entered by the user are sent to our API working at the back-end. After the input is accepted, it is decoded back to the picture and stored. Pictures taken to register faces are always saved in the "known" folder in the format "person_name + .jpg". Images are taken to make recognition and predictions are stored in almost the same way. The camera takes a picture every 3 seconds and sends it to the back-end. The sending procedure takes place in the same way as with the previous one (Figure 3.5) The only difference is that the "unknown" folder is completely emptied before saving and the picture is saved as "check + .jpg". The reason we do this is that the library we use to recognize faces compares two folders and the images inside it to each other, and returns the name of the matching image in the "known" folder.

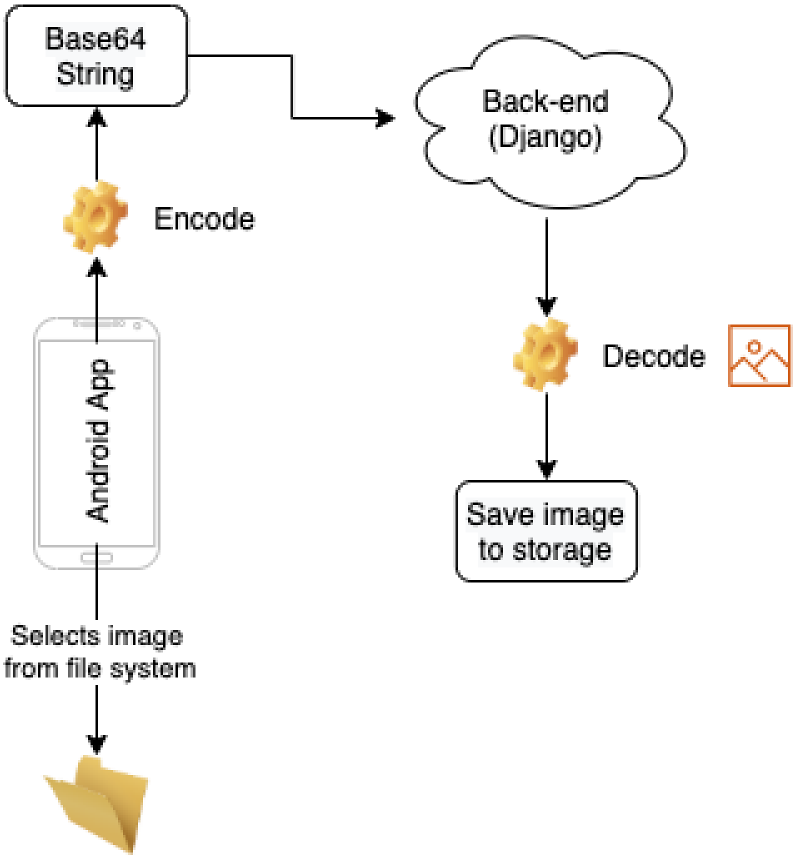


Figure 3.5: Sending Image to server

The Face Recognition [22] library was used for face recognition. This library has 99.38 accuracy on Labeled Faces in the Wild [23]. The method recognize_person() takes the picture previously saved in the unknown folder and compares it to all images in the known folder. Returns "No face detected" if it cannot find the matching image, and returns the name of the matching image if found.

The same Python OpenCV project [24] was used for both age and gender estimation [25]. The predicted gender may be 'Male' or 'Female' and the estimated age is divided into the following ranges - [0 - 2], [4 - 6], [8 - 12], [15 - 20], [25-32], [38- 43], [48-53], [60-100]. It is very difficult to predict the exact age from a single image due to factors such as lighting, makeup, facial expressions, and etc. And so they made it a classification problem rather than a regression. The dataset we used for age and gender estimation is open-source and accessible from here [26]. This dataset contains 26580 photographs taken in eight different age ranges in various environments (lightening, noise, pose). The model used was trained on this data set. As the method for face recognition, the gender_and_age() method takes the picture saved in the previously unknown folder and makes an estimate of age and gender and returns the prediction.

We used code snippets in this [27] article to detect emotions. Emotions in the article are divided into 7 classes: 'Angry': 0, 'Disgust': 1, 'Fear': 2, 'Happy': 3, 'Neutral': 4, 'Sad': 5, 'Surprise': 6. The trained model was retrieved from this [28] source. Like face recognition and gender and age prediction, the recognition_emotion() function takes the picture pre-saved in the unknown folder and returns the facial expression class it predicted from the picture.

3.3 System Analysis and Design

Requirements analysis was conducted where system analysis, feasibility study, planning, modeling, scheduling, and design of the project were discussed. For developing any project, the major problem is requirement gathering, here, it also focused on functional requirements. The procedure for gathering requirements has its own defined procedure according to the complexity of the application. To define the project schedule and processing, different models and techniques were used. Requirements can be defined as a condition or capability that must be processed by a product or an application.

Requirement Analysis

Requirements analysis is the process of planning, forecasting, and studying the overall former needs of the application requirements.

Functional Requirements

Functional requirements are the requirements that should be provided by an application. It is defined as a service statement. Functional requirements tell the programmer how an application should behave in different situations. The functional requirements for our application are as follows:

- REQ-F1: the user should be able to swipe between screens;
- REQ-F2: haptic feedback when changing screens;
- REQ-F3: detect the objects (up to five at the same time) and speak out their names;
- REQ-F4: compare the image with the images stored in the application;
- REQ-F5: convert the given text to speech;
- REQ-F6: produce an audible tone based on light sensitivity;
- REQ-F7: guessing the face expression of the person in front of the camera;
- REQ-F8: guessing the age of the person in front of the camera;
- REQ-F8: guessing the gender of the person in front of the camera;
- REQ-F9: determining the value of the Euro bill in front of the camera;

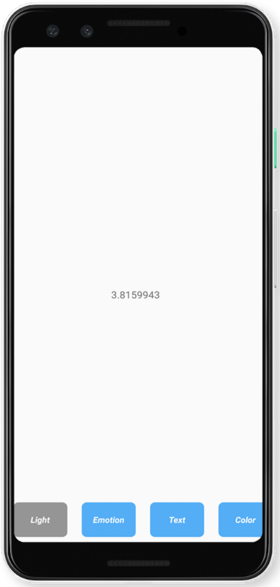
4. Functionality of Tool

The application was implemented on android. A visually impaired persona must have the android phone with the application installed in it. The application has the features of Light detection, Emotion detection, Text to Speech, Face Recognition, Color Detection, Currency Detection. The person opens any feature of the application, the application has the access to camera and it can detect person, object, emotion, currency and then speak about it so the visually impaired person can hear it.

4.1 Software GUI

The GUI of the application is very interactive to the users. The application incorporates 7 main features. The user opens the application and presses the button of the feature which he/she required. The name of the pages is spoken aloud in each page and gives haptic feedback to the user. Besides that, it is possible to switch between features without pressing the screen buttons, just by sliding your finger on the screen to the right or left. Each swipe left and right enables a new feature to be activated, and which feature you are on is heard through the speaker. Every feature has a separate interface in which its information is stored.

Screenshots of each feature are provided below:



3.8159943

Figure 4.1: Light Detection View

As shown in Figure 4.1, the value of light detection is 3.81 and this is an indication that there is a small amount of light in the environment, and the program sounds at low intervals and a low noise. The value displayed on the screen is the light intensity value from the sensor and the sound from the speaker changes accordingly.

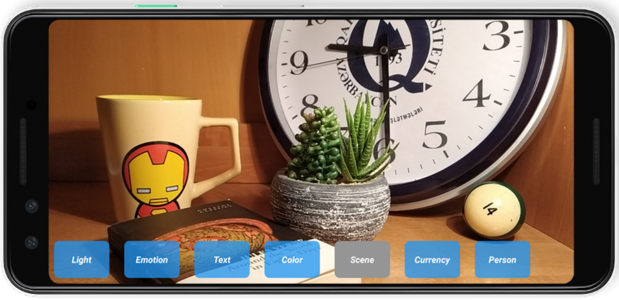
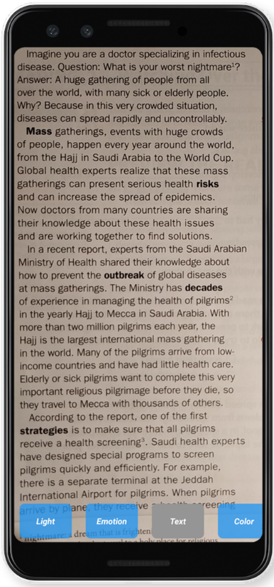


Figure 4.2: Image Labeling View

Figure 4.2 the shows image labeling function. The program can recognize up to 5 objects at the same time and says their names. Here, the program speaks out “clock, pot, flower pot, cup, table”.



Imagine you are a doctor specializing in infectious disease. Question: What is your worst nightmare[1]? Answer: A huge gathering of people from all over the world, with many sick or elderly people. Why? Because in this very crowded situation, diseases can spread rapidly and uncontrollably.

Mass gatherings, events with huge crowds of people, happen every year around the world, from the Hajj in Saudi Arabia to the World Cup. Global health experts realize that these mass gatherings can present serious health **risks** and can increase the spread of epidemics. Now doctors from many countries are sharing their knowledge about these health issues and are working together to find solutions.

In a recent report, experts from the Saudi Arabian Ministry of Health shared their knowledge about how to prevent the **outbreak** of global diseases at mass gatherings. The Ministry has **decades** of experience in managing the health of pilgrims[2] in the yearly Hajj to Mecca in Saudi Arabia. With more than two million pilgrims each year, the Hajj is the largest international mass gathering in the world. Many of the pilgrims arrive from low-income countries and have had little health care. Elderly or sick pilgrims want to complete this very important religious pilgrimage before they die, so they travel to Mecca with thousands of others.

According to the report, one of the first **strategies** is to make sure that all pilgrims receive a health screening[3]. Saudi health experts have designed special programs to screen pilgrims quickly and efficiently. For example, there is a separate terminal at the Jeddah International Airport for pilgrims. When pilgrims arrive by plane, they receive a health screening

[1] nightmare: a dream that is frightening

Figure 4.3: Text-To-Speech View

As shown in Figure 4.3, the text that appears on the camera on this screen is read aloud by the program from start to finish. To read the text again during reading, simply tap the screen once. When you do this, you will hear ”Scanning again” and the camera will start scanning the text on the screen again.

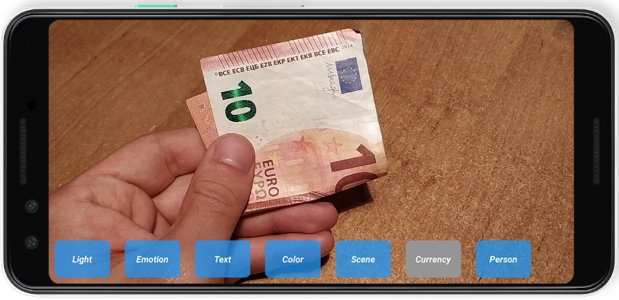


Figure 4.4: Currency Recognition View

Currency identifying function of this App helps users to classify the Euro banknotes. As shown in Figure 4.4, the currency detected is 10 Euro. The app can recognize the 5, 10, 20, 50, 100, 200 and 500-euro banknotes.

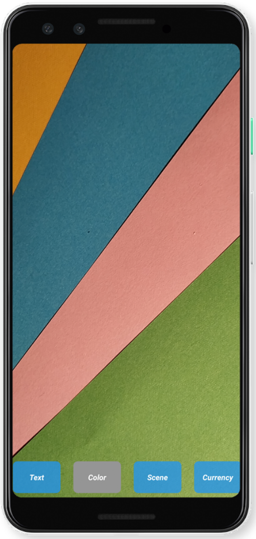


Figure 4.5: Color Recognition View

Figure 4.5 shows Color Recognition View. After the program finds the most dominant color on the screen and compares it with the colors we have previously defined and the program says the color out loud.

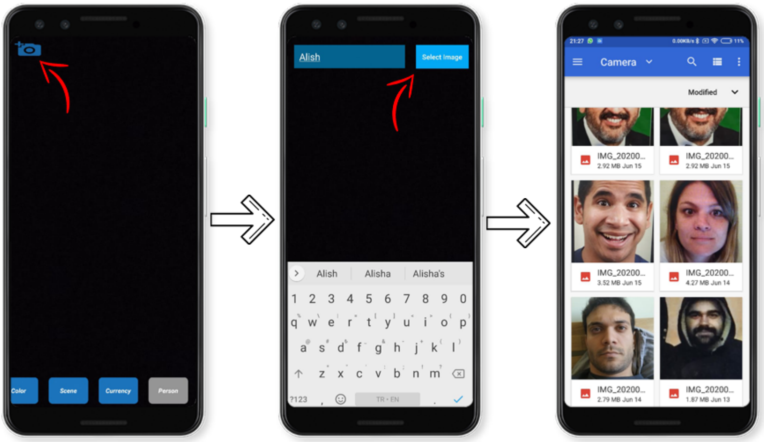


Figure 4.6: Face Recognition View

As shown in Figure 4.6, to use the Face recognition function, firstly we need to upload the picture of the person we want to identify. For this, the above steps must be followed. After the face is added, if the person which added previously seen by the camera, the program tells the name and facial expression of that person. If the person has not been added before, the program tells the person's age, gender, and facial expression. Emotion perception can be made divided into 7 classes (Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise). The predicted gender may be 'Male' or 'Female' and the estimated age is divided into the following ranges - [0 - 2], [4 - 6], [8 - 12], [15 - 20], [25-32],[38- 43], [48-53], [60-100].

5. Evaluation

5.1 User Testing

Chapter 5 comprises the evaluation of the developed App including user and performance testing. The current chapter focuses on data collection tools applied during the evaluation process, describes the procedure, analyses the data and discusses the results.

The usability test was conducted to evaluate the usability of the application. In order to check the efficiency, effectiveness, and satisfaction values, the target audience utilized the application performing the tasks provided. The application was evaluated by ten volunteers who have visual impairments. Four of them were males between the ages of 25 and 55, and the rest were from the opposite gender with the same age range.

After using the application, users were interviewed by phone in order to get their feedback about the functions delivered by the application. They were interviewed in their mother tongue (Azerbaijani) and were translated into English. The target audience were asked to evaluate the usability of each function by the degrees of 'extremely not helpful, somewhat not helpful, somewhat helpful and extremely helpful' providing additional verbal comments on their answers.

As mentioned before, the developed App was supposed to assist unsighted volunteers by distinguishing colours, identifying currency, converting written discourse into speech, recognizing objects around them, recognizing facial expressions and describing people.

The light detection function enables the user to identify the source, direction and the intensity of the light. All of the users reported that they took advantage of this function to check if the lights are on or off in the house and they graded this function as extremely helpful. It has also been stated that this function was not that helpful outside as much as it was inside.

The emotion detector function of this application helps users to find out the psychological state of the person. Two users reported that they had never utilized this kind of App designed for visually impaired people, that is why they found this function extremely helpful. On the other hand, the rest of them claimed that the App recognizes only three emotions in offline mode, namely natural, happy and smiling. It is not possible for the application to distinguish other emotions such as sad and angry except on the online mode. From this point of view, they found it somewhat helpful.

The text-to-speech function assists the target group to convert the written text into audio-lingual format detecting and reading the digital text loudly. Four of the users graded this function as extremely helpful and six of them as somewhat helpful. They expressed that they did not know how to stop reading one material in the middle of the process in order to switch to another text. They should have been informed that if the screen is touched once, it stops reading and prepares itself to read different text material.

The application also helps to recognize the objects around the user. Majority of the volunteers found this function extremely helpful, as it enables them to move around freely and safely.

Colour detector helps to detect the colour of the objects around. All of the volunteers decided that this function is extremely useful. They also stated that they used this function while choosing clothes or doing shopping.

The currency identifying function of this App helps users to classify the Euro banknotes. As the users live in Azerbaijan, they do not use euro, but Manat as currency, therefore they were not able to use this function of the application. However, all of the users liked the idea and asked the developer to add their currency to the system as well.

The face recognizer helps users to identify age, gender, facial expressions and sometimes the name of the person, if the picture belonging to that particular person is uploaded to the system previously. All of the users found the function somewhat not useful, as it only works if the smartphone is connected to the internet.

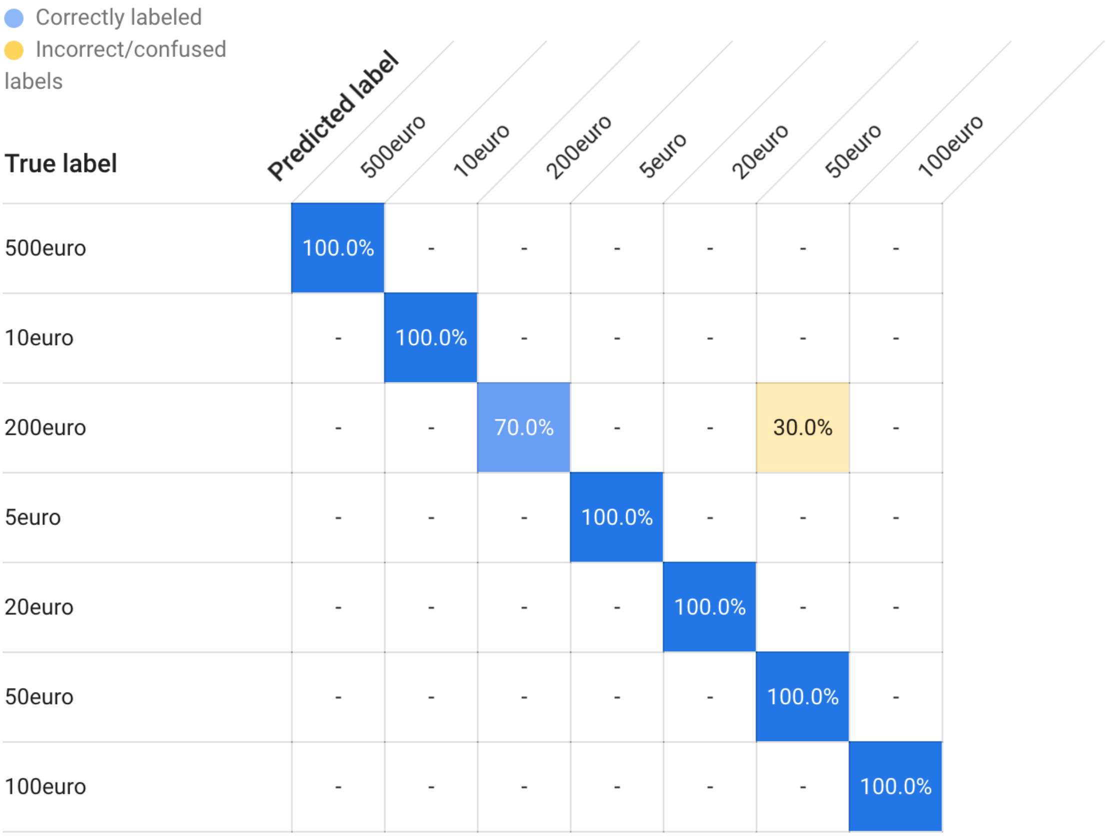
Users also stated that it would be better if there was an audio manual in the application, as they needed somebody else's assistance in order to learn how to use it. Moreover, some functions can be used in particular conditions, for instance, the light detection function works in case the screen of the mobile phone is facing the source of the light.

5.2 Real device and emulator testing

The Google Pixel 1 device was selected for Firebase's performance and accuracy test, and all the following tests related to Firebase were done on the virtual copy of this device at `https://console.firebase.google.com/`.

For the real life test, the Xiaomi MI 6x device was used and all the following tests related to the camera and light sensor were performed on this device.

AutoML currency model evaluation details and model testing



● Correctly labeled
● Incorrect/confused labels

True label \ Predicted label	500euro	10euro	200euro	5euro	20euro	50euro	100euro
500euro	100.0%	-	-	-	-	-	-
10euro	-	100.0%	-	-	-	-	-
200euro	-	-	70.0%	-	-	30.0%	-
5euro	-	-	-	100.0%	-	-	-
20euro	-	-	-	-	100.0%	-	-
50euro	-	-	-	-	-	100.0%	-
100euro	-	-	-	-	-	-	100.0%

Table 5.1: Confusion matrix (Currency detection)

The table 5.1 is the confusion matrix that Auto-ML offers us to evaluate the model and shows how often each label of the model is correctly classified (in blue) and which labels (orange) are the most frequently mixed. As one can be seen from table 5.1 it is difficult for Auto-ML to distinguish between 50 and 200 euros since the important details such as color in visual maintenance are somewhat similar. As a result, the 200 euro banknote has 70% accuracy and is sometimes considered as 50 euro banknote.

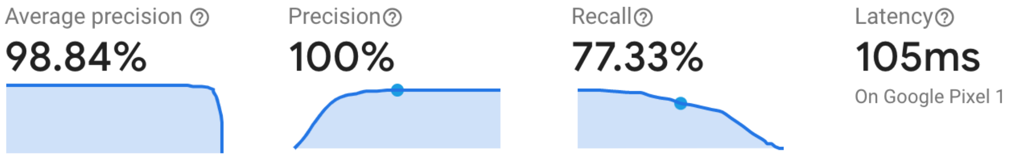


Figure 5.1: Evaluation details

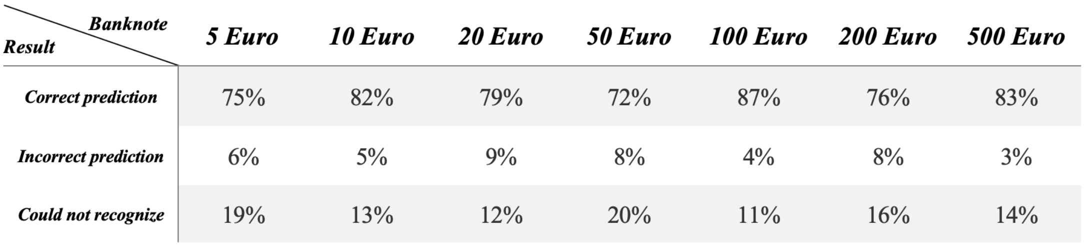
The score threshold is the minimum confidence the model must have for it to assign a label to an image and we chose the threshold as 0.5.

Model performance is measured using two metrics: precision and recall.

In the context of image classification, precision is the ratio of the number of images that were correctly labeled to the number of images the model labeled given the selected threshold. When a model has a high precision value, then it assigns labels incorrectly less often (fewer false positives). As seen from Figure 5.1 , the precision is equal to 100 %.

Recall is the ratio of the number of images that were correctly labeled to the number of images that had content the model should have been able to label. When a model has a high recall, it fails to assign any label less often (fewer false negatives). As seen from Figure 5.1, the recall is equal to 77.73 %.

Latency is the time it takes for a device to make a prediction. In our case, the prediction for currency detection took only 105 ms.

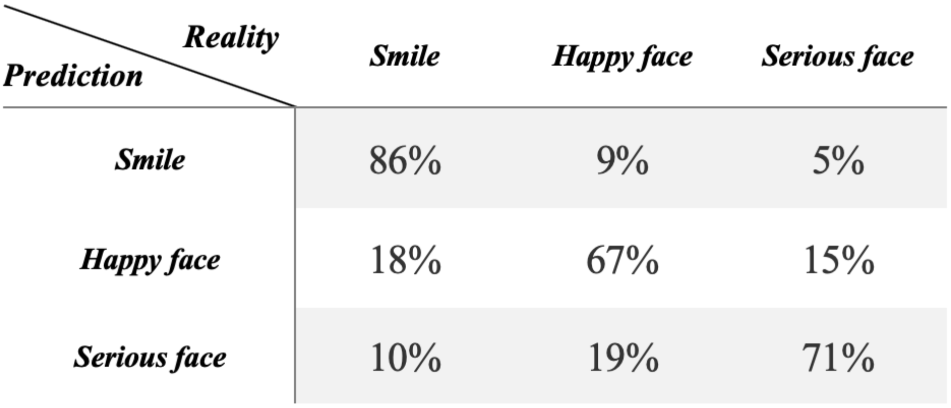


Banknote *Result*	***5 Euro***	***10 Euro***	***20 Euro***	***50 Euro***	***100 Euro***	***200 Euro***	***500 Euro***
Correct prediction	75%	82%	79%	72%	87%	76%	83%
Incorrect prediction	6%	5%	9%	8%	4%	8%	3%
Could not recognize	19%	13%	12%	20%	11%	16%	14%

Table 5.2: Currency Prediction Test Results

Table 5.2 contains a real life test results. As can be seen from the table, when the model is tested with the image of 5, 10, 50, 100, 200, 500 euro banknotes with 100 of each, we see that the model is correctly predicted with an accuracy of 72%, 82%, 79%, 72%, 87%, 76%, 83% respectively.

Emotion Recognition Testing



Reality / Prediction	Smile	Happy face	Serious face
Smile	86%	9%	5%
Happy face	18%	67%	15%
Serious face	10%	19%	71%

Table 5.3: Emotion Recognition Test Results

Table 5.3 contains real life test results. As can be seen from the table, when the function is tested with the image of 60 faces (20 serious, 20 happy, 20 slightly smiling faces), we see that the model is correctly predicted with an accuracy of 86%, 67%, 71% respectively.

Text to Speech Testing

To test the text-to-speech function, the original version of the text was compared with the output of the program via the https://www.diffchecker.com/diff site.

The font of the text greatly affects the result. While reading plain text almost 100% accurately, this ratio decreases for text written in different fonts. In fonts with spaces between letters, they see each letter as a separate word and divides it into letters, and problems arise when reading some math formulas and special characters.

Colour Recognition Testing

The colour recognition feature has many different factors affecting the outcomes and functionality. The reflection of the surface can affect the colour identified by this feature. Also, camera quality and poor lighting can give wrong results. Finally, shadows created by other objects can distort the colour. All these factors contribute to this feature being very hard to evaluate.

Image Labeling Testing

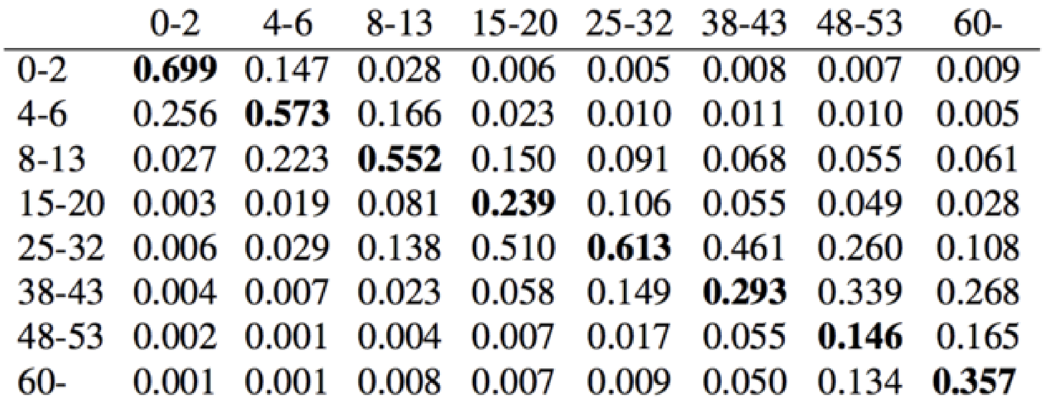
The Table 4.2 below shows the output from Image 4.2. Objects with 4 different labels and their confidence degree are shown in the table.

Face Recognition

The face recognition model is trained with adult photos and doesn't work very well on children. The rate of making mistakes is very high when used on children. The face recognition model works only with source data, as it is trained using a generic data-set of images from websites and celebrity images. These public data-sets are not distributed equally among all individuals in all countries. Face recognition works well with European individuals, but overall accuracy is lower in Asian individuals.

Age Prediction

One of the biggest problems with the trained age prediction model is that it is heavily biased to the 25-32 age group, as shown in the confusion matrix - Table 5.4 [25]:



	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-
0-2	**0.699**	0.147	0.028	0.006	0.005	0.008	0.007	0.009
4-6	0.256	**0.573**	0.166	0.023	0.010	0.011	0.010	0.005
8-13	0.027	0.223	**0.552**	0.150	0.091	0.068	0.055	0.061
15-20	0.003	0.019	0.081	**0.239**	0.106	0.055	0.049	0.028
25-32	0.006	0.029	0.138	0.510	**0.613**	0.461	0.260	0.108
38-43	0.004	0.007	0.023	0.058	0.149	**0.293**	0.339	0.268
48-53	0.002	0.001	0.004	0.007	0.017	0.055	**0.146**	0.165
60-	0.001	0.001	0.008	0.007	0.009	0.050	0.134	**0.357**

Table 5.4: Confusion matrix (Age Prediction)

6. Conclusion

6.1 Summary

In this thesis, studies on how to better use Artificial intelligence and machine learning techniques for the integration of visually impaired people into society have been researched and an application for testing purposes has been developed. With the application, it was aimed to help the visually impaired to obtain more data from their surroundings.

First of all, the idea was to make the program perform predictions during the video. In this way, it was able to get instant feedback without the need for pressing any buttons and waiting. The next goal was to use the program with external tools and/or finger gestures. For this purpose, it has been ensured that commands are given from the screen with swipe gestures and touches. Another goal was to provide haptic and spoken feedback for any event when it changes. The app recognizes objects, colors, currencies, and people's faces when the camera is towards them and also the light source from the sensor.

6.2 Future work

There are parts that need to be added to the application and have to be improved. These parts are left as future work. It is possible to further increase the accuracy of the currency recognition function with more hourly model training. At the same time, it is possible to add more currencies and train the model to enable the program to recognize other currencies. In addition, the use of cloud-based APIs where the program can work online will be considered. In this way, the program can operate with greater accuracy, more power, and more data concentrating on the set. The next goal for the future is to use the program with external tools (external camera and/or external controller to switch between functions).

Bibliography

[1] *Disability and health.* Jan. 2018. URL: https://www.who.int/en/news-room/fact-sheets/detail/disability-and-health (visited on 06/28/2020).

[2] World Health Organization. *International classification of functioning, disability and health : ICF.* 2001.

[3] World Health Organization et al. "Global data on visual impairments 2010". In: *Geneva: WHO* (2012), pp. 3–5.

[4] *Blindness and vision impairment.* Oct. 2019. URL: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (visited on 06/10/2020).

[5] Wafa Elmannai and Khaled Elleithy. "Sensor-Based Assistive Devices for Visually-Impaired People: Current Status, Challenges, and Future Directions". In: *Sensors* 17.3 (2017). ISSN: 1424-8220. DOI: 10.3390/s17030565. URL: https://www.mdpi.com/1424-8220/17/3/565.

[6] Nora Griffin-Shirley et al. "A Survey on the Use of Mobile Applications for People who Are Visually Impaired". In: *Journal of Visual Impairment & Blindness* 111.4 (2017), pp. 307–323. DOI: 10.1177/0145482X1711100402. eprint: https://doi.org/10.1177/0145482X1711100402. URL: https://doi.org/10.1177/0145482X1711100402.

[7] Norbert Streitz and Constantine Stephanidis. *Distributed, Ambient and Pervasive Interactions. Proceedings of the 1st International Conference on Distributed, Ambient and Pervasive Interactions.* Springer International Publishing, July 2013. ISBN: 978-3-642-39350-1.

[8] V. M. Sebastian Raschka. *Python Machine Learning Second Edition.* Packt Publishing Ltd, Aug. 2017, pp. 21–23, 41. ISBN: 978-1-78712-593-3.

[9] *Firebase ML-Kit.* URL: https://firebase.google.com/docs/ml-kit (visited on 07/20/2020).

[10] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.* 2016. arXiv: 1603.04467 [cs.DC].

[11] *Firebase ML-Kit.* May 2018. URL: https://github.com/xiaoyongzhu/SeeingAI-Currency-Detection (visited on 05/18/2020).

[12] Kshitiz Rimal. *Cash Recognition for Visually Impaired.* Jan. 2019. URL: `https://github.com/kshitizrimal/Cash-Recog-Project` (visited on 07/13/2020).

[13] T. Ensar and M. Gunay. "Comparison of face recognition algorithms". In: *2017 25th Signal Processing and Communications Applications Conference (SIU).* 2017, pp. 1–4.

[14] *Vision AI for the Blind and Visually Impaired.* Mar. 2019. URL: `https://www.aipoly.com/` (visited on 06/18/2020).

[15] *Vision AI for the Blind and Visually Impaired.* URL: `https://www.microsoft.com/en-us/ai/seeing-ai` (visited on 07/18/2020).

[16] Griffiths Dawn and Griffiths David. *Head First Kotlin: A Brain-Friendly Guide First Edition.* O'Reilly Media; 1 edition, Mar. 2019. ISBN: 978-1491996690.

[17] *Django Documentation.* English. Version 3.0. Django Software Foundation. July 2020. 1996 pp.

[18] *Android Studio.* URL: `https://developer.android.com/studio` (visited on 04/11/2020).

[19] *Pycharm.* URL: `https://www.jetbrains.com/pycharm/` (visited on 07/20/2020).

[20] *Firebase ML-Kit - label images.* URL: `https://firebase.google.com/docs/ml-kit/android/label-images` (visited on 07/21/2020).

[21] *AutoML Vision Beginner's guide.* URL: `https://cloud.google.com/vision/automl/docs/beginners-guide` (visited on 05/18/2020).

[22] Adam Geitgey. *Face Recognition.* URL: `https://github.com/ageitgey/face_recognition` (visited on 06/11/2020).

[23] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007.

[24] DATAFLAIR TEAM. *Gender and Age Detection with OpenCV.* Jan. 2020. URL: `https://drive.google.com/file/d/1yy_poZSFAPKi0y2e2yj9XDe1N8xXYuKB/view` (visited on 06/25/2020).

[25] G. Levi and T. Hassncer. "Age and gender classification using convolutional neural networks". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 2015, pp. 34–42.

[26] *Adience Benchmark Gender And Age Classification.* URL: `https://www.kaggle.com/ttungl/adience-benchmark-gender-and-age-classification` (visited on 07/23/2020).

[27] “Python super simple to realize human facial emotion recognition”. In: *Python practical collection* (Oct. 2019).

[28] APEKSHA PRIYA. *Emotion Detector Model*. Apr. 2019. URL: `https://github.com/priya-dwivedi/face_and_emotion_detection/tree/master/emotion_detector_models` (visited on 06/14/2020).